www.IrPDF.com

به نام خدا

خود آموز SQL Server 2005

مترجم
مهندس رامین مولاناپور

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Article I.....فـ

هـرست مـطـالـب

Article II.

Article III. مـقـدمـه
Article IV.
Article V. بخش اول: ويـژگـي‌هـاي راهبري پـايـگـاه داده
Article VI.
Article VII. فـصل اول: ويـژگـي‌هـاي مـعـمـاري پـايـگـاه داده و مـوتـور حـافـظه

پشتيباني سخت‌افزاري جديد..... ۱۷
پشتيباني ۶۴ بيتي طبيعي..... ۱۸
پشتيباني NUMA..... ۱۹
پشتيباني از Hyper-Threading..... ۲۰
SQL Server Engine..... ۲۰
يكپارچگي .NET Framework..... ۲۰
پشتيباني چند نمونه‌اي بهبود يافته..... ۲۱
انواع داده جديد..... ۲۱
تصويرگيري و قرينه كردن پايگاه داده..... ۲۱
پشتيباني HTTP طبيعي..... ۲۲
Server Events و DDL Triggers..... ۲۳
بهبودهاي فايل داده پايگاه داده..... ۲۳
پارتيشن‌بندي داده..... ۲٤
بهبودهاي ايندكس..... ۲٦
عمليات ايندكس Online..... ۲۷
بهبودهاي فوق داده و كاتالوگ سيستم..... ۲۷
چند مجموعه نتيجه (MARS)..... ۲۸
بارگذاري داده حجيم..... ۲۸
جستجوي Full-Text..... ۲۹
بهبودهاي پردازشگر پرس‌وجوي T-SQL..... ۲۹

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

امنیت ۲۹
جداسازي كاربر- طرح‌واره ۳۰
زمینه اجراي رویه ذخیره شده ۳۳
كنترل مجوزهاي سنجیده‌تر ۳۵
امنیت كاتالوگ ۳۷

فصل دوم: راهبري پایگاه داده و ابزارهاي برنامه‌نویسي Article VIII.
ابزارهاي مدیریتي و برنامه‌نویسي ۳۹
SQL Server Management Studio ٤۰
Business Intelligence Development Studio ٤۹
زمان استفاده از SQL Server Management Studio و Business Intelligence Development Studio ۵۴
Sqlcmd ۵۵
ابزارهاي كارآیي ۵۸
SQL Server Management Studio Editor: طرح‌هاي اجرا ۵۸
Database Tuning Advisor ٦۰
چارچوب‌هاي مدیریتي جدید ٦۳
SMO ٦٤
AMO ٦۵
RMO ٦۵
WMI ٦٦

فصل سوم: ویژگي‌هاي بازیافت و در دسترس بودن Article IX.
محافظت از خرابي سرور یا پایگاه داده ۶۷
كلاسترینگ Failover بهبود یافته ٦۸
Database Mirroring ۷۰
زمان استفاده از كلاسترینگ یا Database Mirroring ۷۳
بهبودهاي در دسترس بودن پایگاه داده ۷۳
Database Snapshot ۷٤
Early Restore Access ۷٦
عملیات ایندكس به صورت online ۷۸
فرمت‌هاي Online سنجیده ۷۹
اتصال راهبري اختصاصي ۷۹
حافظه Hot-Plug ۸۰
پیكربندي پویاي بهبود یافته ۸۰
سطوح جداسازي تراكنش SQL Server ۸۰
پشتیبان‌گیري و بازیابي ۸۱

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

بازيابي جزيي ۸۱
بهبودهاي قابليت اتكاي رسانه ۸۲
Database Page Checksums ۸۳
پشتيبان‌هاي log و پايگاه داده همزمان ۸۳
پشتيبان‌گيري از كاتالوگ‌هاي Full-Text ۸۳

بخش دوم: ويژگي‌هاي برنامه‌نويسي پايگاه داده
Article X.
Article XI.
Article XII.
فصل چهارم: ويژگي‌هاي قابل برنامه‌نويسي
يكپارچگي CLR ۸۷
اسمبلي‌ها ۸۸
SQL Server .NET Data Provider ۸۹
رويه‌هاي ذخيره شده .NET ۹۲
توابع تعريف شده كاربر .NET ۹۵
تريگرهاي .NET ۹۸
انواع داده تعريف شده كاربر CLR ۱۰۰
انبوهه‌هاي تعريف شده كاربر CLR ۱۰۶
امنيت شيئي پايگاه داده .NET ۱۰۹
زمان استفاده از اشياي پايگاه داده CLR ۱۱۱
بهبودهاي T-SQL ۱۱۱
بهبودهاي TOP ۱۱۱
CTE ۱۱۱
PIVOT و UNPIVOT ۱۱۳
تريگرهاي DDL ۱۱٤
خروجي DML ۱۱۵
WAITFOR ۱۱٦
نوع varchar (max) جديد ۱۱٦
مديريت صرف‌نظر از تراكنش ۱۱۷
زمان استفاده از اشياي پايگاه داده T-SQL ۱۱۸
بهبودهاي ADO.NET ۱۱۸
پشتيباني از مكان‌نماي سرور با استفاده از SQLResultSet ۱۱۸
پشتيباني غيرهمزمان ۱۲۰
MARS ۱۲۱
صفحه‌بندي ۱۲۳
درج يكجا ۱۲٤
مدل اتصال مشترك ۱۲۵

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Article XIII. فصل پنجم: سرویس‌های هشداردهی
مروري بر سرويس‌هاي هشداردهي ۱۲۸
رویدادها ۱۲۸
اشتراك‌ها ۱۲۸
هشدارها ۱۲۹
موتور هشداردهي ۱۲۹
معماري Notification Services ۱۲۹
نوشتن برنامه‌هاي Notification Services ۱۳۱
مراحل برنامه‌نويسي ۱۳۲
برنامه Notification Services نمونه ۱۳۳

Article XIV. فصل ششم: SQL Server Service Broker
مروري بر SQL Server Service Broker ۱۵۰
طراحي برنامه صف ۱۵۰
گفتگوها ۱۵۲
گروه مكالمه ۱۵۲
فعال‌سازي SQL Server Service Broker ۱۵٤
انتقال پيام ۱۵٤
برنامه‌نويسي با SQL Server Service Broker ۱۵۵
مدل برنامه‌نويسي ۱۵۵
DDL و T-SQL DML ۱۵۷
برنامه نمونه SQL Server Service Broker ۱۵۸
راهبري Service Broker ۱٦۳
گزينه‌هاي پيكربندي سيستم ۱٦۳
امنيت گفتگو ۱٦٤
ديدگاه سيستمي ۱٦۵

Article XV. فصل هفتم: يكپارچگي XML
نوع داده XML طبيعي ۱٦۸
انواع داده XML نوع‌دار قوي ۱٦۹
متدهاي نوع داده XML ۱۷۲
پشتيباني XQuery ۱۷۵
ايندكس‌هاي XML ۱۷۵
ايندكس‌هاي XML اصلي ۱۷٦
ايندكس‌هاي XML ثانويه ۱۷٦
بهبودهاي FOR XML ۱۷۷
رهنمود Type ۱۷۷
پرس‌وجوهاي FOR XML تودرتو ۱۷۸

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

توليد طرح واره XSD درون برنامه‌اي ۱۷۹
بهبودهاي OPENXML ۱۸۰
بارگذاري حجيم XML ۱۸۲
دستيابي HTTP SOAP طبيعي ۱۸۳
بهبودهاي XML براي Analysis Server ۱۸٤
XML for Analysis Services ۱۸٤
ديدگاه‌هاي كاتالوگ XML سيستم ۱۸۵

Article XVI. بخش سوم : ويژگي‌هاي هوش تجاري
Article XVII.
Article XVIII. فصل هشتم : Reporting Services
معماري Reporting Services ۱۹۰
اجزاي Reporting Services ۱۹۲
Report Designer ۱۹۲
OLAP Report Designer ۱۹۷
Report Server ۱۹۸
Report Manager ۲۰۰
كلاينت گزارشگيري كاربر نهايي Report Builder ۲۰۱
تأليف گزارش ۲۰۲
مراحل ساخت ۲۰۲
ايجاد يك گزارش Reporting Services ۲۰۳
توزيع يك گزارش Reporting Services ۲۱۳
اجراي يك گزارش Reporting Services ۲۱٤

Article XIX. فصل نهم : Integration Services
معماري Integration Services جديد ۲۱۸
DTP ۲۱۹
DTR ۲۲۱
اجزاي بسته Integration Services ۲۲۲
ويژگي‌هاي بسته Integration Services ۲۲۲
آداپتورهاي داده ۲۲۸
كانتينرها ۲۲۹
وظايف ۲۳۰
تبديلات ۲۳۱
مديريت خطا ۲۳۳
تأمين‌كنندگان Log ۲۳٤
ابزارهاي Integration Services ۲۳٤
ابزارهاي طراحي Integration Services ۲۳۵

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Integration Services Designer ۲۳۷
ابزارهاي بسته‌بندي Integration Services ۲٤٥
Package Migration Wizard ۲٤٦
برنامه سودمند Integration Services Packege Management ۲٤٦
برنامه‌هاي سودمند Integration Services Package Execution ۲٤۷

Article XX. فصل دهم : Analysis Services
مروري بر Analysis Services ۲٥۰
انواع حافظه OLAP ۲٥۱
بهبودهاي موتور Analysis Services ۲٥۲
پشتيباني چند نمونه‌اي ۲٥۲
پشتيباني از كلاسترينگ Failover ۲٥۳
يكپارچگي با .NET Framework ۲٥۳
Unified Dimensional Model ۲٥۳
پشتيباني از تريگر ۲٥٥
پشتيباني رديابي ۲٥٥
پشتيباني از اسكريپت‌نويسي ۲٥٥
بهبودهاي محلي‌سازي ۲٥٥
پشتيباني رديف جدول واقعيت بي‌مرجع ۲٥٦
بهبودهاي مديريتي Analysis Services ۲٥٦
SQL Server Computer Manager ۲٥٦
SQL Server Management Studio ۲٥۷
امنيت ۲٥۸
بهبودهاي پشتيبان‌گيري و بازيابي ۲٥۹
بهبودهاي برنامه‌نويسي ۲٦۰
Business Intelligence Development Studio ۲٦۰
Profiler ۲٦۹
XMLA ۲۷۰
بهبودهاي ODL ۲۷۱
بهبودهاي MDX ۲۷۱
ADOMD.NET ۲۷۲
اشياي مديريتي Analysis Services (AMO) ۲۷۳
Data Mining ۲۷۳
Decision Trees ۲۷٤
Time Series ۲۷٤
Clustering و Sequence Clustering ۲۷٤
Naïve Bayes ۲۷٤
Association Rules ۲۷٤

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Article XXI. بخش چهارم: پیوست‌ها
Article XXII.
Article XXIII. پیوست الف: نصب و ارتقا
ویرایش‌های SQL Server 2005 ..... ۲۷۷
نصب SQL Server 2005 ..... ۲۷۷
ارتقا به SQL Server 2005 ..... ۲۹۵
ارتقا از SQL Server 7 و 2000 ..... ۲۹۵
ارتقا از SQL Server 6.5 (یا ما قبل) ۲۹۶
Article XXIV.
Article XXV. پیوست ب: وقایع سریع

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

فصل اول

ویژگی‌های معماری پایگاه داده و موتور حافظه

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

این نسخه از SQL Server بر طبق معماری ضروری مشابه با آن‌چه که در نسخه ۷ از SQL Server تولید می‌شد، ساخته شده است و مایکروسافت به انجام بهبودهای انقلابی در موتور SQL Server ادامه می‌دهد. طبق قانون مور، ظرفیت سخت‌افزاری در هر ۱۸ ماه دو برابر می‌شود، در حالی که در همان مدت زمانی، هزینه‌های سخت‌افزاری کاهش می‌یابد. هرچند، در حالی که سخت‌افزار ارزان‌تر می‌شود، هزینه‌های مردم بیشتر می‌شود. در نسخه SQL Server 2005، مایکروسافت بهبودهایی را صورت داده است که به SQL Server امکان می‌دهند تا از جدیدترین نسل‌های سخت‌افزار با کارآیی بالا بهره ببرند و این قابلیت را برای آن‌ها فراهم کرده است که خود را به اوج محصولات جهانی برسانند. هم‌زمان، این شرکت ویژگی‌هایی را اضافه کرده است که مدیریت آن‌ها توسط SQL Server آسان‌تر است و هم‌چنین قابلیت‌های جدیدی را فراهم نموده است که به شما امکان می‌دهد تا ارزش بیشتری برای SQL Server در سازمان خود قایل شوید. در این فصل، برخی از مهم‌ترین بهبودها را از نظر معماری که مایکروسافت در SQL Server 2005 صورت داده است، معرفی خواهیم کرد، با این هدف که به شما کمک کنیم تا ببینید این ویژگی‌ها چگونه می‌توانند بهره‌وری شما و قابلیت اعتماد محیط پایگاه داده را بهبود بخشند.

Article XXVI. پشتیبانی سخت‌افزاری جدید

یکی از ویژگی‌های جدید کلیدی که به SQL Server 2005 به عنوان یک موتور پایگاه داده اجازه می‌دهد تا سطوح کارآیی یکسانی با بزرگ‌ترین پایگاه‌های داده UNIX داشته باشد، پشتیبانی سخت‌افزار بهبود یافته آن است. حافظه همیشه یکی از حیاتی‌ترین عواملی بوده است که در کارآیی پایگاه داده تأثیر داشته است و پایگاه‌های داده بزرگ UNIX که رتبه‌بندی کارآیی TPC-C کلاستر فشرده بالایی دارند، همگی به سطح کارآیی با استفاده از یک سخت‌افزار ۶۴ بیتی و محیط سیستم عامل توجه داشته‌اند. هنگامی که یک محیط ۶۴ بیتی قابل مقایسه برای SQL Server به شکل پردازنده Intel Itanium2 ۶۴ بیتی و پشتیبانی ۶۴ بیتی طبیعی Windows Server 2003 در دسترس قرار می‌گیرد، SQL Server بلافاصله به رأس آزمون کارآیی TPC-C منتقل می‌شود. SQL Server 2005 این پشتیبانی ۶۴ بیتی طبیعی را به همراه قابلیتی برای پشتیبانی از NUMA[1] از Windows Server 2003 به ارث می‌برد. SQL Server که بر طبق محیط Windows Server 2003 ساخته شده است، پشتیبانی ۶۴ بیتی را برای معماری Itanium IA64 و معماری AMD x64 فراهم کرده است.

1- Non-Uniform Memory Architecture

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۲۶.۰۱ پشتیبانی ۶۴ بیتی طبیعی

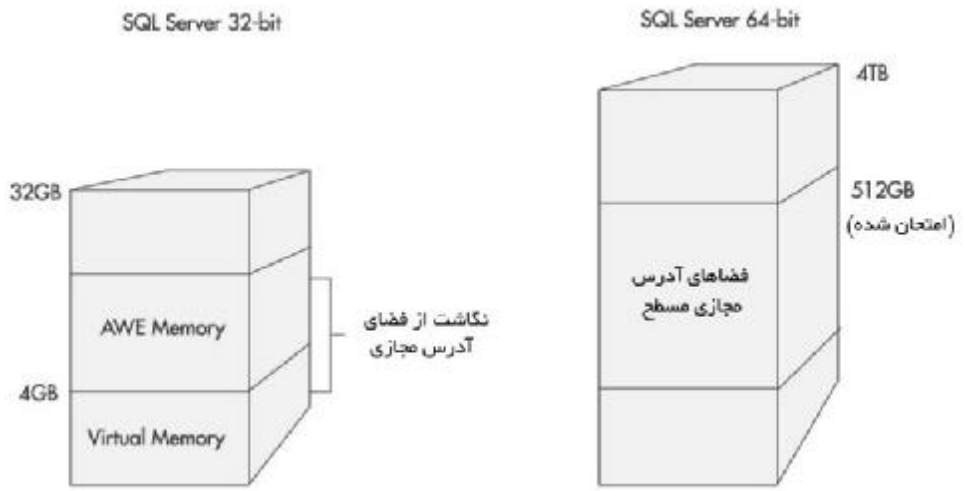
هنگام اجرا در Windows Server 2003 برای سیستم‌های Itanium ۶۴ بیتی، SQL Server 2005 از پردازنده‌های Itanium و Itanium2 اینتل پشتیبانی می‌کند. هنگام اجرا در Windows Server 2003 برای Extended Systems ۶۴ بیتی، SQL Server 2005 از پردازنده‌های Opteron ۶۴ بیتی AMD و AMD Athlon 64 و Xeon اینتل با EMT64[1] پشتیبانی می‌کند. SQL Server 2005 کاملاً از محیط‌های سخت‌افزاری ۳۲ و ۶۴ بیتی برای تمام سرویس‌های عمده آن‌ها، از جمله SQL Server Engine، Analysis Services، SQL Agent و Reporting Services پشتیبانی می‌کند. مزیت واقعی در انتقال یک پایگاه داده به محیط ۶۴ بیتی، توان پردازش سریع‌تر نیست. در عوض، مزیت واقعی در حافظه قابل آدرس‌دهی افزایش یافته نهفته است. شکل ۱-۱ مقایسه‌ای از حداکثر حافظه قابل آدرس‌دهی برای SQL Server ۳۲ بیتی و نگارش ۶۴ بیتی SQL Server 2005 نشان می‌دهد.

معماری ۳۲ بیتی طبیعی به حداکثر 4GB حافظه قابل آدرس‌دهی محدود است. تحت Windows، این حد 4GB حتی بین سیستم عامل و برنامه‌ها تقسیم می‌شود. به عبارت دیگر، 2GB برای سیستم عامل Windows رزرو شده است و 2GB برای برنامه‌ها باقی‌می‌ماند. با استفاده از پشتیبانی AWE در نگارش ۳۲ بیتی، SQL Server می‌تواند حداکثر 32GB از RAM را آدرس‌دهی کند. در حالی که این افزایش واقعی است، باز هم سربار صفحه‌بندی برای داشتن صفحات حافظه مناسب وجود دارد. پیاده‌سازی ۶۴ بیتی طبیعی، محدودیت حافظه را با افزایش حداکثر حافظه قابل دسترسی به 32TB به‌طور مجازی برطرف کرده است. در حال حاضر، هیچ سیستم تولیدی در هیچ جایی، از این مقدار RAM فیزیکی پشتیبانی نمی‌کند. حداکثر مقدار RAM فیزیکی که SQL Server 2005 در زمان عرضه با آن تست شده است، 512GB است. با نسخه بعدی بسته خدماتی بعدی برای Windows Server 2003، انتظار می‌رود حداکثر حافظه مورد پشتیبانی تا 1 TB افزایش یابد.

1- Intel Extended Memory 64 Technology

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

(b) **شکل ۱-۱ مقایسه آدرس‌دهی حافظه ۳۲ و ۶۴ بیتی**

Section۲۶،۰۲ پشتیبانی NUMA

ویژگی جدید دیگری که سیستم عامل Windows Server 2003 انجام می‌دهد، پشتیبانی NUMA[1] است. NUMA معماری سیستمی مورد استفاده برخی تولید کنندگان سیستم از قبیل IBM و Unisys است که کاربرد حافظه و CPU را در سیستم‌های چند پردازنده‌ای کارآمدتر از معماری SMP معمولی مدیریت می‌کند. در سیستم‌های SMP استاندارد، هنگامی که سرعت و تعداد پردازنده‌ها افزایش می‌یابد، رقابت بین پردازنده‌ها برای دستیابی به حافظه موجب رقابت گذرگاه می‌شود. این امر موجب تأخیرهایی در پردازش شده و قابلیت یک سیستم را برای مقیاس‌بندی دچار مشکل می‌کند. نتیجه این امر این است که سیستم‌های SMP در صورت افزایش تعداد پردازنده‌ها، به‌طور خطی مقیاس‌بندی نمی‌شوند. معماری NUMA برای حل این مشکل در سیستم‌های SMP طراحی شده است که در آن، پردازنده‌ها قادر به دستیابی سریع‌تر به داده در RAM گذرگاه سیستمی که می‌تواند آن را فراهم کند، هستند. معماری NUMA، CPU و حافظه را در podهای محلی چند پردازنده گروه‌بندی می‌کند. این podها از طریق یک گذرگاه خارجی به یکدیگر متصل می‌شوند که ترافیک داده چند pod را منتقل می‌کند. این آرایش Pod، مسأله رقابت را با محدود کردن تعداد CPUهای رقیب برای دستیابی به حافظه، برطرف می‌کند. برای تشخیص حداکثر مزایای این معماری، سیستم عامل و برنامه‌ها باید طوری طراحی شوند تا ترافیک داده چند podای را حداقل کرده و دستیابی حافظه داخل pod سریع را حداکثر کنند. اگر سیستم‌عامل و برنامه به‌طور صحیح طراحی شده باشند، معماری

1- Non-Uniform Memory Architecture

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

NUMA مقیاس‌بندی تقریباً خطی را ممکن می‌سازد، هنگامی که پردازنده‌های بیشتری اضافه شوند. Windows Server 2003 و SQL Server 2005 بهبودهای معماری را برای افزایش درجه‌ای که رشته‌ها و حافظه استفاده می‌کنند و در همان pod قرار دارند، مشارکت می‌دهد.

Section ۲۶،۰۳ پشتیبانی از Hyper-Threading

SQL Server 2005 هم‌چنین می‌تواند از Hyper-Threading بهره ببرد و ما باید ممنون مایکروسافت باشیم که Hyper-Threading را به Windows Server 2003 اضافه کرده است. Hyper-Threading یک فناوری CPU است که توسط اینتل توسعه یافته است و دو پردازنده منطقی را برای هر پردازنده فیزیکی در یک سیستم ایجاد می‌کند. هر پردازنده منطقی، قادر به اجرای هم‌زمان رشته‌های مجزاست. هدف Hyper-Threading فراهم کردن مصرف منبع بهتر برای برنامه‌های چند رشته‌ای یا چندین برنامه در حال اجرا روی یک ماشین است. در حالی که Hyper-Threading پتانسیل افزایش بازده در یک سرور را فراهم می‌کند، پردازنده‌های منطقی برای منابع سیستم از قبیل داده در کش پردازنده رقابت می‌کنند. این رقابت برای منابع مانع از انجام کار CPU Hyper-Threading شده و استفاده از دو CPU فیزیکی توسط سیستم قابل مقایسه می‌شود.

SQL Server 2005 دو مزیت عمده را از پشتیبانی Windows Server 2003 برای Hyper-Threading می‌گیرد. اول این که، برخلاف Windows 2000، Windows Server 2003 تنها پردازنده‌های فیزیکی را برای اهداف تعیین مدرک به حساب می‌آورد. بنابراین، یک پردازنده تکی با استفاده از Hyper-Threading، به عنوان یک پردازنده تکی تعیین می‌شود و نه یک پردازنده دوم که تحت Windows 2000 چنین بود. سپس، Windows Server 2003 زمان‌بندی رشته‌ای بهبود یافته را برای سیستم‌های Hyper-Threading شده فراهم کرده است. این تغییرات موجب کارآیی بهتر برای برنامه‌های چند رشته‌ای نظیر SQL Server می‌شوند.

Article XXVII. SQL Server Engine

در حالی که اولین بخش این فصل، تصویر بزرگ‌تری از پشتیبانی سخت‌افزاری جدید فراهم شده در SQL Server 2005 را بررسی کرد، این بخش به بررسی عمقی برخی از مهم‌ترین بهبودهایی می‌پردازد که مایکروسافت در خود SQL Server Engine صورت داده است.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۲۷.۰۱ یکپارچگی NET Framework.

مهم‌ترین بهبود موتور SQL Server در SQL Server 2005، یکپارچگی Microsoft .NET Framework است. یکپارچگی NET Framework. با SQL Server، قابلیت SQL Server 2005 را با امکان ساخت رویه‌های ذخیره شده، توابع تعریف شده کاربر، تریگرها، انبوهه‌ها و انواع تعریف شده کاربر با استفاده از هر یک از زبان‌های NET.، از قبیل VB.NET، C# یا J# توسعه داده است. یکپارچگی NET CLR. با SQL Server 2005، بیش از یک مسأله سطحی است، زیرا موتور پایگاه داده SQL Server از CLR در فرآیند میزبانی می‌کند. می‌توانید در فصل ۴ مطالب بیشتری در این زمینه بیاموزید.

Section ۲۷.۰۲ پشتیبانی چند نمونه‌ای بهبود یافته

بهبود مهم دیگر در ویرایش Enterprise از SQL Server 2005، پشتیبانی برای حداکثر تا ۵۰ نمونه است. این مسأله در SQL Server 2000، ۱۶ نمونه بود. این بهبود مهم خاصی برای میزبانی از فروشندگانی است که چند سرویس SQL Server را به عنوان بخشی از ارایه سرویس‌های وب خود اجاره می‌دهند.

Section ۲۷.۰۳ انواع داده جدید

SQL Server 2005 هم‌چنین از چند نوع داده جدید پشتیبانی می‌کند. در حالی که یکپارچگی NET Framework. ، پشتیبانی برای انواع تعریف شده کاربر را ممکن می‌سازد، SQL Server 2005 نیز یک جفت از انواع داده طبیعی جدید را فراهم کرده است: انواع داده XML و varbinary(max). نوع داده varbinary(max) متد جدیدی را برای استفاده از LOBها با SQL Server فراهم می‌کند. برخلاف انواع داده Image و Text قدیمی‌تر، نوع داده varbinary(max) جدید می‌تواند به عنوان یک متغیر استفاده شود و با برنامه‌نویسی می‌تواند با آن همانند انواع داده کوچک‌تر رفتار کرد که این امر موجب روش‌های کاربرد آسان‌تر و مستحکم‌تر می‌شود.

نوع داده XML جدید برطبق نوع داده varbinary(max)، به شما امکان ذخیره اسناد XML را در پایگاه داده می‌دهد. هرچند، نوع داده XML جدید برای داده XML طراحی شده است و از تأیید طرح‌واره اسناد XML پشتیبانی می‌کند. می‌توانید در فصل ۷ مطالب بیشتری درباره انواع داده جدید مورد پشتیبانی SQL Server 2005 بیاموزید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۲۷،۰۴ تصویرگیری و قرینه کردن پایگاه داده

قرینه کردن پایگاه داده[1] موجب محافظت از خرابی پایگاه داده با دادن یک قابلیت standby پایگاه داده ثابت به SQL Server 2005 می‌شود. قرینه کردن پایگاه داده یک فناوری در دسترس بودن سطح پایگاه داده است که با تمام سخت‌افزارهای استاندارد مورد پشتیبانی SQL Server کار می‌کند. هیچ نیازی برای هر حافظه مشترک بین سرور اصلی و سرور قرینه وجود ندارد و هیچ محدودیت فاصله‌ای نیز وجود ندارد. قرینه کردن پایگاه داده با ارسال ثبت وقایع تراکنش بین سرور اصلی و سرور قرینه کار می‌کند. اصولاً ویژگی قرینه کردن پایگاه داده، یک برنامه انتقال ثبت بلادرنگ را می‌سازد. قرینه کردن پایگاه داده می‌تواند در یک یا چند پایگاه داده تنظیم شود.

Database Snapshot یک تصویر فقط خواندنی از یک پایگاه داده را در نقطه خاصی در زمان فراهم می‌کند. Database Snapshot برای ایجاد کپی‌هایی از یک پایگاه داده برای گزارش‌گیری یا ایجاد کپی‌های پشتیبان از یک پایگاه داده مناسب است که می‌توانید برای roll back کردن یک پایگاه داده تولیدی با حالت قبلی آن استفاده کنید. Database Snapshot می‌تواند با Database Mirroring برای ایجاد یک سرویس گزارش‌گیری برطبق داده موجود در سرور قرینه ترکیب شود. داده در سرور قرینه نمی‌تواند مستقیماً مورد دستیابی قرار گیرد، زیرا سرور قرینه همیشه در حالت بازیافت است. هرچند، می‌توانید یک Database Snapshot را در پایگاه داده قرینه ایجاد کرده و آن‌گاه برای گزارش‌گیری به دیدگاه پایگاه داده دسترسی داشته باشید. در فصل ۳ مطالب بیشتری درباره قرینه کردن و دیدگاه‌های پایگاه داده می‌آموزید.

Section ۲۷،۰۵ پشتیبانی HTTP طبیعی

یکی از بهبودهای مهم دیگر موتور پایگاه داده SQL Server، الحاق پشتیبانی HTTP طبیعی به خود موتور است. قابلیت SQL Server برای پردازش درخواست‌های HTTP ورودی، به SQL Server امکان فراهم کردن اجرای عبارت SQL و احضار رویه ذخیره شده را از طریق پروتکل SOAP می‌دهد. این بدان معنی است که SQL Server 2005 قادر به پردازش درخواست‌های سرویس وب ورودی بدون وجود IIS یا سرور وب دیگری است. پشتیبانی HTTP جدید قابلیت‌های استماع HTTP طبیعی را به SQL Server می‌دهد، از جمله قابلیت پشتیبانی از نقاط انتهایی HTTP که URL پورت و درخواست‌هایی را که پشتیبانی خواهند شد، مشخص می‌کنند. SQL Server هم‌چنین قادر به انتشار سرویس‌های وب به عنوان WSDL[2] برای نقاط انتهاست. پشتیبانی SQL Server HTTP سازگار با استانداردهاست، و از SOAP و نگارش‌های 1.0 و 1.2 و WSDL 1.1 پشتیبانی می‌کند. ویژگی

1- Database Mirroring
2- Web Services Description Language

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

پشتیبانی، HTTP طبیعی جدید، هم‌چنین از تعیین هویت SQL Server و Windows و SSL پشتیبانی می‌کند. برای فعال کردن این ویژگی برای داشتن سازگاری بیشتر با برنامه‌نویسی ردیف میانی، رویه‌های ذخیره شده می‌توانند مجموعه نتایج را به عنوان یک ADO.NET DataSet برگردانند.

Section ۲۷.۰۶ Server Events و DDL Triggers

ویژگی‌های Server Events و DDL Triggers جدید SQL Server 2005 به شما امکان می‌دهند تا با برنامه‌نویسی به تغییرات در سیستم پاسخ دهید. در حالی که هر دوی این ویژگی‌های جدید می‌توانند اعمال مشابهی را انجام دهند، به‌طور کاملاً متفاوتی پیاده‌سازی می‌شوند. تریگرهای DDL، شبیه تریگرهای DML استاندارد، رویدادهای هم‌زمانی هستند که رویه‌های ذخیره شده را اجرا می‌کنند. در فصل ۴، مطالب بیشتر درباره تریگرهای DDL می‌آموزید.

در مقایسه، رویدادهای سرور غیرهم‌زمان هستند. در مدل رویدادی سرور، سرور یک رویداد را برای یک SQL Broker Service ثبت می‌کند و آن‌گاه یک مصرف کننده می‌تواند به‌طور مستقل آن رویداد را بازیابی کند. خود رویداد به عنوان داده XML ثبت می‌شود. هیچ روشی برای roll back کردن یک رویداد وجود ندارد و در صورتی از یک رویداد صرف‌نظر می‌شود که هیچ مصرف کننده‌ای آن را بازیابی نکند. هنگامی که رویدادی رخ می‌دهد، رویداد سیستم فعال می‌شود که می‌تواند شما را از رویداد یا اجرای اختیاری یک روال کد مطلع کند. مثال زیر ساختار دستوری مورد استفاده برای تنظیم یک هشدار رویداد را تشریح می‌کند:

```
CREATE EVENT NOTIFICATION MyDDLEvents
ON SERVER FOR DDL_STATEMENTS TO SERVICE MyDDL_log
```

این مثال رویداد جدیدی را ایجاد کرده و هشدار رویداد را MyDDLEvents می‌نامد و رویداد را به عبارت DDL متصل می‌کند. بخش TO SERVICE مشخص می‌کند که SQL Broker Service با نام MYDLL_log دریافت کننده رویدادها خواهد بود. می‌توانید مطالب بیشتری درباره SQL Service Broker را در فصل ۶ بیابید.

Section ۲۷.۰۷ بهبودهای فایل داده پایگاه داده

SQL Server 2005 هم‌اکنون از توانایی تغییر مسیر فایل‌های داده و ثبت وقایع یک پایگاه داده با استفاده از فرمان ALTER DATABASE پشتیبانی می‌کند. SQL Server 2000 توانایی انتقال فایل‌ها را برای پایگاه داده tempdb فراهم می‌کرد، ولی این مسأله برای سایر پایگاه‌های داده ممکن

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

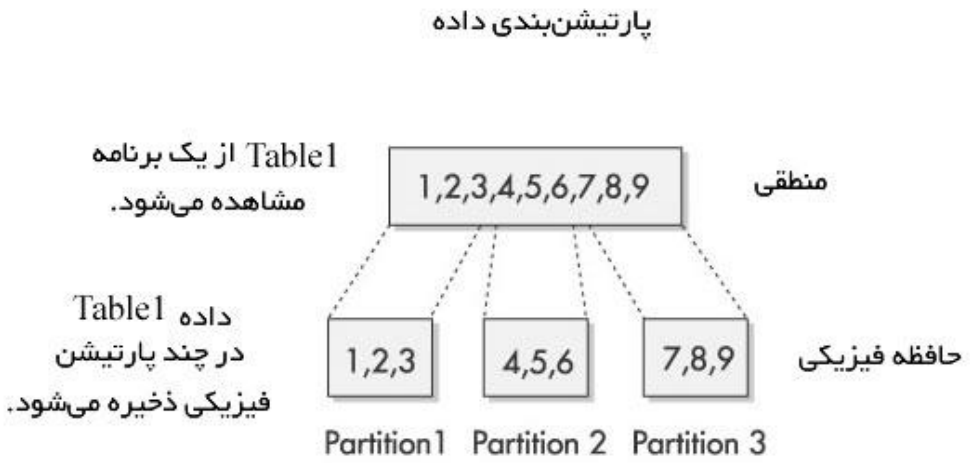
نبود. همان‌گونه که ممکن است انتظار داشته باشید، SQL Server 2005 از انتقال فایل‌ها تنها به عنوان یک عمل offline پشتیبانی می‌کند. مثال بعد ساختار دستوری عبارات ALTER DATABASE جدید را تشریح می‌کند:

```
ALTER DATABASE <database_name>
   MODIFY FILE(name=<'data_file_name'>, filename=<'new path'>)
```

Section ۲۷،۰۸ پارتیشن‌بندی داده

بهبود جدید دیگر در SQL Server 2005، توانایی انجام پارتیشن‌بندی داده است. پارتیشن‌بندی داده[1] به شما اجازه می‌دهد تا یک شئ پایگاه داده از قبیل یک جدول یا ایندکس را به چند قطعه بشکنید. ویژگی پارتیشن‌بندی داده جدید، مدیریت جداول و ایندکس‌های خیلی بزرگ را تسهیل می‌کند. پارتیشن‌بندی برای برنامه‌ها شفاف است که تنها خود شئ پایگاه داده را می‌بینند و از چند پارتیشنی بودن حافظه مرتبط که توسط SQL Server مدیریت می‌شود، آگاه نیست.

پارتیشن‌ها می‌توانند ایجاد و حذف شوند، بدون این که بر در دسترس بودن خود شئ پایگاه داده تأثیر بگذارند. لزوماً، پارتیشن‌بندی به شما امکان می‌دهد تا حافظه داده مرتبط را به چند شئ بشکنید، در حالی که هنوز دیدگاه یک شکلی از شئ و تمام پارتیشن‌های آن را به یک برنامه نشان می‌دهید. شکل ۲-۱ مروری پایه‌ای از پارتیشن‌بندی را نشان می‌دهد.



(a)

(b) شکل ۲-۱ پارتیشن‌بندی داده

1- Data Partitioning

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

SQL Server 2005 از پارتیشن‌بندی داده برای جداول، ایندکس‌ها و دیدگاه‌های ایندکس‌دار پشتیبانی می‌کند. ردیف واحد پایه پارتیشن‌بندی است. پارتیشن‌ها می‌توانند برطبق مقادیر موجود در ستون‌ها در یک ردیف ایجاد شوند. این مسأله، پارتیشن‌بندی افقی[1] نامیده می‌شود. برای نمونه، یک جدول ممکن است برطبق تاریخ پارتیشن‌بندی شود که پارتیشن متفاوتی برای هر سال ایجاد می‌شود. این نوع پارتیشن‌بندی طبق تاریخ به شما امکان می‌دهد تا یک نوع پردازش پنجره‌ای تاریخ اسلایدی را انجام دهید که در آن می‌توانید پارتیشن حاوی داده سال قبل را حذف کنید به‌طوری که بر داده موجود در پارتیشن سال جاری هیچ تأثیری نگذارد.

پارتیشن‌بندی داده یک جفت مزیت مهم را برای هر پایگاه داده بزرگ (VLDBها) فراهم می‌کند. پارتیشن‌های داده می‌توانند مدیریت داده را تسهیل کنند و به شما امکان پشتیبان‌گیری انتخابی را تنها از پارتیشن‌های مشخص شده می‌دهند. مثلاً، در مورد جدول بزرگی که طبق تاریخ پارتیشن‌بندی می‌شود، شاید بخواهید تنها از سال جاری پشتیبان‌گیری کنید نه پارتیشن سال قبل. مزیت دیگر این است که در سیستم‌های چند پردازنده‌ای، می‌توانید یک CPU را به پردازش پارتیشن خودش اختصاص دهید تا بازده بهبود یابد.

دو مرحله پایه‌ای برای پیاده‌سازی پارتیشن‌بندی داده وجود دارد. در اولین مرحله، باید دقیقاً تعیین کنید که چگونه می‌خواهید یک شئ معین را پارتیشن‌بندی کنید. در دومین مرحله، باید هر پارتیشن را به یک محل حافظه فیزیکی نسبت دهید. پارتیشن‌های متفاوت می‌توانند همگی به یک گروه فایل تکی یا پارتیشن‌های متفاوتی که می‌توانند به چند گروه فایل نگاشت شوند، نسبت داده شوند.

مثال بعد ساختار دستوری برای ایجاد یک تابع پارتیشن ساده و طرح‌واره‌ای را که یک جدول را با استفاده از یک بخش Range پارتیشن‌بندی خواهد کرد، نشان می‌دهد:

```
CREATE PARTITION FUNCTION MyPF
  (int) AS RANGE LEFT FOR VALUES (50, 100)
GO
CREATE PARTITION SCHEME MyPS
  AS PARTITION MyPF TO (FileGroup1)
GO
CREATE TABLE MyTable (col1 int, col2 varchar(50))
  ON MyPS(col1)
GO
```

1- Horizontal Partitionong

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

خط اول یک تابع پارتیشن‌بندی به نام MyPF را ایجاد می‌کند. (int) نشان می‌دهد که پارتیشن‌بندی روی ستونی خواهد شد که با استفاده از نوع داده int تعریف شده است. کلمه کلیدی Range مشخص می‌کند که از پارتیشن‌بندی Range استفاده خواهد شد. کلمه کلیدی LEFT کنترل می‌کند که کدام پارتیشن مقادیر خط حاشیه را دریافت خواهد کرد. مقدار LEFT نشان می‌دهد که هر ردیفی که دارای یک مقدار است که با کران پارتیشن تطابق دارد، بلافاصله به پارتیشن سمت چپ منتقل خواهد شد. بخش VALUES برای تعریف محل‌های کران پارتیشن‌ها استفاده می‌شود. توجه به این نکته مهم است که این مقادیر نقاط کران هستند و نه خود پارتیشن‌ها. این مسأله واقعاً موجب ایجاد سه پارتیشن می‌شود: اولی که حاوی مقادیر منفی تا ۵۰ است؛ پارتیشن دوم که حاوی مقادیر ۵۱ تا ۱۰۰ است و پارتیشن سوم که حاوی تمام مقادیر ۱۰۱ به بالا است.

خط دوم یک طرح‌واره پارتیشن‌بندی به نام MyPS را ایجاد می‌کند. بخش AS PARTITION برای مشخص کردن عمل پارتیشن‌بندی استفاده می‌شود که توسط این طرح‌واره به کار خواهد رفت. این مثال از تابع پارتیشن‌بندی MYPF استفاده می‌کند. بخش TO گروه فایل گروه‌های فایلی را تعیین می‌کند که پارتیشن‌ها را ذخیره خواهند کرد. این مثال از یک گروه فایل تکی به نام FileGroup1 استفاده می‌کند.

سپس، طرح‌واره پارتیشن باید به جدولی ضمیمه شود که پارتیشن‌بندی خواهد شد. این مثال دستور CREATE TABLE توسعه یافته را نشان می‌دهد که به جدول امکان پارتیشن‌بندی شدن را می‌دهد. اولین بخش عبارت CREATE TABLE تغییر نکرده است. این بخش نام جدول (در این مثال MyTable) و ستون‌های جدول را مشخص می‌کند. این جدول ساده از دو ستون به نام‌های col1 و col2 استفاده می‌کند. آن‌گاه کلمه کلیدی ON جدیدی برای مشخص کردن طرح‌واره پارتیشن‌بندی مورد استفاده به کار می‌رود. این مثال از طرح‌واره پارتیشن‌بندی MyPS استفاده می‌کند که ایجاد کرده‌اید و ستونی که حاوی داده کلید پارتیشن‌بندی است، در پرانتز قرار می‌گیرد. این مثال از ستون col2 برای کلید پارتیشن‌بندی استفاده می‌کند. این ستون یک نوع داده int است که باید با نوع داده مشخص شده در تابع پارتیشن‌بندی تطابق داشته باشد.

محدودیت‌هایی در انواع ستون‌هایی که می‌توانند برای کلید پارتیشن‌بندی استفاده شوند، وجود دارد. این محدودیت‌ها بسیار شبیه محدودیت ستون‌هایی است که می‌توانند در یک ایندکس به کار روند. متن ntext و انواع داده تصویری نمی‌توانند استفاده شوند. علاوه بر این، ستون‌های timestamp نیز محدود شده‌اند. تنها انواع داده طبیعی "T-SQL" می‌توانند استفاده شوند. نمی‌توانید از یک نوع تعریف شده کاربر به عنوان کلید پارتیشن‌بندی استفاده کنید. هرچند، می‌توانید از نوع داده varchar(max) جدید استفاده کنید. هم‌چنین حد ۱۰۰۰ پارتیشن در هر جدول وجود دارد و تمام پارتیشن‌ها باید در یک گروه وجود داشته باشند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۲۷.۰۹ بهبودهای ایندکس

بهبودهای جدیدی در ایندکس‌ها در SQL Server 2005 وجود دارد. اول این که، بازسازی یک ایندکس کلاستر شده، دیگر تمام ایندکس‌های غیرکلاستری را مجبور به بازسازی نمی‌کند. در SQL Server 2000، هنگام بازسازی یک ایندکس کلاستر شده، تمام ایندکس‌های کلاستر نشده مرتبط نیز بازسازی می‌شدند. دیگر چنین حالتی وجود ندارد، زیرا SQL Server 2005 ایندکس‌های کلاستر نشده را در طی بازسازی ایندکس کلاستر شده صحیح و سالم حفظ می‌کند.

سپس، یک ویژگی ستون‌های موجود وجود دارد که به شما امکان افزودن ستون‌های غیرکلیدی را به ایندکس می‌دهد. این ویژگی جدید به پرس‌وجوهای بیشتر امکان می‌دهد تا توسط ایندکس تحت پوشش قرار گیرند، بنابراین کارآیی پرس‌وجو را با حداقل کردن نیاز برای موتور SQL Server برای رفتن به جدول مرتبط برای تکمیل پرس‌وجو بهبود می‌بخشد. در عوض، موتور می‌تواند الزامات پرس‌وجو را با استفاده از داده در ایندکس تحت پوشش برآورده سازد. یکی از جنبه‌های خوب ویژگی ستون‌های موجود جدید این واقعیت است که ستون‌های موجود که بخشی از کلید نیستند، در اندازه حداکثر ایندکس وجود ندارند که باز هم ۹۰۰ بایت است.

بهبود ایندکس جدید دیگری که مایکروسافت به SQL Server 2005 اضافه کرده است، توانایی غیرفعال کردن یک ایندکس است. غیرفعال کردن یک ایندکس، آن ایندکس را از نگهداری توسط موتور SQL Server متوقف کرده و هم‌چنین مانع از کاربرد ایندکس می‌شود. هنگامی که ایندکسی غیرفعال می‌شود. SQL Server فضای حافظه مورد استفاده ایندکس را می‌گیرد ولی فوق داده ایندکس را حفظ می‌کند. قبل از این که ایندکس غیرفعال بتواند مجدداً فعال شود، باید با استفاده از فرمان ALTER INDEX بازسازی شود.

Section ۲۷.۱۰ عملیات ایندکس Online

نگارش‌های قبلی SQL Server هیچ دستیابی به ایندکس را در هنگام بازسازی ایندکس ممکن نمی‌ساختند. شما باید منتظر می‌ماندید تا وقتی که فرآیند بازسازی کامل شود و جدول بتواند مجدداً بهنگام شود. ویژگی عملیات ایندکس Online جدید SQL Server 2005 به برنامه‌ها امکان دستیابی به ایندکس و انجام عملیات بهنگام‌رسانی، درج و حذف در یک جدول را می‌دهد، در حالی که عملیات بازسازی ایندکس در حال انجام است. می‌توانید اطلاعات بیشتری درباره عملیات ایندکس SQL Server 2005 Online را در فصل ۳ بیابید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۲۷،۱۱ بهبودهای فوق داده و کاتالوگ سیستم

در SQL Server 2000 و نگارش‌های قبلی، کاتالوگ سیستم و فوق داده به عنوان بخشی از هر پایگاه داده در پایگاه داده اصلی ذخیره می‌شدند. در SQL Server 2005، این روش تغییر کرده است و هم‌اکنون فوق داده در پایگاه داده منبع قرار می‌گیرد که سیستم را به عنوان یک شئ sys ذخیره می‌کند. SQL Server 2005 دیگر هیچ دستیابی مستقیمی به جداول سیستم میسر نمی‌سازد. این تغییر ارتقاهای سیستم سریع‌تر و ایمنی بهتری را با مستحکم ساختن فوق داده سیستم ممکن می‌سازند. فوق داده کاتالوگ با استفاده از فیلترهای سطح ردیف ایمن می‌شوند. می‌توانید مطالب بیشتری درباره امنیت سطح ردیف SQL Server 2005 را در بخش بعدی این فصل بیاموزید.

فوق داده جدید کاملاً با قبل سازگار است، مادامی که از جداول سیستمی مستند نشده‌ای استفاده کنید که مایکروسافت به کرات هشدار داده است که از آن‌ها استفاده نشود. فوق داده‌های سیستم در SQL Server 2005 از طریق مجموعه‌ای از دیدگاه‌های کاتالوگ ارایه می‌شوند. دیدگاه‌های کاتالوگ، همانند دیدگاه‌های ANSI INFORMATION_SCHEMA توابع Property و توابع تعبیه شده، نیاز به استفاده از جداول سیستم را که ممکن است در SQL Server 2000 انجام داده باشید، برطرف می‌کنند. در کل، بیش از ۲۵۰ دیدگاه کاتالوگ جدید در SQL Server 2005 وجود دارد و آن‌ها می‌توانند از طرح‌واره sys پایگاه داده هر کاربر مشاهده شوند. می‌توانید دیدگاه‌های سیستم جدید را با استفاده از Microsoft SQL Server Manager Studio برای باز کردن Object Browser بیابید و آن‌گاه به گره Databases | <database> | Views | System Views بروید. هم‌چنین می‌توانید یک پنجره پرس‌وجوی جدید را باز کرده و این پرس‌وجو را وارد کنید:

```
select * from sys.system_views
```

Section ۲۷،۱۲ چند مجموعه نتیجه (MARS)

نگارش‌های قبلی SQL Server محدود به یک مجموعه نتیجه فعال در هر اتصال بودند. SQL Server 2005 هم اینک قادر به پشتیبانی از چند مجموعه نتیجه فعال در یک اتصال است. این ویژگی جدید به شما امکان می‌دهد تا یک اتصال به پایگاه داده را باز کنید، یک پرس‌وجو را اجرا کرده و مقداری نتیجه را پردازش کنید و آن‌گاه پرس‌وجوی دیگری را شروع کرده و نتایج آن را پردازش کنید. برنامه‌های شما می‌توانند به راحتی بین چند مجموعه نتیجه باز، عقب و جلو کنند. در فصل ۴ نحوه استفاده از این ویژگی MARS جدید به وسیله مثال‌هایی نشان داده می‌شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۲۷،۱۳ بارگذاری داده حجیم

SQL Server 2005 برخی بهبودهای مهم را در زمینه افزایش کارآیی در بارگذاری داده حجیم فراهم کرده است. پردازش بارگذاری داده حجیم هم‌اکنون از یک فایل فرمت مبتنی بر XML استفاده می‌کند که تمام عملکرد موجود در نگارش‌های قبلی فایل فرمت BCP[1] را فراهم می‌کند. علاوه بر این، فرمت XML خواندن و درک فایل فرمت BCP را آسان‌تر می‌سازد. برای سازگاری قبلی با برنامه‌های موجود، فایل فرمت BCP قدیمی هنوز می‌تواند استفاده شود.

فرآیند بارگذاری داده حجیم SQL Server 2005 هم‌اکنون از ثبت ردیف‌های بد پشتیبانی می‌کند. این امر به فرآیند بارگذاری داده حجیم امکان می‌دهد تا حتی در صورت مواجه شدن با ردیف‌ها یا داده نامعتبر، به کار خود ادامه دهد. ردیف‌هایی که به‌طور نادرست فرمت شده باشند، به همراه شرحی از شرط خطا، در یک فایل خطا نوشته می‌شوند. ردیف‌هایی که از الزامات تخطی کرده باشند، به یک جدول خطا به همراه شرط خطای خاص آن‌ها هدایت می‌شوند.

Section ۲۷،۱۴ جستجوی Full-Text

پشتیبانی از جستجوی Full-Text نیز در SQL Server 2005 بهبود یافته است. نگارش‌های قبلی SQL Server نیاز به استفاده از رویه‌های ذخیره شده برای ایجاد کاتالوگ‌های جستجوی Full-Text داشتند. در SQL Server 2005، چندین عبارت DDL جدید معرفی شده است تا به شما امکان کار کردن با ویژگی‌های SQL Server Full-Text را بدهند. برای نمونه، دو عبارت DDL جستجوی SQL Full-Text جدید، CREATE FULLTEXT CATALOG و CREATE FULLTEXT INDEX هستند.

بهبودهای دیگر در جستجوی FULL-TEXT در SQL Server 2005 شامل توانایی پشتیبان‌گیری و بازیابی ایندکس‌ها و کاتالوگ‌های جستجوی FULL-TEXT به همراه داده پایگاه داده شما هستند. در ضمن، ایندکس‌ها و کاتالوگ‌های FULL-TEXT می‌توانند به پایگاه‌های داده متناظر خود ضمیمه شده و از آن‌ها جدا شوند. بهبود جالب دیگر پشتیبانی جستجوی FULL-TEXT SQL Server 2005، توانایی استفاده از یک گنج واژه برای یافتن مترادف کلمات جستجو است.

1- Bulk Copy Program

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۲۷،۱۵ بهبودهای پردازشگر پرس‌وجوی T-SQL

بهبودهای متعددی برای پردازشگر پرس‌وجو در SQL Server 2005 وجود دارد، از جمله CTE[1]، یک بخش TOP بهبود یافته، یک عبارت WAITFOR بهبود یافته و یک بخش OUTPUT جدید برای عبارات DML. مثال‌های استفاده از این بهبودها در T-SQL در فصل ۴ ارایه می‌شوند.

Article XXVIII. امنیت

امنیت بزرگ‌ترین نسخه مایکروسافت از زمانی بوده است که شرکت Trustworthy Computing Initiative خود را شروع کرد. هدف Trustworthy Computing Initiative مایکروسافت، ایمن‌تر و قابل اعتمادتر ساختن محصولات مایکروسافت است. همان‌گونه که انتظار می‌رود، SQL Server 2005 که بخشی از Trustworthy Computing Initiative است، دریافت کننده تعدادی از مهم‌ترین بهبودهای امنیتی است. دغدغه امنیتی مایکروسافت برای SQL Server بر ساخت ایمن‌تر و مستحکم‌تر محصول نسبت به طراحی آن از طریق توزیع آن تمرکز دارد. هنگام طراحی بهبودهای امنیت برای SQL Server 2005، مایکروسافت از برخی از اصول امنیتی پیروی می‌کند. ابتدا، باید سیستم را با در نظر گرفتن تمام تنظیمات نصب پیش‌فرض به سمت ایجاد یک محیط ایمن، ایمن کرد. در حالی که گزینه‌هایی برای کاربران باز گذاشته شده است تا تنظیمات امنیتی کمتری را انتخاب کنند، انتخاب این گزینه‌ها نیاز به گزینش‌های آگاهانه‌ای دارد. سپس، مایکروسافت از اصل حداقل امتیازات در طراحی سیستم خود پیروی می‌کند. این سیستم طوری طراحی می‌شود که یک فرد برای انجام یک عمل معین و نه چیز دیگری، باید تنها نیاز به امتیازاتی داشته باشد. بالاخره، مایکروسافت قصد داشت تا ناحیه سطح عرضه بالقوه را با فراهم کردن توانایی نصب تنها اجزای مورد نیاز کاهش دهد.

تمام این ویژگی‌های امنیتی جدید در SQL Server 2005، عمیقاً تحت تأثیر مواردی هستند که مایکروسافت در طی تلاش امنیتی خود در اوایل سال ۲۰۰۲ کشف کرد و در طراحی و پیاده‌سازی SQL Server 2005، آن‌ها را اعمال نمود. برخی از بهبودهای ویژگی امنیتی اصلی در SQL Server 2005 که در این بخش مطالعه خواهید کرد، شامل جداسازی کاربران از طرح‌واره‌ها، زمینه اجرای رویه ذخیره شده جدید، کنترل سنجیده‌تر مجوزها، تقویت خط‌مشی کلمه عبور جدید، تغییراتی در امنیت سطح ردیف و امنیت بهبود یافته برای کاتالوگ‌ها هستند.

Section ۲۸،۰۱ جداسازی کاربر- طرح‌واره

مهم‌ترین تغییر مربوط به امنیت در SQL Server 2005، جداسازی کاربر- طرح‌واره است. یک کاربر یا شاید به‌طور دقیق‌تر، یک اصل، هر موجودیتی است که اشیا پایگاه داده در مقابل آن ایمن

1- Common Tables Expressions

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

می‌شوند. یک اصل می‌تواند یک کاربر Windows، یک کاربر SQL Server، یک نقش برنامه یا یک نقش باشد. در SQL Server 2000، مستقیماً تحت مالکیت اشیای پایگاه داده کاربران قرار می‌گیرد و خود کاربران در جدول سیستمی Sys_Users بودند. همگی این موارد در SQL Server 2000 تغییر کرده‌اند. حال اشیای پایگاه داده تحت مالکیت طرحواره‌ها هستند. کاربران دیگر مستقیماً مالک اشیای پایگاه داده نیستند؛ در عوض آن‌ها مالک طرحواره‌ها هستند. در SQL Server 2000، کاربران و سایر اصول امنیتی را می‌توانید در دیدگاه sys.database_principals جدید بیابید. لیست طرحواره‌های SQL Server 2000 را می‌توانید در دیدگاه sys.schemas جدید بیابید.

یک طرحواره[1] کانتینر اشیا است. طرحواره توسط بخش سوم از ساختار دستوری نام‌گذاری شئ چهار بخشی مورد استفاده SQL Server تعیین می‌شود. مثال بعد ساختار دستوری نام‌گذاری SQL Server 2005 را تشریح می‌کند که هر بخش از نام، به‌طور افزایشی خاص‌تر می‌شود.

Server_name.Database_name.Schema_name.Object_name

در تمام نسخه‌های قبلی SQL Server، نام طرحواره و نام مالک لزوماً یکسان بود. در SQL Server 2005، مالک مجزا از طرحواره است. هنگامی که SQL Server 2000 و نسخه‌های قبلی نام اشیا را تفکیک می‌کردند، SQL Server 2005 ابتدا Database_name.User_name.Object_name را جستجو می‌کرد و در صورت عدم موفقیت، آن‌گاه Database_name.dbo.Object_name را جستجو می‌کند.

دلیل اصلی برای این جداسازی کاربر و طرحواره در SQL Server 2005، سامان بخشیدن به مشکل نیاز به تغییر مالکیت چند شئ پایگاه داده است، چنان‌چه کاربر معینی سازمان را ترک کرده باشد. علاوه بر این، عمل تغییر مالک یک شئ پایگاه داده موجب تغییر نام می‌شود. مثلاً، اگر مالک Table1 در پایگاه داده MyDB از UserA به UserB تغییر کند، آن‌گاه نام معین Table1 از MyDB.UserA.Table1 به MyDB.UserB.Table1 تغییر خواهد کرد. برای کمک به جلوگیری از این مشکل، بعضی از سازمان‌ها، استاندارد مالکیت تمام اشیای پایگاه داده توسط dbo را می‌پذیرند، ولی چیزی در سرور وجود ندارد که این مسأله را اجباری کند.

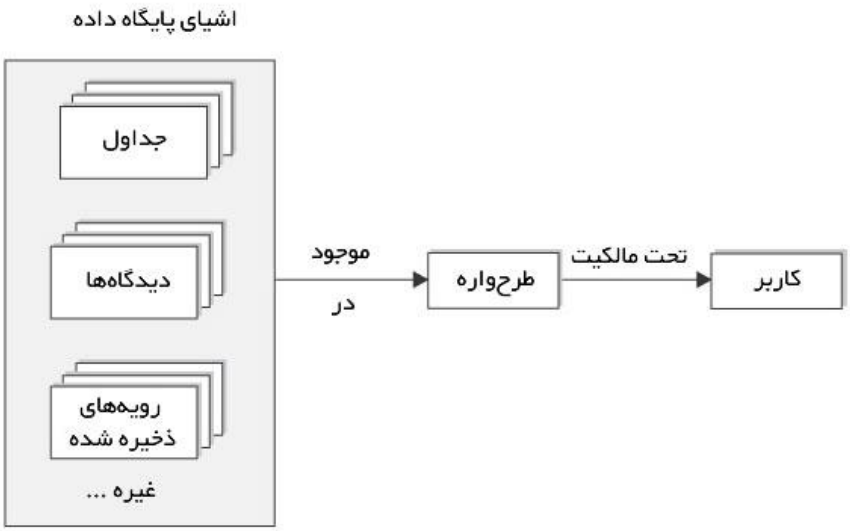
پیاده‌سازی SQL Server 2005 از مفهوم طرحواره یک پایگاه داده، سطحی از انتزاع را در زنجیره مالکیت شئ پایگاه داده معرفی کرده است. می‌توانید زنجیره مالکیت شئ پایگاه داده SQL Server 2005 را در شکل ۳-۱ ببینید.

1- Schema

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

در SQL Server 2005، اشیای پایگاه داده در طرح‌واره‌ها قرار دارند و طرح‌واره به نوبت تحت مالکیت کاربران هستند. این سطح انتزاع جدید، مسأله تغییر مالکیت شئ پایگاه داده را قابل مدیریت‌تر می‌سازد. حذف کاربری که مالک اشیای پایگاه داده در SQL Server 2005 است، بدین معنی است که DBA هم اینک تنها باید مالک طرح‌واره را تغییر دهد و نه تمام تک تک اشیای پایگاه داده را. این امر تعداد اشیایی را که DBA باید برای مالک اشیا در یک پایگاه داده با آن‌ها تماس داشته باشد، به شدت کاهش می‌دهد. برای تغییر مالک تمام اشیا در پایگاه داده SQL Server 2005، تنها مالک طرح‌واره را تغییر داده و آن‌گاه می‌توانید کاربر قدیمی را حذف کنید. تغییر مالک شئ پایگاه داده، نام شئ را تغییر نمی‌دهد، زیرا نام طرح‌واره تغییر نکرده است و تنها مالک آن تغییر کرده است.



(a)

(b) **شکل ۳-۱ زنجیره مالکیت شئ SQL Server 2005**

همان‌گونه که ممکن است انتظار داشته باشید، شئ طرح‌واره جدید، روشی را که SQL Server تفکیک نام شئ پایگاه داده را انجام می‌دهد، تغییر می‌دهد. حال هر کاربر دارای یک طرح‌واره پیش‌فرض مرتبط است و SQL Server 2005 ابتدا نام یک شئ نامعین را با استفاده از طرح‌واره پیش‌فرض کاربر جستجو خواهد کرد. اگر این امر ناموفق بود، SQL Server شئ را با استفاده از نام طرح‌واره dbo جستجو خواهد کرد. مثلاً، اگر UserA دارای یک طرح‌واره پیش‌فرض MySchema1 باشد و آن کاربر پرس‌وجویی را برای جستجوی Table1 انجام دهد، آن‌گاه سرور در ابتدا تلاش خواهد کرد تا نام را با استفاده از MySchema1.Table1 تفکیک کند و آن‌گاه به dbo.Table1 برگردد.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

تنها به دلیل این که پایگاه‌های داده SQL Server 2000 می‌توانستند حاوی چند کاربر و نقش باشند، پایگاه‌های داده SQL Server 2005 می‌توانند حاوی چند طرحواره باشند. هر طرحواره دارای یک مالک اصلی است که معمولاً یک کاربر یا نقش می‌باشد. برای اهداف تفکیک نام، هر کاربر دارای یک طرحواره پیش‌فرض است. آن‌گاه اشیای واقعی پایگاه داده در یک طرحواره قرار می‌گیرند. برای ایجاد اشیای پایگاه داده جدید در یک طرحواره، باید مجوز CREATE را برای خود شئ پایگاه داده و مجوز ALTER یا CONTROL را برای طرحواره داشته باشید. زنجیره مالکیت باز هم بر طبق مالکان اصلی است و نه طرحواره‌ها.

SQL Server 2005 تغییرات DDL چندی را برای سروکار داشتن با جداسازی کاربر و طرحواره جدید معرفی کرده است، از جمله عبارت CREATE/DROP/ALTER برای اشیای USER، ROLE و SCHEMA. لیست زیر نحوه ایجاد و انتساب یک طرحواره پایگاه داده را تشریح می‌کند:

```
/* Create a login */
CREATE LOGIN UserA WITH PASSWORD = 'ABC123#$'
GO
/* Create a user for that login – the schema doesn't need to exist*/
CREATE USER UserA FOR LOGIN UserA
  WITH DEFAULT_SCHEMA = MySchema
GO
/* Create the schema and assign its owner */
CREATE SCHEMA MySchema AUTHORIZATION UserA
GO
/* Create a Table in the new schema */
CREATE TABLE MySchema.Table1 (col1 char (20))
GO
```

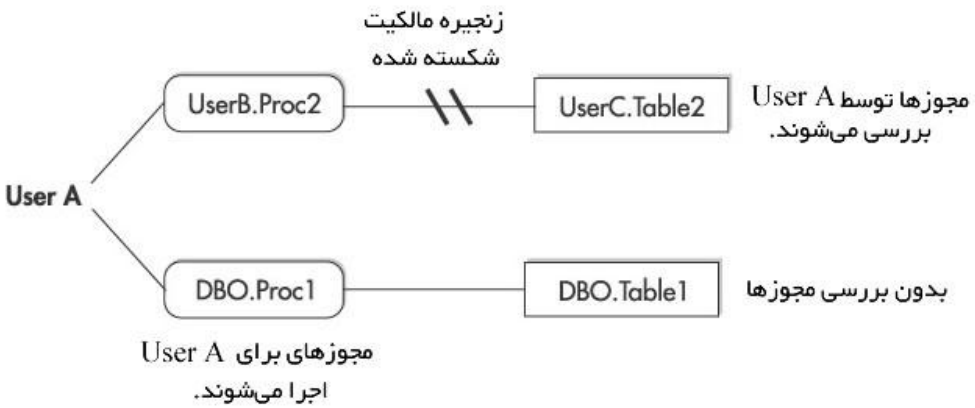
اولین خط در این لیست، login جدیدی به نام UserA را ایجاد کرده و کلمه عبوری را برای آن Login تنظیم می‌کند. خط بعد کاربر جدیدی به نام UserA را برای این login ایجاد کرده و طرحواره پیش‌فرضی را برای UserA تحت عنوان MySchema تنظیم می‌کند. طرحواره واقعی نباید در زمانی که در عبارت CREATE USER مشخص می‌شود، وجود داشته باشد. اگر هنگام ایجاد کاربر جدید، طرحواره پیش‌فرضی را مشخص نکنید، آن‌گاه طرح واره پیش‌فرض با dbo تنظیم خواهد شد. سپس، عبارت CRAETE SCHEMA برای ایجاد یک طرحواره جدید به نام MySchema استفاده می‌شود. بخش AUTHORIZATION مالک طرح واره را برای UserA تنظیم می‌کند. بالاخره، جدولی به نام Table1 در طرحواره به نام MySchema ایجاد می‌شود. مالک MySchema و اشیای آن نظیر Table1، UserA است.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۲۸،۰۲ زمینه اجرای رویه ذخیره شده

در حالی که مایکروسافت به یکی از ویژگی‌های جدید به عنوان زمینه اجرای رویه ذخیره شده مراجعه می‌کند، واقعاً آن را بر ماژول‌ها اعمال می‌کند و نه رویه‌های ذخیره شده. یک ماژول می‌تواند یک رویه ذخیره شده، یک تابع، یا یک اسمبلی باشد. تنظیم زمینه اجرا برای یک ماژول موجب می‌شود تمام عباراتی که در آن ماژول قرار دارند، برای مجوزها در مقابل زمینه اجرای مشخص شده بررسی شوند. به عبارت دیگر، با تنظیم زمینه اجرای یک ماژول معین، موجب می‌شوید که تمام عباراتی که در آن ماژول قرار دارند، با استفاده از هویت کاربری اجرا شوند که مشخص کرده‌اید (به‌جای فراخوانی واقعی ماژول). این ویژگی جدید به شما امکان می‌دهد تا مزایایی مشابه با آن‌چه که از طریق زنجیره مالکیت SQL Server 2000 تشخیص می‌دادید، داشته باشید، ولی این انعطاف‌پذیرتر است زیرا محدودیت‌های مشابهی ندارد. مثلاً، برخلاف زنجیره مالکیت SQL Server 2000 که به شما اجازه تغییر دادن زمینه اجرا را برای SQL پویا نمی‌داد، زمینه اجرای ماژول SQL Server 2005 بر SQL پویا تنها به این دلیل اعمال می‌شود که در SQL ایستا نیز انجام می‌شود. برای درک بهتر این موضوع، شکل ۴-۱ را ببینید که زنجیره مالکیت SQL Server 2000 را تشریح می‌کند.



(a)

(b) شکل ۴-۱ زنجیره مالکیت SQL Server 2000

برای این که UserA، dbo.Proc1 را اجرا کند، باید مجوز اجرای آن شئ را داشته باشد. هرچند، هنگامی که dbo.Proc1 به dbo.Table1 دسترسی دارد، هیچ مجوزی بررسی نمی‌شود، زیرا dbo مالک هر دو شئ است. این مثالی از یک زنجیره مالکیت صحیح بود. در سناریو بعد، برای UserA جهت اجرای UserB.Proc2، UserA باید مجوزهای Execute را برای آن شئ داشته باشد. سپس، هنگامی که UserB.Proc2 تلاش می‌کند تا به UserC.Table2 دستیابی داشته باشد، انتخاب مجوزها از UserA

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

باید بررسی شود. در این مورد، به دلیل این که UserB.Proc2 و UserC.Table2 دارای مالکان متفاوتی هستند، زنجیره مالکیت نقض می‌شود.

اجرای SQL Server 2005 این سناریو را آسان می‌کند، همان‌گونه که در شکل ۵-۱ نشان داده شده است. در این سناریو، هنگامی‌که UserA تلاش می‌کند تا UserB.Proc2 را اجرا کند، SQL Server بررسی می‌کند تا اطمینان یابد که UserA دارای مجوزهای Execute برای UserB.Proc1 است. اگر شئ UserB.Proc1 با یک زمینه اجرایی ایجاد شود که مشخص کند یک رویه ذخیره شده به عنوان UserZ اجرا خواهد شد، آن‌گاه هنگامی که رویه ذخیره شده UserB.Proc1 تلاش می‌کند تا به UserC.Table1 دستیابی داشته باشد، UserB.Proc1 برای مجوزهای Select تنها کاربر مشخص شده در زمینه اجرا را بررسی خواهد کرد که در این مورد UserZ است. هیچ مجوز Selectای برای UserA مورد نیاز نیست، کسی که فراخوان واقعی است.

(c)

شکل ۵-۱ زمینه اجرای SQL Server 2005 (d)

لیست زیر نشان می‌دهد چگونه زمینه اجرای یک رویه ذخیره شده به نام MyProc1 را تغییر می‌دهید:

```
ALTER PROC MySchema.Proc1 WITH EXECUTE AS USER UserB
```

این عبارت بخش WITH EXECUTE جدید را نشان می‌دهد. در این‌جا، رویه ذخیره شده‌ای به نام Proc1 که در MySchema قرار دارد، برای اجرا تحت زمینه UserB تنظیم شده است. باید نام کاربری را برای محتوای اجرا مشخص کنید. نمی‌توانید نام یک نقش را مشخص کنید. تغییرات در زمینه اجرا در دیدگاه Sys.sql_modules جدید ذخیره می‌شود.

Section ۲۸،۰۳ کنترل مجوزهای سنجیده‌تر

SQL Server 2005 هم‌چنین کنترل سنجیده‌تری را روی مجوزها فراهم کرده است. در SQL Server 2005، مایکروسافت مجوزهای بیشتری را در حوزه‌های مختلف اضافه کرده است. این حوزه‌ها

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

برای مجوزهایی که می‌توانند اعمال شوند، عبارتند از: سرور، پایگاه داده، طرح‌واره، شئ و اصل. ایده طراحی تحت مجوزهای بهبود یافته SQL Server 2005 اصل حداقل امنیت و توانایی کنترل دقیق مجوزهای منتسب را به DBA می‌دهند. مجوزهای سنجیده‌تر، خلاص شدن از شر نقش‌های ثابت موجود SQL Server نیست. تمام نقش‌های قدیمی هنوز وجود دارند و می‌توانند بدون هیچ مشکلی به همراه مجوزهای جدید وجود داشته باشند. سناریوی خاصی که این مجوزهای سنجیده‌تر برای سامان دادن به آن در نظر گرفته شده‌اند، حالت ممیزی است. در SQL Server 2000، شما باید عضوی از گروه Sysadmins بودید تا ممیزی را انجام می‌دادید. هرچند، عضویت در این گروه باعث ممکن ساختن کارهای دیگری می‌شد، قابلیت‌هایی بود که رسیدن به آن‌ها مشکل و بعید بود. برخی از مجوزهای جدید موجود در SQL Server 2005، امکان انجام اعمال ممیزی را بدون نیاز به این مسأله که کاربر بخشی از گروه Sysadmins باشد، میسر می‌سازند.

حالات مجوز پایه مشابه GRANT، DENY و REVOKE که در نگارش‌های قبلی SQL Server استفاده می‌شدند، هنوز اعمال می‌شوند. موردی که درباره این روش متفاوت است و SQL Server 2005 از مجوزها استفاده می‌کند، این است که مجوز مشابهی می‌تواند در چندین حوزه اعمال شود. مثلاً اگر مجوزی را برای حوزه پایگاه داده اعمال کنید، بر تمام اشیای موجود در پایگاه داده اعمال می‌شود. اگر مجوزی را برای سطح طرح‌واره به کار برید، تنها بر اشیای موجود در طرح‌واره اعمال می‌شود. به استثنای مجوز DENY، مجوز حوزه بالاتر همیشه استفاده می‌شود. هرچند، یک DENY در هر سطحی همیشه دارای تقدم است. جدول ۱-۱ برخی از مهم‌ترین مجوزهای جدید SQL Server 2005 را فهرست کرده است. مجوزهای سرور در دیدگاه sys.server_permissions وجود دارند، در حالی که مجوزهای پایگاه داده در دیدگاه sys.database_permissions وجود دارد.

(a) جدول ۱-۱ برخی از مجوزهای جدید در SQL Server 2005

مجوز	شرح
ALTER	توانایی تغییر خصوصیات یک شئ را واگذار می‌کند. هم‌چنین توانایی اجرای عبارات CREATE/DROP/ALTER را واگذار می‌کند.
ALTER ANY ‘X’	توانایی تغییر هر شیئی از نوع X را واگذار می‌کند. مثلاً، اگر TABLE را با X جایگزین کنید، این مجوز توانایی تغییر هر جدولی را در پایگاه داده واگذار می‌کند.
ALTER TRACE	توانایی انجام ممیزی و اجرای Profiler را واگذار می‌کند.
CONTROL	مجوزهای شبیه مالک اصلی را واگذار می‌کند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

SELECT	توانایی دستیابی به یک شئ را واگذار می‌کند. هم‌اکنون بر سطوح پایگاه داده و طرح‌واره اعمال می‌شود و نه تنها بر سطح شئ.
EXECUTE	توانایی اجرای یک رویه، اسمبلی یا تابع را واگذار می‌کند. هم‌اکنون بر سطوح پایگاه داده و طرح‌واره اعمال می‌شود و نه تنها سطح شئ.
IMPERSONATE	توانایی جعل هویت کاربری دیگر را به یک login یا کاربر واگذار می‌کند.
TAKE OWNERSHIP	توانایی فرض کردن مالکیت یک شئ را واگذار می‌کند.
VIEW DEFINITION	توانایی مشاهده فوق داده یک شئ را واگذار می‌کند.

ویژگی امنیتی جدید و مهم دیگر در SQL Server 2005، پشتیبانی از خط‌مشی‌های کلمه عبور است. این ویژگی تقویت خط‌مشی جدید از خط‌مشی‌های کلمه عبور محلی Windows پیروی کرده و به شما امکان پیاده‌سازی یک خط‌مشی امنیتی گستره جهانی مستحکم را می‌دهند، نه تنها برای سیستم‌های Windows Server شما، بلکه هم‌چنین برای سیستم‌های پایگاه داده SQL Server شما. SQL Server 2005 هم‌اکنون دارای قابلیت تقویت قدرت کلمه عبور، انقضای کلمه عبور و خط‌مشی‌های Lockout حساب. همان‌گونه که انتظار دارید، خط‌مشی قدرت کلمه عبور، کلمات عبور را مجبور می‌کند تا شامل یک پیچیدگی معین باشند. خط‌مشی انقضای کلمه عبور اطمینان می‌دهد که کلمات عبور منقضی شده و باید در یک فاصله معین باشند. خط‌مشی انقضای کلمه عبور اطمینان می‌دهد که کلمات عبور منقضی شده و باید در یک فاصله معین بازنشانی شوند و خط‌مشی lockout حساب اطمینان می‌دهد که یک حساب بعد از تعداد تلاش‌های بد در مورد کلمه عبور lockout شده است. تمام این خط‌مشی‌های جدید کلمه عبور، در Windows Server 2003 پشتیبانی می‌شوند. هرچند، در Windows 2000 Server تنها از خط‌مشی پیچیدگی کلمه عبور پشتیبانی می‌شود. با پیروی از دغدغه امنیتی مایکروسافت، تمام خط‌مشی‌ها در نصب پیش‌فرض فعال می‌شوند، ولی می‌توانید برطبق هر login، مجدداً پیکربندی کنید. SQL Server 2005 خط‌مشی‌های جدید کلمه عبور را در دیدگاه کاتالوگ sys.sql_logins ذخیره می‌کند.

۲۸،۰۴Section امنیت کاتالوگ

بهبود نهایی مربوط به امنیت که در این بخش به آن می‌پردازیم، امنیت کاتالوگ جدیدی است که توسط SQL Server 2005 فراهم شده است. جداول سیستمی که توسط SQL Server 2000 در پایگاه‌های داده مجزا و در پایگاه داده اصلی استفاده می‌شوند، هم‌اکنون به عنوان دیدگاه‌های کاتالوگ در SQL Server 2005 پیاده‌سازی می‌شوند. فوق داده سرور که در این دیدگاه‌ها عرضه می‌شود، به‌طور پیش‌فرض ایمن است و حداقل مجوزهای عمومی وجود دارد. دیدگاه‌های کاتالوگ SQL Server 2005،

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

امنیت سطح ردیف را به کار می‌گیرند، دستیابی به داده موجود در این دیدگاه‌ها را به تنها اشیایی که مال شماست یا دارای مجوزها هستند، محدود می‌کنند. طبیعتاً، sa استثنایی برای این مسأله است. حساب sa هنوز دارای دستیابی به تمام این اشیا در سرور است.

برای دادن امکان دستیابی به فوق داده به یک کاربر یا نقش، DBA باید از مجوز VIEW DEFINITION جدید استفاده کند. مجوز VIEW DEFINITION می‌تواند در حوزه‌های سرور، پایگاه داده، طرح‌واره و شئ به کار رود.

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

فصل دوم

راهبری پایگاه داده و ابزارهای برنامه‌نویسی

موارد بیشماری برای DBA در SQL Server 2005 تغییر یافته‌اند. زیر سیستم‌های جدید متعددی وجود دارند که باید مدیریت شوند و به علاوه، کل مجموعه ابزار مدیریت ارتقا یافته است. چندین ابزار راهبری که برای مدیریت نگارش‌های قبلی SQL Server استفاده می‌شوند، با ابزارهای جدیدی جایگزین شده‌اند. علاوه بر این، تعداد ابزارهای مدیریتی جدید نیز اضافه شده‌اند. در این فصل، شما را از طریق ویژگی‌های راهبری جدیدی راهنمایی خواهم کرد که مایکروسافت به این نسخه SQL Server 2005 اضافه کرده است.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

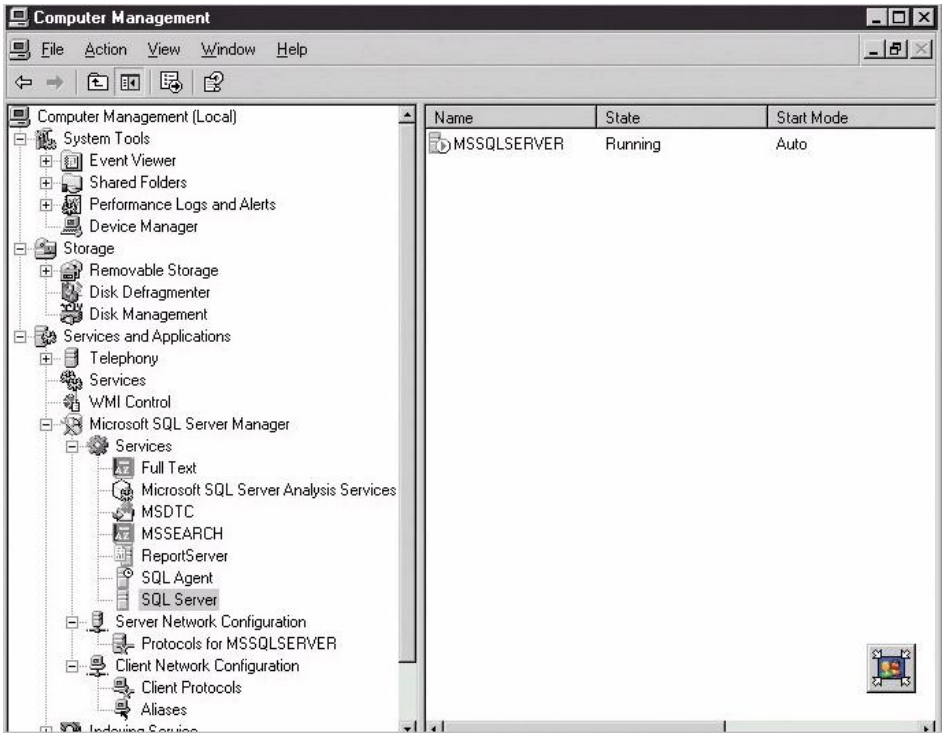
Article XXIX. ابزارهای مدیریتی و برنامه‌نویسی

SQL Server 2005 از یک مجموعه ابزارهای مدیریتی کاملاً جدید نسبت به نگارش‌های قبلی SQL Server استفاده می‌کند. در بخش بعدی این فصل، نگاه دقیق‌تری به این ابزارهای مدیریتی جدید در SQL Server 2005 خواهید داشت.

یکی از اولین مواردی که DBA با تجربه SQL Server درباره ابزارهای مدیریتی SQL Server 2005 جدید توجه خواهد داشت، این است که Server Manager وجود ندارد. در نگارش‌های قبلی SQL Server، Server Manager در System Tray قرار داشت. Server Manager یک نمایش گرافیکی از وضعیت سرویس‌های مختلف SQL Server فراهم می‌کرد، از جمله سرویس SQL Server، سرویس SQL Agent و سرویس Distributed Transaction Coordinator. هم‌چنین می‌توانید از Server Manager برای شروع و متوقف کردن این سرویس‌ها استفاده کنید. در حالی که این ابزار یک ابزار مفید بود، ولی سازگار با لوگوی Windows نبود و دلیل اصلی حذف آن از SQL Server 2005 توسط مایکروسافت، این مسأله بوده است. برای SQL Server 2005، می‌توانید سرویس‌های SQL Server را با استفاده از SQL Computer Manager جدید مشاهده و کنترل کنید. SQL Computer Manager یک برنامه افزودنی MMC است و می‌تواند با استفاده از گزینه My Computer | Manage مورد دستیابی قرار گیرد. می‌توانید SQL Computer Manager جدید را در شکل ۱-۲ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

شکل ۱-۲ SQL Computer Manager (b)

با استفاده از SQL Computer Manager، می‌توانید تمام سرویس‌های SQL Server را ببینید که در سیستم فهرست شده در زیر گره Services اصلی اجرا می‌شود. از پنجره Computer Management برای مدیریت این سرویس‌ها استفاده می‌شود، SQL Server، SQL Agent، ReportServer، MSSearch، MSDTC، Microsoft SQL Server Analysis Services و Full Text.

می‌توانید هر سرویسی را با ابتدا انتخاب آن در پنجره درخت نشان داده شده در سمت چپ صفحه نمایش کنترل کنید. این امر وضعیت سرویس را در پنجره جزییات نشان داده شده در سمت راست نمایش می‌دهد. برای مدیریت سرویس، روی سرویس در پنجره جزییات کلیک راست کرده و سپس گزینه‌ای را برای Start، Stop، Pause، Resume یا Restart برای سرویسی از منوی زمینه بازشو انتخاب کنید.

Section ۲۹.۰۲ SQL Server Management Studio

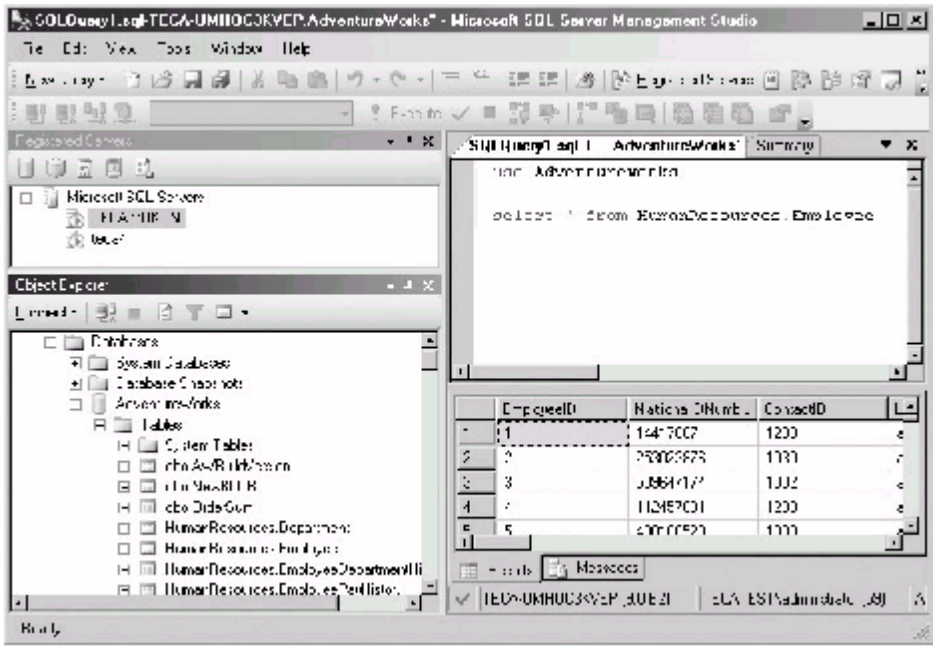
Server Manager تنها ابزار مدیریتی آشنایی نیست که در SQL Server 2005 تغییر کرده است. SQL Server Enterprise Manager که ابزار مدیریتی اصلی برای نگارش‌های ۷ و ۲۰۰۰ از SQL

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Server بود، با SQL Server Management Studio جدید جایگزین شده است. ضمناً، Query Analyzer که ابزار برنامه‌نویسی اصلی T-SQL در نگارش‌های ۷ و ۲۰۰۰ از SQL Server بود، نیز با SQL Server Management Studio جدید جایگزین شده است. SQL Server 2005 هم‌چنین تعدادی از ابزارهای راهبری دیگر برای DBA را فراهم کرده است، از جمله Administrative Console جدید، Database Tuning Advisor و یک Profiler جدید.

SQL Server Management Studio جدید، یک بخش آزاد از ابزارهای راهبری است که در نسخه‌های قبلی SQL Server فراهم شده بودند. SQL Server Management Studio جدید، بیشتر عملکرد مورد استفاده را که توسط Enterprise Manager و Query Analyzer در SQL Server فراهم شده بودند، مشارکت می‌دهد. SQL Server Management Studio با استفاده از گزینه منوی Start\Programs\Microsoft SQL Server\ SQL Server Management Studio مورد دستیابی قرار می‌گیرد. می‌توانید SQL Server Management Studio را در شکل ۲-۲ ببینید.



(a)

شکل ۲-۲ SQL Server Management Studio (b)

SQL Server Management Studio می‌تواند برای مدیریت سیستم‌های SQL Server 2005 و سیستم‌های SQL Server 2000 و SQL Server 7 استفاده شود. این ابزار نمی‌تواند در SQL Server 6.5 یا سیستم‌های قدیمی‌تر استفاده شود. می‌توانید از SQL Server 2000 Enterprise Manager و

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

SQL Server 7 قدیمی‌تر برای مدیریت یک سیستم SQL Server 2005 جدید استفاده کنید، ولی به دلیل برخی تغییرات معماری بین دو نسخه، این مسأله پشتیبانی یا توصیه نمی‌شود. و ابزارهای مدیریتی قدیمی‌تر نمی‌توانند به هیچ یک از ویژگی‌های جدیدی که به SQL Server 2005 اضافه شده‌اند، دستیابی داشته باشند. در حالی که می‌توانید مرور پایه جدول و پایگاه داده را انجام دهید، بیشتر اعمال دیگر موجب خطا خواهند شد. SQL Server Management Studio ابزار انتخاب برای مدیریت سیستم‌های SQL Server 2000 و SQL Server 2005 مرکب است.

SQL Server Management Studio با استفاده از نگارش خاصی از Visual Studio 2005 IDE ساخته شده است. شبیه Visual Studio 2005، SQL Server Management Studio از ایجاد پروژه‌های نرم‌افزاری پشتیبانی می‌کند و به شما اجازه نوشتن، ویرایش، اجرا و اشکال‌زدایی کد را می‌دهد. هم‌چنین برای کنترل نگارش کد منبع با Visual SourceSafe یکپارچه شده است.

هرچند، برخلاف Visual Studio 2005، به شما اجازه کامپایل VB.NET، C#، J# یا VC++ را نمی‌دهد. در عوض، SQL Server Management Studio با T-SQL، MDX، MX و XMLA کار می‌کند.

SQL Server Management Studio تعدادی از بهبودهای مهم را نسبت به ترکیب Query Analyzer و SQL Server Enterprise Manager ارایه می‌دهد. اول این که، تمام کادرهای محاوره‌ای که توسط SQL Server Management Studio نمایش داده می‌شوند، modal نیستند، بدین معنی که نباید قبل از این که بتوانید کار دیگری انجام دهید، به آن کادر محاوره‌ای پاسخ دهید. کادرهای محاوره‌ای که توسط SQL Server Enterprise Manager استفاده می‌شوند، همگی modal بودند و اگر کادری را باز می‌کردید، نمی‌توانستید کار دیگری انجام دهید، مگر این که این کادر محاوره‌ای را می‌بستید. کادرهای محاوره‌ای غیر modal جدید که توسط SQL Server Management Studio استفاده می‌شوند، این مشکل را برطرف کرده‌اند و انجام سایر وظایف مدیریتی را برای DBA ممکن ساخته‌اند، در حالی که یکی از این کادرهای محاوره‌ای در حال نمایش است.

علاوه بر این، SQL Server Management Studio با تعداد زیادی از اشیای پایگاه داده سروکار دارد که بهتر از SQL Server Enterprise Manager قدیمی است. در نگارش‌های قبلی SQL Server، SQL Server Enterprise Manager همیشه تمام اشیای پایگاه داده را هنگام اتصال به یک سرور رجیستر شده می‌شمارد. برای بیشتر حرفه‌های کوچک و متوسط، این یک مشکل نبود. هرچند، برای سازمان‌هایی با تعداد زیادی پایگاه داده حاوی هزاران شئ پایگاه داده، SQL Server Enterprise Manager ممکن بود زمان زیادی را صرف لیست کردن تمام اشیای پایگاه داده و خصوصیات آن‌ها کند (لزوماً، نمایش SQL Server Enterprise Manager غیرقابل استفاده بود تا وقتی که تمام اشیا لیست می‌شدند). در SQL Server 2005، SQL Server Management Studio بسیار هوشمندتر است و

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

اشیای پایگاه داده را طوری مدیریت می‌کند که اشیای پایگاه داده را شمارش نکند، تا وقتی که کاربر بخواهد نمایش یک آیتم پایگاه داده را در Object Browser باز کند. این امر به SQL Server Management Studio اجازه می‌دهد تا هنگام استفاده برای مدیریت پایگاه‌های داده بسیار بزرگ، مسئولیت‌پذیری بهتری را فراهم کند.

SQL Server Management Studio به عنوان میزبان برای اکثر ابزارهای برنامه‌نویسی و مدیریت SQL Server عمل می‌کند. پنجره Registered Servers به شما اجازه می‌دهد تا سیستم‌های SQL Server ای را انتخاب کنید که مدیریت خواهید کرد. پنجره Object Explorer به شما امکان می‌دهد تا کد منبع پایگاه داده خود را در کلکسیون‌های منطقی گروه‌بندی کنید. می‌توانید مطالب بیشتر درباره Registered Servers، Object Explorer، Query Editor و Solution Explorer را در بخش‌های بعدی این فصل بیابید. اطلاعات بیشتر درباره Report Designer در فصل ۸ ارایه شده است. DTS Designer نیز در فصل ۹ بررسی می‌شود.

سرورهای رجیستر شده

بسیار شبیه هنگام استفاده از SQL Server Enterprise Manager، باید سرورها را در SQL Server Management Studio رجیستر کنید، قبل از این که بتوانید از آن برای مدیریت آن‌ها استفاده کنید. از پنجره Registered Servers در SQL Server Management Studio برای رجیستر کردن سیستم‌های جدید SQL Server استفاده می‌شود (این پنجره در بخش سمت چپ بالای شکل ۲-۲ نشان داده شده است). هم‌چنین همانند SQL Server Enterprise Manager، پنجره Registered Servers در SQL Server Management Studio به شما امکان گروه‌بندی سرورهای عادی را با یکدیگر در گروه‌های سرور منطقی می‌دهد. برای رجیستر کردن یک سیستم SQL Server جدید در پنجره Registered Servers، در پنجره کلیک راست کرده و گزینه New\Server Registration را از منوی زمینه انتخاب کنید. به‌طور مشابه، برای ایجاد یک گروه سرور جدید، در پنجره Registered Servers کلیک راست کرده و گزینه New\Server Group را از منوی زمینه انتخاب کنید. یک ویژگی خوب در پنجره Registered Servers، توانایی صادر و وارد کردن تمام سرورهای رجیستر شده است. این امر به شما امکان می‌دهد تا پنجره‌های Registered Servers سایر SQL Server Management Studioها را سریعاً پر کنید، بدون این که مجبور به رجیستر کردن دستی تمام سرورهای مدیریت شده باشید.

پس از رجیستر شدن یک سرور، می‌توانید سرویس‌های آن را با کلیک راست روی سرور در پنجره Registered Servers مدیریت کنید. این امر موجب نمایش منوی زمینه‌ای می‌شود که به شما امکان اتصال به سرور و شروع، توقف، مکث، ادامه دادن یا شروع مجدد سرویس SQL Server را می‌دهد. می‌توانید با اشیای پایگاه داده یک سرور با دابل کلیک کردن روی نام سرور کار کنید، این امر

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

موجب اتصال شما به سرور SQL رجیستر شده و باز شدن خودکار پنجره Object Explorer خواهد شد.

Object Explorer

پنجره Object Explorer در SQL Server Management Studio به شما امکان می‌دهد تا اعمال مدیریتی مشابهی را انجام دهید که با استفاده از SQL Server Enterprise Manager امکان‌پذیر بودند. می‌توانید پنجره Object Explorer را در گوشه سمت چپ پایین شکل ۲-۲ ببینید. با استفاده از Object Explorer، می‌توانید:

- یک سرور را شروع و متوقف کنید
- خصوصیات یک سرور را پیکربندی کنید
- پایگاه‌های داده را ایجاد کنید
- پایگاه‌های داده را متصل کرده و از اتصال خارج کنید
- اشیای پایگاه داده از قبیل جداول، دیدگاه‌ها یا رویه‌های ذخیره شده را ایجاد کنید
- اسکریپت‌های تولید شئ T-SQL را تولید کنید
- Loginهای سرور را تنظیم کنید
- مجوزهای شئ پایگاه داده را مدیریت کنید
- کپی‌سازی را پیکربندی و مدیریت کنید
- سرورهای لینک شده را پیکربندی و مدیریت کنید
- فعالیت سرور را کنترل کنید
- ثبت وقایع سیستم را مشاهده کنید

برای کار کردن با اشیای پایگاه داده که در Object Explorer نمایش داده می‌شوند، روی شئ مناسب در درخت Object Explorer کلیک راست کنید تا منوی زمینه شئ نمایش داده شود. منوی زمینه، مجموعه منحصر به فردی از گزینه‌ها را برای هر یک از اشیای مختلف پایگاه داده فراهم می‌کند. برای نمونه، منوی زمینه پایگاه داده به شما اجازه حذف، تغییر نام، متصل کردن، از حالت اتصال خارج کردن، ممانعت کردن، پشتیبان‌گیری و بازیابی پایگاه‌های داده را می‌دهد، در حالی که منوی زمینه جدول به شما اجازه ایجاد یک جدول جدید، اصلاح یک جدول، باز کردن وابستگی‌های دیدگاه جدول و تنظیم مجوزها را می‌دهد.

ویژگی جالب دیگر Object Explorer جدید، توانایی تولید آسان اسکریپت‌ها برای ایجاد هر یک از اشیا پایگاه داده است که در لیست Object Explorer نشان داده شده‌اند. Script Wizard می‌تواند برای مستند کردن یک طرح‌واره پایگاه داده استفاده شود تا پشتیبانی از یک پایگاه داده را ایجاد کند یا

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

یک کپی آزمایشی از یک پایگاه داده یا اشیای پایگاه داده منتخب ایجاد نماید. می‌توانید ایجاد یک اسکریپت تکی برای تمام اشیای منتخب یا برای ایجاد چند اسکریپت (یکی برای هر شئ) را داشته باشید. اسکریپت‌های T-SQL می‌توانند خروجی را به یک فایل، کلیپ مجدد یا به SQL Server Management Studio Query Editor ارسال کند.

Query Editor

Query Editor جایگزینی برای Query Analyzer است و به شما امکان نوشتن و اجرای اسکریپت‌های T-SQL را می‌دهد. می‌توانید Query Editor را در بخش میانی بالای شکل ۲-۲ ببینید. Query Editor را از SQL Server Management Studio با انتخاب گزینه New Query در صفحه Start یا انتخاب گزینه File | New | SQL Server Query | SQL Server Query اجرا کنید. Query Editor از نوشتن پرس‌وجوها در T-SQL، MDX، DMX، XMLA و SQL Mobile Queries پشتیبانی می‌کند. برخلاف Query Analyzer که همیشه در حالت متصل کار می‌کند، Query Editor جدید دارای گزینه کار کردن با حالت متصل یا غیرمتصل از سرور است. به‌طور پیش‌فرض، به‌طور خودکار و به محض این که بخواهید پرس‌وجوی جدیدی را ایجاد کنید، به سرور متصل می‌شود.

Query Editor شبیه همتای Visual Studio 2005 از کلمات کلیدی کد رنگی پشتیبانی می‌کند، خطاهای دستوری را به صورت ویژوال نشان می‌دهد و به برنامه‌نویس امکان اجرا و اشکال‌زدایی کد را می‌دهد. علاوه بر این، برخلاف Query Analyzer قدیمی، Query Editor از مفهوم پروژه‌ها پشتیبانی می‌کند که گروه‌هایی از فایل‌های مرتبط می‌توانند برای تشکیل یک راه‌حل، با یکدیگر گروه‌بندی شوند. Query Editor جدید پشتیبانی کاملی برای کنترل منبع با استفاده از Visual SourceSafe ارایه می‌دهد. Query Editor هم‌چنین عناصری را اضافه کرده است که در Query Analyzer وجود داشتند و قادر به نمایش نتایج پرس‌وجو در یک مشبک یا به صورت متن است و قادر به ارایه گرافیکی طرح‌های اجرای یک پرس‌وجو است. هم‌چنین گزینه‌ای برای ذخیره اسکریپت‌های T-SQL با استفاده از کنترل نگارش SourceSafe وجود دارد. کنترل نگارش، برنامه‌نویسی گروهی را با ممانعت از تغییر هم‌زمان یک ماژول توسط چند برنامه‌نویس، تسهیل می‌کند. کد منبع باید مخزن کد را بررسی کند، قبل از این که بتواند اصلاح شود و آن‌گاه مجدداً بررسی کند و یک محل مرکزی برای ذخیره کد پایگاه داده به شما بدهد. استفاده از کنترل نگارش با اسکریپت‌های ایجاد پایگاه داده T-SQL، روش ارزنده‌ای را برای جداسازی کد منبع مربوط به هر نسخه از طرح‌واره پایگاه داده فراهم می‌کند. این امر هم‌چنین می‌تواند به عنوان پایه‌ای برای مقایسه طرح‌واره یک پایگاه داده توزیع شده با طرح‌واره مورد انتظاری که با استفاده از کنترل نگارش ذخیره شده است، عمل کند. علاوه بر این،

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

همان‌گونه که بعداً در ادامه فصل خواهید دید، Query Editor هم‌چنین دارای قابلیتی برای ارایه گرافیکی طرح اجرای یک پرس‌وجوست.

پنجره Results

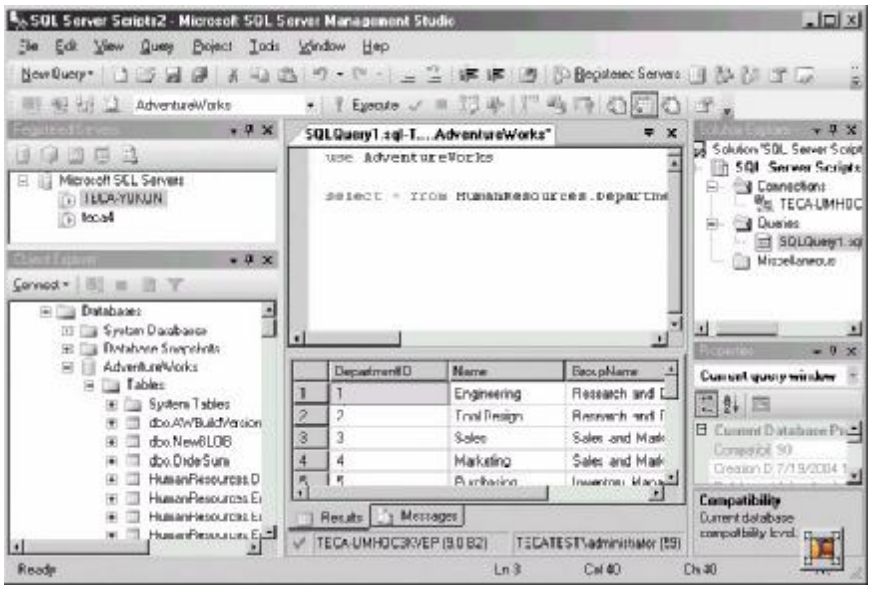
پنجره SQL Server Management Studio Results نتایج پرس‌وجوهایی را نمایش می‌دهد که در Query Editor اجرا می‌شوند. می‌توانید پنجره Results را در بخش سمت راست پایین شکل ۲-۲ ببینید. پنجره Results می‌تواند برای نمایش نتایج پرس‌وجو به شکل متنی یا در یک مشبک تنظیم شود.

Solution Explorer

ابزار مدیریتی مهم دیگری که به عنوان بخشی از SQL Server Management Studio فراهم شده است، Solution Explorer است. می‌توانید Solution Explorer را در گوشه سمت راست بالای شکل ۳-۲ ببینید. Solution Explorer برای فراهم کردن یک دیدگاه درختی و سلسله مراتبی از فایل‌ها و پروژه‌های مختلف در یک راه‌حل استفاده می‌شود. آیتم بالای لیست شده در Solution Explorer، نام راه‌حل SQL Server Management Studio است. به‌طور پیش‌فرض، این نام Solution1 است، ولی می‌توانید آن را با انتخاب راه‌حل و سپس تغییر نام آن در پنجره Properties با هر آن‌چه که می‌خواهید، تغییر دهید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(c)

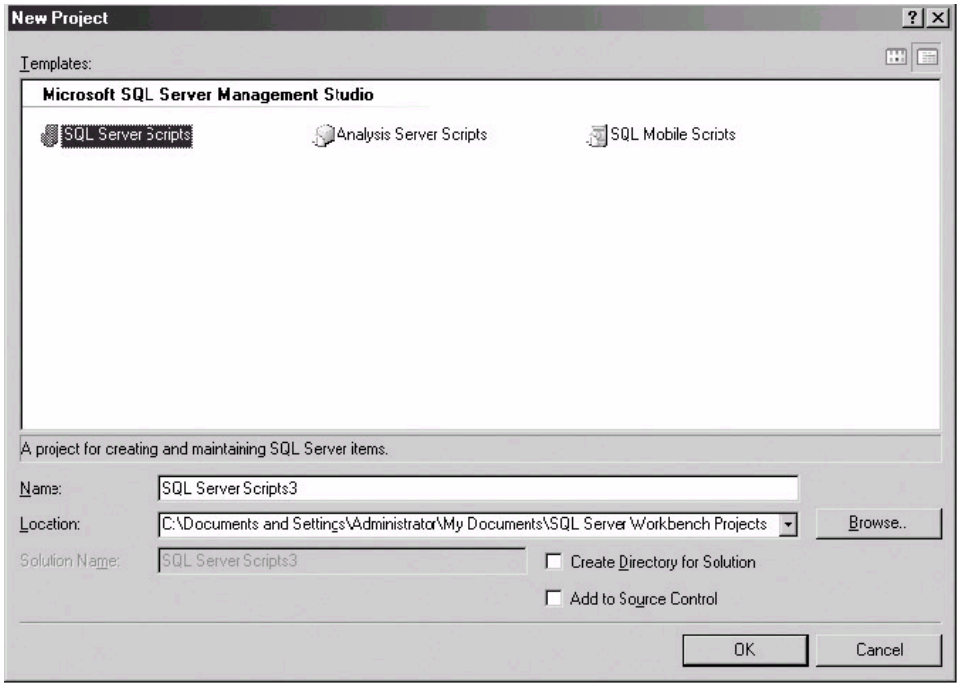
Solution Explorer – SQL Server Management Studio شکل ۳-۲ **(d)**

تحت این راه‌حل، می‌توانید یک یا چند آیتم پروژه یا یک یا چند فایل داشته باشید. فایل‌هایی که در Solution Explorer لیست می‌شوند، می‌توانند به یک پروژه مرتبط شوند یا می‌توانند به خود راه‌حل SQL Server Management Studio مرتبط شوند، بدون این که پروژه واسطه‌ای داشته باشند. فایل‌ها می‌توانند شامل هر نوع فایلی باشند که بتوانند با استفاده از SQL Server Management Studio اصلاح شوند، از جمله پرس‌وجوهای T-SQL، پرس‌وجوهای Analysis Server و پرس‌وجوهای XMLA.

SQL Server Management Studio Solution Explorer از تعدادی از انواع پروژه مختلف پشتیبانی می‌کند، از جمله SQL Server Scripts، Analysis Server Scripts و SQL Mobile Scripts. می‌توانید کادر محاوره‌ای New Project در SQL Server Management Studio را در شکل ۴-۲ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(e)

شکل ۴-۲ New Project – SQL Server Management Studio (f)

SQL Scripts

پروژه‌های SQL Scripts برای گروه‌بندی اسکریپت‌های T-SQL و اتصالات SQL Server مرتبط با یکدیگر استفاده می‌شوند. یک کاربرد متداول برای این نوع پروژه، گروه‌بندی تمام اسکریپت‌های DDL[1] پایگاه داده با یکدیگر است.

Analysis Server Scripts

همانند SQL Scripts که برای فایل‌های مبتنی بر T-SQL استفاده می‌شود، پروژه‌های Analysis Server Scripts در نظر گرفته شده‌اند تا حاوی اتصالات Analysis Server و اسکریپت‌های MDX و XMLA باشند. مجدداً، یک روش ممکن است بهره بردن از این نوع پروژه باشد تا پروژه حاوی تمام اسکریپت‌هایی باشد که انبار داده شما را ایجاد می‌کند و پروژه دیگری ممکن است حاوی اسکریپت‌هایی برای بارگذاری انبار داده باشد.

SQL Mobile Scripts

1- Data Definition Language

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

پروژه‌های SQL Mobile Scripts برای گروه‌بندی اتصالات و پرس‌وجوها با یکدیگر برای یک پایگاه داده SQL Server CE استفاده می‌شود. برای یک پروژه SQL Server CE، یک شئ اتصال، اتصالی به پایگاه داده CE را نشان می‌دهد.

Properties

پنجره Properties که در گوشه سمت راست پایین شکل ۳-۲ نشان داده شده است، در زمان طراحی برای تنظیم خصوصیات اشیای منتخب در Solution Explorer استفاده می‌شود. می‌توانید از پنجره Properties برای تعدادی از وظایف مختلف استفاده کنید، از جمله تنظیم نام راه‌حل‌ها، پروژه‌ها و فایل‌های خود و کنترل تایم‌اوت برای بسته‌های DTS. یک ویژگی خوب درباره پنجره Properties که بلافاصله مشخص نمی‌شود، این واقعیت است که می‌توانید از آن برای تنظیم هم‌زمان خصوصیاتی برای چند آیتم استفاده کنید. برای انجام این کار، چند آیتم را در پنجره Solution Explorer با نگه داشتن کلید CTRL و سپس کلیک کردن آیتم‌ها انتخاب کنید. سپس هنگامی که مقداری را در پنجره Properties تغییر دهید، برای تمام آیتم‌های منتخب تغییر خواهد کرد. همان‌گونه که ممکن است انتظار داشته باشید، آیتم‌های منتخب در پنجره Solution Explorer باید خصوصیات مشترکی را برای این کار به اشتراک بگذارند.

راهنمای پویا

شبیه Visual Studio 2005، SQL Server Management Studio نیز یک پنجره راهنمای پویای جدید را فراهم کرده است. محتویات پنجره راهنمای پویا به‌طور خودکار برطبق محل مکان‌نمای جاری یا اشاره‌گر ماوس در SQL Server Management Studio تغییر می‌کند.

کنترل منبع

کنترل منبع یکپارچه، ویژگی جدید دیگری است که توسط SQL Server Management Studio فراهم شده است. کنترل منبع، برنامه‌نویسی گروهی را با فراهم کردن یک مکانیزم کنترل مرکزی که دستیابی به فایل‌های منبع را مدیریت می‌کند، تسهیل می‌بخشد. برای استفاده از ویژگی کنترل منبع SQL Server Management Studio، باید Visual SourceSafe را قبلاً نصب کرده باشید که همراه با محصول Microsoft Visual Studio 2005 است. ویژگی کنترل منبع SQL Server Management Studio، فرآیندهای check-in و check-out را برای اطمینان از این مسأله فراهم کرده است که دو برنامه‌نویس به‌طور هم‌زمان نتوانند فایل منبع یکسانی را تغییر دهند. سیستم کنترل منبع هم‌چنین قادر به دستیابی به نگارش‌های گذشته هر پروژه است. می‌توانید فایل‌ها و پروژه‌های راه‌حل را

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

با استفاده از گزینه File | Source Control از منوی SQL Server Management Studio به سیستم کنترل منبع Visual SourceSafe اضافه کنید.

پنجره Current Activity

SQL Server Management Studio جدید SQL Server 2005 همچنین یک ویژگی کنترل فرآیند یکپارچه به نام پنجره Current Activity را فراهم کرده است. می‌توانید Current Activity Monitor را با انتخاب گره Database | Management | Activity Monitor از Object Explorer اجرا کنید. پنجره Current Activity مروری از فرآیندهای در حال اجرا در سیستم SQL Server را به شما می‌دهد. این پنجره این موارد را نشان می‌دهد:

- اطلاعات فرآیند
- فرآیندهای بلوکه شده
- اتصالات کاربر
- قفل‌ها
- قفل‌های کاربر

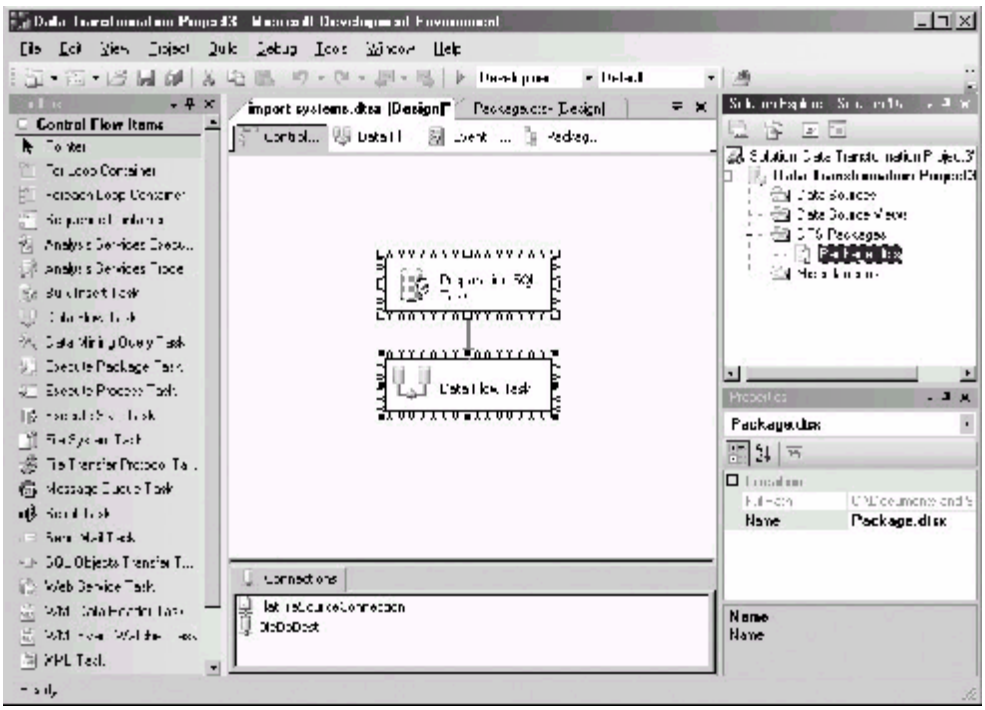
Section ۲۹.۰۳ Business Intelligence Development Studio

شبیه SQL Server Management Studio که برای نوشتن پروژه‌های پایگاه داده رابطه‌ای استفاده می‌شود، Business Intelligence Development Studio جدید برای ایجاد راه‌حل‌های Business Intelligence استفاده می‌شود. برخلاف SQL Server Management Studio، Business Intelligence Development Studio واقعاً به عنوان یک ابزار راهبری طراحی نشده است. از Business Intelligence Development Studio برای کار کردن با پروژه‌های Analysis Server برای نوشتن و توزیع گزارشات Registered Servers و برای طراحی بسته‌های Data Transformation Services استفاده می‌شود.

با توجه به این که Business Intelligence Development Studio دارای هدفی متفاوت از SQL Server Management Studio است، همچنین دارای ظاهر و احساس متفاوتی است. علاوه بر این، در حالی که SQL Server Management Studio به‌طور پیش‌فرض از یک حالت متصل استفاده می‌کند، Business Intelligence Development Studio در یک حالت غیرمتصل شروع می‌شود. Business Intelligence Development Studio با استفاده از گزینه منوی Start | Programs | Microsoft SQL Server | Business Intelligence Development Studio مورد دستیابی قرار می‌گیرد. می‌توانید تصویری از Business Intelligence Development Studio را در شکل ۵-۲ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

شکل ۵-۲ Business Intelligence Development Studio (b)

Business Intelligence Development Studio شبیه SQL Server Management Studio در بالای Visual Studio 2005 IDE ساخته شده است و محیط برنامه‌نویسی راه‌حل‌گرایی را فراهم می‌کند که یک یا چند پروژه در یک راه‌حل وجود دارند. هر پروژه شامل انواعی از اشیاست که برای آن پروژه مناسب هستند. مثلاً، یک پروژه Reporting Services حاوی تعاریف گزارش است، در حالی که یک پروژه DTS حاوی اشیای بسته DTS است. شبیه SQL Server Management Studio، Business Intelligence Development Studio به شما اجازه نمی‌دهد تا VB.NET، C#، J# یا VC++ را کامپایل کنید. در عوض، Business Intelligence Development Studio برای کار کردن با پروژه‌های BI شبیه DTS و Reporting Services طراحی شده است. Business Intelligence Development Studio هم‌چنین با Visual Object Explorer برای کنترل نگارش کد منبع یکپارچه شده است.

جعبه ابزار

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

پنجره جعبه ابزار در Business Intelligence Development Studio در سمت چپ صفحه نمایش در شکل ۵-۲ نشان داده شده است. جعبه ابزار توسط DTS Designer و Reporting Services Designer برای کشیدن و انداختن اجزا در سطوح طراحی مرتبط آن‌ها استفاده می‌شود.

Solution Explorer

شبیه SQL Server Development Studio، Business Intelligence Development Studio نیز دارای یک پنجره Solution Explorer است. می‌توانید Solution Explorer را در گوشه سمت راست بالای صفحه نشان داده شده در شکل ۵-۲ ببینید. Solution Explorer یک نمای درختی و سلسله مراتبی از فایل‌ها و پروژه‌های مختلف فراهم می‌کند که یک راه‌حل Business Intelligence Development Studio را تشکیل می‌دهند. آیتم بالا در سلسله مرتبه Solution Explorer نام راه‌حل است. تحت این راه‌حل، می‌توانید یک یا چند آیتم پروژه و یا چند آیتم داشته باشید. Business Intelligence Development Studio Solution Explorer این قالب‌های پروژه را فراهم می‌کند: Analysis Server Object، Data Transformation Project، Import Analysis Services 9.0، Database، Report Project و Report Project Wizard. شبیه SQL Server Management Studio، راه‌حل‌های Business Intelligence Development Studio محدود به یک پروژه نیستند. می‌توانید راه‌حل‌های چند پروژه‌ای را ایجاد کنید که هر یک از انواع پروژه‌های پشتیبانی شده را تشکیل دهند. می‌توانید کادر محاوره‌ای Business Intelligence Development Studio New Project را در شکل ۶-۲ ببینید.

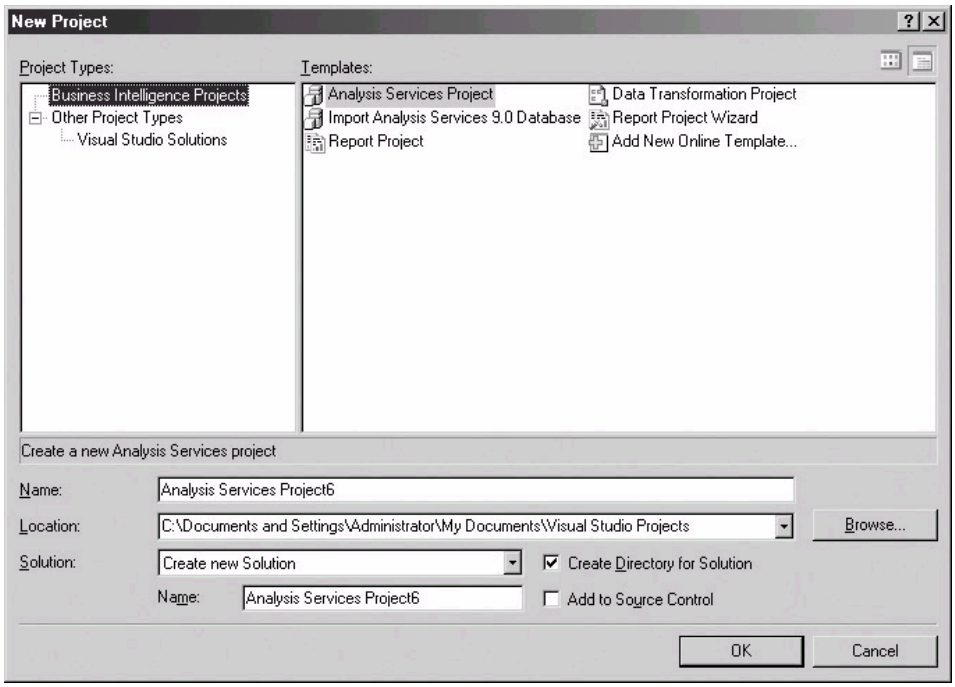
Analysis Services Project

پروژه‌های Analysis Services حاوی تعاریفی برای اشیا در یک پایگاه داده Analysis Services هستند. این‌ها شامل تعاریفی برای منابع داده، دیدگاه‌های منبع داده، مکعب‌ها، ابعاد، نقش‌های مدل‌های mining و اسمبلی‌ها هستند. یکی از ابزارهای اصلی در پروژه‌های Analysis Services، Cube Designer است.

Cube Designer یک ابزار ویژوال برای ساخت مکعب‌های OLAP است. این ابزار با دابل کلیک کردن روی گره Cube اجرا می‌شود که تحت یک پروژه Analysis Services نشان داده می‌شود یا با کلیک راست روی گره Cube و انتخاب View Designer اجرا می‌گردد. اطلاعات مفصل‌تر درباره Cube Designer در فصل ۱۰ ارایه می‌شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(c)

شکل ۶-۲ New Project – Business Intelligence Development Studio (d)

Data Transformation Project

تعریف پروژه برای پروژه‌های Data Transformation به شما امکان ایجاد اشیای مورد استفاده در یک راه‌حل DTS را می‌دهد. این تعاریف شامل ایجاد منابع داده، دیدگاه‌های منبع داده و بسته‌های DTS هستند. DTS Designer که در وسط صفحه شکل ۵-۲ نشان داده شده است، به شما امکان می‌دهد تا بسته‌های DTS را با انتخاب اجزای DTS از نوار ابزار و کشیدن و انداختن آن‌ها در سطح طراحی DTS به‌طور ویژوال ایجاد کنید. با استفاده از DTS Designer، می‌توانید منابع داده را مشخص کنید، انواع تبدیلات مورد استفاده را انتخاب کنید و جریان داده بسته DTS را تنظیم نمایید. در فصل ۹، می‌توانید مطالب بیشتری درباره DTS بیابید.

Import Analysis Services 9.0 Database

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

نوع پروژه Import Analysis Services 9.0 به شما امکان می‌دهد تا پروژه SQL Server 2005 Analysis Services جدیدی را با وارد کردن تعاریف برای یک SQL Server 2000 Analysis Services موجود یا پایگاه داده SQL Server 7 OLAP Server ایجاد کنید.

Report Project

از قالب Report Project برای ایجاد پروژه‌های Reporting Services استفاده می‌شود. این قالب پروژه Reporting Services Designer را اجرا می‌کند که منابع داده را در آن جا انتخاب کرده و گزارش‌ها را به صورت ویژوال طراحی می‌کنید.

Reporting Services Designer به شما امکان می‌دهد تا گزارشات را به‌طور ویژوال طراحی کرده و توزیع آن‌ها را کنترل کنید. با استفاده از Reporting Services Designer، یک DataSet ایجاد کرده و سپس از سطح طراحی ویژوال به همراه کنترل‌هایی از جعبه ابزار Report Items برای طراحی گزارشات استفاده کنید. این گزارشات می‌توانند در یک پایگاه داده SQL Server 2005 و هر منبع پایگاه داده سازگار با ODBC و OLE DB اجرا کنید. Reporting Services و Report Designer در فصل ۸ به‌طور مفصل بررسی می‌شود.

Report Project Wizard

قالب پروژه Report Project Wizard، Report Wizard را اجرا می‌کند که شما را از طریق مجموعه‌ای از صفحات سهل‌الاستفاده راهنمایی می‌کند که به شما امکان تعریف یک گزارش پایه را می‌دهد. Query Builder یکپارچه که در Report Wizard قرار دارد، به شما کمک می‌کند تا پرس‌وجوهای SQL خود را بسازید. هنگامی که گزارشی را با استفاده از Report Wizard ایجاد می‌کنید، می‌توانید برگردید و آن را با استفاده از Report Designer تنظیم کنید. Report Wizard در فصل ۸ بیشتر بررسی می‌شود.

Dynamic Help

شبیه SQL Server Management Studio، Business Intelligence Development Studio نیز دارای یک پنجره Dynamic Help جدید است. محتویات راهنما که در پنجره Dynamic Help نشان داده می‌شوند، برطبق محل مکان‌نمای جاری یا اشاره‌گر ماوس در Business Intelligence Development Studio تغییر می‌کند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

کنترل منبع

Business Intelligence Development Studio هم‌چنین کنترل منبع کاملاً یکپارچه‌ای را با استفاده از Visual SourceSafe فراهم کرده است. کنترل منبع با فراهم کردن یک مکانیزم کنترل مرکزی که دستیابی به فایل‌های منبع را مدیریت می‌کند، برنامه‌نویسی گروهی را تسهیل می‌بخشد. برای استفاده از ویژگی کنترل منبع با Business Intelligence Development Studio، باید Visual SourceSafe را نصب کرده باشید که با Visual Studio 2005 توزیع می‌شود. کنترل منبع فرآیندهای Check-in و Check-out را برای اطمینان از این مسأله فراهم می‌کند که دو برنامه‌نویس نمی‌توانند به‌طور هم‌زمان فایل منبع یکسانی را تغییر دهند و هم‌چنین قادر به دستیابی به نگارش‌های قبلی هر پروژه هستند. می‌توانید راه‌حل‌ها، پروژه‌ها و فایل‌ها را با استفاده از گزینه File | Source Control از Business Intelligence Development Studio به سیستم کنترل منبع اضافه کنید.

Section ۲۹.۰۴ زمان استفاده از SQL Server Management Studio و Business Intelligence Development Studio

هرچند SQL Server Management Studio و Business Intelligence Development Studio در ابتدا شبیه به نظر می‌رسند، برای انجام اهداف متفاوت طراحی شده‌اند. این دو برطبق Visual Studio 2005 IDE هستند و با توجه به این که SQL Server 2005 به همراه یک سرور پایگاه داده رابطه‌ای SQL Server و یک سرور Analysis Services OLAP است، شاید تصور کنید که SQL Server Management Studio برای مدیریت SQL Server استفاده می‌شود، در حالی که از Business Intelligence Development Studio برای مدیریت Analysis Services استفاده می‌گردد. SQL Server Management Studio برای مدیریت SQL Server و Analysis Services و برای نوشتن پرس‌وجوهای T-SQL و MDX استفاده می‌شود. SQL Server Management Studio می‌تواند به نمونه‌هایی از SQL Server، Analysis Services و Reporting Services متصل شود. می‌توانید از آن برای شروع و متوقف کردن این سرویس‌ها و مدیریت آن‌ها با استفاده از SQL Server Management Studio GUI یا با اجرای اسکریپت‌ها استفاده کنید.

Business Intelligence Development Studio یک ابزار راهبری نیست. در عوض، برای دادن این امکان به شما طراحی شده است تا راه‌حل‌های هوشمند تجاری را بنویسید و توزیع کنید. Business Intelligence Development Studio می‌تواند برای طراحی بسته‌های DTS و گزارشات Reporting Services و ایجاد و اجرای پرس‌وجوهای MXD و XMLA استفاده شود، ولی برای مدیریت Analysis Services استفاده نمی‌شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۲۹،۰۵ Sqlcmd

ابزار مدیریتی جدید مهم دیگر در SQL Server 2005، برنامه سودمند Sqlcmd جدید است. این برنامه لزوماً جایگزینی برای برنامه‌های osql قدیمی و isql در نسخه‌های قبلی SQL Server هستند. برنامه isql قدیمی از کتابخانه SQL Server 2005 DB کنار گذاشته شده برای اتصال به SQL Server استفاده می‌کرد، در حالی که برنامه osql از ODBC استفاده می‌کند. برنامه Sqlcmd جدید با استفاده از OLE DB به SQL Server متصل می‌شود. برای سازگاری با قبل، برنامه osql هنوز همراه با SQL Server 2005 است. هرچند، isql از نسخه ۲۰۰۵ کنار گذاشته شده است. برنامه sqlcmd شبیه این ابزارها از اعلان فرمان اجرا شده و به شما امکان وارد کردن و اجرای عبارات T-SQL، رویه‌های ذخیره شده و دسته‌های T-SQL را می‌دهد.

ویژگی مهمی که برنامه Sqlcmd علاوه بر توانایی اجرای فرامین دارد، این واقعیت است که می‌تواند با یک DAC[1] به پایگاه داده متصل شود. DAC به شما اجازه می‌دهد تا در اولویت بالاتری نسبت به فرآیند SQL Server دیگر متصل شده و اجرا کنید و به شما امکان می‌دهد هر فرآیند خارج از کنترل را خاتمه دهید. برای استفاده از DAC، باید برنامه Sqlcmd را با استفاده از سوییچ A– اجرا کنید. می‌توانید اطلاعات بیشتر درباره قابلیت Sqlcmd DAC را در فصل ۴ بیابید. علاوه بر سوییچ A– ، برنامه Sqlcmd از تعدادی سوییچ خط فرمان مفید دیگر پشتیبانی می‌کند. مثلاً، می‌توانید از سوییچ L– برای فهرست کردن تمام سیستم‌های SQL Server رجیستر شده استفاده کنید، به این صورت:

```
Sqlcmd –L
```

سوییچ مفید دیگر، سوییچ p– است که آمارهای کارآیی را برای مجموعه نتیجه به خروجی ارسال می‌کنند. می‌توانید لیست کامل سوییچ‌های خط فرمان پشتیبانی شده را با وارد کردن Sqlcmd/? در اعلان فرمان به دست آورید.

اسکریپت‌نویسی Sqlcmd

علاوه بر پشتیبانی از اجرای عبارات T-SQL استاندارد، برنامه Sqlcmd هم‌چنین از تعدادی توسعه اسکریپت‌نویسی پشتیبانی می‌کند که به شما امکان می‌دهد تا کنترل جریان و متغیرها را در اسکریپت‌های خود داشته باشید. می‌توانید متغیرهای اسکریپت‌نویسی را با استفاده از سوییچ v– به‌طور ضمنی تعریف کنید یا می‌توانید آن‌ها را با استفاده از فرمان setvar پوسته فرمان تنظیم کنید. این مثال

1- Dedicated Administrative Connection

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

نشان می‌دهد که چگونه می‌توانید سوییچ‌های خط فرمان Sqlcmd را با متغیرهای مورد استفاده در اسکریپت‌ها ترکیب کنید:

```
sqlcmd -S MySQLServer –d AdventureWorks -v
                    CUSTOMERID="ALFKI" -i MyScript.sql
```

توجه داشته باشید که برنامه Sqlcmd به‌طور پیش‌فرض از یک اتصال معتبر استفاده می‌کند.

این مثال از سوییچ S– برای مشخص کردن نام سیستم SQL Server استفاده می‌کند. سوییچ d– پایگاه داده‌ای را مشخص می‌کند که به آن متصل خواهید شد. سوییچ v– برای تعریف متغیری به نام VENDORNAME و تأمین آن متغیر با مقدار International استفاده می‌شود. سوییچ I– برای گفتن این مطلب به برنامه Sqlcmd استفاده می‌شود که فرمان T-SQL فایل MyScript.Sql را حاصل خواهد کرد. می‌توانید محتویات فایل MyScript.Sql را در این لیست ببینید:

```
select * from Purchasing.Vendor where Name = '$(VENDORNAME)'
```

علاوه بر فراهم کردن متغیرهای تعریف شده کاربر، مایکروسافت هم‌چنین مجموعه‌ای از متغیرهای از پیش‌تعریف شده را فراهم کرده است که می‌تواند توسط اسکریپت‌های Sqlcmd استفاده شود. جدول ۱-۲ متغیرهای توسعه یافته‌ای را فهرست می‌کند که مایکروسافت به برنامه Sqlcmd اضافه کرده است و سوییچ‌های خط فرمان که می‌توانند برای تأمین مقادیری برای این متغیرها استفاده شوند.

(a) جدول ۱-۲ متغیرهای توسعه یافته Sqlcmd

متغیر	سوییچ خط فرمان
SQLCMDUSER	–U
SQLCMDPASSWORD	–P
SQLCMDSERVER	–S
SQLCMDWORKSTATION	–H
SQLCMDDBNAME	–d
SQLCMDLOGINTIMEOUT	–l
SQLCMDSTATTIMEOUT	–t
SQLCMDHEADERS	–h
SQLCMDCOLSEP	–s
SQLCMDCOLWIDTH	–w
SQLCMDPACKETSIZE	–a
SQLCMDERRORLEVEL	–m

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

علاوه بر فراهم کردن مجموعه‌ای از متغیرهای از پیش‌تعریف شده، برنامه Sqlcmd، لیستی از فرمان‌های توسعه یافته را فراهم کرده است. فرامین برنامه‌های Sqlcmd توسعه یافته در جدول ۲-۲ فهرست شده‌اند.

(b) جدول ۲-۲ برنامه‌های سودمند توسعه یافته Sqlcmd

فرمان	شرح
[:]GO [count]	انتهای یک دسته را مشخص کرده و عبارات کش شده را اجرا می‌کند. افزودن یک مقدار count اختیاری موجب اجرای عبارات به تعداد دفعات مشخص شده می‌شود.
[:]RESET	کش عبارت را پاک می‌کند.
[:]ED	ویرایش بعدی را برای کش عبارت جاری شروع می‌کند.
[:]!!	فرامین سیستم عامل را اجرا می‌کند.
[:]QUIT	برنامه Sqlcmd را خاتمه می‌دهد.
[:]EXIT (results)	از مقدار یک مجموعه نتیجه به عنوان یک مقدار برگشتی استفاده می‌کند.
[:]r <filename>	شامل عبارات Sqlcmd اضافی از فایل مشخص شده است.
:ServerList	سیستم‌های SQL Server پیکربندی شده را فهرست می‌کند.
:List	محتویات کش عبارت را فهرست می‌کند.
:Error <filename>	خروجی خطا را به فایل مشخص شده هدایت می‌کند.
:Out	نتایج پرس‌وجو را به فایل مشخص شده هدایت می‌کند.
:Perftrace <filename>	آمارهای کارآیی را به فایل مشخص شده هدایت می‌کند.
:Connect [timeout]	به یک نمونه SQL Server متصل می‌شود.
:On Error [exit |retry | ignore]	عملی را برای انجام شدن مشخص می‌کند، هنگامی که با خطایی مواجه می‌شوید.
:XML [ON | OFF]	مشخص می‌کند آیا نتایج XML به عنوان یک جریان متوالی به خروجی ارسال خواهند شد.

لیست زیر ایده‌ای از نحوه استفاده فرامین توسعه یافته برنامه‌های Sqlcmd به شما می‌دهد. این مثال به سرور متصل شده و سپس از فرمان EXIT: برای برگرداندن تعداد ردیف‌های جدول Customers استفاده می‌کند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
C:\>sqlcmd
1> use adventureworks
2> :EXIT(select count(*) from sales.customer)
Changed database context to 'AdventureWorks'.
-----------
        91
(1 rows affected)
```

Article XXX. ابزارهای کارآیی

علاوه بر مجموعه کاملی از ابزارهای مدیریتی جدید، SQL Server 2005 هم‌چنین مجموعه بهنگام شده‌ای از ابزارها را برای تحلیل و بهبود کارآیی سرور فراهم کرده است. در پنج سالی که از آخرین عرضه SQL Server می‌گذرد، مایکروسافت قادر به افزودن عملکرد زیادی به ابزارهای مدیریتی شده است که در SQL Server فراهم شده‌اند و همان‌گونه که در این بخش خواهید دید، ابزارهای کارآیی استثنایی نیستند. در این بخش، نگاهی به چهار مورد از مهم‌ترین ابزارهای مدیریت کارآیی در SQL Server 2005 خواهیم داشت: Showplan گرافیکی SQL Server Management Studio Editor، Database Tuning Advisor و آخری Profiler.

Section ۳۰،۰۱: SQL Server Management Studio Editor

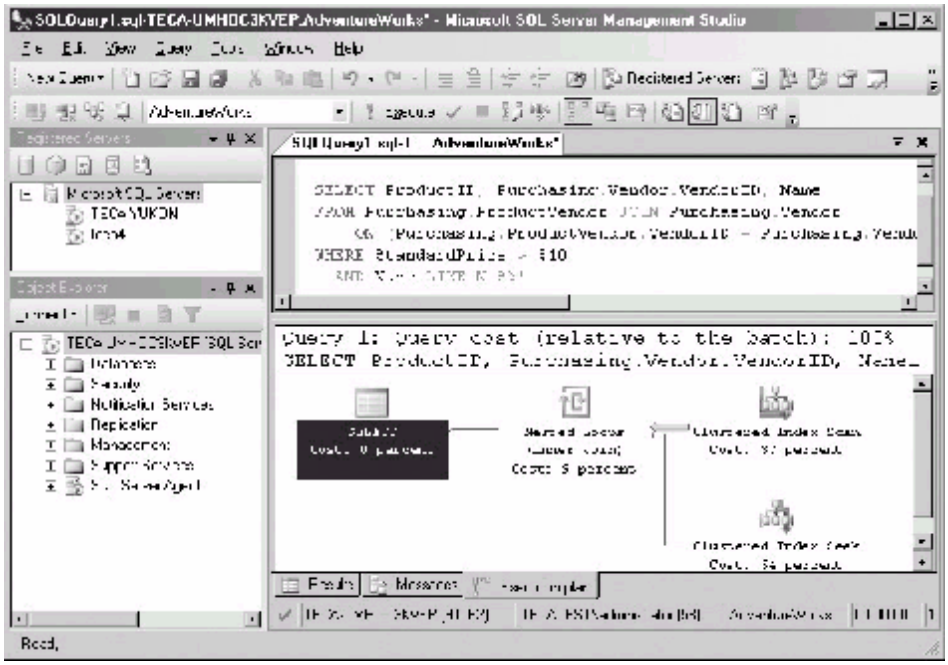
طرح‌های اجرا

شبیه Query Analyzer قدیمی، Query Editor جدید دارای قابلیت نمایش گرافیکی طرح اجرای یک پرس‌وجوست. SQL Server Management Studio Editor جدید نیز این قابلیت را با دادن امکان نمایش یک طرح اجرای واقعی به شما توسعه می‌دهد. می‌توانید یک طرح اجرای نمونه را در شکل ۷-۲ ببینید.

برخی از ویژگی‌های جدیدی که به Showplan گرافیکی Query Editor اضافه شده‌اند، عبارت از مجموعه جدیدی از آیکن‌های مجدداً طراحی شده است که مشهودتر هستند. کدنویسی رنگی جدید از یک الگوی مجموعه‌ای پیروی می‌کند. تمام ساختار تکرار آبی، مکان‌نماها زرد و ساختارهای زبان SQL سبز هستند. مقدار پارامترهایی که برای پرس‌وجوهای پارامتری استفاده می‌شوند، در Showplan نمایش داده خواهند شد. اطلاعات بهبود یافته‌ای برای پرس‌وجوهایی که به‌طور موازی اجرا می‌شوند، وجود دارد که Shoplan می‌تواند تعداد ردیف‌های تولید شده توسط هر رشته را نشان دهد. علاوه بر این، اطلاعات Showplan جدید هم‌چنین نشان خواهند داد که چه هنگام توسعه‌های تعریف شده کاربر CLR (UDX) به عنوان بخشی از یک پرس‌وجو استفاده می‌شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

شکل ۷-۲ Query Editor Showplan (b)

XML Showplan قابل صدور

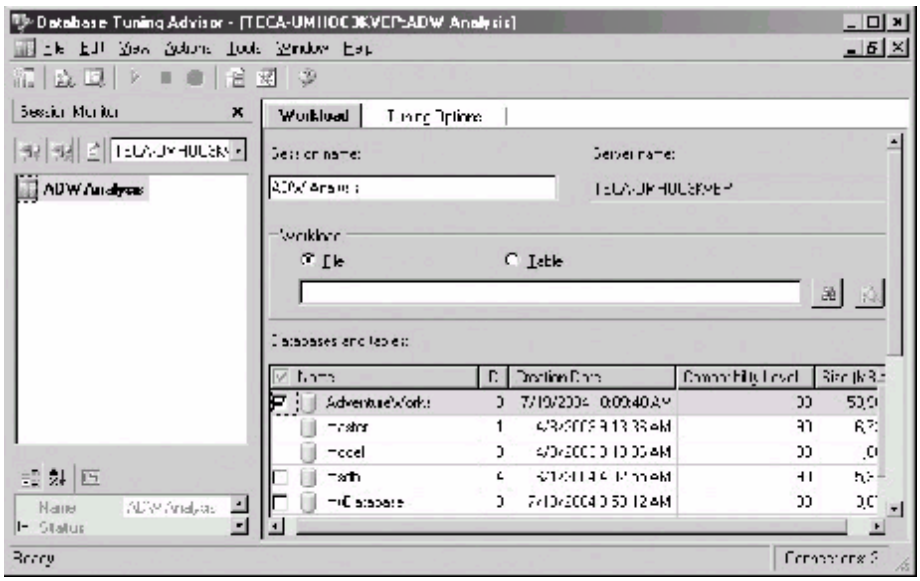
برخلاف Query Analyzer قدیمی، Query Editor هم‌چنین دارای توانایی صادر کردن نتایج Showplan به عنوان یک سند XML است. آن‌گاه این سند Showplan XML می‌تواند با پشتیبانی مایکروسافت یا شخص پشتیبان فنی دیگری برای عیب‌یابی مباحث کارآیی پرس‌وجو وارد شود. Showplanهای XML تولید شده، طرح‌واره تأیید شده هستند و فرمت XML Showplan جدید به سادگی قابل حمل بوده و می‌تواند برای مرور به یک سایت راه دور e-mail شود، می‌تواند وارد شود و توسط یک کاربر راه‌دور بدون نیاز به دستیابی به پایگاه داده منبع، گرافیکی شود. استفاده از فرمت XML برای خروجی Showplan، مزایای خوبی نسبت به فرمت‌های متنی و گرافیکی دارد و می‌تواند با استفاده از هر فناوری XML از قبیل XQuery، XPath، DOM یا SAX پردازش شود. اگر از پایگاه‌های داده موبایل پشتیبانی می‌کنید، یک نکته جالب دیگر این است که SQL Server CE از همین فرمت استفاده می‌کند. نکته مهمی که باید به آن توجه داشته باشید این است که در حالی که صدور جدید به فرمت XML در دسترس است، لزوماً نباید از آن استفاده کنید. هنوز از فرمت‌های Showplan گرافیکی و متنی قدیمی‌تر پشتیبانی می‌شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۳۰،۰۲ Database Tuning Advisor

Database Tuning Advisor ویژگی جدید دیگری است که در SQL Server 2005 وجود دارد که به شما امکان تنظیم کارایی برنامه‌های پایگاه داده را می‌دهد. Database Tuning Advisor لزوماً نگارش بهنگام شده‌ای از Index Tuning Wizard است که در نسخه‌های قبلی SQL Server فراهم شده بود. هرچند، Database Tuning Advisor جدید بسیار فراتر از حدس زدن ایندکس‌های جدید است و یکی از دلایل این مسأله، نام آن است. Database Tuning Advisor قادر به کار کردن با بخش‌هاست و از اجرای زمان‌بندی شده و تحلیل what-if ارزیابی بهبود یافته پشتیبانی می‌کند. دلیل دیگر این واقعیت است که Database Tuning Advisor جدید هم‌اکنون یک برنامه پر ویژگی است (به‌جز مجموعه‌ای از کادرهای محاوره‌ای ویزارد). می‌توانید مثالی از Database Tuning Advisor را در شکل ۸-۲ ببینید.



(a)

شکل ۸-۲ Database Tuning Advisor (b)

توصیه‌های پارتیشن‌بندی

علاوه بر توصیه‌های ایندکس‌ها، Database Tuning Advisor جدید پشتیبانی کاملی برای پارتیشن‌بندی داده SQL Server 2005 فراهم کرده است. Database Tuning Advisor می‌تواند کاربرد پارتیشن‌های تراز شده و نشده را توصیه کند (یک پارتیشن تراز شده از یک ایندکس کلاستر شده

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

استفاده می‌کند که به همان ترتیب جدول است، در حالی که یک پارتیشن تراز نشده از یک ایندکس کلاستر نشده استفاده می‌کند که به ترتیبی متفاوت از جدول پایه است). علاوه بر توانایی توصیه افزودن پارتیشن‌ها، Database Tuning Advisor هم‌چنین می‌تواند حذف پارتیشن‌های موجود را توصیه کند. برای کسب اطلاعات بیشتر در مورد پارتیشن‌بندی داده در SQL Server 2005، به فصل ۲ مراجعه کنید.

توقف و ادامه

Index Tuning Wizard قدیمی شبیه یک برنامه ویزارد مناسب، در یک حالت اجرای یک زمانه کار می‌کرد که از طریق صفحات ویزارد از ابتدا تا انتها، آن را به صورت مرحله‌ای انجام می‌دادید. Database Tuning Advisor جدید، در مقایسه، می‌تواند متوقف شود و با حالت ذخیره شده، مجدداً شروع شود. حتی می‌تواند از حالتی در ایستگاه‌های کاری کلاینت متفاوت ادامه دهد.

تنظیم زمان‌بندی شده

Database Tuning Advisor از یک جفت سناریو متفاوت برای تنظیم زمان‌بندی شده پشتیبانی می‌کند. ابتدا، برخلاف Index Tuning Wizard قدیمی که می‌توانست برای دوره‌های زمانی طولانی اجرا شود، در حالی که با توصیه‌های ایندکس بهینه‌ای مطرح می‌شد، Database Tuning Advisor جدید می‌تواند برای اجرای مدت زمان معینی تنظیم شود و سپس یک توصیه تنظیم "بهترین تلاش" را برطبق زمان اجرای داده شده تنظیم کند. اگر Index Tuning Wizard قدیمی مجاز به خاتمه نبود، قادر به تولید هیچ توصیه‌ای نبود. در مقایسه، می‌توانید Database Tuning Advisor را برای اجرا به مدت چهار ساعت تنظیم کنید و بهترین توصیه‌ای را بدهد که بتواند آن چیزی را بدهد که قادر به انجام شدن در آن زمان معین باشد. علاوه بر این، Database Tuning Advisor دارای توانایی زمان‌بندی اجراهای آن بدون نیاز به اجرای محاوره‌ای بودن ابزار یا انجام زمان‌بندی با استفاده از اسکریپت‌نویسی مشکلی است.

تنظیم what-if

ویژگی جدید و جالب دیگر Database Tuning Advisor، توانایی انجام تنظیم what-if همراه با ارزیابی است. این ویژگی به شما اجازه تست کردن یک سناریو اجرای داده شده را برحسب تست کردن انتخابی توصیه‌های آن می‌دهد. مثلاً، اگر Database Tuning Advisor توصیه کند که چهار ایندکس متفاوت را اضافه کنید، ولی شما احساس کرده باشید که واقعاً نیازی به چهار ایندکس ندارید، می‌توانید

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

بار کار را با استفاده از ایندکس‌های منتخب خود مجدداً اجرا کرده و ببینید که چگونه انتخاب شما با توصیه‌های اصلی Database Tuning Advisor مقایسه می‌شود.

بهبودهای Profiler

در حالی که Profiler ویژگی جدیدی نیست، ولی بهبودهای چندی در نسخه جدید آن داده شده است. یکی از مهم‌ترین تغییرات اساسی در Profiler در سطح معماری نهفته است. Profiler دقیقاً به موتور SQL Server گره خورده است و در نسخه‌های قبل، هنگام اعمال بهنگام‌رسانی‌ها بر موتور SQL Server که بر اطلاعات مورد استفاده Profiler تأثیر می‌گذارند، یک بهنگام‌رسانی متناظر باید بر سیستم‌های کلاینتی اعمال شود که برای اجرای Profiler استفاده می‌شوند. در نسخه SQL Server 2005، مایکروسافت تعریف ستون‌ها و رویدادهای معتبر را از موتور SQL Server اقتباس کرده و آن‌ها را در فایل‌های XML ذخیره می‌کند. هنگامی که Profiler در ابتدا شروع می‌شود، نگارش SQL Serverای را بررسی می‌کند که به آن متصل شده است. اگر Profiler به نگارش شناخته شده‌ای از SQL Server متصل شود، آن‌گاه از تعاریف کش شده آن استفاده می‌کند. اگر به نگارش بهنگام شده‌ای از SQL Server متصل شود، آن‌گاه اطلاعات پروفایلی XML جدید را دان‌لود می‌کند. این امر به Profiler امکان می‌دهد تا بدون عیب و نقص با نگارش‌های بهنگام شده SQL Server بدون نیاز به بهنگام‌رسانی‌ها برای کلاینت Profiler در هر زمان که بهنگام‌رسانی برای موتور SQL Server اعمال می‌شود، کار می‌کند.

پروفایل کردن Analysis Services

یکی از بزرگ‌ترین تغییرات در Profiler برای SQL Server 2005، توانایی آن برای پروفایل کردن عبارات MDX است که در مقابل Analysis Services اجرا می‌شود. Profiler جدید می‌تواند در یک نمونه SQL Server 2005 Analysis Services به همان روشی کار کند که با موتور SQL Server رابطه‌ای در نسخه‌های قبلی کار می‌کرد.

دیدگاه‌های انبوهه

ویژگی دیدگاه‌های انبوهه SQL Server 2005 Profiler جدید به شما امکان گروه‌بندی رویدادها را با یکدیگر و نشان دادن تعداد کل برای کل گروه را می‌دهد. مثلاً، می‌توانستید انتخاب کنید که رویدادهای Profiler را برطبق ID فرآیند سیستم (SPID) و Login Event ببینید و تعداد کل Loginهای ناموفق را با یک SPID خاص مشاهده کنید. این ویژگی می‌تواند به شما امکان مشاهده آسان‌تر رویدادها و تمایلات مهم را می‌دهد، بدون این که نیاز به گرفتن خروجی Profiler داشته باشید، آن را در یک جدول ذخیره کنید و سپس پرس‌وجوها را در آن اجرا کنید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

مجوز Trace جدید

بهبود مهم دیگر در SQL Server 2005 که مستقیماً بر Profiler تأثیر می‌گذارد، مجوز ردیابی جدید. در نسخه‌های قبلی SQL Server، برای اجرای Profiler، باید یک راهبر سیستم بودید. در حالی که برای بیشتر فعالیت‌های عیب‌یابی کارآیی طبیعی، مسأله نبود، اثبات شده است که مانعی برای اجرای وارسی‌هاست که لزوماً نباید امتیازات راهبری سیستم کامل بازرس را داشته باشید. SQL Server 2005 شامل یک نقش Trace جدید است که می‌توانید آن را به یک Login نسبت دهید و توانایی اجرای Profiler را به آن Login بدهید، بدون این که عضوی از گروه راهبری سیستم باشید. بدیهی است، به دلیل این که Profiler به‌طور بالقوه دستیابی به تمام داده‌ها را میسر می‌سازد، باید درباره واگذاری مجوز Trace هوشیار باشید.

اقتباس داده توسط Event Type

Profiler جدید SQL Server 2005 هم‌چنین امکان اقتباس اطلاعات از خروجی Profiler را برطبق نوع رویداد دارد. مثلاً، این ویژگی به شما امکان اقتباس تمام عبارات T-SQL را از یک جلسه Profiler معین و سپس نوشتن این رویدادها به عنوان یک فایل sql. که بتواند با استفاده از SQL Server Management Studio اجرا شود، می‌دهد.

Profiler این قابلیت مشابه را برای عبارات MDX و DMX فراهم کرده است. کاربرد واقعاً جالب دیگر برای این ویژگی، توانایی اقتباس اطلاعات بن‌بست است. آن‌گاه این اطلاعات بن‌بست می‌توانند در نمای گرافیکی با استفاده از Profiler ارایه شوند.

ایجاد Trace Wizard

ویژگی جدیدی دیگری که در SQL Server 2005 وجود دارد، معرفی مجدد Trace Wizard است. Trace Wizard در اصل در SQL Server 7 معرفی شد، ولی از نسخه SQL Server 2000 حذف شد. مایکروسافت Create Trace Wizard را در SQL Server 2005 برگردانده است.

لزوماً، Create Trace Wizard اجرای ردیابی‌های را با ردیابی گام به گام توسط راهبر آسان‌تر می‌کند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Article XXXI. چارچوب‌های مدیریتی جدید

علاوه بر ابزارهای مدیریتی جدید در SQL Server 2005، چارچوب کاری مدیریت SQL Server هم‌چنین به‌طور کامل اصلاح شده است. یک چارچوب کاری مدیریتی مبتنی بر NET. جدید به نام SMO[1] می‌تواند برای نوشتن برنامه‌های مدیریت سرور اختصاصی برای SQL Server 2005 استفاده شود. APIهای مدیریتی مبتنی بر NET. جدید هم‌چنین برای Analysis Services و مدیریت کپی‌سازی اضافه شده‌اند. تغییر جدید مهم دیگر برای مجموعه ابزار راهبری SQL Server 2005، پشتیبانی از رویدادهای WMI و پیکربندی WMI است. در بخش بعد این فصل، مطالب بیشتری درباره هر یک از این چارچوب‌های کاری مدیریت جدید در SQL Server 2005 خواهید یافت.

Section ۳۱.۰۱ SMO

شبیه ما قبل خود، یعنی DMO[2]، چارچوب کاری SMO طراحی شده است تا مدیریت هر جنبه SQL Server را با برنامه‌نویسی ممکن سازد. هرچند، برخلاف چارچوب کاری شیئی DMO قدیمی‌تر که بر اساس CPM بود، چارچوب کاری شیئی SMO جدید به عنوان یک کتابخانه کلاس NET. پیاده‌سازی شده است. این بدان معنی است که SMO نیازمند این است که NET Framework. بر روی سیستم‌هایی که برای اجرای برنامه‌های مدیریتی SMO استفاده می‌شوند، نصب شده باشد. SMO می‌تواند برای مدیریت سیستم‌های SQL Server 7 و SQL Server 2000 نیز به مانند SQL Server 2005 استفاده شود.

برای سازگاری با قبل، SQL Server 2005 به پشتیبانی از DMO ادامه داده است، ولی برای پشتیبانی از ویژگی‌های جدید موجود در نسخه جدید بهبود یافته است. مثلاً، DMO نمی‌تواند برای مدیریت SQL Service Broker، ویژگی نقاط انتهایی HTTP، انواع داده varbinary (max) و XML جدید یا هر یک از ویژگی‌های جدید دیگری که مایکروسافت به نسخه ۲۰۰۵ اضافه کرده است، استفاده شوند. به عبارت دیگر، DMO تنها محدود به پشتیبانی از ویژگی‌هایی است که در نسخه‌های قبلی SQL Server وجود داشتند. چارچوب کاری شیئی SMO جدید، بیش از ۱۵۰ کلاس جدید را برای نشان دادن ویژگی‌های جدید موجود در SQL Server 2005 فراهم کرده است. برای آسان کردن انتقال برنامه‌های DMO قدیمی به SMO، چارچوب کاری شیئی مورد استفاده SMO تقریباً برطبق همین چارچوب کاری شیئی است که توسط DMO استفاده می‌شد. در حالی که چارچوب‌های کاری شیئی یکسان هستند، آن‌ها کاملاً شبیه هستند. برای نمونه، SMO و DMO هر دو مالک اشیایی هستند که سرورها، پایگاه‌های داده، جداول، ستون‌ها و سایر اشیای اصلی پایگاه داده را نشان می‌دهند.

1- System Management Objects
2- Distributed Management Objects

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

علاوه بر توانایی دستیابی به تمام ویژگی‌های جدید موجود در SQL Server 2005، چارچوب کاری شیئی SMO جدید، یک جفت ویژگی مهم دیگر را فراهم کرده است. ابتدا، SMO از نمونه‌سازی بهینه اشیا استفاده می‌کند، بدین معنی که می‌تواند نمونه‌ای را ایجاد کند که یک شئ سطح بالا شبیه یک شئ سرور بدون نیاز به بازیابی تمام خصوصیاتی که آن شئ را تشکیل می‌دهند، نشان می‌دهد. در مورد شیئی شبیه سرور، این بدان معنی است که SMO نباید تمام اطلاعات پایگاه داده را به علاوه تمام اطلاعات شیئی پایگاه داده را برای تمام پایگاه‌های داده در سرور هنگام ایجاد شئ سرور بازیابی کند. نمونه‌سازی کامل به تأخیر می‌افتد تا وقتی که به‌طور صریح به شئ مراجعه شود. نمونه‌سازی بهینه موجب کارآیی خیلی بهتری، به‌خصوص برای پیاده‌سازی‌های VLDB می‌شود. SMO هم‌چنین دارای توانایی اکتساب و دسته‌ای کردن گروه‌هایی از عبارات SQL برای کارآیی بهتر هستند. هم‌چنین یک قابلیت اسکریپت‌نویسی بهبود یافته وجود دارد که توسط یک کلاس Scripter جدید ممکن می‌شود. کلاس Scripter قادر به کشف و ایجاد اسکریپت‌های ایجاد برای اشیای پایگاه داده و وابستگی‌های آن‌هاست.

SMO در فضاهای نام Microsoft.SqlServer.Mamagement.Smo، Microsoft.SqlServer.Mamagement.Smo.Common و Microsoft.SqlServer.Mamagement.Smo.Agent پیاده‌سازی می‌شود.

Section ۳۱،۰۲ AMO[1]

AMO شبیه SMO برای مدیریت سرورهای SQL Server برای مدیریت Analysis Services استفاده می‌شود. در بین سایر آیتم‌ها، AMO می‌تواند برای مدیریت سرورهای Analysis Services، منابع داده، مکعب‌ها، ابعاد، سنجش‌ها و مدل‌های data mining استفاده شود. AMO شبیه SMO یک چارچوب کاری شیئی مبتنی بر NET. است که برای اجرا نیاز به CLR دارد. AMO از لحاظ مفهومی برطبق کتابخانه DSO[2] مبتنی بر COM است. همانند DMO، SQL Server 2005 هنوز از DSO برای سازگاری با قبل برنامه‌های موجود پشتیبانی می‌کند. AMO پیام‌های مبتنی بر XML را تولید می‌کند که می‌توانند برای ایجاد یا تغییر اشیای Analysis Server استفاده شود. این پیام‌های XML می‌توانند به عنوان اسناد XML ذخیره شوند. AMO در فضای نام Microsoft.SqlServer.Managment.AnalysisServices پیاده‌سازی می‌شود.

1- Analysis Management Objects
2- Decision Support Objects

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۳۱.۰۳ RMO[1]

دقیقاً شبیه SMO که برای کنترل تمام جنبه‌های مدیریت و پیکربندی SQL Server طراحی شده است، RMO مبتنی بر NET. جدید برای پیکربندی و مدیریت فرآیند کپی‌سازی پایگاه داده در SQL Server 2005 طراحی شده است. برنامه‌های RMO شبیه SMO و AMO نیاز به وجود CLR دارند. چارچوب کاری شیئی RMO جدید حاوی کلاس‌هایی است که مدیریت کپی‌سازی‌ها، راهبری، هم‌زمان‌سازی و کنترل SQL Server را ممکن می‌سازند. RMO در فضای نام Microsoft.SqlServer.Replication پیاده‌سازی شده است.

Section ۳۱.۰۴ WMI[2]

SQL Server 2005 هم‌چنین می‌تواند با استفاده از WMI API مدیریت شود. WMI استاندارد مدیریت چند محصولی مایکروسافت است، ولی قبل از نسخه ۲۰۰۵، WMI در مدیریت SQL Server شبیه DMO یا SQL Server Enterprise Manager مؤثر نبود. برای فروشگاه‌هایی که سعی دارند چارچوب کاری مدیریت WMI را استاندارد کنند، این امر توزیع‌های جهانی SQL Server را مشکل می‌کند، زیرا اغلب نیاز به مداخله دستی دارد. برای سامان بخشیدن به مسأله قابلیت مدیریت جهانی، قابلیت‌های مدیریتی WMI SQL Server 2005 با یک WMI Configuration Provider جدید و رویدادهای WMI جدید توسعه یافته است.

WMI Configuration Provider

WMI Configuration Provider جدید SQL Server 2005 کاربرد مدیریت راه‌دور را برای اتصال به SQL Server و انجام اعمال مدیریتی ممکن می‌سازد. API پیکربندی WMI برای کار کردن به الگویی غیرمتصل طراحی شده است. تأمین کننده WMI در فضای نام Microsoft.SqlServer.Smo.Wmi پیاده‌سازی شده است.

رویدادهای WMI

رویدادهای WMI جدید SQL Server 2005 به SQL Server امکان می‌دهند تا توسط MOM[3] و برنامه‌های مدیریتی سازگار با WMI شخص ثالث دیگر کنترل شود. SQL Server 2005 می‌تواند رویدادهای WMI را برای تمام عملیاتی که می‌توانند با استفاده از Profiler ردیابی شوند، به‌طور مجازی به وجود آورد. مثلاً، رویدادها می‌توانند برای ایجاد پایگاه داده، حذف پایگاه داده، تغییر پایگاه داده و

1- Replication Management Objects
2- Windows Management Instrumentation
3- Microsoft Operations Manager

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

تمام عملیات دیگری که می‌توانند ردیابی شوند، تولید گردند. شبیه هنگام استفاده از Trace موقع استفاده از رویدادها، مقداری سربار تولید می‌شود، ولی آن‌ها اطلاعات مدیریتی بسیار غنی را فراهم می‌کنند. رویدادهای WMI می‌توانند توسط SQL Server، Analysis Services، DTS و Reporting Services به وجود آیند.

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

فصل سوم

ویژگی‌های بازیافت و در دسترس بودن

بهبودهای جدید در SQL Server 2005 برای بهبود دسترس‌پذیری SQL Server با افزایش قابلیت‌های کلاسترینگ آن و فراهم کردن گزینه‌های پایگاه داده جدیدی که قابلیت‌های در دسترس بودن متوالی و بازیافت پایگاه داده بهبود یافته را ارایه می‌کنند، ادامه می‌یابد. دستیابی به در دسترس بودن بالا در یک مقیاس کوچک نسبتاً آسان است، ولی هنگام افزایش اندازه سیستم، به‌طور نمایی مشکل‌تر می‌شود. SQL Server 2005 تعدادی از ویژگی‌های جدیدی را که در دسترس بودن و قابلیت بازیافت پایگاه داده را با سامان بخشیدن به موانع اصلی که از در دسترس بودن پایگاه داده سطح جهانی جلوگیری می‌کنند، بهبود می‌بخشد. برخی از عواملی که از در دسترس بودن پایگاه داده در نگارش‌های قبلی SQL Server جلوگیری می‌کردند، شامل مواردی از قبیل زمان‌های failover همراه با تأخیر، نیاز به داشتن دستیابی پایگاه داده انحصاری برای عملیات نگهداری منتخب و گاهی اوقات مشکل بودن پاسخ دادن سرور در طی زمان مصرف بالای CPU، از قبیل هنگامی که مقداری از کد کاربر در یک

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

حلقه بی نهایت می‌افتد، می‌باشند. عامل دیگری که می‌تواند تأثیر زیادی بر در دسترس بودن پایگاه داده بگذارد، ارتقاهای سخت‌افزاری است. حتی اگر ارتقاهای سخت‌افزاری، رویدادهای طرح‌ریزی شده باشند، نیازمند زمان بیکاری سیستم برای انجام ارتقا هستند. SQL Server 2005 قابلیت جدیدی را برای کاهش زمان بیکاری مورد نیاز با یکی از متداول‌ترین ارتقاهای سخت‌افزاری طراحی کرده است. در این فصل، نگاهی به این گزینه‌های بازیافت و در دسترس بودن موجود در SQL Server 2005 خواهیم انداخت، بنابراین می‌توانید نحوه استفاده از این ویژگی‌ها را برای پیاده‌سازی در SQL Server 2005 در یک محیط پایگاه داده تولیدی قابل بازیافت و در دسترس درک کنید.

Article XXXII. محافظت از خرابی سرور یا پایگاه داده

خرابی سرور پایگاه داده، یک از کار افتادگی است که با خرابی یک سخت‌افزار یا مشکل نرم‌افزاری به وجود می‌آید که فعال نبودن سرور را برای مدت زمانی نمایش می‌دهد. خرابی سرور پایگاه داده هم‌چنین می‌تواند با عوامل محیطی از قبیل یک سانحه به وجود آید. در این بخش، مطالبی درباره برخی از مهم‌ترین ویژگی‌های جدیدی که SQL Server 2005 برای سامان بخشیدن به خرابی‌های سرور پایگاه داده فراهم کرده است، می‌آموزید.

Section ۳۲.۰۱ کلاسترینگ Failover بهبود یافته

یک مزیت در دسترس بودن بالای کلیدی که SQL Server 2005 در بخشی از پشتیبانی آن برای Windows Server 2003 پشتیبانی برای کلاسترینگ failover را اشتقاق می‌کند، است. با بهره بردن از پشتیبانی کلاسترینگ بهبود یافته در Windows Server 2003، SQL Server 2005 هم‌اینک می‌تواند در کلاسترهای تا هشت گره در Windows Server 2003 Datacenter Edition باشد. علاوه بر این، SQL Server 2005 از کلاسترینگ چهار گرهه در Windows Server 2003 Enterprise Edition و Windows Server 2003 Datacenter Server پشتیبانی می‌کند. از یک حداکثر کلاسترینگ دو گرهه در Windows 2000 Advanced Server پشتیبانی می‌شود.

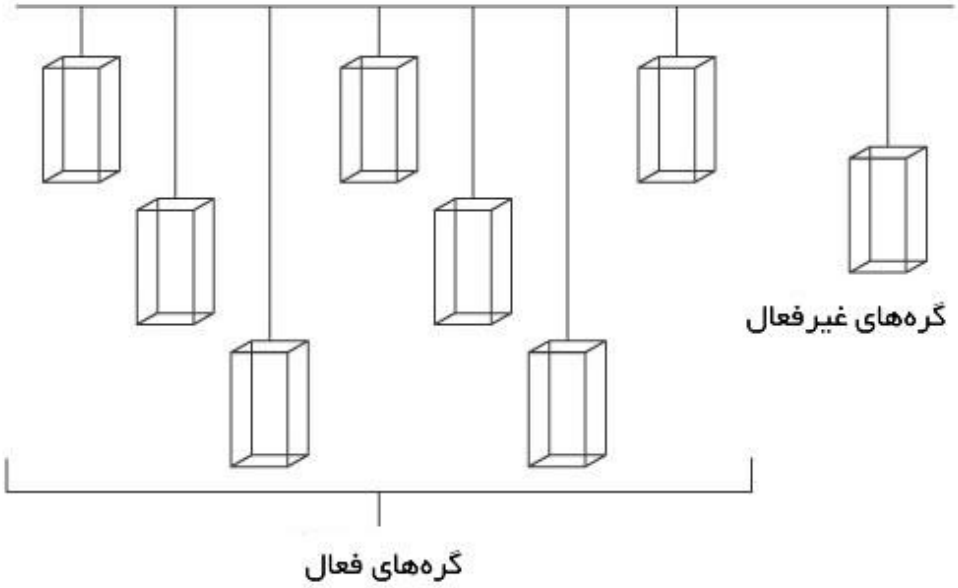
با سرویس‌های کلاسترینگ Windows، هر سرور در کلاستر، یک گره[1] نامیده می‌شود. تمام گره‌ها در یک کلاستر در حالتی از ارتباط ثابت هستند. اگر یکی از گره‌ها در یک کلاستر در دسترس نباشد، گره دیگری وظایف آن را در نظر خواهد گرفت و تدارک سرویس‌هایی یکسان را با گره خراب شده، شروع می‌کند. این فرآیند failover نامیده می‌شود. کاربرانی که به کلاستر دسترسی دارند، به‌طور خودکار به گره جدید سوییچ می‌کنند.

1- Node

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

کلاسترینگ می‌تواند به دو روش اساسی تنظیم شود: کلاسترینگ Active-Active که تمام گره‌ها کاری انجام می‌دهند یا Active-Passive که یکی از گره‌ها غیرفعال است، تا وقتی که گره فعال خراب شود. Windows Server 2003 هم‌چنین از پیکربندی‌های N+I (N فعال با I زاپاس) پشتیبانی می‌کند که یک روش کلاسترینگ با هزینه کارآمد و انعطاف‌پذیری بالا فراهم می‌کند تا برنامه‌ها در دسترس قرار گیرند. مثلاً، با یک کلاستر هشت گرهه در یک پیکربندی N+I، می‌توانید هفت گره از هشت گره را برای در دسترس بودن تنظیم کرده و سرویس‌های متفاوتی را فراهم کنید، در حالی که هشت گره حالت غیرفعال هستند که می‌تواند سرویس‌هایی از هر یک از هفت گره فعال را در نظر داشته باشد. شکل ۱-۳ مثالی از کلاستر هشت گرهی را نشان می‌دهد که هفت گره فعال بوده و یک گره آماده کار است تا در صورت خراب شدن هر یک از هفت گره فعال، شروع به کار کند.



(a)

شکل ۱-۳ پشتیبانی کلاستر هشت گرهی (b)

برخی از بهبودهای خاص کلاسترینگ در SQL Server 2005 شامل پشتیبانی برای یک تنظیم کلاستر ناموجه است. علاوه بر این، تمام سرویس‌های مختلف در SQL Server 2005 کاملاً از کلاستر آگاه هستند، از جمله:

- Database Engine
- Analysis services

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

- Reporting services
- Notification services
- SQL Server Agent
- Full-Text Search
- Service Broker
- SQLiMail

علاوه بر این، تمام ابزارهای مدیریتی عمده در SQL Server 2005 نیز آگاه از کلاستر هستند، از جمله:

- SQL Server Management Studio
- Service Control Manager
- SQL Profiler
- SQL Query Analyzer

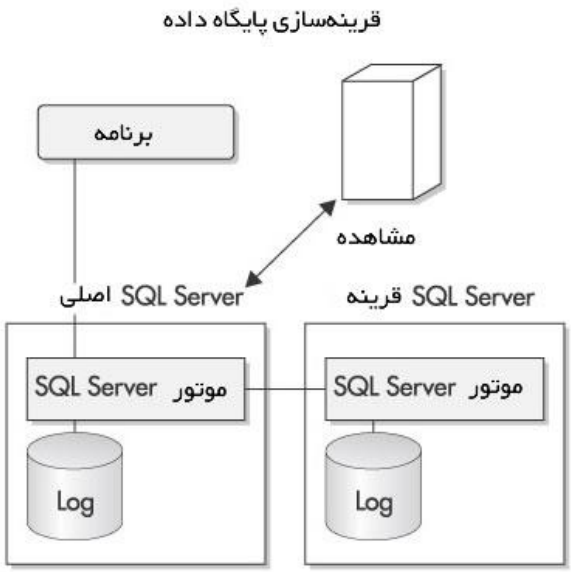
می‌توانید اطلاعات بیشتر درباره ابزارهای مدیریت SQL Server 2005 را در فصل ۱ بیابید.

Section ۳۲،۰۲ Database Mirroring

احتمالاً بزرگ‌ترین ویژگی جدید در ناحیه در دسترس بودن، پشتیبانی SQL Server 2005 از Database Mirroring است. ویژگی Database Mirroring جدید از خرابی سرور یا پایگاه داده با دادن قابلیت آماده به کار ثابت توسط SQL Server 2005 محافظت می‌کند. Database Mirroring لزوماً failover سطح پایگاه داده را فراهم می‌کند. در رویدادی که پایگاه داده اصلی خراب شود، قرینه پایگاه داده، یک پایگاه داده دوم آماده کار را در حدود ۲ تا ۳ ثانیه در دسترس قرار می‌دهد. Database Mirroring فقدان داده صفر را مهیا می‌کند و پایگاه داده قرینه همیشه با تراکنش جاری که در سرور پایگاه داده اصلی پردازش می‌شود، بهنگام می‌شود. تأثیر اجرای Database Mirroring بر بازده تراکنش حداقل صفر است. پشتیبانی failover پایگاه داده می‌تواند برای انجام خودکار یا دستی تنظیم شود. حالت failover خودکار را به پایگاه‌های داده تولیدی خود بدهید. Database Mirroring با تمام آیتم‌های سخت‌افزاری استاندارد که از SQL Server پشتیبانی می‌کنند، کار می‌کند (هیچ نیازی برای سیستم‌های خاص نیست و سرور اصلی و سرور قرینه نباید یکی باشند). علاوه بر این، بر خلاف راه‌حل‌های کلاسترینگ در دسترس بودن بالا، هیچ نیازی برای حافظه مشترک بین سرور اصلی و سرور قرینه وجود ندارد. می‌توانید مروری از نحوه کار ویژگی Database Mirroring را در شکل ۲-۳ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

(b)

شکل ۲-۳ Database Mirroring

Database Mirroring با استفاده از سه سیستم پیاده‌سازی می‌شود: سرور اصلی، سرور قرینه و شاهد. با این نام‌ها دچار تشویش نشوید. سرور اصلی تنها نام سیستمی است که در حال حاضر سرویس‌های پایگاه داده را فراهم می‌کند. تمام اتصالات کلاینت وارده برای سرور اصلی شکل می‌گیرند. کار سرور قرینه، نگهداری یک کپی از پایگاه داده قرینه سرور اصلی است. بسته به پیکربندی سرور قرینه، سرور اصلی و قرینه می‌توانند بدون عیب و نقص به نقش‌ها سوییچ کنند. شاهد لزوماً به عنوان یک شخص ثالث مستقل عمل می‌کند و کمک می‌کند تا تعیین کنید کدام سیستم نقش سرور اصلی را در نظر خواهد داشت. هر سیستمی که سرور اصلی خواهد بود، یک رأی می‌گیرد. شاهد دو رأی می‌گیرد تا در مورد سرور اصلی تصمیم بگیرد. این مسأله بسیار مهم است، زیرا امکان‌پذیر است که ارتباطات بین سرور اصلی و سرور قرینه بتوانند قطع باشند که در این مورد، هر سیستم خود را برای کار کردن به عنوان سرور اصلی انتخاب می‌کند. شاهد رأی تصمیم‌گیری را تبدیل می‌کند.

Database Mirroring با ارسال ثبت تراکنش‌ها بین سرور اصلی و سرور قرینه کار می‌کند. بنابراین، طبق ضرورت، ویژگی Database Mirroring جدید، یک برنامه ارسال ثبت بلادرنگ است. Database Mirroring می‌تواند توسط یک پایگاه داده تنظیم شود یا می‌تواند برای چند پایگاه داده تنظیم گردد. هنگامی که یک سیستم کلاینت، درخواستی را به سرور اصلی می‌نویسد، آن درخواست واقعاً در فایل ثبت سرور اصلی نوشته می‌شود، قبل از این که در فایل داده نوشته شود، زیرا SQL

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Server از یک ثبت Write-ahead استفاده می‌کند. سپس، آن رکورد تراکنش به سرور قرینه ارسال می‌شود که در ثبت تراکنش سرور قرینه نوشته می‌شود. بعد از این که سرور قرینه، رکورد را در ثبت خود نوشت، پیام تأییدی را به سرور اصلی مبنی بر دریافت رکورد ارسال می‌کند که نشان می‌دهد هر دو سیستم در فایل ثبت خود دارای داده یکسانی هستند. در مورد عملیات Commit، سرور اصلی منتظر دریافت پیام تأییدی از سرور قرینه می‌ماند، قبل از این که پاسخ خود را به کلاینت برگرداند و کامل شدن عمل را به او بگوید. برای بهنگام نگه داشتن فایل‌های داده در سرور قرینه، لزوماً در حالتی از بازیافت مستمر قرار دارد و داده را از ثبت گرفته و فایل داده را بهنگام می‌کند.

Database Mirroring با گرفتن یک پشتیبان از پایگاه داده‌ای مقداردهی می‌شود که باید در سرور اصلی قرینه شود و آن‌گاه آن پشتیبان را برای سرور قرینه خود بازیابی کنید. به عبارت دیگر، پایگاه داده قرینه باید در سرور قرینه وجود داشته باشد، قبل از این که فرآیند قرینه‌سازی بتواند شروع شود. این امر داده و طرح واره پایگاه داده مرتبط را در یک‌جا قرار می‌دهد. فرآیند بازیابی و پشتیبان‌گیری می‌تواند از هر یک از انواع رسانه استاندارد SQL Server استفاده کند، از جمله نوار یا دیسک. سپس، از فرمان ALTER DATABASE استفاده کنید، همان‌گونه که در این‌جا برای شروع فرآیند قرینه‌سازی استفاده می‌کند:

```
ALTER DATABASE <database name> SET PARTNER = '<partner server name>'
```

فرمان ALTER DATABASE ابتدا باید در سرور قرینه اجرا شود، آن را به نام SQL Server اصلی در بخش SET PARTNER اشاره می‌کند. سپس فرمان ALTER DATABASE در زمان بعدی اجرا می‌شود (این بار در سرور اصلی). هنگامی که فرمان ALTER DATABASE در سرور اصلی اجرا می‌شود، نام سرور قرینه در بخش SET PARTNER تأمین می‌شود. هنگامی که فرمان ALTER DATABASE در سرور اصلی کامل می‌شود، قرینه پایگاه داده در حال حرکت تنظیم شده و سرور اصلی، انتقال ثبت‌ها را به سرور قرینه پایگاه داده در حال حرکت تنظیم شده و سرور اصلی، انتقال ثبت‌ها را به سرور قرینه شروع خواهد کرد. مرحله بعدی در تنظیم Database Mirroring، تنظیم شاهد است. بیشتر اوقاتی که سرور اصلی و قرینه را تنظیم می‌کنید، شاهد با اجرای مجدد فرمان ALTER DATABASE در سرور اصلی تنظیم می‌شود، همان‌گونه که در این لیست نشان داده شده است:

```
ALTER DATABASE <database name> SET WITNESS = '<witness server name>'
```

در این مورد، نام پایگاه داده با نام پایگاه داده مورد استفاده در فرامین قبل یکسان است. هرچند، این بار بخش SET WITNESS برای مشخص کردن سرور شاهد استفاده می‌شود. هنگامی که سرور شاهد تنظیم شد، Database Mirroring دارای تمام اطلاعاتی است که باید failover پایگاه داده

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

را انجام دهند. Database Mirroring لزوماً یک پایگاه داده مجازی با تحمل خرابی را به شما می‌دهد. هرچند، این داستان کامل Database Mirroring نیست. پیاده‌سازی یک پایگاه داده آماده کار ثابت، به پایگاه داده امکان می‌دهد تا سریعاً بعد از یک خرابی در دسترس قرار گیرد، ولی ارتباطات کاربر برای آن پایگاه داده خراب شده، چه می‌شود؟ این‌جا محلی است که ویژگی جدید Transparent Client Redirect نقش خود را ایفا می‌کند.

Transparent Client Redirect

ویژگی Transparent Client Redirect بسیار نزدیک به ویژگی Database Mirroring کار می‌کند و به سیستم‌های کلاینت اجازه هدایت خودکار به سرور قرینه را در زمان در دسترس نبودن سرور اصلی می‌دهد. این ویژگی جدید در MDAC[1] SQL Server 2005 جدید پیاده‌سازی شده است و هیچ تغییری برای برنامه‌های لایه داده یا کلاینت مورد نیاز نیست. میان‌افزار MDAC از هر دو سرور اصلی و قرینه آگاه است. MDAC نام سرور قرینه را طبق اتصال اولیه آن با سرور اصلی به دست می‌آورد. هنگامی که اتصال به سرور اصلی از بین می‌رود، MDAC ابتدا تلاش خواهد کرد تا مجدداً به سرور اصلی متصل شود. اگر اولین تلاش برای اتصال ناموفق باشد، آن‌گاه MDAC به‌طور خودکار تلاش اتصال دوم آن را به سرور قرینه هدایت می‌کند. همانند کلاسترینگ، اگر اتصال در میانه یک تراکنش از بین برود، آن‌گاه آن تراکنش roll back خواهد شد و باید بعد از اتصال کلاینت به سرور قرینه، مجدداً انجام شود.

Section ۳۲،۰۳ زمان استفاده از کلاسترینگ یا Database Mirroring

کلاسترینگ و Database Mirroring، هر دو راه‌حل‌های آماده جهانی هستند که قادر به فراهم کردن یک سیستم پشتیبان در مورد خرابی سرور یا پایگاه داده هستند. آن‌ها دارای شباهت‌هایی هستند. هر دو قادر به رساندن فقدان داده به صفر هستند، هر دو دارای تشخیص خطای خودکار هستند، هر دو failover خودکار را فراهم می‌کنند و هر دو اتصال مجدد کلاینت را ممکن می‌سازند. هرچند، تفاوت‌های مهمی نیز دارند. Database Mirroring همان‌گونه که از نام آن مشخص است، در سطح پایگاه داده کار می‌کند، در حالی که کلاسترینگ در سطح سرور کار می‌کند. Database Mirroring تقریباً زمان failover ۲ تا ۳ ثانیه‌ای ثابتی را فراهم می‌کند، در حالی که کلاسترینگ دارای زمان failover حدود ۳۰ ثانیه است (حتی گاهی طولانی‌تر که این امر وابسته به سطح فعالیت پایگاه داده و اندازه پایگاه‌های داده در سرور خراب شده است). Database Mirroring هم‌چنین خرابی‌های

1- Microsoft Data Access Components

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

دیسک را به‌طور کاملاً بهتری محافظت می‌کند، زیرا هیچ حافظه دیسک مشترکی وجود ندارد، درست همانند راه‌حل کلاسترینگ. به‌طور مجازی، هیچ حد فاصله‌ای برای Database Mirroring وجود ندارد، در حالی که کلاسترینگ دارای حد تقریباً ۱۰۰ مایل است، زیرا یک حد زمانی برای انتقال ضربان قلب بین گره‌های کلاستر وجود دارد. علاوه بر این، Database Mirroring یک فناوری ساده‌تر است که پیاده‌سازی آن نسبت به کلاسترینگ آسان‌تر است. از طرف دیگر، Database Mirroring نمی‌تواند برای پایگاه‌های داده سیستم استفاده شود، در حالی که کلاسترینگ پایگاه داده سیستم و پایگاه‌های داده کاربر را محافظت می‌کند. اصولاً، کلاسترینگ راه‌حل بهتری برای محافظت از یک سرور کامل است، در حالی که Database Mirroring راه‌حل بهتری برای محافظت از یک برنامه یا پایگاه داده بحرانی است.

Article XXXIII. بهبودهای در دسترس بودن پایگاه داده

هرچند، همه روش‌های قبل، عوامل مهمی را سامان می‌بخشیدند که در دسترس بودن پایگاه داده را کاهش می‌دهد، احتمالاً هیچ عاملی وجود ندارد که بر در دسترس بودن و قابلیت بازیافت بیش از نگهداری پایگاه داده تأثیر گذارد. در حالی که در سال‌های آینده قطعاً بدون خرابی پایگاه داده امکان‌پذیر است، نگهداری پایگاه داده بحث روزانه‌ای است که نمی‌توانید از آن دوری کنید. SQL Server 2005 چالش‌هایی از این ناحیه حیاتی را به علاوه دو ویژگی جدید بر عهده می‌گیرد: Database Snapshots و دستیابی بازیابی قدیمی. در بخش بعدی این فصل، نگاهی مفصل به هر یک از این ویژگی‌های در دسترس بودن و قابلیت بازیافت خواهیم داشت.

Section ۳۳.۰۱ Database Snapshot

ویژگی Database Snapshot جدید یک تصویر فقط خواندنی از پایگاه داده را در محل خاصی از زمان فراهم می‌کند. یک Database Snapshot برای ایجاد کپی‌هایی از یک پایگاه داده برای گزارشگیری یا برای ایجاد یک کپی پشتیبان از پایگاه داده‌ای که می‌تواند برای برگرداندن یک پایگاه داده تولیدی به حالت قبل استفاده شود، مناسب‌تر است. هنگامی که یک Database Snapshot ایجاد می‌شود، سیستم یک کپی فوق داده از پایگاه داده مشخص شده در آن نقطه خاص از زمان می‌سازد. هر تغییر بعدی که برای پایگاه داده اصلی صورت می‌گیرد، در Database Snapshot منعکس نمی‌شود. برنامه‌ها دقیقاً به یک Database Snapshot در صورتی متصل می‌شود که پایگاه داده دیگری باشد. یک Database Snapshot می‌تواند برای هر پایگاه داده‌ای ایجاد شود. Database Snapshotها به عنوان بخشی از عبارت CREATE DATABASE DDL ایجاد می‌شوند. می‌توانید نحوه ایجاد یک دیدگاه را در پایگاه داده AdventureWorks نمونه در این لیست ببینید:

```
CREATE DATABASE AdWSnapShot_080104_0700 ON
( NAME = AdventureWorks_Data,
  FILENAME = 'C:\Program Files\Microsoft SQL Server
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
    \MSSQL.1\MSSQL\DATA\AdventureWorks_data_040422_1800.ss')
AS SNAPSHOT OF AdventureWorks;
```

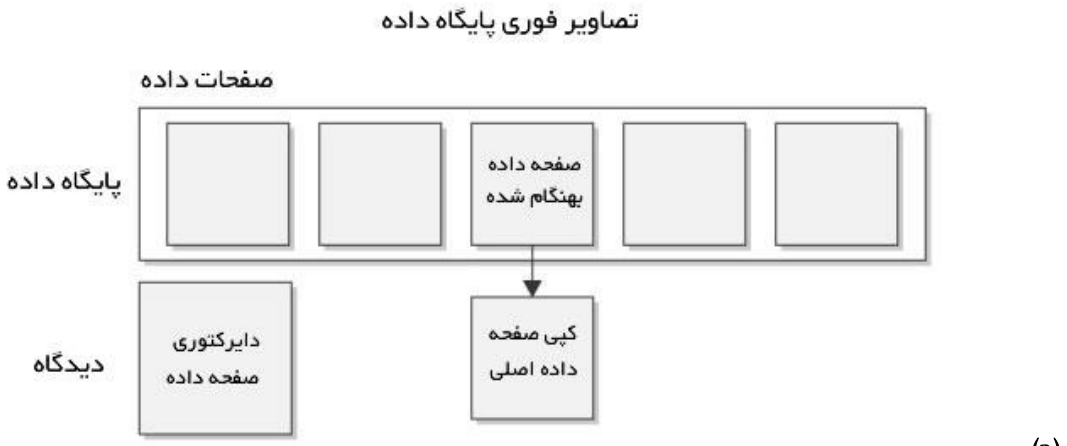
در این مثال، می‌توانید ببینید که یک Database Snapshot به نام AdWSnapShot_080104_0700 در پایگاه داده AdventureWorks ایجاد شده است. Database Snapshotها با استفاده از بخش AS SNAPSHOT OF ایجاد می‌شوند که می‌توانید در لیست می‌بینید. هنگام ایجاد یک Database Snapshot، باید تمام فایل‌های داده‌ای را مشخص کنید که در پایگاه داده منبع استفاده می‌شوند. با توجه به این که پایگاه داده AdventureWorks شامل یک فایل داده است، تنها فایل AdventureWorks_Data باید به عبارت CREATE DATABASE مشخص شود. فایل‌های ثبت را مشخص نکنید.

ایجاد یک Database Snapshot یک عمل کم هزینه است، همان‌گونه که سرور اصولاً از فوق داده در الحاق با بازیافت برای ایجاد نقطه دید استفاده می‌کند. درک این مسأله مهم است که Database Snapshotها، کپی کاملی از یک پایگاه داده نیستند. در عوض، ایجاد یک Database Snapshot یک عمل فقط فوق داده است. یک Database Snapshot از همان صفحات داده مشابه پایگاه داده اصلی استفاده می‌کند، بنابراین نیازی به مقدار زیادی فضای دیسک اضافی نیست. Database Snapshotها با استفاده از فناوری کپی و نوشتن ساخته می‌شوند که هر زمانی که تغییری در یکی از صفحات داده پایگاه داده منبع صورت می‌گیرد، یک کپی از آن صفحه برای Database Snapshot ذخیره شده و سپس صفحه بهنگام شده به همان روش طبیعی نوشته می‌شود.

هنگامی که Database Snapshot مورد دستیابی قرار می‌گیرد، از صفحات داده مشترک استفاده می‌کند، تا وقتی که به صفحه تغییر یافته برسد و سپس صفحاتی را بررسی خواهد کرد که کپی شده‌اند و نه صفحات داده‌ای که حاوی داده بهنگام شده هستند. در این روش، Database Snapshot نیاز به حافظه‌ای برای تنها صفحاتی دارد که از زمان ایجاد Database Snapshot تغییر یافته‌اند. شکل ۳-۳ چگونگی کار کردن Database Snapshotها را نشان می‌دهد.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

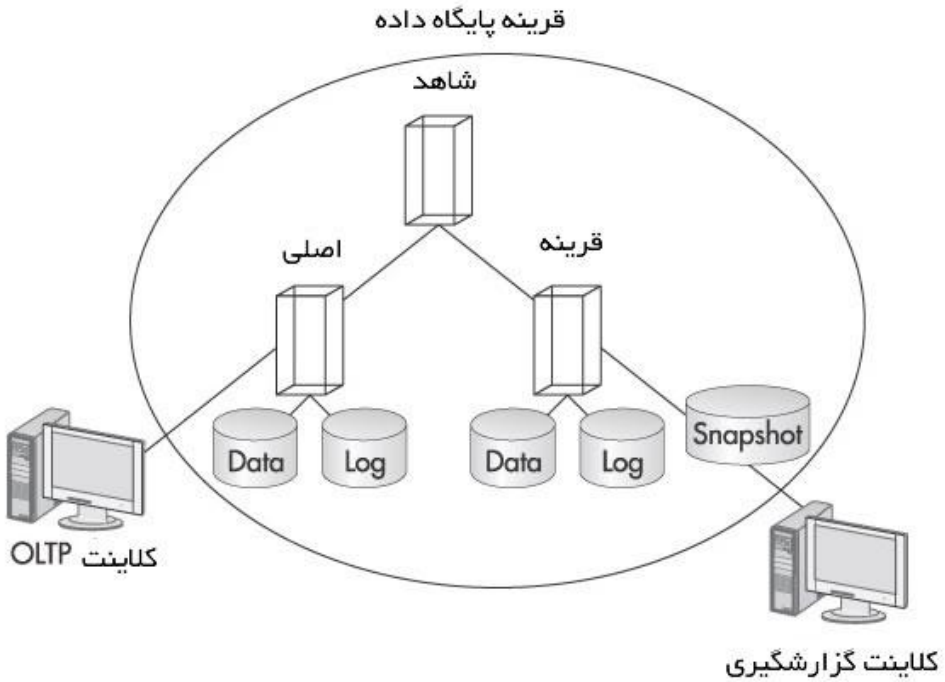
شکل ۳-۳ Database Snapshotها (b)

Database Snapshotها می‌توانند با Database Mirroring برای ایجاد یک سرور گزارشگیری برطبق داده‌ای که در سرور قرینه وجود دارد، ترکیب شوند (شکل ۴-۳). طبیعتاً، داده در سرور قرینه همیشه در حالت بازیافت است، بنابراین نمی‌تواند توسط یک برنامه مورد دستیابی قرار گیرد. هرچند، می‌توانید یک Database Snapshot را ایجاد کنید که بر طبق پایگاه داده قرینه باشد و آن Database Snapshot می‌تواند در حالت فقط خواندنی برای گزارشگیری مورد دستیابی قرار گیرد.

هر چند پایگاه داده در سرور قرینه نمی‌تواند مستقیماً مورد دستیابی قرار گیرد، زیرا در یک حالت بازیافت در حال پیشرفت است که بر Database Snapshot تأثیری نمی‌گذارد که به یک تصویر فوری از صفحات داده در پایگاه داده سرور قرینه دستیابی دارد. ایجاد یک Database Snapshot در سرور قرینه به شما امکان می‌دهد تا توان پردازشی سرور قرینه را با امکان انتقال گزارش ایستا به آن سرور بهتر به کار برید. هنگام استفاده از Database Snapshotها در یک Database Mirroring باید به رویداد یک failover توجه داشته باشید و Database Snapshotها موجودی که در سرور قرینه ایجاد شده‌اند، بی‌عیب و نقص باقی خواهند ماند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(c)
(d) **شکل ۴-۳ Database Mirroring و Database Snapshot یک سرور گزارشگیری را ایجاد می‌کنند.**

تشخیص این نکته حایز اهمیت است که Database Snapshotها یک ویژگی در دسترس بودن هستند و یک ویژگی failover آن‌ها روش‌های بیشتری برای دستیابی به داده شما فراهم می‌کنند و در دسترس بودن داده شما را افزایش می‌دهند. آن‌ها یک ویژگی failover نیستند (اگر پایگاه داده اصلی در دسترس نباشد، Database Snapshot نیز در دسترس نخواهد بود).

Section ۳۳،۰۲ Early Restore Access

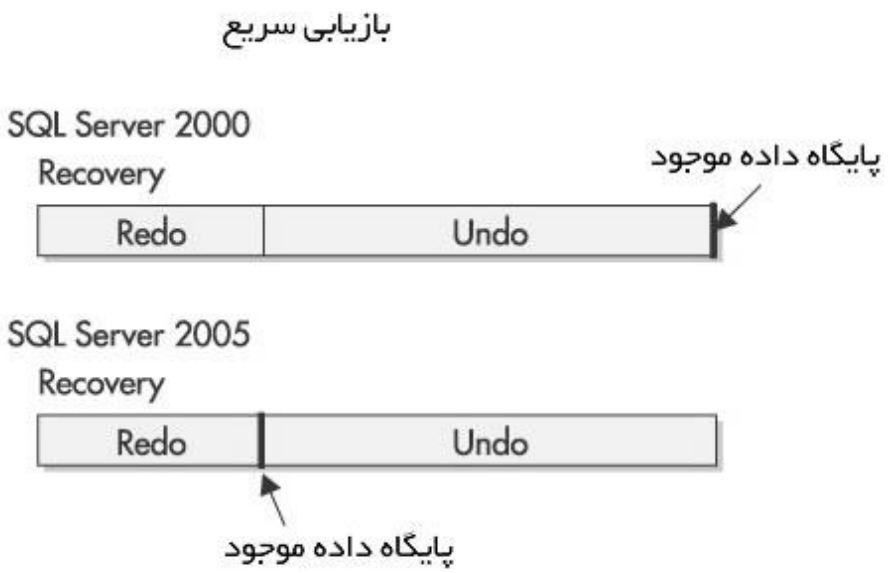
هر بار که پایگاه داده SQL Server شروع می‌شود یا یک پایگاه داده بازیابی می‌گردد، آن پایگاه داده باید به دوره‌ای از بازیافت برود که هر تراکنشی که در ثبت است، بر فایل‌های داده اعمال می‌شود. فاز بازیافت معمولاً redo گفته می‌شود. به محض این که بخش redo فرآیند بازیافت کامل می‌شود، تمام عملیات undo انجام می‌گردد، زیرا تمام تراکنش‌های ناقص در ثبت وقایع roll back می‌شوند. در SQL Server 2000، پایگاه داده در دسترس نبود، تا وقتی که تمام عملیات redo و undo کامل می‌شدند. اگر پایگاه داده‌ای مختل می‌شد، در حالی که فعال بود و آن پایگاه داده بسیار مشغول بود،

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

آن‌گاه قبل از این که پایگاه داده مجدداً در دسترس قرار گیرد، یک تأخیر طولانی را داشتیم، زیرا باید تا زمانی که تمام ورودی‌ها در ثبت وقایع پردازش می‌شدند، منتظر می‌ماندید.

Early Restore Access جدید SQL Server 2005 به پایگاه‌های داده امکان می‌دهد تا بلافاصله بعد از کامل شدن بخش redo فرآیند بازیافت، در دسترس قرار گیرد. در SQL Server 2005، هنگامی‌که پایگاه داده مجدداً شروع می‌شود، تمام تراکنش‌های بازی که در ثبت وقایع هستند، مجدداً انجام شده و سپس پایگاه داده بلافاصله در دسترس قرار می‌گیرد. نتیجه دقیق این است که پایگاه داده خیلی زودتر در دسترس قرار می‌گیرد. برای تراکنش‌هایی که Commit نشده باشند، SQL Server هنوز قفل‌هایی را روی صفحات داده مورد استفاده این تراکنش‌ها نگهداری می‌کند، بنابراین حتی اگر پایگاه داده در حال استفاده باشد، مستحکم باقی خواهند ماند. سپس SQL Server 2005 فرآیند undo کردن این تراکنش‌ها را در حالت فعال بودن پایگاه داده شروع می‌کند. کاربران می‌توانند عملیات خواندن/نوشتن پایگاه داده را در طی فاز undo بازیافت شروع کنند. هرچند، هر تلاشی در طی فرآیند Undo، برای دستیابی به داده‌ای که SQL Server قفل کرده است، موجب بلوکه شدن خواهد شد تا وقتی که فرآیند undo قفل‌های روی آن داده را رها کند. شکل ۵-۳ تفاوت در دسترس‌پذیری بین SQL Server 2000 و SQL Server 2005 را نشان می‌دهد.



(a)

شکل ۵-۳ Early Restore Access (b)

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

عمل RESTORE حالا سنجیده‌تر است و برد عمل redo را تعریف می‌کند. می‌توانید پشتیبان‌های کامل، جزیی یا گروه فایل را بازیابی کنید. اگر نشان دهید که می‌خواهید یک گروه فایل را بازیابی کنید، آن‌گاه تنها آن داده گروه فایل به عمل roll-forward اضافه می‌شود. اگر نشان دهید که می‌خواهید یک پشتیبان کامل را بازیابی کنید، آن‌گاه تمام داده در مجموعه پشتیبان، برای عمل redo استفاده خواهد شد.

Section ۳۳،۰۳ عملیات ایندکس به صورت online

در نگارش‌های قبلی SQL Server هنگامی که ایندکسی بازسازی می‌شد، نمی‌توانستید هیچ عمل بهنگام‌رسانی را روی جدول انجام دهید تا وقتی که بازسازی ایندکس خاتمه می‌یافت. ویژگی عملیات ایندکس online جدید SQL Server 2005، در دسترس بودن SQL Server را با دادن امکان بهنگام‌رسانی، درج و حذف ردیف‌ها از جدول در هنگام بازسازی ایندکس به برنامه‌ها، توسعه داده است. ویژگی ایندکس online جدید، این جادو را با حفظ دو کپی از ایندکس انجام می‌دهد: یکی که برنامه‌ها می‌توانند به استفاده از آن ادامه دهند و ایندکس موقت دومی که هنگام بازسازی ایندکس استفاده می‌شود. موتور SQL Server هر دو ایندکس را با تمام تغییرات هنگام انجام بازسازی نگهداری می‌کند. هنگام اتمام بازسازی، ایندکس قدیمی حذف شده و با ایندکس جدید جایگزین می‌شود. بازسازی ایندکس Online برای عبارات CREATE INDEX، ALTER INDEX REBUILD، ALTER INDEX DISABLE، DROP INDEX، ALTER TABLE ADD CONSTRAINT و ALTER TABLE DROP CONSTRAINT پشتیبانی می‌شود. هر چند SQL Server 2005 هم‌اکنون از بازسازی ایندکس Online پشتیبانی می‌کند، ولی سرباری اضافی را متحمل می‌شود، زیرا می‌توانید بازسازی ایندکس‌های خود را به صورت offline نیز داشته باشید.

لیست زیر نشان می‌دهد که چگونه ویژگی ایندکس‌گذاری online استفاده می‌شود:

```
CREATE INDEX MyIndex ON Person.Contact(LastName) WITH (ONLINE=ON)
```

در این‌جا می‌توانید ببینید که عبارت CREATE INDEX استاندارد برای ایجاد ایندکسی به نام MyIndex در ستون LastName جدولی به نام PersonContact استفاده شده است که در پایگاه داده AdventureWorks است. بخش ONLINE=ON به ایندکس امکان ایجاد به صورت Online را می‌دهد.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۳۳،۰۴ فرمت‌های Online سنجیده

ویژگی جدید دیگری که در دسترس بودن SQL Server 2005 را بهبود بخشیده است، توانایی انجام بازیابی‌های سنجیده است. در حالی که این ویژگی بازیابی‌های سطح جدولی که افراد مختلف از زمان SQL Server 6.5 تاکنون دیده‌اند، نیست، ویژگی بازیابی سنجیده در SQL Server 2005 به شما امکان بازیابی گروه فایل‌های منتخب را در یک پایگاه داده می‌دهد، در حالی که بقیه پایگاه داده به دسترس‌پذیری خود ادامه می‌دهد. در SQL Server 2000، واحد اصلی در دسترس بودن پایگاه داده است. تمام اجزای پایگاه داده باید سالم باشند، قبل از این که پایگاه داده در دسترس باشد. در SQL Server 2005، هم‌اکنون واحد در دسترس بودن گروه فایل است. SQL Server 2005 به شما امکان می‌دهد تا یک گروه فایل را در یک زمان یا حتی یک صفحه یا گروهی از صفحات را در یک زمان بازیابی کنید و بقیه پایگاه داده می‌تواند به در دسترس بودن ادامه دهد، مادامی که گروه فایل اصلی بالا باشد.

ردیابی صفحه آسیب دیده

یک ویژگی جدید دقیقاً مرتبط که با توانایی انجام بازیابی‌های سطح صفحه با SQL Server 2005 گره خورده است، توانایی ردیابی صفحات آسیب دیده است. در یک عمل خواندن، با صفحات بدی که مواجه می‌شویم، در یک جدول ردیابی می‌شوند و با استفاده از قابلیت بازیابی سنجیده، می‌توانید بازیابی را به صورت صفحه به صفحه بازیابی کنید، در حالی که پایگاه داده به صورت online باقی می‌ماند. هر تراکنشی که از داده یک صفحه آسیب دیده استفاده کند، roll back می‌شود. اگر صفحه بدی در طی rollback تراکنش برگردد، آن‌گاه پایگاه داده باید مجدداً شروع شود.

Section ۳۳،۰۵ اتصال راهبری اختصاصی

ویژگی جالب دیگر در SQL Server 2005 که به فراهم کردن دسترس‌پذیری بهتر کمک می‌کند، اتصال راهبری اختصاصی جدید است. اتصال راهبری اختصاصی، DBA را با دستیابی به سرور، بدون توجه به کار بار جاری سرور مهیا می‌کند. این امر به DBA امکان دستیابی به سرور و کشتن فرآیندهای خارج از کنترل را می‌دهد. برای شروع ابزار SQLCMD جدید در حالت راهبری اختصاصی، این فرمان را وارد کنید:

```
C:\Sqlcmd –A
```

برای کسب اطلاعات بیشتر درباره استفاده از اتصال راهبری اختصاصی به فصل ۱ مراجعه کنید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۳۳،۰۶ حافظه Hot-Plug

ویژگی در دسترس بودن جدید دیگری که می‌تواند از Windows Server 2003 مشتق شود، حافظه Hot-Plug است. حافظه Hot-Plug به شما اجازه می‌دهد تا RAM را در حالی اضافه کنید که سیستم در حال اجراست. همان‌گونه که ممکن است انتظار داشته باشید، این ویژگی نیاز به پشتیبانی OS مرتبط و محیط سخت‌افزاری دارد. در مورد SQL Server 2005، این بدان معنی است که SQL Server باید در Windows Server 2003 اجرا شود تا بتواند از این ویژگی در دسترس بودن بهره ببرد. اگر محیط از این ویژگی پشتیبانی کند، می‌توانید حافظه را اضافه کنید و SQL Server 2005 قادر خواهد بود به طور پویا RAM اضافی را تشخیص دهد، بدون این که نیاز به reboot سرور یا downtime داشته باشد. در حالی که حافظه Hot-Plug به شما اجازه می‌دهد تا به‌طور پویا RAM را اضافه کنید، امکان برداشتن آن میسر نیست.

Section ۳۳،۰۷ پیکربندی پویای بهبود یافته

تمام پیکربندی در SQL Server 2005 هم اینک پویاست، از جمله وابستگی I/O و CPU. در SQL Server 2005، هم‌اکنون می‌توانید این مقادیر را تغییر دهید و اثرات آن بلافاصله انجام خواهند شد.

دیگر نیازی به شروع مجدد سرور نیست. علاوه بر این، تغییراتی در AWE[1] نیز پویاست. SQL Server 2005 می‌تواند به‌طور خودکار تغییرات اندازه AWE فیزیکی را به‌طور پویا تنظیم کند. این ویژگی برای کار کردن در الحاق با توانایی استفاده از حافظه Hot-Plug طراحی شده است. این ویژگی نیاز به Windows Server 2003 دارد.

Section ۳۳،۰۸ سطوح جداسازی تراکنش SQL Server

ناحیه دیگری که بر قابلیت پایگاه داده تأثیر می‌گذارد، سطح جداسازی تراکنش است که توسط برنامه مورد استفاده قرار می‌گیرد. SQL Server 2005 سطح جدیدی از تراکنش به نام Snapshot Isolation را فراهم کرده است که انعطاف‌پذیری بیشتری را برای دستیابی داده به آن می‌دهد. چهار سطح جداسازی ANSI استانداردی که در نگارش‌های قبلی SQL Server از آن‌ها پشتیبانی می‌شد (Serializable، Repeatable، Read Committed، Read Incommitted) همگی از تراکنش‌های یکی دیگر با قفل‌گذاری روی ردیف‌های داده مرتبط محافظت می‌کنند، بنابراین سایر برنامه‌ها نمی‌توانند داده را تغییر دهند. هنوز هم از این سطوح تراکنش پشتیبانی می‌شود. Snapshot Isolation جدید در

1- Address Windowing Extensions

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

دسترس بودن داده بیشتری را برای خواندن برنامه‌ها فراهم می‌کند. با Snapshot Isolation، SQL Server 2005 نوعی از قفل‌گذاری بهینه را انجام می‌دهد که SQL Server هیچ قفلی را روی ردیف‌های موجود در یک تراکنش نمی‌گذارد. در عوض، رد حالت ردیف را هنگام باز شدن تراکنش حفظ می‌کند. هنگامی که یک تراکنش Snapshot باز می‌شود، سیستم لزوماً داده ردیف را برای تراکنش کپی می‌کند و به برنامه شما امکان می‌دهد تا مشاهده داده ردیف را ادامه دهد، زیرا در زمان شروع تراکنش قرار داشته است. هیچ قفلی روی ردیف‌ها قرار داده نمی‌شود. این امر خواندن‌های مستحکم بلوکه نشده را در یک برنامه OLTP ممکن می‌سازد. به دلیل این که Snapshot Isolation هیچ قفلی را نگهداری نمی‌کند، نویسنده‌های داده، خواننده‌ها و خواننده‌ها، نویسنده‌ها را بلوکه نمی‌کنند. سطح Snapshot Isolation جدید واقعاً برای برنامه‌هایی است که تأکید بر عملیات خواندن دارند. هنگامی که برنامه شما، ردیف‌ها را در یک تراکنش می‌خواند، ممکن است داده قدیمی را بخواند، زیرا سایر برنامه‌ها قادر به تغییر داده هستند. در حالی که این سطح جداسازی جدید، نوشتن‌ها را میسر نمی‌سازد، سربار اضافی در تمام بهنگام‌رسانی‌ها وجود دارد. Snapshot Isolation هنگامی مفید است که نیاز به یک دیدگاه مستحکم زمانی از تمام داده در پایگاه داده خود دارید. برای برنامه‌های بهنگام‌رسانی، Snapshot Isolation در روش‌هایی که هزینه قفل‌گذاری بیش از هزینه roll back کردن یک تراکنش است، بهترین کاربرد را دارد.

Article XXXIV. پشتیبان‌گیری و بازیابی

SQL Server 2005 هم‌چنین چند ویژگی جدید را در ناحیه فناوری پشتیبان‌گیری و بازیابی در برمی‌گیرد. در حالی که تعدادی از ویژگی‌های در دسترس بودن قبل مربوط به برخی از این ویژگی‌های پشتیبان‌گیری و بازیابی بود (به ویژه عملیات Online سنجیده)، این بخش در اصل بر تغییراتی تمرکز دارد که مایکروسافت برای مستحکم‌تر و آسان‌تر ساختن پیاده‌سازی فرآیند پشتیبان‌گیری و بازیابی صورت داده است.

Section ۳۴.۰۱ بازیابی جزیی

در SQL Server 2000، پایگاه داده در طی عمل بازیابی در دسترس نبود. در SQL Server 2005، پایگاه داده به محض بازیابی گروه فایل اصلی، online است. تنها داده بازیابی شونده در دسترس نیست. اگر کاربر به داده‌ای دستیابی داشته باشد که در گروه فایل اصلی باشد، عمل به صورت موفق انجام می‌شود، بدون این که چیزی به کاربر نشان داده شود و بازیابی بقیه پایگاه داده را بیان کند. اگر

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

کاربر تلاش کند تا به داده یک گروه فایل دسترسی داشته باشد که هم اینک در حال بازیابی است یا هنوز offline است، آن‌گاه پایگاه داده پیامی را برخواهند گرداند که بیانگر offline بودن داده است.

Section ۳۴،۰۲ بهبودهای قابلیت اتکای رسانه

SQL Server 2005 هم‌چنین بهبودهایی را در روش ارتباط با رسانه فراهم کرده است. به ویژه، هم اینک انجام پشتیبان‌گیری‌ها برای وسایل قرینه را میسر می‌سازد، جامعیت checksum بهبود یافته را فراهم می‌کند و ادامه عمل بازیابی را ممکن می‌سازد، حتی اگر با خطایی مواجه شود.

پشتیبان‌گیری قرینه‌سازی رسانه

یکی از ویژگی‌های قابلیت اتکای رسانه جدید در SQL Server 2005، توانایی پشتیبانی از قرینه‌سازی رسانه است. قرینه‌سازی رسانه به شما امکان می‌دهد تا به‌طور هم‌زمان یک پشتیبان‌گیری را برای چند وسیله پشتیبان‌گیری انجام دهید. مثلاً، می‌توانید از پایگاه داده خود برای tape1 و tabe2 در یک زمان پشتیبان‌گیری کنید و دو کپی یکسان از مجموعه پشتیبان داشته باشید. رسانه پشتیبان‌گیری اضافی، یک ضامن استاندارد است که می‌تواند به محافظت از سازمان شما از خطاهای رسانه‌ای کمک کند و کمک می‌کند تا از توانایی انجام یک بازیابی موفق مطمئن شوید. این مثال، کاربرد ویژگی قرینه‌سازی رسانه جدید را تشریح می‌کند:

```
BACKUP DATABASE AdventureWorks TO
 TAPE='\\.\tape1'
MIRROR TO
 TAPE='\\.\tape2'
WITH FORMAT, MEDIANAME = 'ADWBackup'
```

این مثال نشان می‌دهد که پایگاه داده AdventureWorks بر روی tape1 پشتیبان‌گیری کرده و بر روی tape2 قرینه‌سازی نموده است. کارآیی پشتیبان‌گیری متأثر از افزودن یک قرینه پشتیبان نیست. می‌توان تا چهار قرینه را بر یک پشتیبان اعمال کرد.

تأیید پشتیبان‌های بهبود یافته

در SQL Server 2000، استفاده از فرمان RESTORE VERIFYONLY تنها باعث می‌شود SQL Server نوار را بدون هیچ کار اضافه‌ای در فرآیند بازیابی بخواند. استفاده از فرمان RESTORE VERIFYONLY در SQL Server 2005 هر کاری در مورد بازیابی واقعی در رسانه بازیابی را به‌طور مختصر انجام می‌دهد و ایده بهتری در مورد امکان موفقیت داده در مجموعه پشتیبان به شما می‌دهد.

ادامه خطاهای بازیابی گذشته

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

بهبود قابلیت اتکای رسانه جدید دیگر، توانایی عمل بازیابی برای ادامه دادن خطاهای رسانه‌ای گذشته است. نگارش‌های قبلی SQL Server، در صورت مواجه شدن با خطا در طی فرآیند بازیابی، از عمل بازیابی صرف‌نظر می‌کردند. هرچند، در مواردی که این تنها نگارش رسانه موجود بود، ممکن بود مانع جدی برای بازیافت داده موجود باشد. با این ویژگی جدید، SQL Server 2005 ادامه فرآیند بازیابی را تا حد امکان میسر می‌سازد. و از خطاهای رسانه‌ای که با آن‌ها مواجه می‌شود، صرف‌نظر می‌کند. بعد از خاتمه فرآیند بازیابی، پایگاه داده می‌تواند به‌طور دستی مرمت شود.

Section ۳۴.۰۳ Database Page Checksums

ویژگی Database Page Checksums جدید به پایگاه داده امکان می‌دهد تا خطاهای I/O و دیسک را که توسط زیرسیستم دیسک گزارش نشده‌اند، تشخیص دهد. هنگامی که ویژگی Database Page Checksums فعال می‌شود، SQL Server یک مقدار Checksum را هنگام نوشتن صفحه روی دیسک محاسبه کرده و آن Checksum را به همراه داده می‌نویسد. هنگامی که صفحه خوانده می‌شود، Checksum دیگری محاسبه شده و با Checksum اولی که با داده نوشته شده است، مقایسه می‌کند. اگر آن دو متفاوت باشند، خطایی گزارش می‌شود. فرمان زیر نشان می‌دهد که چگونه Database Page Checksums می‌تواند به یک پایگاه داده موجود اضافه شود:

```
ALTER DATABASE AdventureWorks
 SET PAGE_VERIFY CHECKSUM
```

Section ۳۴.۰۴ پشتیبان‌های log و پایگاه داده هم‌زمان

ویژگی جدید دیگر در ناحیه پشتیبانی SQL Server 2005، توانایی انجام هم‌زمان پشتیبان‌های log و پایگاه داده است. با استفاده از SQL Server 2000، باید منتظر پشتیبان‌گیری log بمانید تا وقتی که پشتیبان‌گیری پایگاه داده کامل شود. در SQL Server 2005، پشتیبان‌های پایگاه داده دیگر پشتیبان‌های block log نیستند. در ضمن، هم‌چنین می‌توانید از فایل‌ها و گروه‌های فایل در همان زمانی پشتیبان‌گیری کنید که از log تراکنش پشتیبان‌گیری می‌کنید. هرچند، محدود به انجام پشتیبان‌گیری از یک فایل داده در زمانی در هر پایگاه داده هستید.

Section ۳۴.۰۵ پشتیبان‌گیری از کاتالوگ‌های Full-Text

یک بهبود پشتیبان‌گیری دیگری که بخشی از SQL Server 2005 است، توانایی پشتیبان‌گیری از کاتالوگ‌های Full Text است. در SQL Server 2000، داده کاتالوگ Full Text خارج از SQL Server توسط سرویس جستجوی مایکروسافت نگهداری می‌شد. پشتیبان‌گیری از پایگاه داده SQL

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Serverای که حاوی داده Full-Text است که به‌طور خودکار موجب پشتیبان‌گیری از کاتالوگ Full-Text نمی‌شود. این امر امکان خارج از هم زمانی بودن کاتالوگ را در مورد داده Full-Text بازیابی شونده مطرح می‌کند. SQL Server 2005 این مشکل را با داشتن توانایی پشتیبان‌گیری خودکار از تمام کاتالوگ‌های Full-Text خارجی در همان زمان پشتیبان‌گیری از پایگاه داده برطرف می‌کند. این امر اطمینان می‌دهد که کاتالوگ Full-Text و داده بین عملیات پشتیبان‌گیری و بازیابی مستحکم باقی می‌مانند. فرامین DATABASE ATTACH و DEATTACH نیز دارای گزینه‌ای برای داشتن تمام کاتالوگ‌های Full-Text هستند. لینک‌ها در پایگاه داده به فایل‌های خارجی مورد استفاده اشاره می‌کنند. هنگامی که از پایگاه داده پشتیبان‌گیری می‌شود، از این فایل‌های خارجی نیز پشتیبان‌گیری می‌شود که استحکام پایگاه داده را اطمینان می‌دهد.

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

فصل چهارم

ویژگی‌های قابل برنامه‌نویسی

ویژگی‌های برنامه‌نویسی جدید در SQL Server 2005 تجمعی از تلاش‌های چندین ساله گروه برنامه‌نویسی SQL Server و گروه برنامه‌نویسی .NET Framework است. مهم‌ترین این ویژگی‌های برنامه‌نویسی جدید، یکپارچگی CLR[1] .NET است. یکپارچگی CLR آن را به میزبان کاملی از قابلیت‌های جدید تبدیل کرده است، از جمله قابلیتی برای ایجاد اشیای پایگاه داده با استفاده از هر یک

1- Common Language Run-time

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

از زبان‌های سازگار با NET.، از جمله C#، VB.NET و Managed C++. در این فصل، مقدمه‌ای بر این ویژگی‌های یکپارچگی NET CLR. جدید خواهیم داشت و مثال‌هایی را خواهید دید که نحوه استفاده آن‌ها را نشان می‌دهند. سپس، این فصل به عنوانی می‌پردازد که برای برنامه‌نویسان DBAهای SQL Server آشناتر است: ویژگی‌های جدید در T-SQL. سپس، این فصل مروری بر سمت کلاینت خواهد داشت و تعدادی از ویژگی‌های جدید برنامه‌نویسی در NET Framework Data Provider. بهنگام شده برای SQL Server را که همراه با SQL Server 2005 هستند، ارایه می‌دهد.

Article XXXV. یکپارچگی CLR

بدون شک مهم‌ترین ویژگی جدید در SQL Server 2005، یکپارچگی Microsoft .NET Framework است. یکپارچگی CLR با SQL Server، قابلیت SQL Server را به روش‌های مهم مختلفی توسعه می‌دهد. در حالی که T-SQL، زبان دستکاری و دستیابی داده موجود، برای عملیات دستیابی داده مجموعه‌گرا مناسب است، محدودیت‌هایی نیز دارد. T-SQL که بیش از یک دهه از طراحی آن می‌گذرد، یک زبان رویه‌ای است و نه یک زبان شئ‌گرا. یکپارچگی CLR با SQL Server 2005، توانایی ایجاد اشیای پایگاه داده را با استفاده از زبان‌های شئ گرایی نظیر VB.NET و C# به آن می‌دهد. در حالی که این زبان‌ها همان قدرت طبیعت مجموعه‌گرا را مثل T-SQL ندارند، از منطق پیچیده‌ای پشتیبانی می‌کنند، قابلیت‌های محاسبه بهتری دارند، دستیابی آسان‌تری به منابع خارجی فراهم می‌کنند، استفاده مجدد کد را تسهیل می‌بخشند و یک محیط برنامه‌نویسی کلاس بالا دارند که توانی بیش از Query Analyzer قدیمی فراهم می‌کند.

یکپارچگی NET CLR. با SQL Server 2005، برنامه‌نویسی رویه‌های ذخیره شده، توابع تعریف شده کاربر، تریگرها، انبوهه‌ها و انواع تعریف شده کاربر را با استفاده از هر یک از زبان‌های NET. ممکن می‌سازد. یکپارچگی NET CLR. با SQL Server 2005 بیش از یک مسأله سطحی است. در واقع، موتور پایگاه داده SQL Server 2005 میزبان CLR در فرآیند است. با استفاده از مجموعه‌ای از APIها، موتور SQL Server تمام مدیریت حافظه را برای برنامه‌های CLR میزبانی شده انجام می‌دهد.

کد مدیریت شده با استفاده از ADO.NET در الحاق با NET Data Provider. SQL Server به پایگاه داده دستیابی دارد. یک شئ SQL Server جدید به نام اسمبلی[1] واحد توزیع برای اشیای NET. با پایگاه داده است. برای ایجاد اشیای پایگاه داده CLR، ابتدا باید یک DLL با استفاده از Visual Studio 2005 ایجاد کنید. سپس آن DLL را به عنوان یک اسمبلی در SQL Server وارد کنید. بالاخره، آن اسمبلی را به یک شئ پایگاه داده نظیر یک رویه ذخیره شده یا تریگر لینک کنید. در بخش

1- Assembly

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

بعد، نحوه استفاده واقعی از ویژگی‌های CLR را در SQL Server 2005 به‌طور مفصل‌تر بررسی می‌کنیم.

Section ۳۵.۰۱ اسمبلی‌ها

برای ایجاد اشیای پایگاه داده NET.، باید کد مدیریت شده‌ای بنویسید و آن را در یک اسمبلی NET. کامپایل کنید. متداول‌ترین روش انجام این کار، استفاده از Visual Studio 2005 و سپس ایجاد یک پروژه SQL Server جدید و کامپایل آن در یک DLL است. جزییات بیشتر درباره نحوه ایجاد یک پروژه کد مدیریت شده جدید با Visual Studio 2005، در بخش "رویه‌های ذخیره شده NET." در ادامه فصل ارایه می‌شود.

بعد از ایجاد اسمبلی، می‌توانید آن را با استفاده از فرمان T-SQL CREATE ASSEMBLY در SQL Server بارگذاری کنید، همان‌گونه که در این‌جا می‌بینید:

```
CREATE ASSEMBLY MyCLRDLL
FROM '\\SERVERNAME\CodeLibrary\MyCLRDLL.dll'
```

فرمان CREATE ASSEMBLY پارامتری می‌گیرد که حاوی مسیری برای DLLای است که در SQL Server بارگذاری خواهد شد. این مسیر می‌تواند یک مسیر محلی باشد، ولی اغلب مسیری به یک اشتراک فایل شبکه‌ای خواهد بود. هنگامی که CREATE ASSEMBLY اجرا می‌شود، DLL در پایگاه داده اصلی کپی می‌شود.

اگر یک اسمبلی بهنگام شده یا کنار گذاشته شود، آن‌گاه می‌توانید با استفاده از فرمان DROP ASSEMBLY به این صورت آن را حذف کنید:

```
DROP ASSEMBLY MyCLRDLL
```

به دلیل این که اسمبلی‌ها در پایگاه داده قرار دارند، هنگامی که کد منبع برای آن اسمبلی اصلاح شده و اسمبلی مجدداً کامپایل می‌شود، این اسمبلی ابتدا باید با استفاده از فرمان DROP ASSEMBLY از این پایگاه داده حذف شده و سپس با استفاده از فرمان CREATE ASSEMBLY قبل از انعکاس بهنگام‌رسانی‌ها در اشیای پایگاه داده SQL Server، مجدداً بارگذاری شود.

می‌توانید از دیدگاه sys.assemblies برای مشاهده اسمبلی‌هایی استفاده کنید که به SQL Server 2005 اضافه شده‌اند، همان‌گونه که در این‌جا نشان داده شده است:

```
SELECT * FROM sys.assemblies
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

با توجه به این که اسمبلی‌ها با استفاده از فایل‌های خارجی ایجاد می‌شوند، ممکن است بخواهید فایل‌های مورد استفاده برای ایجاد این اسمبلی‌ها را نیز مشاهده کنید. می‌توانید این کار را با استفاده از دیدگاه sys.assembly_files انجام دهید، به این صورت:

```
SELECT * FROM sys.assembly_files
```

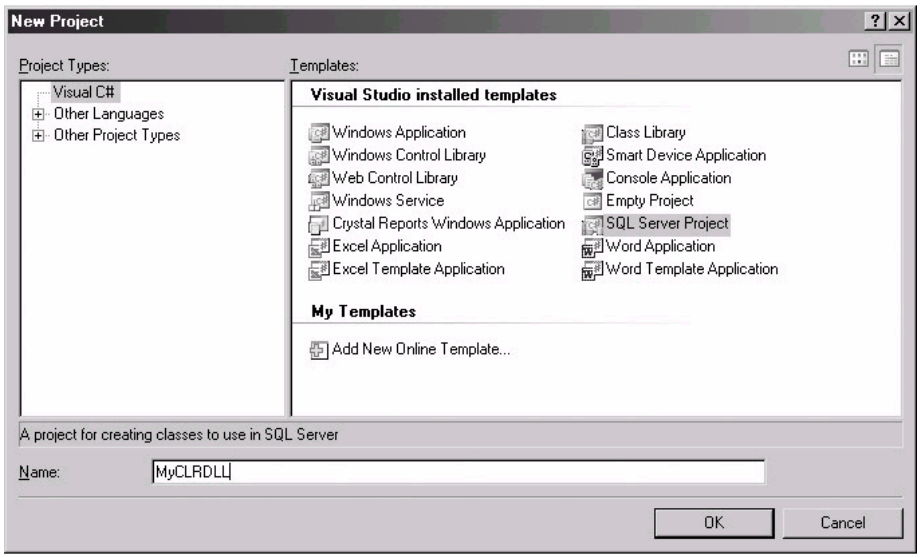
Section ۳۵،۰۲ SQL Server .NET Data Provider

اگر با ADO.NET آشنا هستید، شاید متعجب شوید که چگونه این رویه‌های پایگاه داده CLR جدید به پایگاه داده متصل می‌شوند. روی هم رفته، ADO.NET اتصال پایگاه داده خود را با استفاده از تأمین کننده‌های داده NET. مبتنی بر کلاینت نظیر NET Framework Data Provider. برای SQL Server که به پایگاه داده SQL Server متصل می‌شود یا NET Framework Data Provider. برای Oracle که برنامه‌های ADO.NET را به پایگاه‌های داده Oracle متصل می‌کند. در حالی که این امر برای یک برنامه شبکه‌ای عالی است، حرکت در بین کتابخانه‌های شبکه‌ای، کارآمدترین حالت اتصال برای کدی نیست که مستقیماً روی سرور اجرا می‌شود، برای رفع این مشکل، مایکروسافت SQL Server .NET Data Provider را فراهم کرده است. SQL Server .NET Data Provider، یک اتصال درون حافظه‌ای برای پایگاه داده SQL Server تولید می‌کند.

مرجعی به SQL Server .NET Data Provider جدید، به‌طور خودکار هنگام ایجاد یک پروژه SQL Server جدید، به برنامه‌های Visual Studio 2005 شما اضافه می‌شود. برای ایجاد یک نوع پروژه SQL Server جدید، ابتدا Visual Studio 2005 را باز کرده و سپس گزینه File | New Project را از منوی انتخاب کنید. سپس، از کادر محاوره‌ای New Projects، نوع پروژه را انتخاب کرده (مثلاً، Visual Basic Projects، Visual C# Projects) و سپس در لیست قالب‌ها حرکت کنید تا وقتی که قالب پروژه SQL Server نشان داده شده در شکل ۱-۴ را ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

(b)

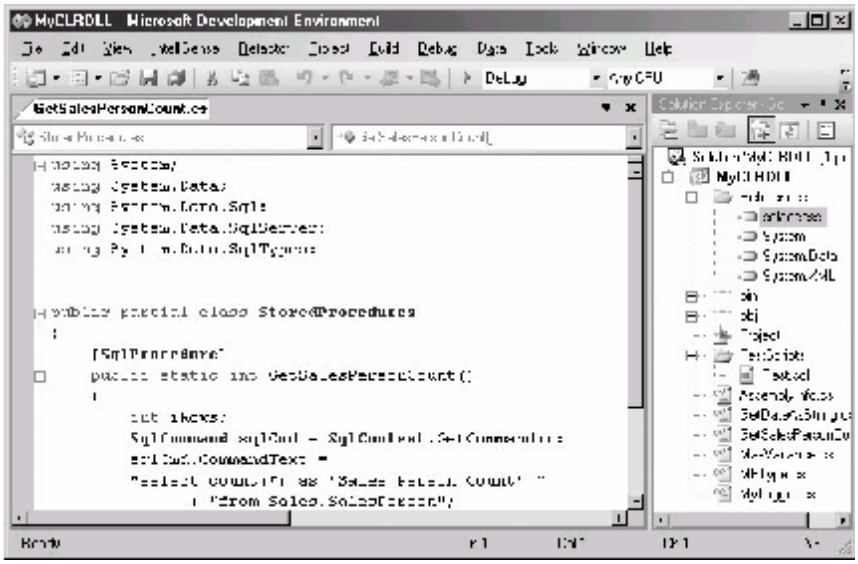
شکل ۱-۴ ایجاد یک پروژه SQL Server جدید با Visual Studio 2005

توجه : از ایجاد پروژه‌ها SQL Server در Visual Studio 2005 پشتیبانی می‌شود (کدی به نام Whidbey) که برای عرضه در همان زمان SQL Server 2005 برنامه‌ریزی شده است.

بعد از انتخاب قالب پروژه SQL Server، نامی به پروژه خود داده و OK را برای ایجاد پروژه کلیک کنید. تمام مراجع مورد نیاز به‌طور خودکار به پروژه SQL Server شما اضافه می‌شوند. SQL Server .NET Data Provider به عنوان مرجع SQLaccess اضافه می‌شود که می‌توانید در شکل ۴-۲ ببینید که متمایز شده است. علاوه بر این، می‌توانید مرجع System.Data را ببینید که پشتیبانی برای ADO.NET و اشیای داده‌گرای آن از قبیل DataSet و انواع داده SQL Server فراهم می‌کند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(c)

شکل ۲-۴ مرجع SQL Server .NET Data Provider (d)

علاوه بر افزودن مراجع مناسب، یکی از مهم‌ترین مواردی که قالب‌های Visual Studio 2005 SQL Server به‌طور خودکار برای شما انجام می‌دهند، اضافه کردن رهنمودهای ورود صحیح است. هنگامی که اشیای پایگاه داده SQL Server CLR را ایجاد می‌کنید، باید مطمئن باشید که یک عبارت ورود برای فضای نام System.Data.SQLServer در پروژه خود داشته باشید. فضای نام System.Data.SQLServer حاوی کلاس‌های NET. ای است که SQL Server .NET Data Provider را تشکیل می‌دهند. رهنمود ورود به شما امکان می‌دهد تا به کلاس‌هایی در فضای نام System.Data.SQLServer با استفاده از نام کلاس کوتاه آن‌ها مراجعه کنید و نه نام‌های کاملاً معین طولانی که همیشه نام فضای نام قبل از آن‌ها قرار گرفته باشد (مثلاً، System.Data.SQLServer). برای یک پروژه #C، رهنمود ورود چنین است:

```
using System.Data.SqlServer;
```

برای یک پروژه VB.NET، فضای نام System.Data.SQLServer چنین است:

```
Imports System.Data.SqlServer
```

برخلاف پروژه‌های ADO.NET مناسب که باید به‌طور صریح اتصالی به یک نمونه SQL Server معین با استفاده از متد Open شئ اتصال باز کند، SQL Server .NET Data Provider به‌طور ضمنی

www.IrPDF.com www.IrPDF.com www.IrPDF.com

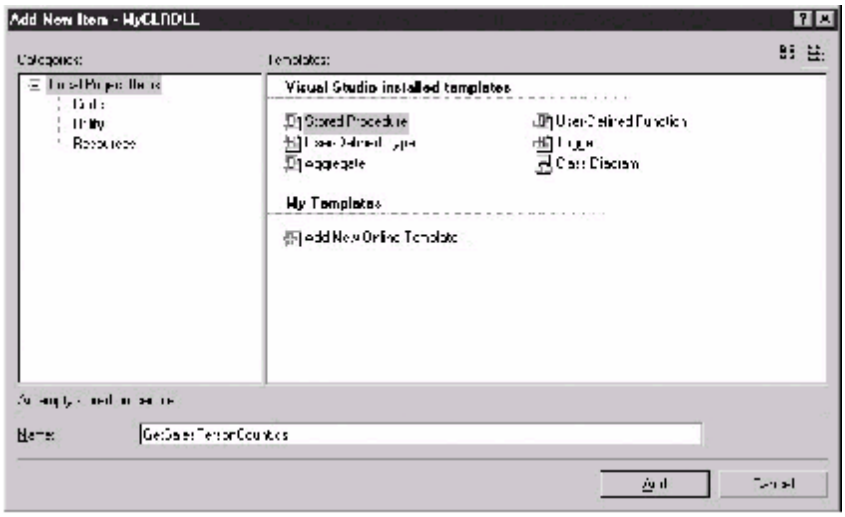
www.IrPDF.com

اتصالی به سیستم SQL Server محلی به وجود می‌آورد و همان‌گونه که در مثال‌های بعد خواهید دید، هیچ نیازی به ایجاد یک شئ ADO.NET Connection و احضار متد Open آن نیست.

با صراحت می‌گوییم که Visual Studio 2005 برای ایجاد اشیای پایگاه داده NET. برای SQL Server 2005 مورد نیاز نیست. می‌توانید اشیای پایگاه داده CLR را با استفاده از NET Framework. 2.0 و NET Framework SDK. توسعه دهید. هرچند، Visual Studio 2005 قالب‌های پروژه و گزینه‌های توزیع پروژه‌ای را فراهم می‌کند که مزایای مهمی نسبت به ایجاد دستی این اشیا با استفاده تنها از NET SDK. به آن می‌دهد.

Section ۳۵،۰۳ رویه‌های ذخیره شده NET.

رویه‌های ذخیره شده[1]، یکی از محتمل‌ترین اشیای پایگاه داده هستند که بخواهید با استفاده از یکی از زبان‌های NET. مدیریت شده ایجاد کنید، زیرا رویه‌های ذخیره شده اغلب حاوی منطقی پیچیده و قوانین تجاری نمایان هستند که برای بیان در T-SQL مشکل هستند. برای ایجاد یک رویه ذخیره شده CLR در Visual Studio 2005، می‌توانید از گزینه Project | Add Stored Procedure برای نمایش کادر محاوره‌ای قالب‌های نصب شده Visual Studio استفاده کنید که در شکل ۳-۴ نشان داده شده است.



(a)

شکل ۳-۴ افزودن یک رویه ذخیره شده CLR (b)

1- Stored Procedures

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

از کادر محاوره‌ای Add New Item، گزینه Stored Procedure را از لیست قالب‌های نمایش داده شده در لیست Templates انتخاب کرده و سپس نام رویه ذخیره شده را در فیلد Nameای فراهم کنید که می‌توانید در پایین صفحه نمایش ببینید. در این‌جا، می‌توانید ببینید که رویه ذخیره شده به نام GetSalesPersonCount است. Visual Studio 2005 کلاس جدیدی را برای رویه ذخیره شده به پروژه شما اضافه می‌کند. فایل کلاس تولید شده بعد از نام رویه ذخیره شده نام‌گذاری شده و شامل تمام رهنمودهای ورود مورد نیاز و کد برجسته‌ای است که رویه ذخیره شده را نام‌گذاری می‌کند. پر کردن بقیه کد که کار رویه ذخیره شده را می‌سازد، بر عهده شماست. این مثال کد منبع مورد نیاز برای ایجاد یک رویه ذخیره شده CLR را تشریح می‌کند:

```
using System;
using System.Data;
using System.Data.Sql;
using System.Data.SqlServer;
using System.Data.SqlTypes;

public partial class StoredProcedures
{
   [SqlProcedure]
   public static int GetSalesPersonCount()
   {
      int iRows;
      SqlCommand sqlCmd = SqlContext.GetCommand();
      sqlCmd.CommandText =
      "SELECT COUNT(*) AS 'Sales Person Count' "
          + "FROM Sales.SalesPerson";
      iRows = (int)sqlCmd.ExecuteScalar();
      return iRows;
   }
};
```

اولین نکته مهم که در این کد توجه برانگیز است، رهنمودی است که فضای نام System.Data.SQLServer را وارد می‌کند. این امر به پروژه MyCLRDLL امکان استفاده از SQL Server .NET Data Provider را بدون نیاز همیشگی به مراجعه به نام کاملاً معین می‌دهد. دومین نکته توجه برانگیز صفت [SQLProcedure] است که قبل از نام متد قرار گرفته است و به کامپایلر می‌گوید که این متد به عنوان یک رویه ذخیره شده SQL Server ارایه می‌شود. سپس، می‌توانید ببینید که نام کلاس پیش‌فرض برای این رویه ذخیره شده با StoredProcedure تنظیم شده است. این کلاس حاوی یک متد ایستا به نام GetSalesPersonCount است که یک نوع داده int را برمی‌گرداند. برای C#، این متد باید به صورت ایستا تعریف شود. برای کد VB.NET، این متد باید به صورت

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Shared تعریف شود. کد این‌جا لزوماً تعداد ردیف‌های جدول Sales.SalesPerson را در پایگاه داده AdventureWorks نمونه بازیابی می‌کند. توجه داشته باشید که در این کلاس، متد Open استفاده نمی‌شود. این یک انحراف آزاد برای کد ADO.NET مبتنی بر کلاینت است. در عوض، SQL Server .NET Data Provider به‌طور خودکار اتصالی به سرور محلی باز می‌کند.

بعد از کامپایل کد منبع رویه ذخیره شده CLR در یک اسمبلی، می‌توانید آن اسمبلی را به پایگاه داده اضافه کرده و رویه ذخیره شده CLR را ایجاد کنید. می‌توانید این کار را به دو روش انجام دهید: اگر در Visual Studio 2005 برنامه‌نویسی می‌کنید، آن‌گاه می‌توانید از گزینه Build | Deploy Solution برای نصب رویه ذخیره شده CLR جدید در پایگاه داده SQL Server استفاده کنید. یا می‌توانید مراحل توزیع را به‌طور دستی انجام دهید. برای کمک به درک نحوه استفاده از اشیای CLR در پایگاه داده، مراحل توزیع دستی را در بخش بعد ببینید. بعد از تولید DLL، مرحله بعد استفاده از آن DLL برای ایجاد یک شئ SQL Server جدید به نام اسمبلی است. این کد ایجاد یک اسمبلی را برای MyCLRDLL.DLL تشریح می‌کند:

```
CREATE ASSEMBLY MyCLRDLL
FROM '\\MyFileShare\Code Library\MyCLRDLL.dll'
```

فرمان CREATE ASSEMBLY از اولین آرگومان برای نام‌گذاری اسمبلی استفاده می‌کند. در این‌جا، آن را MyCLRDLL نامیده‌ایم که شبیه نام NET DLL. واقعی است، ولی استفاده از نام‌های مشابه الزامی نیست. آرگومان بعد از بخش FROM به عبارت CREATE ASSEMBLY می‌گوید که DLL فیزیکی در دیسک کجا قرار دارد. این مکان می‌تواند درایو محلی یا یک مسیر UNC باشد.

توجه : در زمان نوشتن این کتاب، اولین باری که فرمان CREATE ASSEMBLY را به‌طور دستی اجرا کردم، Microsoft.VisualStudio.DataTools.SQLAttributes.dll نیز باید در همان دایرکتوری حاوی NET DLL. ای باشد که باید به عنوان یک اسمبلی اضافه کنید.

هنگام ایجاد اسمبلی، DLL در پایگاه داده SQL Server مقصد کپی شده و اسمبلی رجیستر می‌شود. این کد ایجاد رویه ذخیره شده GetSalesPersonCount را تشریح می‌کند که از اسمبلی MyCLRDLL استفاده می‌کند:

```
CREATE PROCEDURE GetSalesPersonCount
AS EXTERNAL NAME
MyCLRDLL.StoredProcedures.GetSalesPersonCount
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

بخش EXTERNAL NAME در SQL Server 2005 جدید است. در این‌جا، بخش EXTERNAL NAME مشخص می‌کند که رویه ذخیره شده GetSalesPersonCount با استفاده از یک اسمبلی NET. ایجاد خواهد شد. یک اسمبلی می‌تواند حاوی چندین کلاس و متد باشد؛ عبارت EXTERNAL NAME از این ساختار دستوری برای تعیین متد و کلاس صحیح مورد استفاده اسمبلی استفاده می‌کند:

Assembly Name.ClassName.MethodName

در مورد مثال قبل، اسمبلی رجیستر شده MyCLRDLL نامیده می‌شود. کلاس در اسمبلی StoredProcedure است و متد در کلاسی که اجرا خواهد شد، GetSalesPersonCount است.

بعد از ایجاد رویه ذخیره شده CLR، می‌تواند به‌طور صریح شبیه هر رویه ذخیره شده T-SQL فراخوانی شود، همان‌گونه که در این مثال نشان داده شده است:

```
DECLARE @mycount INT
EXEC @mycount = GetSalesPersonCount
PRINT @mycount
```

Section ۳۵،۰۴ توابع تعریف شده کاربر NET.

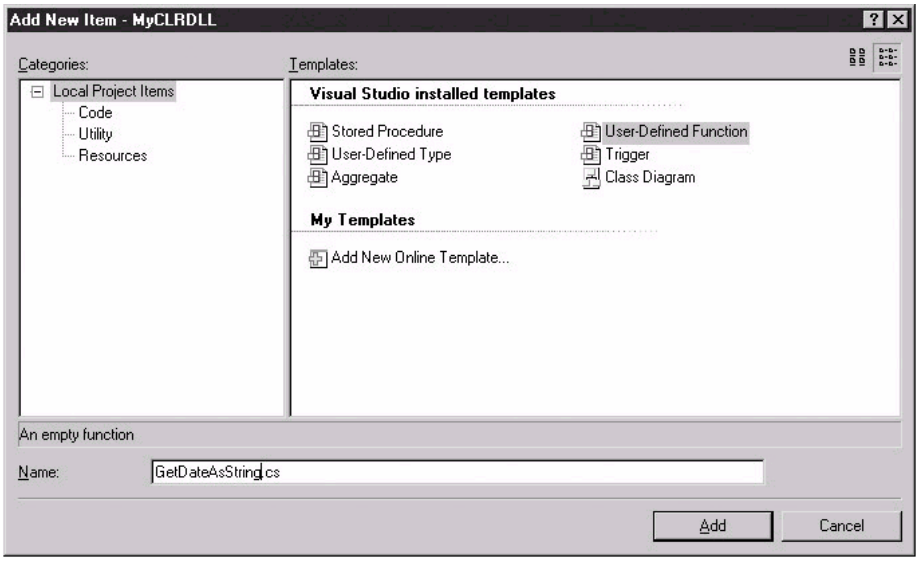
ایجاد توابع تعریف شده کاربر[1] (UDFها)، ویژگی جدید دیگری است که با یکپارچگی NET. CLR امکانپذیر شده است. توابع تعریف شده کاربر که انواع عددی را برمی‌گردانند، باید یک نوع داده NET. را برگردانند که می‌تواند به‌طور ضمنی به یک نوع داده SQL Server تبدیل شود. توابع عددی نوشته شده با NET Framework. ، می‌توانند به‌طور مهمی، T-SQL را در روش‌های خاصی انجام دهند، زیرا برخلاف توابع T-SQL، توابع NET. با استفاده از کد کامپایل شده ایجاد می‌شوند. توابع تعریف شده کاربر، هم‌چنین می‌تواند انواع جدول را برگرداند که در مورد این تابع، باید یک مجموعه نتیجه را برگرداند.

برای افزودن یک UDF با استفاده از Visual Studio 2005، می‌توانید از گزینه منوی Project | Add User-Defined Function که در شکل ۴-۴ نشان داده شده است، استفاده می‌کند.

1- User-defined functions

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

شکل ۴-۴ افزودن یک تابع تعریف شده کاربر CLR (b)

می‌توانید این را به یک پروژه موجود اضافه کنید، همان‌گونه که من انجام داده‌ام (آن را به MyCLRDLL نمونه که در مثال قبل ایجاد کردیم، اضافه کنید) یا یک پروژه SQL Server جدید را ایجاد کنید. مثال بعد، یک UDF ساده به نام GetDateAsString را نشان می‌دهد که یک تبدیل تاریخ به رشته پایه را انجام می‌دهد:

```
using System;
using System.Data.Sql;
using System.Data.SqlTypes;

public partial class UserDefinedFunctions
{
   [SqlFunction]
   public static SqlString GetDateAsString()
   {
      DateTime CurrentDate = new DateTime();
      return CurrentDate.ToString();
   }
};
```

در این‌جا توجه داشته باشید که فضای نام System.Data.SQLServer مورد نیاز نیست، همان‌گونه که این تابع خاص هیچ دستیابی داده‌ای را انجام نمی‌دهد. سپس، می‌توانید ببینید که به‌طور پیش‌فرض کلاس UserDefinedFunctions را ایجاد کرده است تا حاوی تمام متدهایی باشد که این

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

اسمبلی به عنوان UDF ارایه خواهد کرد. هم‌چنین می‌توانید ببینید که صفت [SQLFunction] برای تعیین متد GetDateAsString به عنوان یک UDF استفاده می‌شود. کد این‌جا تاریخ سیستم را به یک نوع داده رشته‌ای تبدیل می‌کند.

برای ایجاد تابع، اسمبلی ابتدا باید همانند آن‌چه که در مثال رویه ذخیره شده بیان شد، ایجاد شود. اگر از Visual Studio 2005 استفاده می‌کنید، می‌توانید گزینه Build | Deploy Solution را انتخاب کنید. اگر این کار را به‌طور دستی انجام می‌دهید و این در یک اسمبلی با اشیای CLR دیگر است، ابتدا باید این اشیا را حذف کنید، سپس اسمبلی را حذف کرده و بالاخره اسمبلی و اشیا را مجدداً ایجاد کنید. با توجه به این که این متد به MyCLRDLL اضافه شده است که هم‌اکنون در یک اسمبلی و یک رویه ذخیره شده استفاده می‌شود، عبارت DROP PROCEDURE و عبارت DROP ASSEMBLY باید ابتدا اجرا شوند تا اشیای پایگاه داده وابسته قبل از بارگذاری مجدد NET. DLL بهنگام شده در یک اسمبلی حذف شوند. بعد از حذف اشیای موجود، عبارت CREATE ASSEMBLY زیر می‌تواند برای بارگذاری مجدد DLL جدید اجرا شود:

```
CREATE ASSEMBLY MyCLRDLL
FROM '\\MyFileShare\Code Library\MyCLRDLL.dll'
```

سپس عبارت CREATE FUNCTION برای ایجاد تابع SQL Server جدیدی استفاده می‌شود که متد مناسب را در اسمبلی اجرا می‌کند. لیست زیر نحوه ایجاد یک تابع تعریف شده کاربر NET. توسط عبارت CREATE FUNCTION را نشان می‌دهد:

```
CREATE FUNCTION GetDateAsString()
RETURNS nvarchar(256)
EXTERNAL NAME
MyCLRDLL.UserDefinedFunctions.GetDateAsString
```

برای توابع تعریف شده کاربر، عبارت CREATE FUNCTION با بخش EXTERNAL NAME توسعه یافته است که لزوماً نام تابع تعریف شده کاربر را به متد مناسب در اسمبلی NET. لینک می‌کند. در این مثال، تابع GetDateAsString از اسمبلی به نام MyCLRDLL استفاده می‌کند. در آن اسمبلی، از کلاس UserDefinedFunctions و متد GetDateAsString در آن کلاس استفاده می‌شود.

بعد از ایجاد تابع، می‌تواند شبیه یک تابع عادی SQL Server فراخوانی شود. می‌توانید نحوه اجرای تابع GetDateAsString را در این مثال ببینید:

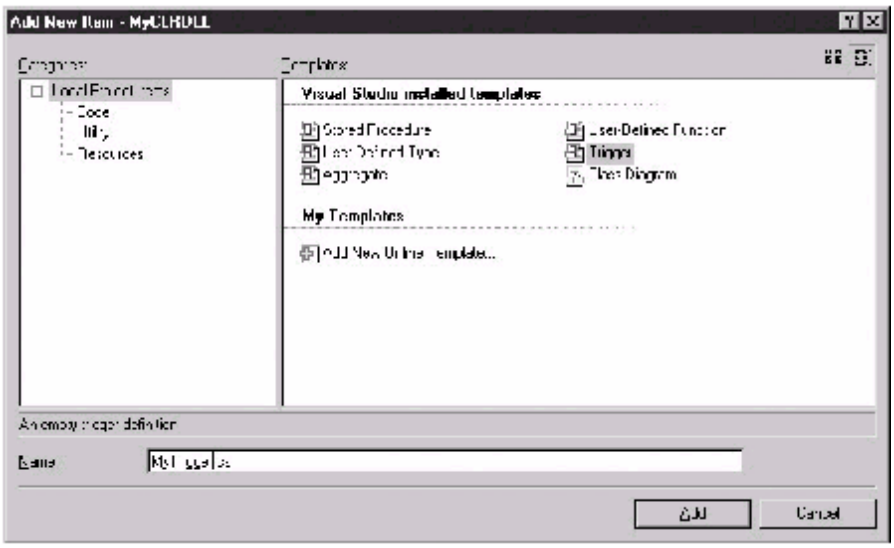
```
SELECT dbo.GetDateAsString()
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۳۵،۰۵ تریگرهای NET.

علاوه بر رویه‌های ذخیره شده و توابع تعریف شده کاربر، قابلیت‌های یکپارچگی NET. در SQL Server 2005، هم‌چنین توانایی ایجاد تریگرهای تعریف شده کاربر[1] (UDTهای) NET. را فراهم می‌کند. برای افزودن یک UDT با استفاده از Visual Studio 2005، می‌توانید از گزینه منوی Project | Add Trigger استفاده کنید، همان‌گونه که در شکل ۵-۴ نشان داده شده است.



(a)

شکل ۵-۴ افزودن یک تریگر CLR (b)

همانند اشیای پایگاه داده CLR دیگر، گزینه Trigger را از لیست قالب‌ها انتخاب کرده و سپس نام تریگر را در اعلام نام مهیا کنید. Visual Studio 2005 یک فایل برجسته را تولید خواهد کرد که می‌توانید به کد خود اضافه کنید. لیست کد مثال بعد یک تریگر CLR ساده به نام MyTrigger را تشریح می‌کند:

```
using System;
using System.Data;
using System.Data.Sql;
using System.Data.SqlServer;
using System.Data.SqlTypes;

public partial class Triggers
{
```

1- User-defined triggers

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
    // Enter existing table or view for the target
    //  and uncomment the attribute line
    [SqlTrigger (Name="MyTrigger",
     Target="Person.ContactType", Event="FOR INSERT")]
    public static void MyTrigger()
    {
        SqlTriggerContext oTriggerContext =
            SqlContext.GetTriggerContext();
        SqlPipe sPipe = SqlContext.GetPipe();
        SqlCommand sqlCmd = SqlContext.GetCommand();
        if (oTriggerContext.TriggerAction == TriggerAction.Insert)
        {
            sqlCmd.CommandText = "SELECT * FROM inserted";
            sPipe.Execute(sqlCmd);
         }
    }
}
```

شبیه سایر مثال‌ها، این فایل برجسته شامل رهنمود ورود مناسب و تولید یک کلاس است که در این مورد Triggers نامیده می‌شود و متدی با صفت متد مناسب است. این مثال کد از یک جفت شئ ADO.NET جدید استفاده می‌کند: شئ SQLTriggerContext و شئ SQLPipe. شئ SQLTriggerContext اطلاعاتی درباره عمل تریگری که فعال شده و ستون‌هایی که متأثر شده‌اند، فراهم می‌کند. شئ SQLTriggerContext همیشه توسط شئ SQLContext نمونه‌سازی می‌شود. معمولاً، شئ SQLContext اطلاعاتی درباره زمینه فراخوان فراهم می‌کند. بخصوص، در این مورد، شئ SQLContext به کد امکان دستیابی به جدول مجازی را می‌دهد که در طی اجرای تریگر ایجاد شده است. این جدول مجازی، داده‌ای را که موجب فعال شدن تریگر شده است، ذخیره می‌کند.

سپس، شئ SQLPipe ایجاد می‌شود. شئ SQLPipe معبری را نشان می‌دهد که اطلاعات بین CLR و کد فراخوان ارسال می‌شوند. در این‌جا، شئ SQLPipe به تریگر توسعه یافته امکان می‌دهد تا با فراخوان خارجی ارتباط برقرار کند. سپس شئ SQLContext برای تعیین این مسأله استفاده می‌شود، آیا عمل تریگر یک عمل درج بوده است. اگر چنین است، آن‌گاه محتویات جدول تریگر مجازی بازیابی شده و با استفاده از متد Execute شئ SQLPipe به فراخوان ارسال کنید.

هنگامی‌که کد ایجاد شد، می‌توانید آن را با استفاده از گزینه | Visual Studio 2005 Build Deploy Solution برای پایگاه داده توزیع کنید و اسمبلی و هر شئ مرتبط به آن را حذف کرده و مجدداً ایجاد کنید. برای ایجاد دستی یک تریگر CLR، می‌توانید از حالت CREATE TRIGGER استفاده کنید که در مثال بعد آن را می‌بینید. این کد نحوه ایجاد تریگر توسعه یافته را در جدول Person.ContactType در پایگاه داده AdventureWorks را نشان می‌دهد:

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
CREATE TRIGGER MyTrigger
ON Person.ContactType
FOR INSERT
AS EXTERNAL NAME
MyCLRDLL.Triggers.MyTrigger
```

شبیه مثال‌های NET. دیگر، تریگر توسعه یافته با استفاده از عبارت CREATE TRIGGER ایجاد می‌شود. عبارت CREATE TRIGGER با بخش AS EXTERNAL NAME توسعه یافته است که تریگر را به متدی در یک اسمبلی مرتبط می‌کند. در این‌جا بخش EXTERNAL NAME به متدی در اسمبلی مرتبط می‌شود. در این‌جا، بخش EXTERNAL NAME به اسمبلی به نام MyCLRDLL اشاره می‌کند. در کلاس Triggers آن فضای نام، متد MyTrigger حاوی کدی است که هنگام فعال شدن تریگر توسعه یافته اجرا خواهد شد.

تریگر NET. برای هر عمل درجی که در جدول Job انجام می‌شود، فعال خواهد شد. مثلاً، این عبارت INSERT ردیفی را به جدول Person.ContactType اضافه خواهد کرد که موجب فعال شدن تریگر NET. خواهد شد:

```
INSERT INTO Person.ContactType VALUES(102, 'The Big Boss',
  '2004-07-20 00:00:00.000')
```

تریگر مثال MyTrigger یک عبارت Select را در مقدار ردیف درج شده انجام می‌دهد. سپس از شئ SQLPipe برای ارسال نتایج به فراخوان استفاده می‌کند. در این مثال، تریگر محتویات مقادیر ردیف درج شده را به فراخوان برمی‌گرداند.

Section ۳۵.۰۶ انواع داده تعریف شده کاربر CLR

ویژگی مهم دیگر در SQL Server 2005 که با یکپارچگی NET CLR. فعال می‌شود، توانایی ایجاد انواع تعریف شده کاربر[1] (UDTهای) واقعی است. با استفاده از UDTها، می‌توانید انواع ردیف فراهم شده توسط SQL Server را توسعه داده و انواع داده‌ای را که خاص برنامه یا محیط شماست، اضافه کنید.

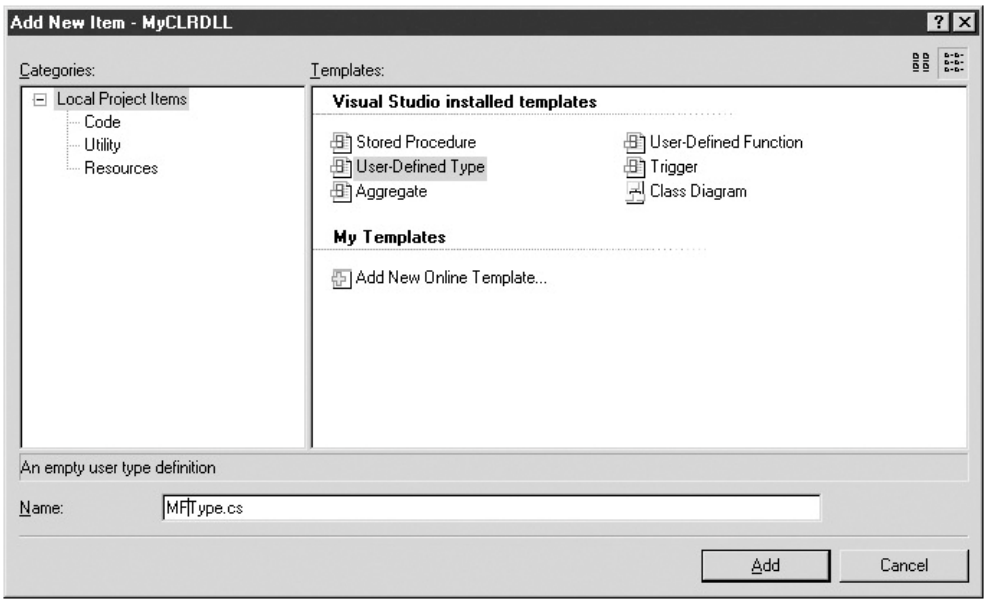
در مثال بعد، نحوه ایجاد یک UDT را خواهید دید که یک کد جنسیت را نشان می‌دهد: M برای مرد و F برای زن. در حالی که می‌توانید این داده را در یک فیلد کاراکتری تک بایتی استاندارد ذخیره کنید، استفاده از یک UDT اطمینان می‌دهد که این فیلد تنها این دو مقدار را بدون نیاز اضافی به تریگرها، الزامات یا سایر تکنیک‌های تأثیر اعتبار داده، خواهد پذیرفت.

1- User-defined types

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

اگر از Visual Studio 2005 استفاده می‌کنید، بهترین روش برای ایجاد UDT، استفاده از قالب‌های SQL Server است. برای ایجاد یک UDT جدید، روی پروژه خود در Visual Studio 2005 کلیک راست کرده و Add | Add Class را از منوی زمینه انتخاب کنید. این امر کادر محاوره‌ای Add New Item را نمایش خواهد داد که در شکل ۶-۴ نشان داده شده است.



(a)

شکل ۶-۴ ایجاد یک NET SQL Server UDT. (b)

User-Defined Type را از لیست قالب‌های SQL Server انتخاب کنید. نامی را وارد کنید که برای انتساب به کلاس در نظر دارید و سپس Open را برای تولید یک فایل برجسته برای UDT توسط Visual Studio کلیک کنید. این فایل برجسته، چهار متدی را پیاده‌سازی می‌کند که SQL Server 2005 برای تمام UDTها نیاز دارد. این متدها برای برآورده کردن الزامات قرارداد SQL Server UDT ژنریک هستند (افزودن کد برای ساخت UDT که اعمال با معنی را انجام دهد، به عهده شما گذاشته می‌شود). چهار متد UDT مورد نیاز در جدول ۱-۴ فهرست شده‌اند.

جدول ۱-۴ متدهای UDT مورد نیاز (c)

متد	شرح
IsNullable	این متد مورد نیاز برای نشان دادن قابل null بودن شئ استفاده می‌شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

	SQL Server 2005 برای پیاده‌سازی null پذیری، نیاز به تمام UDTها دارد، بنابراین این متد همیشه باید true را برگرداند.
Parse	این متد مورد نیاز، یک پارامتر رشته‌ای را گرفته و آن را به عنوان یک UDT ذخیره می‌کند.
ToString	این متد مورد نیاز، محتویات UDT را به یک رشته تبدیل می‌کند.
Default constructor	این متد مورد نیاز، نمونه جدیدی از UDT را ایجاد می‌کند.

می‌توانید کلاس MFType کامل شده‌ای را ببینید که برای پیاده‌سازی یک UDT برای کدهای M (مرد) و F (زن) در این لیست استفاده می‌شود:

```
using System;
using System.Data.Sql;
using System.Data.SqlTypes;

[Serializable]
[SqlUserDefinedType(Format.SerializedDataWithMetadata,
MaxByteSize=512)]
public class MFType: INullable
{
   string m_value;
   public override string ToString()
   {
      string s = "null";
      if (m_value != null)
      {
         s = m_value.ToString();
         return s;
      }
      else return m_value.ToString();
   }
   public bool IsNull
   {
      get
      {
         if (m_value == null)
            return true;
         else return false;
      }
   }

   public static MFType Null
   {
      get
      {
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
        MFType h = new MFType();
        return h;
    }
  }

  public static MFType Parse(SqlString s)
  {
    if (s.IsNull || s.Value.ToLower() == "null")
      return Null;
    MFType u = new MFType();
    u.Value = s.ToString();
    return u;
  }

  // Create a Value Property
  public SqlString Value
  {
    get
    {
      return (m_value);
    }
    set
    {
      if (value == "M" || value == "F")
      {
        m_value = value.ToString();
      }
      else
      {
        throw new ArgumentException
          ("MFType data type must be M or F");
      }
    }
  }
}
```

اولین بخش این کد لزوماً قالبی است که توسط انواع تعریف شده کاربر مورد نیاز است. صفت کلاس باید قابل سریالی شدن باشد، کلاس باید رابط INullable را پیاده‌سازی کند و نام کلاس با نام UDT تنظیم می‌شود. می‌توانید رابط IComparable را به‌طور اختیاری اضافه کنید. در این مثال، MFType نام کلاس است. یک متغیر رشته‌ای به نام m_value برای نگهداری محتویات کد معرفی شده است. سپس، می‌توانید متد ToString مورد نیاز را ببینید. متد ToString بررسی می‌کند تا ببیند

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

آیا محتویات متغیر m_value، null است. اگر چنین است، آن‌گاه رشته "null" برگردانده می‌شود. در غیر این صورت، متد ToString متغیر m_value، مقدار رشته‌ای محتویات را برمی‌گرداند.

بخش بعدی کد، خصوصیت IsNull را تعریف می‌کند. متد get این خصوصیت، محتویات متغیر m_value را بررسی کرده و در صورت null بودن m_value، مقدار true را برمی‌گرداند. در غیر این صورت، متد Get مقدار false را برمی‌گرداند. سپس، می‌توانید متد Null را ببینید که توسط قالبی برای برآورده کردن شرط UDT برای null پذیری تولید شده است.

متد Parse یک آرگومان رشته‌ای را می‌گیرد که در خصوصیت Value شئ ذخیره می‌شود. می‌توانید تعریفی برای خصوصیت value را کمی پایین‌تر در کد ببینید. متد Parse باید به صورت ایستا معرفی شود یا اگر از VB.NET استفاده می‌کنید، باید یک خصوصیت Shared باشد. خصوصیت Value خاص این پیاده‌سازی است. در این مثال، خصوصیت Value برای ذخیره و بازیابی مقدار UDT استفاده می‌شود و هم‌چنین مسئول ویرایش مقادیر مجاز است. در متد Set، می‌توانید ببینید که تنها مقادیر M یا F مجاز هستند. تلاش برای استفاده از هر مقدار دیگری، موجب رخ دادن استثنایی می‌شود که به فراخوان هشدار می‌دهد که "نوع داده MFType باید M یا F باشد". بسیار شبیه یک تابع یا رویه ذخیره شده CLR، بعد از تکمیل کد، در یک DLL کامپایل می‌شود. آن‌گاه آن DLL به عنوان یک اسمبلی SQL Server با استفاده از عبارت CREATE ASSEMBLY یا گزینه Visual Studio 2005 Deploy Solution وارد می‌شود (گزینه Visual Studio 2005 Deploy Solution اسمبلی و UDT را ایجاد می‌کند).

برای افزودن دستی UDT به پایگاه داده، می‌توانید از یک عبارت CREATE TYPE شبیه این استفاده کنید:

```
CREATE TYPE MFType EXTERNAL NAME MyCLRDLL.MFType
```

همانند هنگام ایجاد سایر اشیای پایگاه داده .NET. ، کلمه کلیدی EXTERNAL NAME برای مشخص کردن اسمبلی و فضای نام برای UDT استفاده می‌شود. در این مورد، با توجه به این که خود UDT به عنوان یک کلاس پیاده‌سازی می‌شود، نیاز به نام متد نیست. مقدار MyCLRDLL اسمبلی را مشخص می‌کند و مقدار MFType کلاس UDT را مشخص می‌کند. برای مشاهده UDTهایی که برای یک پایگاه داده ایجاد کرده‌اید، می‌توانید دیدگاه sys.Types را پرس‌وجو کنید، همان‌گونه که در این‌جا نشان داده شده است:

```
SELECT * FROM sys.Types
```

هنگامی که UDT ایجاد می‌شود، می‌توانید از آن در T-SQL شبیه انواع داده طبیعی SQL Server استفاده کنید. هرچند، با توجه به این که UDTها حاوی متدها و خصوصیات هستند،

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

تفاوت‌هایی وجود دارد. این مثال نشان می‌دهد که چگونه MFType UDT می‌تواند به عنوان یک متغیر استفاده شود و چگونه خصوصیت Value آن می‌تواند مورد دستیابی قرار گیرد:

```
DECLARE @mf MFType
SET @mf.Value='N'
PRINT @mf.Value
```

در این لیست، متغیر UDT با استفاده از عبارت T-SQL DECLARE استاندارد معرفی می‌شود. می‌توانید به اعضای UDT با پیشوند قراردادن آن‌ها با نماد (.) دستیابی داشته باشید. در این لیست، عبارت SET برای انتساب مقدار N به خصوصیت value متغیر UDT استفاده می‌شود. به دلیل این که N یک مقدار معتبر نیست، این خطا تولید می‌شود:

```
.Net SqlClient Data Provider: Msg 6522, Level 16, State 1, Line 2
A CLR error occurred during execution of 'MFType':
System.ArgumentException: MFType data type must be M or F
at MFType.set_Value(SqlString value)
```

همان‌گونه که UDTها می‌توانند به عنوان متغیر مورد استفاده قرار گیرند، هم‌چنین می‌توانند برای ایجاد ستون‌ها استفاده شوند. این لیست، ایجاد جدولی را نشان می‌دهد که از MFType UDT استفاده می‌کند:

```
CREATE TABLE MyContacts
(ContactID int,
FirstName varchar(25),
LastName varchar(25),
Gender MFType)
```

در حالی که ایجاد ستون‌ها با نوع UDT شبیه هنگام استفاده از یک نوع داده طبیعی است، انتساب مقادیر به UDT کمی متفاوت از انتساب ستون استاندارد است. UDTهای پیچیده می‌توانند حاوی چندین مقدار باشند. در این مورد، با توجه به این که UDT از یک مقدار ساده استفاده می‌کند، می‌توانید مقادیری را به آن نسبت دهید، همان‌گونه که می‌توانید در مورد هر نوع داده تعبیه شده‌ای این کار را انجام دهید. این مثال نحوه درج ردیفی در جدول نمونه MyContacts را نشان می‌دهد که حاوی MFType UDT است:

```
INSERT INTO MyContacts VALUES(1, 'Michael', 'Otey', 'M')
```

برای بازیابی محتویات UDT با استفاده از عبارت SELECT، باید از نمادگذاری UDT.Member هنگام مراجعه به یک ستون UDT استفاده کنید، همان‌گونه که در این‌جا نشان داده شده است:

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
SELECT ContactID, LastName, Gender.Value FROM MyContacts
```

Section ۳۵،۰۷ انبوهه‌های تعریف شده کاربر CLR

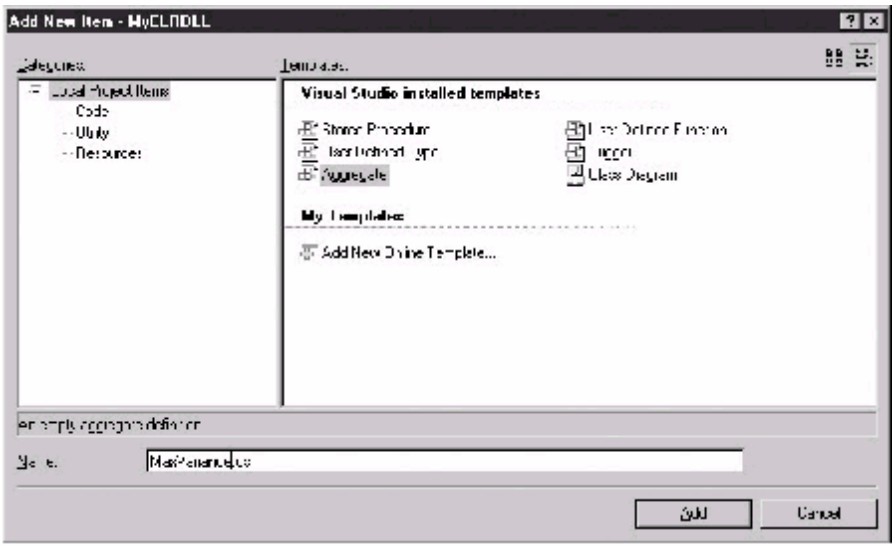
یک انبوهه تعریف شده کاربر[1] (UDAGG)، نوع جدید دیگری از شئ پایگاه داده NET. است که در SQL Server 2005 معرفی شده است. لزوماً، یک انبوهه تعریف شده کاربر، یک تابع قابلیت توسعه است که به شما امکان می‌دهد تا مقادیر را روی گروهی در طی پردازش یک پرس‌وجو انبوهه کنید. SQL Server همیشه مجموعه پایه‌ای از توابع انبوهه شبیه MIN، MAX و SUM را فراهم کرده است که می‌توانید در یک پرس‌وجو استفاده کنید. انبوهه‌های تعریف شده کاربر به شما امکان توسعه این گروه از توابع انبوهه را با انبوهه‌های اختصاصی می‌دهد. شبیه توابع انبوهه طبیعی، انبوهه‌های تعریف شده کاربر به شما اجازه اجرای محاسبات در مجموعه‌ای از مقادیر را می‌دهند و یک مقدار تکی را برمی‌گردانند. هنگامی که یک UDAGG را ایجاد می‌کنید، منطقی را مهیا کنید که انبوهه را انجام خواهد داد. در این بخش، نحوه ایجاد یک UDAGG ساده را خواهید دید که مقدار میانه را برای مجموعه‌ای از اعداد محاسبه می‌کند.

برای ایجاد یک UDAGG جدید با استفاده از Visual Studio 2005، گزینه Project | Add Aggregate را از منویی برای نمایش کادر محاوره‌ای Add New Item انتخاب کنید که می‌توانید آن را در شکل ۷-۴ ببینید.

1- User-Defined Aggregate

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

شکل ۷-۴ افزودن یک انبوهه تعریف شده کاربر (b)

Aggregate را از لیست قالب‌های SQL Server انتخاب کرده و سپس نامی برای کلاس وارد نموده و Open را کلیک کنید. Visual Studio یک فایل برجسته برای کلاس Aggregate تولید خواهد کرد. بیشتر شبیه یک UDT، فایل برجسته کلاس Aggregate چهار متدی را پیاده‌سازی می‌کند که SQL Server 2005 برای تمام انبوهه‌های تعریف شده کاربر نیاز دارد. چهار متد مورد نیاز برای تمام UDAGGها در جدول ۲-۴ لیست شده‌اند.

جدول ۲-۴ متدهای انبوهه مورد نیاز (c)

متد	شرح
Init	این متد شئ را مقداردهی می‌کند. این متد برای هر انبوهه یک بار احضار می‌شود.
Accumulate	این متد مورد نیاز برای هر آیتم در مجموعه انبوهه شده، یک بار احضار می‌شود.
Merge	این متد مورد نیاز هنگامی احضار می‌شود که سرور یک پرس‌وجو را با استفاده از توازن اجرا می‌کند. این متد برای ادغام داده از نمونه‌های موازی متفاوت با یکدیگر استفاده می‌شود.
Terminate	این متد مورد نیاز، نتایج انبوهه را برمی‌گرداند. این متد بعد از پردازش تمام

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

	آیتم‌ها، یک بار احضار می‌شود.

می‌توانید کلاس MaxVariance مثال را ببینید که برای پیاده‌سازی یک انبوهه MaxVar تعریف شده کاربر در این لیست استفاده می‌شود:

```
[Serializable]
[SqlUserDefinedAggregate(Format.SerializedDataWithMetadata, MaxByteSize = 512)]
public class MaxVariance
{
   int m_LowValue;
   int m_HighValue;

   public void Init()
   {
      m_LowValue = 999999999;
      m_HighValue = -999999999;
   }
   public void Accumulate(SqlInt32 Value)
   {
      if (Value > m_HighValue) m_HighValue = (int)Value;
      if (Value < m_LowValue) m_LowValue = (int)Value;
   }
   public void Merge (MaxVariance Group)
   {
      if (Group.GetHighValue() > m_HighValue)
         m_HighValue = Group.GetHighValue();
      if (Group.GetLowValue() < m_LowValue)
         m_LowValue = Group.GetLowValue();
   }
   public SqlInt32 Terminate ()
   {
      return m_HighValue - m_LowValue;
   }
   // Helper methods
   public int GetLowValue()
   {
      return m_LowValue;
   }
   public int GetHighValue()
   {
      return m_HighValue;
   }
}
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

در بالای این کلاس، می‌توانید صفت سریالی‌سازی را ببینید که توسط کلاس‌های UDAGG مورد نیاز است. سپس، دو متغیر برای نگهداری مقادیر حداقل و حداکثری معرفی می‌شوند که توسط انبوهه مواجه می‌شویم. بعد از آن، در متد Init، مقادیر بالا و پایین به دو متغیر نسبت داده می‌شوند و اطمینان حاصل می‌شود که آن‌ها مقادیر منتسب از لیست خواهند بود. در حالی که متد Init فقط یک بار فراخوانی می‌شود، متد Accumulate برای هر ردیف در مجموعه نتیجه، یک بار فراخوانی می‌شود. در این مثال، متد Accumulate، مقدار ورودی را با مقادیر ذخیره شده در متغیرهای m_HighValue و m_LowValue مقایسه می‌کند. اگر مقدار ورودی بیشتر از مقدار بالای جاری باشد، در متغیر m_HighValue ذخیره می‌شود. اگر مقدار کوچک‌تر از مقدار m_LowValue باشد، در m_LowValue ذخیره می‌شود. در غیر این صورت، هیچ عملی برای این UDAGG توسط متد Accumulate انجام نمی‌شود.

توجه: به دلیل این که UDAGGها سریالی می‌شوند، باید از الزام حافظه کل UDAGG آگاه باشید. UDAGG بعد از هر بار احضار متد Accumulate سریالی می‌شود و نمی‌تواند از اندازه ستون حداکثر ۸۰۰۰ بایت تجاوز کند.

متد Merge هنگامی استفاده می‌شود که UDAGG به‌طور موازی پردازش می‌شود که معمولاً چنین حالتی در بیشتر پرس‌وجوها نیست. اگر Merge فراخوانی شود، کار آن وارد کردن مقادیر انبوهه جاری از نمونه موازی است. در این‌جا می‌توانید ببینید که این کار با استفاده از دو متد کمکی انجام می‌شود که لزوماً مقادیر متغیرهای m_HighValue و m_LowValue را صادر می‌کنند. این مقادیر با مقادیر موجود مقایسه شده و چنان‌چه بیشتر یا کمتر باشند، جایگزین مقادیر جاری در m_HighValue و m_LowValue خواهند شد.

متد Terminate یک بار بعد از پردازش تمام نتایج فراخوانی می‌شود. برای این مثال، متد Terminate کمترین مقدار موجود را از بیشترین مقدار موجود کم کرده و اختلاف را به فراخوان برمی‌گرداند.

بعد از کامپایل کلاس NET. در یک DLL، می‌توانید DLL را به عنوان یک اسمبلی SQL Server با استفاده از گزینه Visual Studio 2005 Deploy Solution یا به‌طور دستی با استفاده از عبارت CREATE ASSEMBLY و عبارات CREATE AGGREGATE وارد کنید. عبارت CREATE AGGREGATE دستی در این‌جا نشان داده شده است:

```
CREATE AGGREGATE MaxVariance(@MyInt int)
RETURNS int
EXTERNAL NAME MyCLRDLL.MaxVariance
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

این مثال ایجاد یک انبوهه به نام MaxVariance را نشان می‌دهد. این انبوهه می‌تواند با انواع داده صحیح استفاده شود و یک عدد صحیح را برگرداند. در بخش EXTERNAL NAME، می‌توانید کدی را برای این UDAGG ببینید که در کلاس MaxVariance اسمبلی MyCLRDLL قرار دارد.

می‌توانید از UDAGG درست شبیه توابع انبوهه تعبیه شده SQL Server استفاده کنید. یک تفاوت کوچک این است که باید قبل از UDAGG، نام طرح‌واره بیاید تا به سیستم اجازه یافتن آن را بدهد. این خط استفاده از MaxVariance UDAGG را نشان می‌دهد:

```
SELECT dbo.MaxVariance(MinQty) FROM Sales.SpecialOffer
```

نتیجه این UDAGG تفاوت بین مقادیر بالا و پایین موجود در ستون SalesSpecialOffer است.

Section ۳۵،۰۸ امنیت شیئی پایگاه داده .NET

هیچ بحثی در مورد ویژگی‌های CLR جدید بدون مباحث امنیتی مربوط به استفاده از اسمبلی‌های .NET و SQL Server CLR کامل نمی‌شود. برخلاف T-SQL که هیچ امکان طبیعی برای مراجعه به منابع خارج از پایگاه داده ندارد، اسمبلی‌های .NET کاملاً قادر به دستیابی به منابع سیستم و شبکه هستند. بنابراین، ایمن کردن آن‌ها جنبه مهمی از برنامه‌نویسی آن‌هاست. در SQL Server 2005، مایکروسافت مدل امنیتی SQL Server مبتنی بر کاربر را با مدل ایمنی CLR مبتنی بر مجوزها یکپارچه کرده است و با پیروی از مدل امنیتی SQL Server، کاربران امکان دستیابی به تنها اشیای پایگاه داده را دارند (از جمله آن‌هایی که از اسمبلی‌های .NET ایجاد شده‌اند) تا حقوق کاربری داشته باشند. امنیت CLR این مسأله را با فراهم کردن کنترل روی نوع منابع سیستمی که می‌توانند توسط کد .NET در حال اجرا روی سرور مورد دستیابی قرار گیرند، توسعه می‌دهد. مجوزهای امنیتی CLR در زمانی مشخص می‌شوند که اسمبلی با استفاده از بخش WITH PERMISSION_SET عبارت CREATE ASSEMBLY ایجاد می‌شود. جدول ۳-۴ گزینه‌هایی را برای مجوزهای امنیتی پایگاه داده CLR خلاصه می‌کند که می‌توانند بر اشیای پایگاه داده SQL Server اعمال شوند.

(a) جدول ۳-۴ گزینه‌های امنیتی شیئی پایگاه داده CLR

امنیت CLR	دستیابی خارجی مجاز است	فراخوانی برای کد مدیریت نشده
SAFE	بدون دستیابی خارجی	بدون فراخوانی برای کد مدیریت

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

		نشده
EXTERNAL_ACCESS	دستیابی خارجی از طریق APIهای مدیریتی مجاز است.	بدون فراخوانی برای کد مدیریت نشده
UNSAFE	دستیابی خارجی مجاز است.	فراخوانی برای کد مدیریت نشده مجاز است.

استفاده از مجوز SAFE، تمام دستیابی خارجی را ممنوع می‌کند. مجوز EXTERNAL_ACCESS مقداری دستیابی خارجی به منابع را با استفاده از APIهای مدیریت شده ممکن می‌سازد. SQL Server برای دستیابی منابع خارجی نقش فراخوان را بازی می‌کند. باید مجوز EXTERNAL_ACCESS جدیدی برای ایجاد اشیا با این مجموعه مجوز داشته باشید. مجوز UNSAFE اصولاً طوری است که می‌توان به تمام منابع سیستم دسترسی داشت و فراخوانی کد مدیریت شده و نشده میسر است. تنها راهبران سیستم می‌توانند اشیایی با مجوزهای UNSAFE ایجاد کنند.

Section ۳۵،۰۹ زمان استفاده از اشیای پایگاه داده CLR

اشیای پایگاه داده CLR که با استفاده از CLR ایجاد می‌شوند، برای جایگزینی رویه‌های ذخیره شده توسعه یافته‌ای که نیاز به دستیابی به منابع سیستم خارجی برای ایجاد اشیای پایگاه داده دارند که نیاز به منطق پیچیده‌ای دارند یا برای اشیای پایگاه داده‌ای که به‌طور بالقوه بین پایگاه داده و لایه ردیف داده یک برنامه قابل انتقال هستند، مناسب می‌باشند. آن‌ها برای دستیابی داده خام و اعمال بهنگام‌رسانی نظیر T-SQL چندان مناسب نیستند.

Article XXXVI. بهبودهای T-SQL

با تمام ویژگی‌های جدید مربوط به .NET ، شاید تعجب کنید اگر بدانید مایکروسافت برای برداشتن پشتیبانی از T-SQL برنامه‌ریزی کرده باشد، ولی قطعاً چنین نیست. T-SQL هنوز بهترین زبان برای استفاده جهت دستیابی داده خام است و همان‌گونه که ممکن است توجه کرده باشید، ساختار دستوری T-SQL انجام داده است که در این بخش به آن‌ها می‌پردازیم.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۳۶.۰۱ بهبودهای TOP

در SQL Server 2000، مجبور به استفاده از یک مقدار ثابت در الحاق با بخش TOP بودید. به عبارت دیگر، می‌توانستید ردیف‌های TOP 5 یا TOP 10 را انتخاب کنید که مقدار ۵ یا ۱۰ یک ثابت بود. در SQL Server 2005، تابع TOP هم اینک کاربرد یک عبارت را در الحاق با بخش TOP ممکن می‌سازد. یک عبارت می‌تواند هر عبارت T-SQL مجازی باشد، از جمله یک متغیر یا زیر پرس‌وجوی عددی. بخش TOP هم‌چنین در عبارات INSERT، UPDATE و DELETE پشتیبانی می‌شود. این امر انعطاف‌پذیری بیشتری نسبت به قبل به بخش TOP می‌دهد. مثالی از کاربرد بخش TOP جدید چنین است:

```
USE AdventureWorks
DECLARE @MyTop INT
SET @MyTop = 15
SELECT TOP (@MyTop) CustomerID, SalesPerson FROM Sales.Customer
```

Section ۳۶.۰۲ CTE[1]

ویژگی T-SQL جدید دیگر، پشتیبانی از CTEهاست. CTEها بسیار شبیه دیدگاه‌ها هستند؛ هرچند، آن‌ها در یک پرس‌وجو تعبیه می‌شوند. دلیل اصلی مایکروسافت برای CTEها برای SQL Server 2005، فراهم کردن مکانیزمی برای مدیریت پرس‌وجوهای بازگشتی است. بازگشتی با این واقعیت به دست می‌آید که یک CTE برای مراجعه به خود مجاز است. برای جلوگیری از امکان همه‌گیر بودن سیستم با یک پرس‌وجوی بازگشتی با ساختار ضعیف، SQL Server یک محدودیت گستره سرور را در مورد حداکثر سطح بازگشتی مجاز با یک حداکثر پیش‌فرض ۱۰۰ سطح پیاده‌سازی کرده است. یک CTE به عنوان بخشی از کلمه کلیدی WITH پیاده‌سازی می‌شود و می‌تواند با عبارات SELECT، INSERT، UPDATE و DELETE استفاده شود. برای پیاده‌سازی پرس‌وجوهای بازگشتی با استفاده از CTE جدید، باید از یک ساختار خاص استفاده کنید، همان‌گونه که در مثال کد ساده‌ای در ادامه نشان داده شده است. این مثال یک پرس‌وجوی بازگشتی ساده را با استفاده از جدول HumanResources.Employee در پایگاه داده نمونه AdventureWorks انجام می‌دهد:

```
USE AdventureWorks
WITH EmployeeChart(EmployeeID, ManagerID, Title)
AS
(SELECT EmployeeID, ManagerID, Title
 FROM HumanResources.Employee
 WHERE EmployeeID = 3
 UNION ALL
```

1- Common Table Expressions

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
SELECT L2.EmployeeID, L2. ManagerID, L2.Title
 FROM HumanResources.Employee AS L2
 JOIN EmployeeChart
  ON L2.ManagerID = EmployeeChart.EmployeeID)
SELECT * from EmployeeChart
```

برای استفاده از یک CTE، ابتدا یک بخش WITH بنویسید که برای نام‌گذاری CTE استفاده می‌کنید و ستون‌هایی را برای انقیاد به یک عبارت SELECT مشخص کنید. باید یک سمی‌کالون در جلوی کلمه کلیدی WITH قرار دهید، چنان‌چه اولین عبارت در یک دسته نباشد. اولین عبارت SELECT عضو لنگر[1] نامیده می‌شود و نباید به خودش مراجعه کند. در این مورد، ستون‌های EmployeeID، ManagerID و Title را از جدول AdventureWorks Employee بازیابی می‌کند. دومین عبارت SELECT به CTE مراجعه کرده و عضو بازگشتی[2] نامیده می‌شود. در این مورد، عضو بازگشتی ستون‌هایی یکسان را بازیابی می‌کند و به عضو لنگر در ستون ManagerID ملحق می‌شود. می‌توانید نتایج این CTE را در این لیست ببینید:

EmployeeID	ManagerID	Title
3	12	Engineering Manager
4	3	Senior Tool Designer
9	3	Design Engineer
11	3	Design Engineer
158	3	Research and Development Manager
263	3	Senior Tool Designer
267	3	Senior Design Engineer
270	3	Design Engineer
5	263	Tool Designer
265	263	Tool Designer
79	158	Research and Development Engineer
114	158	Research and Development Engineer
217	158	Research and Development Manager

(13 row(s) affected)

Section ۳۶،۰۳ PIVOT و UNPIVOT

اضافه شدن عملگرهای رابطه‌ای PIVOT و UNPIVOT ویژگی جدید دیگری در SQL Server 2005 T-SQL است. عملگرهای PIVOT و UNPIVOT برای روش‌های OLAP که در آن‌ها با داده جدولی و نه داده رابطه‌ای سروکار داریم، مفید هستند. عملگر PIVOT مجموعه‌ای از

1- Anchor Member
2- Recursive Member

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

ردیف‌ها را به ستون‌ها تبدیل می‌کند. همان‌گونه که ممکن است انتظار داشته باشید، عملگر UNPIVOT معکوس عملگر PIVOT است و ستون‌های محوری را به ردیف‌ها تبدیل می‌کند. هرچند، بسته به موقعیت، عمل UNPIVOT ممکن است دقیقاً معکوس عمل PIVOT نباشد. این موقعیت به این دلیل رخ می‌دهد که عمل PIVOT چنین چیزی را تنظیم می‌کند و مقادیر خاصی را از قلم خواهد انداخت. اگر مقداری در طی عمل PIVOT از قلم افتاده باشد، بدیهی است که نمی‌تواند از حالت محوری خارج شود. بنابراین، عملگر UNPIVOT همیشه موجب یک تصویر قرینه دقیق از شرط محوری اصلی نمی‌شود.

می‌توانید در لیست زیر ببینید که عملگر PIVOT چگونه کار می‌کند. Select ساده‌ای را در فایلی به نام OrderSum در نظر داشته باشید، می‌توانید مجموعه‌ای از سفارشات را برای یک CustomerID برابر ۱ ببینید که هر سفارش دارای یک سال مرتبط است.

```
OrderId     CustomerID      OrderYear
----------- --------------- -----------
100         1               2000
101         1               2000
102         1               2000
103         1               2001
104         1               2001
105         1               2002
106         1               2003
107         1               2004

(8 row(s) affected)
```

با استفاده از عملگر PIVOT جدید SQL Server 2005، می‌توانید این مجموعه نتیجه را تبدیل کنید که هر سال را به‌طور عمودی در مجموعه نتیجه‌ای که سال‌ها را به‌طور افقی برای هر مشتری لیست کرده و تعداد سفارشات را برای هر سال جمع می‌بندد، لیست کند. عمل PIVOT نمونه در این لیست نشان داده شده است:

```
SELECT * FROM OrderSum
 PIVOT (COUNT(OrderID)
   FOR OrderYear IN([2000], [2001], [2002], [2003], [2004]))
   AS P
 WHERE CustomerID = 1
```

در این‌جا، عمل PIVOT با عبارت SELECT برای ایجاد یک مجموعه نتیجه جدید استفاده می‌شود. اولین مقدار عملگر PIVOT مقداری را تعیین می‌کند که در ستون محور قرار خواهد گرفت. در

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

این مثال، انبوهه COUNT(OrderID) تعداد سفارشات را برای هر مقدار محور جمع می‌زند. کلمه کلیدی FOR ستون را تعیین می‌کند که مقادیر آن محوری خواهند شد. این مثال عمل PIVOTای را نشان می‌دهد که روی ستون OrderYear انجام می‌شود. مقادیر تعیین شده توسط لیست کلمه کلیدی IN، مقادیری از ستون محوری شده هستند که به عنوان سرستون‌ها استفاده خواهند شد. می‌توانید مجموعه نتیجه محوری شده را در این لیست ببینید:

```
CustomerID      2000        2001        2002        2003        2004
---------------  ----------- ----------- ----------- ----------- -----------
1               3           2           1           1           1
Warning: Null value is eliminated by an aggregate or other SET operation.

(1 row(s) affected)
```

Section ۳۶،۰۴ تریگرهای DDL

نگارش‌های قبلی SQL Server تنها به تریگرها امکان می‌دادند تا با رویدادهای دستکاری داده نظیر درج یا بهنگام‌رسانی یک ردیف استفاده شوند. SQL Server 2005 این مسأله را با مجاز ساختن تریگرها برای قرار گرفتن در رویدادهای DDL از قبیل ایجاد و حذف اشیای پایگاه داده از قبیل جداول، دیدگاه‌ها، رویه‌ها و loginها توسعه داده است. تریگرهای DDL می‌توانند با عبارات CREATE، ALTER و DROP مرتبط شوند. این امر به DBA امکان قرار دادن محدودیت‌هایی روی نوع عملیاتی DDLای را می‌دهد که می‌توانند در یک پایگاه داده معین انجام شوند یا می‌توانید از این تریگرها برای ارسال پیام‌های هشدار با توجه به تغییرات مهم طرح‌واره‌ای که در پایگاه داده انجام شده است، استفاده کنید. این مثال نحوه افزودن یک تریگر DDL به نام NoTableUpdate را به عبارات DDL ، DROP Table و ALTER Table نشان می‌دهد:

```
CREATE TRIGGER NoTableUpdate
ON DATABASE FOR DROP_TABLE, ALTER_TABLE
AS
PRINT 'DROP TABLE and ALTER TABLE statement are not allowed'
ROLLBACK
```

در این‌جا، می‌توانید ببینید که چگونه تریگر DDL جدید می‌تواند برای محدود کردن کاربرد عبارات DROP TABLE و ALTER TABLE استفاده می‌شوند. اگر یک عبارت DROP TABLE و ALTER TABLE صادر شود، تریگر NoTableUpdate یک پیام خطا را چاپ خواهد کرد و عمل

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

DDL مذکور roll back می‌شود. یک تلاش برای صادر کردن یک عبارت ALTER TABLE در پایگاه داده حاوی تریگر NoTableUpdate در این‌جا نشان داده شده است:

```
DROP TABLE and ALTER TABLE statement are not allowed
.Net SqlClient Data Provider: Msg 3609, Level 16, State 2, Line 1
Transaction ended in trigger. Batch has been aborted.
```

برای اعمال تغییرات جداول در یک پایگاه داده بعد از انجام این تریگر، ابتدا باید تریگر DDL را حذف کنید.

Section ۳۶،۰۵ خروجی DML

ویژگی T-SQL جدید دیگر در SQL Server 2005، توانایی تولید خروجی از عبارات T-SQL DML (INSERT، UPDATE و DELETE) است. بخش Output جدید، داده اصلاح شده را برمی‌گرداند. مثلاً، این عبارت DELETE تمام ردیف‌ها را از جدول OrderSum حذف می‌کند:

```
DECLARE @MyOrderSumTVar TABLE(
   OrderID int,
   CustomerID int,
   OrderYear int);
DELETE FROM OrderSum
OUTPUT DELETED.* INTO @MyOrderSumTVar

SELECT * FROM @MyOrderSumTVar
```

توجه : جدول نمونه OrderSum در مثال قبلی این فصل ایجاد شد.

در این‌جا بخش OUTPUT DELETE.* مشخص می‌کند که تمام ردیف‌های حذف شده، به خروجی خواهند رفت. در نگارش‌های قبلی SQL Server، تنها تعدادی ردیف را می‌دیدید که توسط این عبارت تحت تأثیر قرار می‌گرفتند. می‌توانید نتایج بخش T-SQL DML Output را در این‌جا ببینید:

```
(8 row(s) affected)
OrderID     CustomerID          OrderYear
----------- -----------         -----------
100         1                   2000
101         1                   2000
102         1                   2000
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
103     1     2001
104     1     2001
105     1     2002
106     1     2003
107     1     2004

(8 row(s) affected)
```

Section ۳۶،۰۶ WAITFOR

ویژگی T-SQL جدید دیگر در SQL Server 2005، پشتیبانی بهبود یافته برای فرمان WAITFOR است. در نگارش‌های قبلی SQL Server، فرمان WAITFOR امکان انتظار برای تنها یک زمان از پیش تعریف شده را دارد. در SQL Server 2005، فرمان WAITFOR امکان انتظار برای نتایج یک عبارت RECEIVE را دارد. دلیل اصلی پشت این ویژگی، تسهیل پشتیبانی برنامه‌نویسی T-SQL برای قابلیت‌های صف‌بندی جدید فراهم شده توسط زیرسیستم SQL Service Broker است (می‌توانید مطالب بیشتر درباره این مسأله را در فصل ۶ مطالعه کنید). لیست بعد نشان می‌دهد که فرمان WAITFOR جدید چگونه می‌تواند در الحاق با یک عبارت RECEIVE استفاده شود:

```
WAITFOR (RECEIVE TOP (1) * FROM dbo.ServiceBrokerQueue)
```

Section ۳۶،۰۷ نوع varchar (max) جدید

نوع داده varchar (max) جدید روش دیگری را برای نوع داده متنی/ تصویری فراهم کرده است. نوع داده varchar (max) جدید، توسعه‌ای برای انواع داده varchar، nvarchar و varbinary است. شبیه انواع داده text، ntext و image، نوع داده varchar (max) تا 2GB داده را پشتیبانی می‌کند. هرچند، برخلاف انواع داده text، ntext، و image موجود، نوع داده varchar (max) می‌تواند حاوی داده کاراکتری و باینری باشد. ضمناً، هیچ پشتیبانی برای اشاره‌گرهای متنی ندارد.

مایکروسافت نوع داده varchar (max) جدید را برای کار کردن با انواع داده بزرگ معرفی کرد که بیشتر شبیه کار کردن با داده رشته‌ای استاندارد است. تمام توابع رشته‌ای در انواع داده varchar (max) کار می‌کنند و توابع SUBSTRING می‌توانند برای خواندن مقادیر زیادی داده استفاده شوند. علاوه بر این، عبارت T-SQL UPDATE برای پشتیبانی از بهنگام‌رسانی قطعات داده در یک نوع داده varchar (max) بهبود یافته است. می‌توانید ستونی را با استفاده از نوع داده varchar (max) جدید ایجاد کنید، به این صورت:

```
CREATE TABLE NewBLOB
(
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
   DataID INT IDENTITY NOT NULL,
   BLOBData VARCHAR(MAX) NOT NULL
)
```

Section ۳۶،۰۸ مدیریت صرف‌نظر از تراکنش

پیشرفت مهم دیگر در T-SQL در SQL Server 2005، مدیریت صرف‌نظر از تراکنش بهبود یافته است. در SQL Server 2005، یک مدل Try/Catch جدید به تراکنش اضافه شده است. این ساختار Try/Catch جدید خطاهای صرف‌نظر از تراکنش گیر افتاده را بدون فقدان زمینه تراکنش ممکن می‌سازد. در SQL Server 2000، هرچند می‌توانستید از یک تراکنش صرف‌نظر کنید، ولی هیچ روشی برای حفظ زمینه تراکنش وجود نداشت، بنابراین می‌توانید تراکنش صرف‌نظر شده را به‌طور کامل بازیافت کنید. مدیریت صرف‌نظر تراکنش Try/Catch جدید SQL Server 2005 به شما امکان نگهداری زمینه کامل تراکنش صرف‌نظر شده را می‌دهد و گزینه ایجاد مجدد تراکنش را به شما می‌دهد. این لیست کد ساختار T-SQL Try/Catch پایه را نشان می‌دهد:

```
BEGIN TRY
   <SQL Statements>
END TRY
BEGIN CATCH TRAN_ABORT
   <SQL Statements>
END CATCH
```

تراکنش در بلوک Try قرار گرفته است. اگر RAISERROR در عبارت TRAN_ABORT در بلوک Try صادر شود، کنترل به بلوک Catch منتقل می‌شود. در بلوک Catch، متغیر @@error می‌تواند برای تعیین شرط خطا ارزیابی شود.

Section ۳۶،۰۹ زمان استفاده از اشیای پایگاه داده T-SQL

اشیای پایگاه داده T-SQL در SQL Server به تدریج کامل نشده‌اند. در واقع، T-SQL هنوز بهترین گزینه برای اشیایی است که باید عملیات دستیابی داده خام را انجام دهند. برای نمونه، اگر رویه‌های ذخیره شده‌ای داشته باشید که کار اصلی آن‌ها درج، بهنگام‌رسانی یا حذف ردیف‌های داده باشد، آن‌گاه این اشیا باید با استفاده از T-SQL توسعه یابند و نه یکی از زبان‌های NET. .

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Article XXXVII. بهبودهای ADO.NET

علاوه بر ویژگی‌های T-SQL و CLR جدید، SQL Server 2005 هم‌چنین تعدادی بهبود جامع در سمت کلاینت با دسته‌بندی کردن یک بهنگام‌رسانی برای ADO.NET فراهم کرده است. هنگامی که ADO.NET به پخته‌تر شدن ادامه می‌دهد، بالاخره ویژگی‌هایی را از دست می‌دهد که در نسخه قبلی آن وجود داشته است، ADO مبتنی بر COM، به همراه یک جفت ویژگی جدید. همان‌گونه که خواهید دید، ویژگی‌های جدیدی را بیان می‌کنم که مایکروسافت به ADO.NET اضافه کرده است، بیشتر آن‌ها برخی از قابلیت‌های جدید را ارایه می‌دهند که به موتور پایگاه داده SQL Server اضافه شده‌اند. علاوه بر ویژگی‌های عمده جدید که در بخش‌های بعد لیست شده‌اند، همان‌گونه که انتظار می‌رود، ADO.NET جدید هم‌چنین از انواع داده XML و T-SQL varchar (max) جدید پشتیبانی می‌کند.

Section ۳۷.۰۱ پشتیبانی از مکان‌نمای سرور با استفاده از SQLResultSet

یکی از مهم‌ترین ویژگی‌های جدید فراهم شده در ADO.NET جدید، پشتیبانی از مکان‌نماهای سمت سرور[1] است. این یکی از نواحیی است که ADO.NET، ویژگی‌هایی را در ADO مبتنی بر COM نداشت. نگارش‌های قبلی ADO.NET تنها از مکان‌نماهای سمت کلاینت پشتیبانی می‌کردند، از قبیل آنی که محیط کلاینت باید در مورد کار نگهداری مجموعه نتیجه انجام دهد. در مکان‌نماهای سمت سرور، می‌توانید آن کار را به سرور منتقل کنید. مایکروسافت این ویژگی جدید را برای پشتیبانی از SQL Server .NET Data Provider درون فرآیندی اضافه کرده است که روی سرور اجرا می‌شود. مایکروسافت این ویژگی را برای الزامات سمت سرور درون فرآیندی اضافه کرده است، زیرا در سرور، نیاز به اسکرول پویا از طریق نتایج مجموعه نتایج با عمر کوتاه است و کمی همی ارتباط متقابل کاربر مورد نیاز است. مکان‌نماهای سمت سرور حالت نگهداری را انجام می‌دهند که مقیاس‌پذیری را کاهش داده و نیاز به رفت و برگشت‌ها به سرور را افزایش می‌دهند. هرچند، به دلیل مسائل مقیاس‌پذیری، مایکروسافت این را تنها به فضای نام System.Data.SQLServer سمت سرور اضافه کرده است. پشتیبانی مکان‌نمای سمت سرور، بخشی از فضای نام System.DataSQLClient سمت کلاینت نیست.

در ADO.NET 2.0، شئ SQLResultSet مکان‌نماهای سمت سرور را برای برنامه شما ارایه می‌دهد. این مکان‌نماها هم قابل بهنگام شدن و هم قابل اسکرول به صورت پویا هستند. مکان‌نماهای سمت سرور جدید توسط

1- Server-Side Cursors

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

متد ExecuteResultSet جدید در شئ System.Data.SQLServer SQLCommand نمونه‌سازی می‌شوند. این مثال کاربرد شئ ADO.NET SQLResultSet را تشریح می‌کند:

```
using System;
using System.Data;
using System.Data.Sql;
using System.Data.SqlServer;
using System.Data.SqlTypes;

public partial class StoredProcedures
{
   [SqlProcedure]
   public static void GetProductName()
   {
      SqlPipe myPipe = SqlContext.GetPipe();
      myPipe.Send("GetProductName: Opening server cursor");
      SqlCommand cmd = SqlContext.GetCommand();
      cmd.CommandType = CommandType.Text;
      cmd.CommandText = "SELECT Name FROM Production.Product WHERE
MakeFlag = 1";

      SqlResultSet resultset = cmd.ExecuteResultSet
         (ResultSetOptions.Scrollable |
         ResultSetOptions.Updatable);

      if (resultset.HasRows == true)
      {
         while (resultset.Read())
         {
            myPipe.Send(resultset.GetString(0));
            // You could optionally update with
            //resultset.Update();
            // or scroll back using
            //resultset.ReadLast();
         }
         resultset.Close();
      }
      myPipe.Send("GetProductName: Server cursor closed");
   }
};
```

در این‌جا، می‌توانید رویه ذخیره شده CLR جدید به نام GetProductName را ببینید. ابتدا، شئ SQLPipe برای ارسال یک پیام پیشرفت به کلاینت استفاده می‌شود. سپس، شئ SQLCommand نمونه‌سازی می‌شود و محتویات ستون Name را در جدول Production.Product پایگاه داده AdventureWorks نمونه بازیابی خواهد کرد. سپس، نمونه‌ای از شئ SQLResultSet ایجاد می‌شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

بیشتر شبیه SQLDataReader، شئ SQLResultSet با استفاده از شئ SQLCommand نمونه‌سازی می‌شود. در این مورد، یک مکان‌نمای قابل بهنگام شدن و قابل اسکرول با استفاده از متد ExecuteResultSet باز می‌شود. بعد از باز شدن مکان‌نما، برنامه شما می‌تواند به سمت جلو و به سمت عقب اسکرول کند و بهنگام‌رسانی‌ها را انجام دهد. با توجه به این که مکان‌نماهای سمت سرور حالت را نگهداری می‌کنند و منابع سرور را برای مادامی که باز هستند، مصرف می‌کنند، اطمینان از بسته شدن آن‌ها در صورت عدم نیاز به آن‌ها، بسیار مهم است.

۳۷،۰۲Section پشتیبانی غیرهم‌زمان

ویژگی دیگری که در ADO وجود داشت که در نسخه‌های قبلی ADO.NET نبود، پشتیبانی برای پرس‌وجوهای غیرهم‌زمان[1] است. پرس‌وجوهای غیرهم‌زمان برنامه‌های کلاینتی را فراهم می‌کنند که توانایی واگذاری پرس‌وجوها را بدون بلوکه کردن مداخله کاربر دارند. در ردیف میانی برنامه‌ها، پشتیبانی غیرهم‌زمان ADO.NET جدید، توانایی صدور چندین درخواست پایگاه داده را در رشته‌های مختلف بدون بلوکه کردن رشته‌ها برای برنامه‌های سرور فراهم می‌کند. این پشتیبانی غیرهم‌زمان جدید هم‌چنین با نگارش‌های قبلی SQL Server کار می‌کند. در SQL Server 2005، ADO.NET پشتیبانی غیرهم‌زمان در NET Framework. است. عملیات غیرهم‌زمان با استفاده از متد BEGINxxx شئ شروع شده و با استفاده از متد Endxxx خاتمه می‌یابد. شئ IAsyncResult برای بررسی وضعیت تکمیل فرمان استفاده می‌شود.

```
SqlConnection cn = new SqlConnection
 ("SERVER=TECA-YUKON;INTEGRATED SECURITY=True;"
  + "DATABASE=AdventureWorks;async=True");
SqlCommand cmd = new SqlCommand("SELECT * FROM Production.Product", cn);
cmd.CommandType = CommandType.Text;
try
{
   cn.Open();
   IAsyncResult myResult = cmd.BeginExecuteReader();
   while (!myResult.IsCompleted)
   {
      // Perform other code actions
   }
   // Process the contents of the reader
   SqlDataReader rdr = cmd.EndExecuteReader(myResult);

   // Close the reader
   rdr.Close();
```

1- Asyncrounous

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
}
catch (Exception ex)
{
   MessageBox.Show(ex.Message);
}
finally
{
   cn.Close();
}
```

پشتیبانی غیرهم‌زمان ADO.NET 2.0 در فضاهای نام کلاینت نظیر فضای نام System.DataSQLClient پیاده‌سازی می‌شود. اولین نکته مهم در این مثال رشته اتصال است. برای پیاده‌سازی پشتیبانی غیرهم‌زمان، رشته اتصال باید حاوی کلمه کلیدی ASYNC=true باشد. سپس، توجه داشته باشید که شئ IAsyncResult در بلوک Try است. متد BeginExecuteReader شئ SQLCommand برای شروع یک پرس‌وجوی غیرهم‌زمان استفاده می‌شود که تمام ردیف‌ها را در جدول Production.Product برمی‌گرداند. کنترل بلافاصله بعد از اجرای عبارت به برنامه برگردانده می‌شود. برنامه نیازی به انتظار برای اتمام پرس‌وجو ندارد. سپس، یک حلقه while برای بررسی وضعیت شئ IAsyncResult استفاده می‌شود. هنگامی که فرمان غیرهم‌زمان کامل می‌شود، خصوصیت IsCompleted با true تنظیم می‌شود. در این لحظه، حلقه while تکمیل شده و فرمان EndExecuteReader برای انتساب پرس‌وجوی غیرهم‌زمان به یک SQLDataReader برای پردازش استفاده می‌شود.

Section ۳۷.۰۳ MARS[1]

توانایی بهره بردن از ویژگی MARS جدید SQL Server 2005، بهبود دیگری در نگارش جدید ADO.NET است. در نگارش‌های قبلی ADO.NET و SQL Server، محدود به یک مجموعه نتیجه فعال در هر اتصال بودید. و در حالی که ADO مبتنی بر COM و OLE DB دارای یک ویژگی بودند که پردازش چند مجموعه نتیجه را به برنامه اجازه می‌داد، این ویژگی برای پردازش فرامین اضافی، نیاز به اتصالات جدیدی داشت. ویژگی MARS جدید در ADO.NET از قابلیت SQL Server 2005 برای داشتن چندین فرمان فعال در یک اتصال بهره می‌برد. در این مدل، می‌توانید یک اتصال به پایگاه داده باز کنید، سپس به اولین فرمان برگشته و نتایج دیگری را پردازش کنید. می‌توانید با خیالی آسوده بین فرامین مختلف سوییچ کنید. هیچ بلوکه کردنی بین فرامین وجود ندارد و هر دو فرمان از یک اتصال به پایگاه داده استفاده می‌کنند. این ویژگی، کارآیی و مقیاس‌پذیری بالایی را برای برنامه‌های ADO.NET

1- Multiple Active Result Sets

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

2.0 به همراه دارد. با توجه به این که این ویژگی وابسته به پایگاه داده SQL Server 2005 است، تنها می‌تواند با پایگاه‌های داده SQL Server 2005 به کار رود و با نگارش‌های قبلی SQL Server کار نمی‌کند. این مثال کاربرد MARS را تشریح می‌کند:

```
SqlConnection cn = new SqlConnection
  ("SERVER=TECA-YUKON;INTEGRATED SECURITY=True;"
    + "DATABASE=AdventureWorks");
SqlCommand cmd1 =
  new SqlCommand("SELECT * FROM HumanResources.Department", cn);
cmd1.CommandType = CommandType.Text;
try
{
    cn.Open();
    SqlDataReader rdr = cmd1.ExecuteReader();
    while (rdr.Read())
    {
        if (rdr["Name"].ToString() == "Production")
        {
            SqlCommand cmd2 = new SqlCommand
              ("SELECT * FROM HumanResources.Employee "
                + "WHERE DepartmentID = 7", cn);
            cmd2.CommandType = CommandType.Text;
            SqlDataReader rdr2 = cmd2.ExecuteReader();
            while (rdr2.Read())
            {
                // Process results
            }
            rdr2.Close();
        }
    }
    rdr.Close();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    cn.Close();
}
```

در این مثال، می‌توانید ببینید که cmd1 و cmd2 شئ SQLConnection یکسانی به نام cn را به اشتراک گذاشته‌اند. شئ cmd1 برای باز کردن یک SQLDataReader استفاده می‌شود که تمام ردیف‌های جدول HumanResources.Department را می‌خواند. هنگامی که Department ای به نام Production پیدا می‌شود، شئ SQLCommand دوم به نام cmd2، برای خواندن محتویات جدول

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

HumanResources.Employee استفاده می‌شود. نکته مهم این است که SQLCommand به نام cmd2 قادر به اجرا با استفاده از شئ SQLConnection فعال است که هم‌چنین به شئ cmd1 سرویس می‌دهد.

Section ۳۷.۰۴ صفحه‌بندی

پشتیبانی یکپارچه برای صفحه‌بندی، ویژگی جدید جالب دیگری در SQL Server 2005 در مورد ADO.NET است. صفحه‌بندی همیشه مسأله مشکلی برای کار کردن در برنامه‌های کلاینت بوده است و ADO.NET جدید، پشتیبانی بنیادی برای صفحه‌بندی را با دادن امکان انتخاب و انقیاد به محدوده‌ای از ردیف‌های یک مجموعه نتیجه به برنامه فراهم کرده است. برای مقیاس‌پذیری، پیاده‌سازی صفحه‌بندی جدید هیچ حالتی را در سرور نگهداری نمی‌کند. هرچند، این نیز بدان معنی است که عضویت در مجموعه صفحه‌بندی برای تغییر بین اجراها امکان‌پذیر است. این بدان معنی است که ویژگی صفحه‌بندی جدید برای داده‌ای مناسب است که نسبتاً ثابت باشد و اغلب تغییر نکند. پشتیبانی صفحه‌بندی در ADO.NET جدید مبتنی بر ترتیب است و برای استفاده از آن، باید ردیف شروعی را در مجموعه نتیجه و تعداد ردیف‌ها را برای قرار گرفتن در صفحه مشخص کنید. ردیف‌های مجموعه صفحه با استفاده از DataReader استاندارد خوانده می‌شوند. این مثال از زیرروال PageProductsTable، استفاده از عملکرد صفحه‌بندی ADO.NET جدید را تشریح می‌کند:

```
private SqlDataReader PageProductsTable(int nStartRow, int nPageSize)
{
    SqlConnection cn = new SqlConnection
      ("SERVER=TECA-YUKON;INTEGRATED SECURITY=True;"
        + "DATABASE=AdventureWorks");
    SqlCommand cmd =
      new SqlCommand("SELECT * FROM Production.Product", cn);
    cmd.CommandType = CommandType.Text;
    cn.Open();
    return cmd.ExecutePageReader
      (CommandBehavior.Default, nStartRow, nPageSize);
}
```

در این مثال، متد PageProductsTable دو عدد صحیح را که مکان شروع و تعداد ردیف‌های خوانده شونده صفحه‌بندی را تعریف می‌کنند، به عنوان آرگومان می‌گیرد. این متد یک شئ ADO.NET SQLDataReader را برمی‌گرداند. در داخل این روال، اشیای SQLConnection و SQLCommand به صورت طبیعی ایجاد می‌شوند. بعد از باز شدن شئ SQLConnection، متد

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

ExcutePageReader شئ SQLCommand برای بازیابی صفحه‌ای از نتایج فراخوانی می‌شود. اولین آرگومان متد ExcutePageReader، شمارشگر CommandBehaviorDefault است که به شئ SQLCommand نحوه مدیریت اتصال را هنگام خاتمه عمل می‌گوید. دومین آرگومان ترتیبی است که ردیف شروع را مشخص می‌کند. سومین آرگومان تعداد ردیف‌های برگردانده شونده را مشخص می‌کند. می‌توانید از قابلیت‌های صفحه‌بندی ADO.NET 2.0 استفاده کنید، همان‌گونه که در این مثال می‌بینید:

```
DataTable dt = new DataTable("Products");
dt.Load(PageProductsTable(10, 10));
dataGridView1.DataSource = dt;
```

در این‌جا، یک شئ DataTable جدید ایجاد شده و سپس متد Load برای ارسال نتایج SQLDataReader به DataTable استفاده می‌شود. سپس DataTable به یک شئ dataGridView مقید می‌شود.

Section ۳۷،۰۵ درج یکجا

بهبود مهم دیگر در ADO.NET 2.0، شئ SQLBulkCopy جدید است. شئ SQLBulkCopy یک متد با کارآیی بالا برای انتقال اشیا بین پایگاه‌های داده مختلف با سیستم‌های SQL Server متفاوت فراهم می‌کند. این مثال نحوه استفاده از شئ SQLBulkCopy را تشریح می‌کند:

```
// Create source & destination connections
SqlConnection cnSource = new SqlConnection
  ("SERVER=TECA-YUKON;INTEGRATED SECURITY=True;"
    + "DATABASE=AdventureWorks");
SqlConnection cnDest = new SqlConnection
  ("SERVER=TECA-YUKON;INTEGRATED SECURITY=True;"
    + "DATABASE=AdventureWorks2");

cnSource.Open();
cnDest.Open();

// Read the source data
SqlCommand cmd = new SqlCommand
  ("SELECT * FROM Sales.SpecialOffer", cnSource);
SqlDataReader rdr = cmd.ExecuteReader();

// Create SqlBulkCopy object and write the destination data
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
SqlBulkCopy bulkData = new SqlBulkCopy(cnDest);
bulkData.DestinationTableName = "SpecialOffers";
bulkData.WriteToServer(rdr);

bulkData.Close();
cnSource.Close();
cnDest.Close();
```

در این مثال، دو شئ اتصال ایجاد می‌شود که به پایگاه‌های داده مختلفی در یک سیستم اشاره می‌کنند. اولین شئ اتصال از پایگاه داده AdventureWorks استفاده می‌کند و دومین شئ از یک کپی به نام AdventureWorks2 استفاده می‌کند. هر دو شئ اتصال باز شده و سپس یک SQLDataReader برای خواندن داده از اتصال منبع استفاده می‌شود. سپس یک شئ SQLBulkCopy ایجاد شده و به شئ اتصال مقصد متصل می‌شود. سپس، متد WriteToServer شئ SQLBulkCopy با استفاده از SQLDataReader ای که به اتصال منبع ضمیمه شده است، فراخوانی می‌شود. متد WriteToServer داده را از منبع به مقصد کپی می‌کند. توجه به این نکته مهم است که شئ مقصد باید در اتصال مقصد وجود داشته باشد. متدهایی اضافی در شئ SQLBulkCopy وجود دارند که می‌توانید از آن‌ها برای انجام نگاشت طرح‌واره اختصاصی بین جداول منبع و مقصد استفاده کنید.

Section ۳۷.۰۶ مدل اتصال مشترک

یکی از مشکلات ADO.NET 1.0 این واقعیت بود که از تأمین کننده آگاه نبود. به عبارت دیگر، باید از تأمین کننده خاصی برای اتصال به یک محیط پایگاه داده مقصد خاص استفاده می‌کردید. مثلاً، SQLClient تنها می‌توانست به سیستم‌های SQL Server متصل شود و نه به سیستم‌های Oracle. ضمناً، OracleClient تنها می‌توانست به سیستم‌های Oracle متصل شود و نه به سیستم‌های SQL Server. در حالی که می‌توانستید کد خود را برای بارگذاری تأمین کننده صحیحی بارگذاری کنید، نتیجه عالی نبود و راحت نیز نبود. ADO.NET 2.0 این مشکل را با افزودن یک قابلیت Provider Factory جدید که قادر به نمونه‌سازی تأمین کننده مناسب در زمان اجراست، حل کرده است. این مثال نحوه استفاده از Provider Factory جدید را تشریح می‌کند:

```
DbDataReader rdr;
DbProviderFactory provider =
   DbProviderFactories.GetFactory("System.Data.SqlClient");

using (DbConnection cn = provider.CreateConnection())
{
   using (DbCommand cmd = provider.CreateCommand())
   {
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
        cmd.CommandText = "SELECT * FROM Production.Location";
        cmd.Connection = cn;
        cn.ConnectionString =
          ("SERVER=TECA-YUKON;INTEGRATED SECURITY=True;"
            + "DATABASE=AdventureWorks");
        cn.Open();
        rdr = cmd.ExecuteReader(CommandBehavior.CloseConnection);
        DataTable dt = new DataTable("Product Locations");
        dt.Load(rdr);
        dataGridView1.DataSource = dt;
    }
}
```

در این‌جا، می‌توانید ببینید که متد GetFactory برای ایجاد نمونه‌ای از داده System.Data.SQLClient فراهم شده در زمان اجرا استفاده می‌شود. سپس، یک شئ DbCommand برای اجرای فرمانی برای بازیابی محتویات جدول Production.Location استفاده می‌شود که به DbDataReader ارسال می‌شود. بالاخره، نتایج DbDataReader برای یک DataTable بارگذاری می‌شود که به شئ dataGridView مقید شده است.

تـــوجـه : **باید فضای نام System.Data.Common را برای استفاده از اشیای DbProvider داشته باشید.**

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

فصل پنجم

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

سرویس‌های هشداردهی

سرویس‌های هشداردهی[1] ابتدا در سال ۲۰۰۲ به عنوان یک download وب برای SQL Server 2000 معرفی شدند. در SQL Server 2005، سرویس‌های هشداردهی به عنوان زیر سیستمی جدید به محصول پایه اضافه شده‌اند. سرویس‌های هشداردهی یک چارچوب کاری هستند که به شما امکان ساخت برنامه‌های پیام‌رسانی اختصاصی را می‌دهند که این برنامه‌ها قادر به قرار دادن اطلاعات اختصاصی برای چندین مشترک و وسیله هستند. سرویس‌های هشداردهی توسط انواعی از شرکت‌های معروف برای تحویل اطلاعات شخصی به مشتریان آن‌ها استفاده می‌شوند. برای نمونه، موبایل MSN از سرویس‌های هشداردهی برای ارسال اخبار خصوصی و سایر اطلاعات به انواع وسایل موبایل استفاده می‌کند. پیاده‌سازی سرویس‌های هشداردهی معروف دیگر در ESPN.com است که سرویسی را فراهم کرده است که مشتریان می‌توانند برای یافتن نتایج مسابقات ورزشی استفاده کنند. سرویس‌های هشداردهی SQL Server 2005 به شما اجازه می‌دهند تا این انواع برنامه‌های شیوه هشداردهی یکسان را بسازید. در حالی که می‌توانید برنامه‌های هشداردهی را از صفر بسازید، سرویس‌های هشداردهی SQL Server 2005 شروعی عالی در این نوع پروژه را با فراهم کردن یک چارچوب کاری مستحکم، مقیاس‌پذیر و تست شده به شما می‌دهند که می‌توانید از آن‌ها به عنوان پایه‌ای برای برنامه‌های هشداردهی خود استفاده کنید.

در این فصل، مطالبی درباره زیرسیستم سرویس‌های هشداردهی جدید می‌آموزید. در اولین بخش این فصل، مروری بر این زیر سیستم جدید خواهید داشت. سپس، مطالبی درباره نحوه ساخت برنامه‌های سرویس‌های هشداردهی می‌آموزید. برای شروع، مروری بر فرآیند برنامه‌نویسی خواهید داشت و سپس مقداری کد نمونه را می‌بینید که نحوه استفاده از چارچوب کاری سرویس‌های هشداردهی را برای ساخت برنامه‌های شیوه هشداردهی تشریح می‌کنند.

Article XXXVIII. مروری بر سرویس‌های هشداردهی

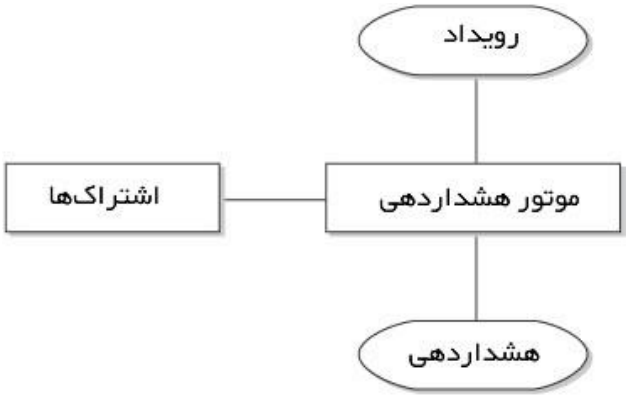
یک برنامه سرویس‌های هشداردهی یک لایه نرم‌افزاری است که بین یک منبع اطلاعات و دریافت کننده مورد نظر آن اطلاعات قرار دارد. برنامه سرویس‌های هشداردهی رویدادهای از پیش‌تعریف شده خاص را کنترل کرده و می‌تواند به‌طور هوشمند اطلاعاتی درباره این رویدادها را برای

1- Notification Services

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

انواع وسایل مقصد مختلف با استفاده از یک زمانبند تحویل شخصی فیلتر کرده و مسیریابی می‌کند. برنامه‌های سرویس‌های هشداردهی شامل سه جزء اصلی هستند: رویدادها، اشتراک‌ها و هشدارها. شکل ۱-۵ مروری سطح بالا از یک برنامه سرویس‌های هشداردهی را فراهم می‌کند.



(a)

شکل ۱-۵ مروری بر سرویس‌های هشداردهی (b)

Section ۳۸،۰۲ رویدادها

در یک برنامه سرویس‌های هشداردهی، رویدادها تنها چیزی هستند که به نظر می‌رسند (مواردی اتفاق می‌افتد که باید از آن‌ها مطلع شوید). در مورد مثال ESPN.com، یک رویداد ممکن است چیزی شبیه نتایج نهایی برخی رخدادهای ورزشی باشد. برای NASDAR، یک رویداد ممکن است افزایش بها به سطح خاصی باشد. در سرویس‌های هشداردهی SQL Server 2005، رویدادها به عنوان ردیف‌هایی در یک جدول از پایگاه داده Notification Services ذخیره می‌شوند.

Section ۳۸،۰۳ اشتراک‌ها

اشتراک‌ها[1] نحوه بیان کاربران یک برنامه Notification Services به سیستم درباره نوع رویدادهایی هستند که در آن‌جا درج می‌شوند. مثلاً، در مثال Notification Services مبتنی بر ورزش، یک کاربر ممکن است اشتراکی را برای داشتن نتایج نهایی بازی‌های Los Angeles Lakers ایجاد کند. در مثال Notification Services قیمت‌های سهام، یک کاربر ممکن است اشتراکی را برای مطلع شدن ایجاد کند، هنگامی که قیمت سهام مایکروسافت به بیش از ۵۰ دلار در هر اشتراک برسد. SQL

1- Subscriptions

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Server 2005 Notification Services اشتراک‌ها را نظیر رویدادها به عنوان ردیف‌هایی در یک جدول ذخیره می‌کنند.

Section ۳۸.۰۴ هشدارها

یک هشدار[1] لزوماً پیامی است که به کاربر نهایی ارسال خواهد شد و حاوی اطلاعات مربوط به رویدادی است که کاربر مشترک آن است. برای مثال در سرویس‌های هشداردهی ESPN، این ممکن است یک پیام e-mail باشد که تیم دلخواه شما بر رقبای خود پیروز شده و این نتیجه نهایی است. هشدارها می‌توانند به فرمت‌های مختلف برای انواع وسایل مقصد مختلف تحویل شوند.

Section ۳۸.۰۵ موتور هشداردهی

کار برنامه Notification Services، کنترل رویدادهای خارجی و یافتن تطابق‌هایی بین رویدادها و اشتراک‌های ثبت شده است. هنگامی که رویدادی با اشتراکی تطابق دارد، موتور Notification Services هشداری را به کاربر نهایی ارسال می‌کند. مقیاس‌پذیری یک برنامه Notification Services وابسته به بخش بزرگی از نحوه تطابق رویدادها با اشتراک‌ها توسط موتور Notification Services است. مایکروسافت چارچوب کاری Notification Services مرتبط را به صورت مقیاس‌پذیر در سطح اینترنت طراحی کرده است، بدین معنی که با محیطی مناسب، SQL Server 2005 Notification Services می‌تواند برای مدیریت میلیون‌ها رویداد، اشتراک و هشدار رشد کند. برای انجام این کار، Notification Services از موتور پایگاه داده رابطه‌ای بسیار کارآمد SQL Server 2005 برای الحاق ردیف‌هایی از جدول رویدادها با ردیف‌های جدول اشتراک‌ها برای تطابق رویدادها با اشتراک‌ها بهره ببرند.

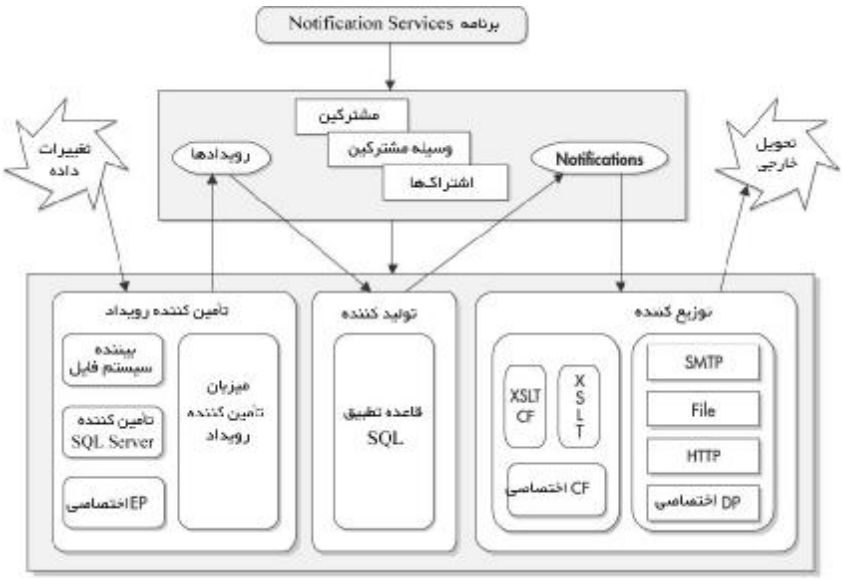
Article XXXIX. معماری Notification Services

SQL Server 2005 Notification Services محیطی است که طراحی شده است تا به شما امکان نوشتن و توزیع برنامه‌های هشداردهی با مقیاس‌پذیری بالا را می‌دهد. برنامه‌های SQL Server 2005 Notification Services با استفاده از T-SQL و XML ایجاد می‌شوند و با استفاده از .NET Framework یکپارچه‌ای اجرا می‌شوند که در SQL Server 2005 ساخته شده است. نگارش‌های ۳۲ و ۶۴ بیتی محیط Notification Services وجود دارند. می‌توانید مروری بر معماری SQL Server 2005 Notification Services را در شکل ۲-۵ ببینید.

1- Notification

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

(b)

شکل ۲-۵ مروری بر معماری Notification Services

در بالای شکل ۲-۵، می‌توانید برنامه Notification Services را ببینید. این برنامه توسط برنامه‌نویس نوشته می‌شود؛ وظیفه اصلی آن، افزودن اطلاعات اشتراک در جدول Notification Services Subscriptions است. این برنامه می‌تواند یک برنامه وب ASP.NET، یک برنامه ویندوزی Win32 استاندارد یا هر نوع برنامه‌ای که قادر به فراخوانی APIهای مدیریت شده .NET است (به‌طور مستقیم یا از طریق کلاس‌های .NET COM Interop)، باشد.

در سمت چپ شکل ۲-۵، می‌توانید بخش رویدادهای معماری SQL Server 2005 Notification Services را ببینید. رویدادها توسط تأمین کنندگان رویداد[1] به سیستم اضافه می‌شوند. این تأمین کنندگان رویداد، منابع دستیابی خارجی را برای بررسی تغییرات کنترل می‌کنند. SQL Server 2005 همراه با دو تأمین کننده رویداد تعبیه شده است: یک ناظر سیستم فایل[2] که تشخیص تغییرات در فایل‌های سیستم عامل را ممکن می‌سازد و تأمین کننده رویداد SQL Server که یک پایگاه داده Analysis Services یا SQL Server را برای تغییرات کنترل می‌کند. Notification Services هم‌چنین می‌تواند توسعه یابد تا سایر منابع داده خارجی را از طریق کاربرد تأمین کنندگان

1- Event Providers
2- File System Watcher

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

رویداد اختصاصی بررسی کنید. هنگامی که یک تأمین کننده رویداد با تغییر داده‌ای مواجه می‌شود، آن رویداد را در جدول events پایگاه داده Notification Services ثبت می‌کند.

مولد (یا موتور) Notification Services که در وسط شکل ۲-۵ نشان داده شده است، از منطق فراهم شده توسط برنامه‌نویس Notification Services برای تطبیق رویدادهای ذخیره شده توسط تأمین کنندگان رویداد برای اشتراک‌های وارد شده توسط کاربران نهایی برنامه استفاده می‌کند. هنگامی که رویدادی با یک اشتراک تطابق دارد، مولد ردیفی حاوی اطلاعات مشترک و رویداد را در جدول notifications پایگاه Notification Services اضافه می‌کند.

توزیع کننده[1] که در سمت راست پایین شکل ۲-۵ نشان داده شده است، جدول notifications را برای ورودی‌های جدید کنترل می‌کند. هنگامی که مولد، ورودی جدیدی را به جدول notifications اضافه می‌کند، توزیع کننده آن ورودی را بازیابی کرده و آن را برای تحویل فرمت می‌کند. به‌طور پیش‌فرض، توزیع کننده، هشدار را با استفاده از XSLT فرمت خواهد کرد که خروجی را به روشی نمایش می‌دهد که با وسیله خروجی مشخص شده سازگار است. هنگامی که هشدار فرمت شده باشد، آن‌گاه توزیع کننده آن را برای تحویل به کاربر نهایی به تأمین کننده توزیع ارسال می‌کند. SQL Server 2000 همراه با سه تأمین کننده توزیع تعبیه شده است: یک تأمین کننده SMTP، یک تأمین کننده فایل و یک تأمین کننده HTTP که می‌تواند برای SOAP، NET Alerts. و هشدارهای SMS استفاده شود. همان‌گونه که احتمالاً حدس زده‌اید، تأمین کننده SMTP، هشدارها را از طریق یک سرور e-mail از قبیل Exchange تحویل می‌دهد. تأمین کننده فایل، هشدار را به شکل یک فایل سیستم عامل تحویل می‌دهد. تأمین کننده HTTP هشدار را به عنوان یک صفحه وب HTML یا با استفاده از یکی از پروتکل‌های مرتبط با HTTP دیگر تحویل می‌دهد. علاوه بر این موارد، هم‌چنین می‌توانید قابلیت‌های تحویل توزیع کننده را با افزودن یک تأمین کننده توزیع اختصاصی توسعه دهید.

Article XL. نوشتن برنامه‌های Notification Services

در اولین بخش این فصل، مروری بر SQL Server 2005 Notification Services جدید داشتیم. در این بخش، به بررسی عمقی‌تر خواهیم پرداخت، بنابراین مطالبی درباره مراحل واقعی موجود در نوشتن برنامه‌های SQL Server 2005 Notification Services می‌آموزید. ابتدا مروری بر فرآیند برنامه‌نویسی خواهیم داشت و سپس یک برنامه Notification Services بسیار ساده را می‌سازیم.

1- Distributor

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۴۰،۰۱ مراحل برنامه‌نویسی

فرآیند نوشتن برنامه‌های Notification Services با تعریف طرح‌واره و قوانینی شروع می‌شود که نحوه کار برنامه را مشخص می‌کنند. سپس، باید برنامه را کامپایل کنید. سپس باید رابطی بسازید که به کاربر اجازه افزودن اشتراکاتی را به برنامه می‌دهند. بالاخره، باید اجزایی اختصاصی را اضافه کنید که شاید توسط برنامه مورد نیاز باشند. اجازه دهید نگاهی به هر یک از این مراحل به‌طور مفصل داشته باشیم.

تعریف طرح‌واره و قوانین

برنامه‌نویس Notification Services از ترکیبی از XML و T-SQL برای تعریف قوانین و طرح‌واره برنامه استفاده می‌کند. هنگامی که طرح‌واره و قوانینی را برای یک برنامه Notification Services تعریف می‌کنید، لزوماً رویدادهایی را که برنامه کنترل خواهد کرد و اشتراک‌های برنامه، هشدارهای آن و منطقی که برای تطابق رویدادها با اشتراک‌ها استفاده خواهد شد، توصیف می‌کند. قوانین و طرح‌واره برنامه Notification Services در اصل در دو فایل تعریف می‌شوند. Adf.xml و config.xml. می‌توانید این فایل‌ها را با استفاده از یک ویراستار متنی استاندارد یا یک ویراستار مبتنی بر XML نظیر Visual Studio 2005 یا XMLSpy ایجاد کنید. اطلاعات مفصل‌تر درباره محتویات خاص فایل‌های adf.xml و config.xml در ادامه فصل ارایه می‌شود.

کامپایل برنامه

بعد از ایجاد طرح‌واره و قوانین، مرحله بعدی در ساخت یک برنامه Notification Services، کامپایل تمام کد و رجیستر کردن سرویسی است که برنامه‌های Notification Services را اجرا خواهد کرد. برای کامپایل برنامه، SQL Server 2005 برنامه سودمند خط فرمان nscontrol را فراهم کرده است که برای ایجاد، رجیستر کردن و هم‌چنین بهنگام‌رسانی برنامه Notification Services استفاده می‌شود. علاوه بر این، SQL Server Management Studio، یک کادر محاوره‌ای را فراهم کرده است که به شما امکان ایجاد محاوره‌ای برنامه‌های Notification Services را می‌دهد.

ساخت برنامه‌های مدیریتی اشتراک هشدار

دو مرحله اول، موتور اصلی برنامه Notification Services را می‌سازند. هرچند، کاربران هنوز نیاز به روشی برای افزودن اطلاعات اشتراک برای برنامه Notification Services دارند. برای این که به کاربران امکان وارد کردن اطلاعات اشتراکشان را بدهید، برنامه Notification Services نیاز به یک

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

رابط مدیریت اشتراک دارد که معمولاً یک برنامه وب یا GUI است که با استفاده از فناوری‌های ASP.NET، VB.NET در الحاق با یک محیط برنامه‌نویسی نظیر Visual Studio 2005 ساخته می‌شود.

افزودن اجزای اختصاصی

بالاخره، آخرین مرحله در ساخت برنامه Notification Services، افزودن اختیاری هر جزء اختصاصی است که ممکن است مورد نیاز برنامه باشد. اجزای اختصاصی شامل هر تأمین کننده رویداد مورد نیاز، فرمت‌های محتوا یا پروتکل‌های تحویل هشداری هستند که در محصول SQL Server 2005 Notification Services پایه نیستند.

Section ۴۰،۰۲ برنامه Notification Services نمونه

برنامه Notification Services نمونه که در بخش بعدی این فصل ارایه می‌شود، مقدار یک ستون را در یک جدول SQL Server کنترل می‌کند. هنگامی که مقدار آن ستون از آستانه خاصی تجاوز می‌کند، برنامه Notification Services نمونه، یک هشدار e-mail را به کاربر نهایی ارسال خواهد کرد. این مثال نگارش ساده‌ای از مثال Stock است که در برنامه‌های نمونه SQL Server 2000 Notification Services وجود داشت. برای این که همه موارد کار کنند، این برنامه با استفاده از تأمین کننده SQL Server 2005 برای بررسی یک جدول در پایگاه داده شروع می‌کند. یک رویداد باید ایجاد شده و مقدار ستونی را در آن جدول بررسی کند، کاربر باید اشتراکی را برای آن رویداد وارد کند و یک قانون باید اضافه شود تا به SQL Server اجازه دهد تا رویدادها را با اشتراک‌ها تطبیق دهد. هنگامی که رویدادی با قانون رویدادی تطابق داشته باشد، تأمین کننده توزیع e-mail یک برنامه با فرمت XML را به کاربر نهایی ارسال خواهد کرد. حال که مروری بر برنامه Notification Services نمونه داشتید، اجازه دهید تا ساخت آن را ببینیم.

ایجاد فایل‌های تعریف و پیکربندی برنامه

برنامه‌های Notification Services شامل دو فایل اصلی هستند: یک فایل تعریف برنامه[1] (ADF) و یک فایل پیکربندی برنامه[2] (ACF) که هر دو فایل XML هستند و باید برطبق طرح‌واره‌های xsd آن‌ها ساخته شوند. طرح‌واره‌های XSD برای اطمینان از این مسأله استفاده می‌شوند که هر دو سند مالک عناصر و صفات مورد نیاز هستند. فایل ACF نام برنامه Notification Services و نام نمونه

1- Application Definition File
2- Application Configuration File

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

آن و مسیر دایرکتوری برنامه را تعریف می‌کند. نام نمونه لزوماً نام یک سرویس ویندوز است که برنامه Notification Services را اجرا می‌کند.

فایل ADF فایل اصلی برای Notification Services است؛ بخش‌های متفاوت ADF، رویداد، اشتراک، قوانین و ساختار هشدار را که توسط برنامه Notification Services به کار خواهند رفت، توصیف می‌کنند. برای SQL Server 2000، باید تمام کدنویسی XML را به‌طور دستی برای ایجاد این دو فایل انجام دهید. این امر هم‌چنین برای بتاهای اولیه SQL Server 2005 درست است. هرچند، SQL Server 2005 نهایی دارای ابزارهای گرافیکی خواهد بود که به شما امکان ایجاد فایل‌های ACF و ADF را بدون کدنویسی دستی XML در یک ویراستار متنی می‌دهد.

علاوه بر این دو فایل اصلی که ساختار و برنامه Notification Services را تعریف می‌کنند، یک برنامه Notification Services نیاز به یک برنامه مدیریت اشتراک دارد که مسئول افزودن اطلاعات اشتراک برای پایگاه داده برنامه Notification Services است. در حالی که ADF و ACF در اصل با استفاده از XML و T-SQL ساخته می‌شوند، برنامه مدیریت اشتراک معمولاً با استفاده از یک زبان سطح بالا با یک رابط کاربر گرافیکی ساخته می‌شود.

نکته: ساختار راه‌حل چند پروژه‌ای Visual Studio .NET به شما امکان گروه‌بندی ساده فایل‌های تعریف برنامه XML و کد مدیریت اشتراک را به عنوان دو پروژه‌ای که بخشی از یک راه‌حل مشترک هستند، می‌دهد.

این لیست فایل ACF (config.xml) را نشان می‌دهد که در برنامه Notification Services نمونه این فصل استفاده شد:

```
<?xml version="1.0" encoding="utf-8"?>
<NotificationServicesInstance xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.microsoft.com/MicrosoftNotificationServices
    /ConfigurationFileSchema">
    <InstanceName>NSSampleInstance</InstanceName>
      <SqlServerSystem>tecayukon</SqlServerSystem>
    <Applications>
      <Application>
        <ApplicationName>NSSample</ApplicationName>
        <BaseDirectoryPath>C:\NSSample</BaseDirectoryPath>
          <ApplicationDefinitionFilePath>ADF.xml
          </ApplicationDefinitionFilePath>
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
    </Application>
  </Applications>
  <DeliveryChannels>
    <DeliveryChannel>
      <DeliveryChannelName>EmailChannel
        </DeliveryChannelName>
        <ProtocolName>SMTP</ProtocolName>
    </DeliveryChannel>
  </DeliveryChannels>
</NotificationServicesInstance>
```

می‌بینید که ACF یک سند نسبتاً ساده است. این فایل می‌تواند با استفاده از هر ویراستار متنی یا مبتنی بر XML ایجاد شود. مهم‌ترین نکات توجه برانگیز تگ‌های SqlServerSystem، InstanceName، ApplicationName، BaseDirectoryPath و ApplicationDefinitionFilePath هستند. همان‌گونه که ممکن است حدس زده باشید، تگ نام SqlServerSystem حاوی نام سیستمی SQL Serverای است که میزبان پایگاه‌های داده Notification Services خواهد بود، تگ InstanceName، نام نمونه‌ای را برای برنامه تعریف می‌کند و تگ ApplicationName نام برنامه Notification Services را تعریف می‌کند. BaseDirectoryPath به کامپایلر محل یافتن فایل ADF را می‌گوید و تگ ApplicationDefinitionFilePath نام سند XML حاوی کد ADF را مهیا می‌سازد. علاوه بر این آیتم‌های پایه، ACF هم‌چنین از تگ DeliveryChannel برای تعریف نحوه تحویل هشدارها استفاده می‌کند. در این مثال، تگ DeliveryChannel از پروتکل SMTP برای تحویل هشدارهای e-mail استفاده می‌کند.

تعریف طرح‌واره رویداد ADF

تعاریف اصلی که کنترل می‌کنند چگونه یک برنامه Notification Services کار می‌کند، در ADF قرار دارند. اولین کاری که باید برای ساخت برنامه نمونه انجام دهید، ساخت طرح‌واره‌ای برای رویدادهاست. در فایل ADF، عنصر EventClasses حاوی کد XML مورد استفاده برای تعریف رویدادهای Notification Services است. عنصر EventClasses می‌تواند حاوی چندین تعریف رویداد باشد. هر تعریف رویداد در یک زیرعنصر EventClass مجزا توصیف می‌شود. قطعه کد زیر از فایل adf.xml مثال، کد XML مورد استفاده برای ایجاد نمونه را تشریح می‌کند:

```
<?xml version="1.0" encoding="utf-8" ?>
<Application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.microsoft.com/MicrosoftNotificationServices/
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
ApplicationDefinitionFileSchema">
 <!-- Describe the Events  -->
 <EventClasses>
   <EventClass>
     <EventClassName>DataEvents</EventClassName>
     <Schema>
       <Field>
         <FieldName>DataID</FieldName>
         <FieldType>int</FieldType>
         <FieldTypeMods>not null</FieldTypeMods>
       </Field>
       <Field>
         <FieldName>DataValue</FieldName>
         <FieldType>int</FieldType>
         <FieldTypeMods>null</FieldTypeMods>
       </Field>
     </Schema>
     <IndexSqlSchema>
       <SqlStatement>CREATE INDEX DataEventsIndex
         ON DataEvents ( DataID )</SqlStatement>
     </IndexSqlSchema>
     <Chronicles>
       <Chronicle>
         <ChronicleName>DataEventsTable</ChronicleName>
         <SqlSchema>
           <SqlStatement>
             IF EXISTS(SELECT name FROM dbo.sysobjects
             WHERE name = 'DataEventsTable')
             DROP TABLE dbo.DataEventsTable
             CREATE TABLE DataEventsTable
             (
               [DataID]    int,
               [DataValue]    int
             )
           </SqlStatement>
         </SqlSchema>
       </Chronicle>
     </Chronicles>
   </EventClass>
 </EventClasses>
```

تمام فایل‌های ADF باید با عناصر برنامه شروع شوند که همان‌گونه که ممکن است حدس زده باشید، برنامه Notification Services را ارایه می‌دهند. عناصر اصلی در عنصر برنامه‌ای که برنامه را

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

تعریف می‌کند، عناصر EventClasses، SubscriptionClasses و NotificationClasses هستند. لیست کد قبل حاوی بخش EventClasses از ADF است. به دلیل این که این برنامه نمونه تنها از یک رویداد استفاده می‌کند، عنصر EventClasses تنها حاوی یک عنصر EventClass به نام DataEvents است. بخش طرح‌واره رویداد را تعریف می‌کند که برنامه Notification Services کنترل خواهد کرد. در این مورد، دو ستون تعریف می‌شوند: ستون DataID که یک شناسه ساده است و ستون DataValue که حاوی یک مقدار مرتبط است. شباهت این دو تعریف برای ستون‌های SQL Server، تصادفی نیست. Notification Services از این تعاریف برای ایجاد ستون‌ها در پایگاه داده Notification Services استفاده می‌کند. برای اطمینان از کارآیی خوب، تگ IndexSqlSchema برای ایجاد ایندکسی روی ستون DataID به کار می‌رود.

بخش Chronicles جداولی را تعریف می‌کند که داده رویداد را برای استفاده از اشتراک‌های زمان‌بندی شده ذخیره می‌کند. بخش Chronicles دارای دو تگ مورد نیاز است: تگ ChronicleName و تگ SqlSchema. تگ ChronicleName نام جدول Chronicle را مهیا می‌کند و تگ SqlSchema حاوی عبارات T-SQL است که جدول را ایجاد می‌کنند. به دلیل این که این کد در طی ایجاد Notification Services و بهنگام‌رسانی فرآیندها استفاده می‌شود، ابتدا باید وجود جدول را بررسی کند و چنانچه آن را یافت، آن را حذف کند. سپس خط بعدی T-SQL، جدول را ایجاد می‌کند.

تعریف تأمین‌کننده‌های ADF

بعد از تعریف رویدادهایی که برنامه را کنترل خواهند کرد، مرحله بعدی در ایجاد برنامه Notification Services، تعریف تأمین کننده‌ای است که این رویدادها را به برنامه تحویل خواهد داد. در این قطعه کد از بخش بعدی فایل adf.xml، می‌توانید تعریفی برای تأمین کننده رویداد SQL Server مورد استفاده برای اتصال برنامه Notification Services به SQL Server ببینید:

```
<!-- Specify the SQL Server Event Provider -->
<Providers>
  <HostedProvider>
    <ProviderName>SQLData</ProviderName>
    <ClassName>SQLProvider</ClassName>
    <SystemName>tecayukon</SystemName>
    <Schedule>
      <Interval>P0DT00H00M60S</Interval>
    </Schedule>
    <Arguments>
      <Argument>
        <Name>EventsQuery</Name>
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
        <Value>SELECT DataID, DataValue FROM DataEventsTable</Value>
      </Argument>
      <Argument>
        <Name>EventClassName</Name>
        <Value>DataEvents</Value>
      </Argument>
    </Arguments>
  </HostedProvider>
</Providers>
```

بخش Providers از ADF تأمین کننده‌های رویداد مورد استفاده برنامه Notification Services را توصیف می‌کند. در این مثال، عنصر HostedProvider تأمین کننده رویداد SQL Server را تعریف می‌کند. عنصر ProviderName برای انتساب نامی به تأمین کننده استفاده می‌شود و عنصر SystemName نام سیستم SQL Server ای را که تأمین کننده به آن متصل خواهد شد، مهیا می‌کند.

عنصر Schedule تعریف می‌کند که تأمین کننده چند وقت به چند وقت به سیستم متصل خواهد شد؛ این فاصله زمانی با مقدار تعریف شده در عنصر Interval کنترل می‌شود. مقدار موجود در عنصر Interval از نوع داده مدت زمانی XML استفاده می‌کند. بخش 0DT این مقدار، یک فاصله زمانی تاریخی را با یک مقدار 0 نشان می‌دهد. بخش 00HR یک فاصله زمانی ساعتی را با مقدار 0 نشان می‌دهد. بخش 00M یک فاصله زمانی دقیقه‌ای را با مقدار 0 نشان می‌دهد. بخش 60S یک فاصله زمانی ثانیه‌ای را با مقدار 60 نشان می‌دهد. به عبارت دیگر، این فاصله زمانی با 60 ثانیه تنظیم می‌شود.

عنصر Arguments پرس‌وجویی را مهیا می‌کند که برای اقتباس داده از منبع رویداد استفاده خواهد شد. در این مثال، محتویات ستون DataID and DataValue در DataEventsTable هر ۶۰ ثانیه برای کلاس رویداد به نام DataEvents که در عنصر EventClass قبلی تعریف شده بود، بازیابی خواهد شد.

تعریف طرح‌واره اشتراک ADF

هنگامی که رویدادها توصیف شدند، مرحله بعدی در ایجاد فایل ADF، تعریف اشتراک‌هاست. این لیست کد، بخش بعدی فایل adf.xml را نشان می‌دهد که اشتراک‌های مورد استفاده برنامه Notification Services نمونه را نشان می‌دهد:

```
<!-- Describe the Subscription -->
<SubscriptionClasses>
  <SubscriptionClass>
    <SubscriptionClassName>DataSubs</SubscriptionClassName>
    <Schema>
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
    <Field>
      <FieldName>DeviceName</FieldName>
      <FieldType>nvarchar(255)</FieldType>
      <FieldTypeMods>not null</FieldTypeMods>
    </Field>
    <Field>
      <FieldName>SubLocale</FieldName>
      <FieldType>nvarchar(10)</FieldType>
      <FieldTypeMods>not null</FieldTypeMods>
    </Field>
    <Field>
      <FieldName>DataID</FieldName>
      <FieldType>int</FieldType>
      <FieldTypeMods>not null</FieldTypeMods>
    </Field>
    <Field>
      <FieldName>DataTriggerValue</FieldName>
      <FieldType>int</FieldType>
      <FieldTypeMods>not null</FieldTypeMods>
    </Field>
  </Schema>
  <IndexSqlSchema>
    <SqlStatement>CREATE INDEX DataSubIndex ON DataSubs
     ( DataID )</SqlStatement>
  </IndexSqlSchema>
  <EventRules>
    <EventRule>
      <RuleName>DataSubEventRule</RuleName>
      <Action>
        SELECT dbo.DataNotificationsNotify
          (s.DeviceName, s.SubLocale, e.DataID, e.DataValue)
          FROM DataSubs s JOIN DataEvents e
          ON s.DataID = e.DataID
          LEFT OUTER JOIN DataEventsTable t
          ON s.DataID = t.DataID
          WHERE s.DataTriggerValue &lt;= e.DataValue
          AND (s.DataTriggerValue &gt; t.DataValue
          OR .DataValue IS NULL)

        INSERT INTO DataEventsTable (DataID, DataValue)
        SELECT e.DataID, e.DataValue
          FROM DataEvents e
          WHERE e.DataID NOT IN
            (SELECT DataID from DataEventsTable)

        UPDATE DataEventsTable
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
            SET DataValue = e.DataValue
            FROM DataEvents e, DataEventsTable t
            WHERE e.DataID = t.DataID
            AND e.DataValue &gt; t.DataValue
        </Action>
        <EventClassName>DataEvents</EventClassName>
      </EventRule>
    </EventRules>
  </SubscriptionClass>
</SubscriptionClasses>
```

بخش SubscriptionClasses سند ADF شبیه EventClasses می‌تواند چندین اشتراک را توصیف کند که هر اشتراک در یک عنصر SubscriptionClass مجزا توصیف می‌شود. این مثال از یک SubscriptionClass به نام DataSubs استفاده می‌کند. بخش Schema اشتراک را توصیف می‌کند. فیلد DeviceName نوع وسیله مقصد را نشان می‌دهد. SubLocale برای تغییر اختیاری زبانی که مشترک برای بازیابی هشدار به کار خواهد برد، استفاده می‌شود. فیلدهای DataID و DataTriggerValue رویدادی را مشخص خواهند کرد که مشترک آن خواهند شد و مقدار داده‌ای که یک هشدار را تریگر خواهد کرد. باز هم عنصر IndexSqlSchema شبیه EventClass برای ایجاد ایندکسی روی ستون DataID برای کارآیی بهتر استفاده می‌شود. Notification Services از این توضیحات برای ایجاد ستون‌های پایگاه داده در SQL Server استفاده می‌کند. بعد از تنظیم اشتراک‌ها، بخش بعدی کد در عنصر EventRules منطقی را تعریف می‌کند که برنامه Notification Services برای تطابق رویدادها با اشتراک‌ها به کار خواهد برد. در حالی که اطلاعات Event و Subscription با استفاده از XML تعریف می‌شوند، قوانین رویداد با استفاده از کد T-SQL ایجاد می‌گردند که در عنصر EventRulesAction ذخیره می‌شوند. در این مثال، مهم‌ترین نکته قابل توجه این است که جدول DataSubs با جدول DataEvents تلفیق می‌شود که DataValue از جدول DataEvents بزرگ‌تر از DataTriggerValue است ولی کوچک‌تر از مقدار جدید است. هنگامی که شرط تلفیق برآورده می‌شود، ردیفی برای مشترک ایجاد خواهد شد. عبارات INSERT و UPDATE برای بهنگام‌رسانی جدول Chronicle هنگام پیدا نشدن یک مقدار یا هنگامی که مقدار بزرگ‌تر از مقدار Chronicle موجود باشد، استفاده می‌شود.

طرح‌واره هشدار ADF

بخش بعدی تعریف فایل ADF، تنظیم طرح‌واره NotificationClasses است. NotificationClasses نحوه تحویل اطلاعات هشداردهی را توصیف می‌کند. این لیست کد بخش آخر فایل adf.xml است که حاوی تعریف NotificationClass است. عنصر NotificationClasses می‌تواند

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

چندین نوع هشداردهی را توصیف کند که هر نوع در عنصر NotificationClass خاص خود توصیف می‌شود. به دلیل این که این برنامه نمونه تنها از یک نوع هشداردهی استفاده می‌کند، بخش NotificationClasses حاوی یک عنصر NotificationClass است.

```
<!-- Describes the Notifications -->
<NotificationClasses>
  <NotificationClass>
    <NotificationClassName>DataNotifications</NotificationClassName>
    <Schema>
      <Fields>
        <Field>
          <FieldName>DataID</FieldName>
          <FieldType>int</FieldType>
        </Field>
        <Field>
          <FieldName>DataValue</FieldName>
          <FieldType>int</FieldType>
        </Field>
      </Fields>
    </Schema>
    <!-- Specify the Content Format XSLT -->
    <ContentFormatter>
      <ClassName>XsltFormatter</ClassName>
      <Arguments>
        <Argument>
          <Name>XsltBaseDirectoryPath</Name>
          <Value>C:\NSSample</Value>
        </Argument>
        <Argument>
          <Name>XsltFileName</Name>
          <Value>NSSample.xslt</Value>
        </Argument>
      </Arguments>
    </ContentFormatter>
  </NotificationClass>
</NotificationClasses>
<Generator>
  <SystemName>tecayukon</SystemName>
</Generator>
<Distributors>
  <Distributor>
    <SystemName>tecayukon</SystemName>
  </Distributor>
</Distributors>
<ApplicationExecutionSettings>
  <PerformanceQueryInterval>PT5S</PerformanceQueryInterval>
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
</ApplicationExecutionSettings>

</Application>
```

در این لیست، می‌توانید ببینید که کلاس هشداردهی DataNotifications نامیده می‌شود. عنصر Schema کلاس‌های DataNotification اطلاعاتی را تعریف می‌کند که به مشترک ارسال خواهد شد. در این‌جا می‌توانید ببینید که مقدار فیلدهای DataID و DataValue به عنوان بخشی از هشداردهی ارسال خواهند شد.

عنصر ContentFormatter نحوه فرمت‌بندی هشداردهی را هنگام ارسال به مشترک تعریف می‌کند. این مثال کاربرد XSLFormatter را تشریح می‌کند. عنصر Arguments دایرکتوری محل فایل XSLT و نام فایل را توصیف می‌کند. در این لیست، می‌توانید ببینید که فایل XSLT در دایرکتوری C:\NSSample قرار دارد و NSSample.xslt نامیده می‌شود.

عناصر Generator، Distributor و ApplicationExecutionSettings سیستم SQL Serverای که برای تولید هشدارها استفاده خواهد شد، سیستمی که برای توزیع هشدارها استفاده خواهد شد و فاصله زمانی را که در آن، شمارنده‌های کارآیی سیستم بهنگام خواهند شد، مشخص می‌کنند.

فرمت‌بندی خروجی هشداردهی

در لیست قبل، گفتیم که هشداردهی با استفاده از شیوه‌نامه NSSample.xslt فرمت‌بندی می‌شود. می‌توانید در لیست زیر ببینید که این شیوه‌نامه نمونه چه شکلی است:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0"
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:template match="notifications">
  <HTML>
  <BODY>
   <xsl:apply-templates />
   <I>This message was generated using
   <BR/>Microsoft SQL Server Notification Services</I><BR/><BR/>
  </BODY>
  </HTML>
 </xsl:template>
 <xsl:template match="notification">
  <P>The data value of <B><xsl:value-of select="DataID"/></B>
  <BR/>reached the value <B><xsl:value-of select="DataValue"/></B>
  </P>
 </xsl:template>
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
</xsl:stylesheet>
```

شیوه‌نامه مورد استفاده برای فرمت‌بندی خروجی برنامه Notification Services، یک شیوه‌نامه HTML استاندارد است. در بخش قالب، می‌توانید ببینید کجا فیلدهای DataID و DataValue از NotificationClass در هشداردهی نمایش داده می‌شود.

ایجاد برنامه Notification Services

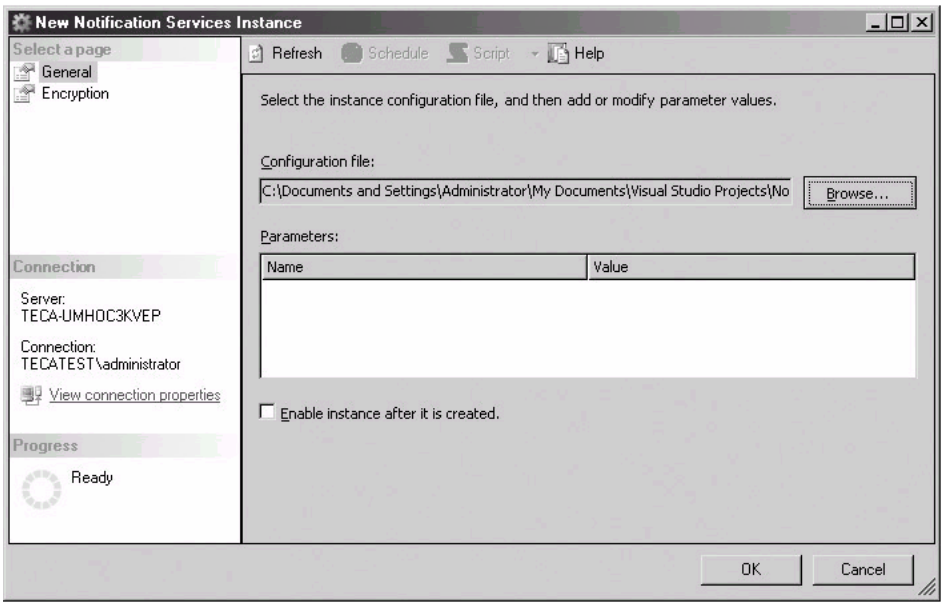
بعد از ایجاد کد برنامه T-SQL و XML مورد نیاز، آماده ساخت برنامه Notification Services هستید. برنامه‌های Notification Services می‌توانند به صورت محاوره‌ای با استفاده از SQL Server Management Studio ایجاد شوند یا می‌توانند با استفاده از برنامه سودمند nscontrol ایجاد گردند.

ایجاد برنامه‌های Notification Services با استفاده از SQL Server Management Studio

بعد از این که فایل‌های config.xml و adf.xml که Notification Services را تعریف می‌کنند، ایجاد شدند، می‌توانید از آن‌ها برای ساخت برنامه Notification Services خود از SQL Server Management Studio با باز کردن Object Browser و کلیک راست روی گره Notification Services استفاده کنید. سپس، می‌توانید گزینه New Notification Services Instance از منوی زمینه را برای نمایش صفحه‌ای شبیه شکل ۳-۵ انتخاب کنید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

شکل ۳-۵ کادر محاوره‌ای New Notification Services Instance (b)

برای ایجاد یک برنامه NotificationClasses جدید با استفاده از کادر محاوره‌ای New Notification Services Instance، روی Browse کلیک کرده و به دایرکتوری حاوی فایل config.xml برنامه خود بروید. سپس، فایل config.xml را انتخاب کرده و OK را کلیک کنید. اگر بخواهید برنامه بلافاصله بعد از ایجاد فعال شود، باید کادر انتخاب Enable Instance After It Is Created را در حالت تأیید قرار دهید.

ایجاد برنامه‌های Notification Services با استفاده از Nscontrol

Nscontrol یک ابزار خط فرمان است که برای ایجاد و راهبری برنامه‌های Notification Services استفاده می‌شود. Nscontrol تعداد فرامین عمل مختلفی را که می‌توانید برای کار کردن با برنامه‌های Notification Services استفاده کنید، می‌داند. جدول ۱-۵ فرامین عمل nscontrol موجود را فهرست کرده است.

جدول ۱-۵ فرامین nscontrol (c)

فرمان Nscontrol	شرح

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

nscontrol create	یک برنامه Notification Services و پایگاه‌های داده آن را ایجاد می‌کند.
nscontrol delete	یک برنامه Notification Services و پایگاه‌های داده آن را حذف می‌کند.
nscontrol disable	یک برنامه Notification Services را غیرفعال می‌کند.
nscontrol displayargumentkey	کلید مورد استفاده برای رمزگذاری داده رویداد را نمایش می‌دهد.
nscontrol enable	برنامه Notification Services را فعال می‌کند.
nscontrol listversions	نگارش Notification Services و هر یک از برنامه‌های رجیستر شده را نمایش می‌دهد.
nscontrol register	یک برنامه Notification Services را رجیستر می‌کند.
nscontrol status	وضعیت یک برنامه Notification Services را نمایش می‌دهد.
nscontrol unregister	یک برنامه Notification Services را از حالت رجیستر شده بر می‌دارد.
nscontrol update	یک برنامه Notification Services را بهنگام می‌کند.

ایجاد یک برنامه Notification Services یک فرآیند چند مرحله‌ای است. ابتدا، برنامه نیاز به ایجاد با استفاده از فرمان nscontrol create دارد. این امر پایگاه داده مورد استفاده برنامه Notification Services را ایجاد می‌کند. سپس، برنامه باید با استفاده از فرمان nscontrol register رجیستر شود. این امر سرویسی را ایجاد می‌کند که برای اجرای برنامه استفاده می‌شود. فایل دسته‌ای زیر، توالی فرمان مورد نیاز برای ایجاد برنامه NSSample Notification Services نمونه را تشریح می‌کند:

```
echo off
cls
set NSdir="C:\Program Files\Microsoft SQL Server\90\NotificationServices\9.0.242\bin"
echo ========================================
echo Beginning NSSampleInstance Creation
echo ========================================
echo .
echo Create the application databases
%NSdir%\nscontrol create -in config.xml

echo Register the application
%NSdir%\nscontrol register -name NSSampleInstance –service

echo Enable the application
%NSdir%\nscontrol enable -name NSSampleInstance

echo start the NS app as a service
net start NS$NSSampleInstance
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
echo Display the status of the app
%NSdir%\nscontrol status -name NSSampleInstance
```

آرگومان n– فرمان nscontrol create، نام Notification Services ACF را مشخص می‌کند. در این مثال، ACF، config.xml نامیده می‌شود. اجرای فرمان nscontrol create باعث ایجاد دو پایگاه داده در سرور می‌شود: NSSampleInstanceMain و NSSampleInstanceNSSample که رویدادهای داده و تعریف برنامه Notification Services را ذخیره می‌کنند.

فرمان nscontrol register از آرگومان name– برای تعیین نام نمونه برنامه Notification Services برای رجیستر شدن استفاده می‌کند. سوییچ service– آن را برای رجیستر کردن سرویسی به نام NS$NSSampleInstance هدایت می‌کند. فرمان nscontrol enable از پارامتر name– برای تعیین نام نمونه برنامه‌ای که فعال خواهد شد، استفاده می‌کند.

هنگامی که برنامه فعال می‌شود، سرویس آن می‌تواند با استفاده از فرمان net start اجرا شود. برای امتحان، هم‌چنین می‌توانید برنامه NS$NSSampleInstance از اعلان فرمان یا کادر محاوره‌ای Run استفاده کنید.

افزودن مشترکین برنامه

مشترکین با استفاده از APIهای کد مدیریت شده‌ای که مایکروسافت در SQL Server 2005 Notification Services فراهم کرده است، به برنامه‌های Notification Services اضافه می‌شوند. APIهای Microsoft .NET Framework به شما امکان اضافه کردن، بهنگام‌رسانی و حذف مشترکین و وسایل و اشتراک‌های مشترکین را می‌دهد. در حالی که Notification Services API از طریق کلاس‌های کد مدیریت شده فراهم شده است، هم‌چنین می‌توانید با استفاده از برنامه‌های COM مبتنی بر Win32 از کد مدیریت نشده به آن API دستیابی داشته باشید.

Notification Services API در Microsoft.SqlServer.NotificationServices.dll قرار دارد و باید به عنوان یک مرجع به پروژه .NET اضافه شود. سپس، می‌توانید از کلاس‌های Notification Services برای مدیریت اشتراک‌ها برای برنامه‌های Notification Services خود استفاده کنید. این نمونه کد، نحوه افزودن یک اشتراک را با استفاده از API کد مدیریت شده نشان می‌دهد:

```
using Microsoft.SqlServer.NotificationServices;
using System.Text;

public class NSSubscriptions
{
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
  private string AddSubscription(string instanceName, string
    applicationName, string subscriptionClassName, string subscriberId)
  {
    // Create the Instance object
    NSInstance myNSInstance = new NSInstance(instanceName);

    // Create the Application object
    NSApplication myNSApplication = new NSApplication
     (myNSInstance, applicationName);

    // Create the Subscription object
    Subscription myNSSubscription = new Subscription
      (myNSApplication, subscriptionClassName);
    myNSSubscription.Enabled = true;
    myNSSubscription.SubscriberId = subscriberId;

    // Set the subscription data fields
    myNSSubscription["DeviceName"] = "MyDevice";
    myNSSubscription["SubscriberLocale"] = "USA";
    myNSSubscription["DataID"] = 1;
    myNSSubscription["DataTriggerValue"] = 100;

    // Add the subscription
    string subscriptionId = myNSSubscription.Add();
    return subscriptionId;
  }
}
```

در بالای این لیست، می‌توانید یک رهنمود import را برای فضای نام Microsoft.SqlServer.NotificationServices ببینید. استفاده از رهنمود import به شما امکان استفاده از کلاس‌ها را در فضای نام NotificationServices بدون نیاز به تعیین کامل اسامی می‌دهد. سپس، در داخل متد AddSubscriptions، می‌توانید کدی را ببینید که اشتراکی را اضافه می‌کند. ابتدا یک شئ نمونه Notification Services ایجاد شده و بعد از آن شئ Application می‌آید. سپس، شئ Application برای ایجاد یک شئ اشتراک به نام myNSSubscriptions استفاده می‌شود. مقادیری به خصوصیات شئ Subscription نسبت داده شده و سپس متد Add آن برای افزودن اشتراکی به پایگاه داده فراخوانی می‌شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

فصل ششم

SQL Server Service Broker

SQL Server Service Broker یک زیرسیستم جدید است که پشتیبانی صف‌بندی غیرهم‌زمان تضمین شده را به SQL Server 2005 اضافه کرده است. صف‌بندی غیر هم‌زمان، بُعدی از مقیاس‌پذیری را به SQL Server 2005 اضافه می‌کند. صف‌بندی غیرهم‌زمان که در بیشتر برنامه‌های با مقیاس‌پذیری بالای دیگر وجود دارد، شامل زیرسیستم‌های I/O سیستم عامل، سرورهای وب و حتی عملیات داخلی خود موتور پایگاه داده SQL Server است. افزودن صف‌بندی غیرهم‌زمان به SQL Server 2005، قابلیت مدیریت صف‌بندی غیرهم‌زمان را برای برنامه‌های پایگاه داده کاربر نهایی نیز به همراه دارد. صف‌بندی غیرهم‌زمان عامل مهمی برای مقیاس‌پذیری است، زیرا به برنامه اجازه پاسخگویی به درخواست‌هایی بیشتر از محیط را که ممکن است قادر به مدیریت فیزیکی باشند، می‌دهد. برای نمونه، درمورد یک سرور وب، چنان‌چه ده هزار کاربر به‌طور هم‌زمان منابعی را از سرور درخواست کنند، بدون صف‌بندی غیرهم‌زمان، سرور وب مستأصل خواهد شد، زیرا تلاش می‌کند تا رشته‌هایی را برای مدیریت تمام درخواست‌های وارده مدیریت کند. صف‌بندی غیرهم‌زمان، صف‌بندی تمام درخواست‌ها را برای قرار گرفتن در یک صف ممکن می‌سازد. بنابراین به‌جای مستأصل شدن، سرور وب می‌تواند ورودی‌ها را از صف با حداکثر سطوح کارآیی پردازش کند. در مورد سرور وب، صف‌بندی غیرهم‌زمان به سرور امکان

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

مدیریت مؤثر تعداد اتصالات کاربر بیشتری را می‌دهد. SQL Server Service Broker به شما امکان ساخت این نوع مقیاس‌پذیری مشابه را در برنامه‌های پایگاه داده می‌دهد.

در این فصل، مطالبی درباره ویژگی‌های جدید فراهم شده توسط SQL Server Service Broker می‌آموزید. مروری بر زیرسیستم جدید خواهید داشت و مطالبی درباره اجزای اصلی آن می‌آموزید. سپس اصول ایجاد برنامه‌های SQL Server Service Broker را خواهید دید. در این‌جا مطالبی درباره فوق داده SQL Server Service Broker و مدل برنامه‌نویسی آن می‌آموزید. ابتدا، نحوه ایجاد صف‌ها و انواع پیام را خواهید دید. سپس، نحوه استفاده از فرامین T-SQL جدید را خواهید دید که به برنامه‌های پیام‌رسانی امکان افزودن ورودی‌ها به یک صف و دریافت ورودی‌هایی که به صف اضافه شده‌اند، می‌دهد. بالاخره، مطالبی درباره برخی از ویژگی‌های راهبری موجود در زیرسیستم SQL Server Service Broker می‌آموزید.

Article XLI. مروری بر SQL Server Service Broker

SQL Server Service Broker **توانایی انجام صف‌بندی غیر هم‌زمان را به** SQL Server 2005 اضافه کرده است. قابلیت صف‌بندی جدید در موتور SQL Server ساخته شده است و کاملاً تراکنشی است. تراکنش‌ها می‌توانند رویدادهای صف‌بندی شده را مشارکت دهند و می‌توانند commit یا roll back شوند. می‌توانید با استفاده از عبارات SQL جدید، به SQL Server Service Broker دستیابی داشته باشید. مثال‌هایی از این فرامین در بخش بعدی این فصل ارایه شده است. علاوه بر این، SQL Server Service Broker جدید هم‌چنین از تحویل قابل اتکای پیام‌ها به صف‌های راه دور پشتیبانی می‌کند. این بدان معنی است که برنامه‌های صف‌بندی که با استفاده از SQL Server Service Broker ساخته شوند می‌توانند چندین سیستم SQL Server را به وجود آورند و باز هم تحویل پیام تضمین شده را فراهم می‌کنند، حتی برای صف‌های راه‌دور. پیام‌هایی که در بین صف‌ها ارسال می‌شوند، می‌توانند بسیار بزرگ باشند (تا 2GB). SQL Server Service Broker به روش‌های مورد نیاز برای تقسیم‌بندی پیام‌های بزرگ در قطعات کوچک‌تری انجام خواهد داد که در بین شبکه ارسال شده و سپس در نهایت آن‌ها را مجدداً اسمبل می‌کند. هر دو نگارش ۳۲ و ۶۴ بیتی SQL Server Service Broker وجود دارد.

Section ۴۱.۰۱ طراحی برنامه صف

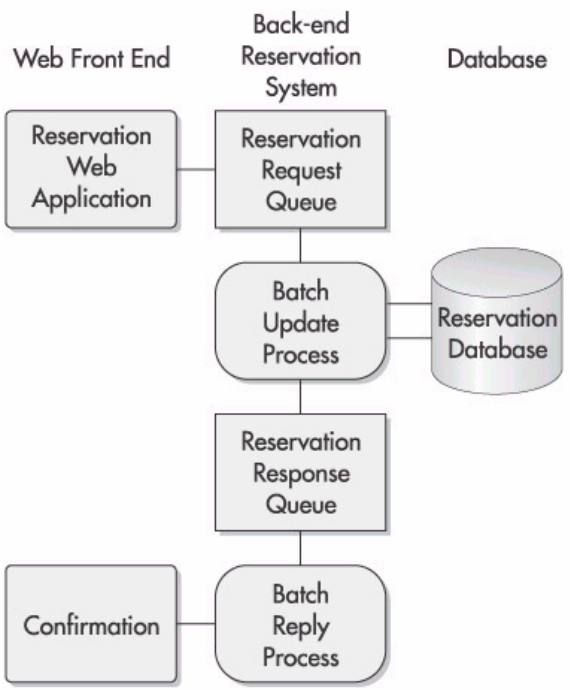
در حالی که ایده صف‌بندی در برنامه‌ها ممکن است برای بیشتر طراحان پایگاه داده کمی عجیب به نظر رسد، صف‌ها معمولاً در برنامه‌های با مقیاس‌پذیری بالا هستند. یکی از معروف‌ترین این نوع برنامه‌ها، سیستم‌های رزرواسیون خطوط هوایی مورد استفاده توسط خطوط هواپیمایی عمده نظیر

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

United، Delta و American و توسط دلالان مسافرتی نظیر Expedia و CheapTickets.com است. برای داشتن ایده‌ای در مورد نحوه استفاده صف‌بندی در یکی از این برنامه‌ها، می‌توانید به شکل ۱-۶ مراجعه کنید که می‌توانید طراحی یک برنامه صف‌بندی شده ساده را ببینید.

شکل ۱-۶ مروری سطح بالا از یک سیستم رزرواسیون هواپیمایی نمونه را نشان می‌دهد. در این‌جا می‌توانید ببینید که لایه ارایه برنامه توسط برنامه‌ای که روی یک بستر وب اجرا می‌شود، به مرور به کاربر نهایی تحویل می‌شود. این برنامه می‌تواند با استفاده از ASP.NET یا تعدادی از سایر زبان‌های برنامه‌نویسی نوشته شود. سپس برنامه front end با سیستم رزرواسیون واقعی محاوره خواهد کرد که به‌طور طبیعی روی سیستم دیگری اجرا می‌شود.



(a)

شکل ۱-۶ طراحی برنامه صف‌بندی شده (b)

به دلیل این که برنامه‌هایی نظیر این‌ها باید از هزاران کاربر هم‌زمان پشتیبانی کنند، آن‌ها نمی‌توانند قفل ردیف‌ها را فراهم کنند، در حالی که یک کاربر معین منتظر می‌ماند تا در مورد جزییات نهایی یک پرواز تصمیم‌گیری کند یا حتی یک رزرواسیون را شروع کند و سپس اجرا شود و برای

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

خاتمه برنامه‌ریزی کند. قفل‌گذاری ردیفی در این نوع روش، به‌طور جدی مانع از مقیاس‌پذیری و حتی قابلیت استفاده برنامه می‌شود. صف‌بندی این مشکل را با دادن امکان انجام یک درخواست غیرهم‌زمان برای یک رزرواسیون برطرف می‌کند و درخواستی را به سیستم رزرواسیون back-end ارسال کرده و بلافاصله برای کار دیگر آزاد می‌شود. در هیچ جایی از فرآیند تعیین رزرواسیون، هیچ قفلی روی جداول پایگاه داده نداریم. سیستم رزرواسیون back-end که لزوماً در حالت دسته‌ای کار می‌کند، درخواست رزرواسیون را خارج از صف خواهد گرفت و سپس بهنگام‌رسانی پایگاه داده را انجام می‌دهد. با توجه به این که بهنگام‌رسانی در حالت دسته‌ای و بدون محاوره کاربر انجام می‌شود و سریعاً اتفاق می‌افتد و حداقل زمان برای قفل کردن ردیف‌ها مورد نیاز است، در حالی که بهنگام‌رسانی انجام می‌شود. اگر درخواست موفق باشد، رزرواسیون کاربر نهایی تأیید می‌شود. در غیر این صورت، اگر درخواستی به این دلیل پذیرفته نشود که تمام صندلی‌ها رزرو شده‌اند و یا به دلایل دیگر، آن‌گاه رزرواسیون پذیرفته نخواهد شد و کاربر از این وضعیت مطلع می‌شود.

Section ۴۱،۰۲ گفتگوها

گفتگوها یک جزء ضروری SQL Server Service Broker جدید مایکروسافت هستند. لزوماً، گفتگوها[1] پیام‌رسانی دو روشه را بین دو نقطه انتهایی فراهم می‌کنند. نقاط انتهایی برای این پیام‌ها می‌توانند دو برنامه‌ای باشند که روی نمونه‌ها یا سرورهای متفاوتی اجرا می‌شوند یا می‌توانند دو برنامه‌ای باشند که روی یک سرور اجرا می‌شوند. شکل ۲-۶ گفتگوی SQL Server Service Broker را نشان می‌دهد.

(a)

شکل ۲-۶ گفتگوی SQL Server Service Broker (b)

هدف اصلی یک گفتگوی SQL Server Service Broker، فراهم کردن دنباله‌ای مرتب از رویدادهاست. گفتگوهای SQL Server Service Broker ترتیب رویداد قابل اتکایی را در بین سرور و یا رشته‌ها حفظ می‌کنند، حتی اگر خرابی شبکه، برنامه یا سرور وجود داشته باشد و به‌طور موقت پردازش رویدادهای صف‌بندی شده را مختل کرده باشد. هنگامی که پردازش صفی بازیابی می‌شود، رویدادها به

1- Dialogs

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

ترتیب به پردازش شدن ادامه می‌دهند که از محل آخرین رویداد صف‌بندی شده پردازش انجام می‌شود. گفتگوها به پیام‌های صف‌بندی شده امکان می‌دهند تا به همان ترتیبی خوانده شوند که در صف قرار دارند. گفتگوها می‌توانند برای پردازش رویدادها به حالت full-duplex یا حالت half-duplex تنظیم شوند.

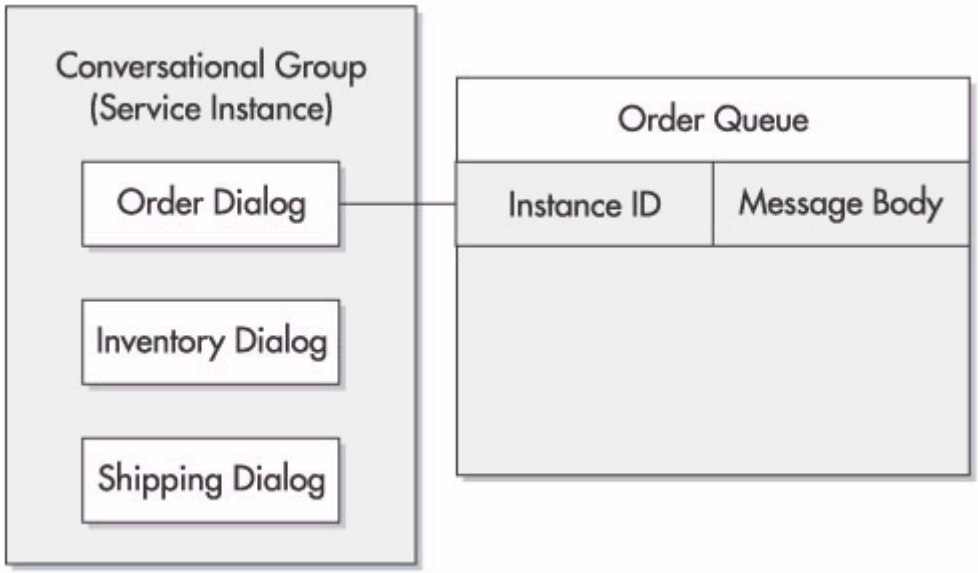
Section ۴۱،۰۳ گروه مکالمه

جزء اصلی دیگر زیر سیستم SQL Server Service Broker، گروه مکالمه است. گروه مکالمه[1] به مجموعه‌ای از گفتگوهای SQL Server Service Broker مرتبط است. یک برنامه ممکن است برای تکمیل یک وظیفه، نیاز به چندین گفتگو داشته باشد و یک گروه مطالعه به شما امکان می‌دهد تا تمام این گفتگوهای مرتبط را با یکدیگر به‌طور منطقی گروه‌بندی کنید. مثلاً، یک برنامه ورودی سفارش ممکن است گفتگویی برای پردازش سفارشات داشته باشد، دیگری صورت‌حساب را پردازش کند، دیگری خرید را مدیریت کند و باز هم دیگری درخواست‌های حواله‌ای را مدیریت کند. تمام این گفتگوهای مرتبط می‌توانند با یکدیگر گروه‌بندی شوند تا یک گروه مکالمه را تشکیل دهند. در این مورد، گروه مکالمه اصولاً تمام گفتگوها را برای یک برنامه نشان می‌دهد. شکل ۳-۶ رابطه گروه مکالمه و گفتگوی SQL Server Service Broker را نشان می‌دهد.

1- Conversation Group

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

(b)

شکل ۳-۶ گروه مکالمه SQL Server Service Broker

هدف اصلی تحت گروه مکالمه، فراهم کردن یک مکانیزیم قفل‌گذاری برای صف‌های مرتبط است. در این صحنه، مهم درک این نکته است که قفل‌گذاری بر هر جدول پایگاه داده مرتبط اعمال نمی‌شود. در عوض، قفل‌گذاری که توسط گروه مکالمه میسر می‌شود، بر موجودیت‌هایی در صف‌ها اعمال می‌شود. در مورد یک سفارش، این امر ممکن است بدین معنی باشد که هیچ فرآیندی نمی‌تواند برای هر یک از صف‌ها در گروه مکالمه شما، سفارشات را بخواند، در حالی که برنامه شما قفلی را روی این ورودی‌ها گذاشته است.

علاوه بر قفل‌گذاری، گروه مکالمه به برنامه امکان نگهداری حالت را نیز می‌دهد. نگهداری حالت همیشه یکی از مشکل‌ترین جنبه‌های یک برنامه غیرهم‌زمان بوده است، زیرا می‌تواند باعث تأخیرهای طولانی بین رسیدن پیام‌ها شود. باز نگه داشتن رشته‌های فعال در طی آن دوره زمانی یک راه‌حل نیست، زیرا تمام رشته‌های بی‌استفاده، منابع سیستم را به شدت مصرف می‌کنند و مقیاس‌پذیری را پایین می‌آورند. در عوض، گروه مکالمه حالت یک موجودیت را با استفاده از یک جدول حالت نگهداری می‌کند. گروه مکالمه از یک ID نمونه (یک GUID) به عنوان کلید اصلی برای تعیین ورودی‌ها در گروه صف‌ها استفاده می‌کند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۴۱،۰۴ فعال‌سازی SQL Server Service Broker

فعال‌سازی SQL Server Service Broker ویژگی منحصر به فرد دیگری از زیرسیستم SQL Server Service Broker است. فعال‌سازی به شما امکان ایجاد رویه ذخیره شده‌ای را می‌دهد که مربوط به یک صف ورودی داده شده است. هدف رویه ذخیره شده، پردازش خودکار پیام‌ها از آن صف است. هنگامی که پیام جدیدی می‌رسد، رویه ذخیره شده مرتبط به‌طور خودکار برای مدیریت پیام‌های ورودی اجرا می‌شود. اگر رویه ذخیره شده با خطایی مواجه شود، می‌تواند استثنایی را به وجود آورده و به‌طور خودکار بازیافت شود.

به‌طور پریودیک، SQL Server Service Broker وضعیت صف ورودی را بررسی می‌کند آیا رویه ذخیره شده پیام‌های ورودی را در صف ورودی نگه می‌دارد. اگر SQL Server Service Broker تعیین کند که پیام‌های در حال انتظاری در صف وجود دارند، آن‌گاه به‌طور خودکار نمونه دیگری از خواننده صف را برای پردازش پیام‌های اضافی شروع می‌کند. این فرآیند شروع خودکار خواننده‌های صف اضافی می‌تواند تا وقتی ادامه یابد که به مقدار MAX_QUEUE_READERS از پیش تنظیم شده برسد. ضمناً، هنگامی که SQL Server Service Broker تعیین می‌کند که هیچ پیامی در صف باقی نمانده است، کاهش تعداد خواننده‌های صف فعال را به‌طور خودکار شروع می‌کند.

صف‌ها لزوماً نباید به رویه‌های ذخیره شده مرتبط شوند. پیام‌هایی که نیاز به پردازش پیچیده‌تری دارند، هم‌چنین می‌توانند به رویه‌های ردیف میانی مرتبط شوند. برای انجام فعال‌سازی خودکار فرآیندهای خارجی، SQL Server Service Broker هم‌چنین از اجرای یک رویداد SQL Server پشتیبانی می‌کند. این رویدادها می‌توانند برای استفاده از WMI[1] مشترک شوند. در این مورد، هنگامی که رویداد در صف می‌آید، رویداد SQL Server اجرا شده و توسط مشترک WMI خوانده می‌شود که به نوبت خواننده صف خارجی را شروع می‌کند.

Section ۴۱،۰۵ انتقال پیام

پروتکل انتقال پیام SQL Server Service Broker به پیام‌ها امکان ارسال روی شبکه را می‌دهد. انتقال پیام SQL Server Service Broker بر طبق TCP/IP است و معماری کلی انتقال پیام SQL Server Service Broker کمی شبیه معماری مورد استفاده TCP/IP و FTP است. انتقال پیام SQL Server Service Broker مرکب از دو پروتکل است: پروتکل Binary Adjacent Broker که یک

1- Windows Management Instrumentation

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

پروتکل سطح پایین شبیه TCP است و پروتکل Dialog که یک پروتکل سطح بالاتر شبیه FTP است و در بالای پروتکل Binary Adjacent Broker قرار دارد.

پروتکل Binary Adjacent Broker

پروتکل Binary Adjacent Broker یک پروتکل TCP/IP سطح پایین بسیار کارآمد است که انتقال پیام پایه را فراهم می‌کند. این یک پروتکل دو جهته و مولتی پلکس شده است و بنابراین می‌تواند انتقال پیام را برای چند گفتگوی SQL Server Service Broker مدیریت کند. پروتکل Binary Adjacent Broker درباره ترتیب پیام یا تأیید تحویل پیام نگرانی ندارد. همه چیز توسط پروتکل Dialog مدیریت می‌شود. در عوض، پروتکل Binary Adjacent Broker به سادگی پیام‌ها را روی شبکه و با سرعت ممکن ارسال می‌کند.

پروتکل Dialog

پروتکل Dialog ارتباطات انتها به انتها را برای یک گفتگوی SQL Server Service Broker مدیریت می‌کند. این پروتکل برای تحویل طبق ترتیب و تنها یک دفعه‌ای پیام‌ها طراحی شده است، ارسال پیام‌ها و تأیید آن‌ها را مدیریت می‌کند. هم‌چنین مدیریت خرابی متقارن را فراهم می‌کند که هر دو گره انتها از خرابی تحویل هر پیامی مطلع می‌شوند. علاوه بر این، پروتکل Dialog مسئول تعیین هویت و رمزگذاری پیام‌هاست.

Article XLII. برنامه‌نویسی با SQL Server Service Broker

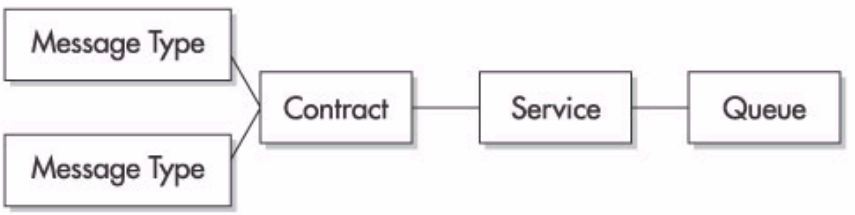
همان‌گونه که در اولین بخش این فصل بیان شد، SQL Server Service Broker زیرسیستمی است که برنامه‌نویسی برنامه‌های پیام‌رسانی مبتنی بر پایگاه داده را ممکن می‌سازد. در حالی که اولین بخش این فصل شما را با مروری از اجزای اصلی زیرسیستم SQL Server Service Broker مهیا می‌سازد و ایده‌ای از توابع و ارتباطات متقابل این اجزا را به شما می‌دهد، این بخش به بررسی عمقی‌تر می‌پردازد و شما را با یک ارایه خلاصه از نحوه نوشتن برنامه‌ها با استفاده از SQL Server Service Broker مهیا می‌کند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۴۲.۰۱ مدل برنامه‌نویسی

SQL Server Service Broker مجموعه‌ای از فوق داده را برای توصیف انواع پیام‌هایی که یک برنامه می‌تواند دریافت کند به کار می‌برد که جهت‌های ارسال و نحوه ارتباط پیام‌ها هستند. لزوماً، فوق داده SQL Server Service Broker شالوده پیام‌رسانی مورد استفاده برنامه را توصیف می‌کند. علاوه بر این، این فوق داده دارای مجوزهای مرتبطی است که می‌توانند برای ایمن ساختن برنامه پیام‌رسانی استفاده شوند. شکل ۴-۶ مدل فوق داده SQL Server Service Broker را تشریح می‌کند.



(a)

شکل ۴-۶ فوق داده SQL Server Service Broker (b)

نوع پیام[1]، اساسی‌ترین شئ در فوق داده SQL Server Service Broker است.

این نوع پیام برای بیان پیام‌هایی که به کار خواهند رفت، استفاده می‌شود. برای پیام‌های XML، انواع پیام می‌توانند یک طرح‌واره اختیاری مربوط به پیام داشته باشند. اگر یک طرح‌واره مرتبط وجود داشته باشد، SQL Server Service Broker اطمینان خواهد داد که پیام با تعریف طرح‌واره کامپایل می‌شود. اگر هیچ طرح‌واره‌ای وجود نداشته باشد، با پیام‌ها به عنوان داده باینری رفتار می‌شود.

شئ اصلی بعدی در فوق داده SQL Server Service Broker، قرارداد[2] است. انواع پیام‌ها در قراردادها گروه‌بندی می‌شوند. قرارداد تمام پیام‌هایی را توصیف می‌کند که می‌توانند با استفاده از گفتگوی خاصی دریافت شوند. مثلاً، اگر یک گفتگوی درخواست و پاسخ رزرواسیون ساده را تشکیل دهد، قرارداد مشخص خواهد کرد که این دو نوع پیام با یکدیگر گروه‌بندی شده و به سیستم رزرواسیون مرتبط می‌شوند. قرارداد لزوماً مشخص می‌کند کدام نوع پیام‌ها توسط یک گفتگوی معین مجاز است.

قراردادها برای تشکیل یک سرویس[3] با یکدیگر گروه‌بندی می‌شوند که لزوماً تمام گفتگوهایی را نشان می‌دهد که برای پردازش پیام‌ها برای آن سرویس مورد نیاز هستند. یک سرویس به یک یا چند

1- Message Type
2- Contract
3- Service

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

صف مرتبط می‌شود و شئ اصلی است که توسط برنامه SQL Server Service Broker سامان می‌یابد. برنامه یک گفتگو را برای یک سرویس باز می‌کند. برنامه با جزییات فیزیکی نحوه پیاده‌سازی سرویس کاری ندارد، زیرا جزییات پیاده‌سازی فیزیکی توسط فوق داده تعریف می‌شود. این صف حاوی پیام‌هایی است که توسط یک برنامه SQL Server Service Broker دریافت می‌شوند.

Section ۴۲،۰۲ DDL و T-SQL DML

T-SQL با چندین عبارت جدید که یکپارچگی طبیعی پیام‌رسانی SQL Server Service Broker را با رویه‌های پایگاه داده متداول ممکن می‌سازند، بهبود یافته است. جدول ۱-۶ عبارات DDL جدید T-SQL را خلاصه کرده است که برای ایجاد اشیای SQL Server Service Broker استفاده می‌شوند.

(a) جدول ۱-۶ عبارات T-SQL برای اشیای SQL Server Service Broker

T-SQL DDL	شرح
CREATE MESSAGE TYPE	نوعی پیام جدیدی ایجاد می‌کند.
CREATE CONTRACT	قرارداد جدیدی را در پایگاه داده ایجاد می‌کند.
CREATE QUEUE	صف جدیدی را در پایگاه داده ایجاد می‌کند.
CREATE ROUTE	مسیر جدیدی را در پایگاه داده ایجاد می‌کند.
CREATE SERVICE	سرویس جدیدی را در پایگاه داده ایجاد می‌کند.
ALTER MESSAGE TYPE	نوع پیام را تغییر می‌دهد.
ALTER CONTRACT	قرارداد پیام را تغییر می‌دهد.
ALTER QUEUE	صف پیام را تغییر می‌دهد.
ALTER ROUTE	مسیر پیام را تغییر می‌دهد.
ALTER SERVICE	سرویس پیام را تغییر می‌دهد.
DROP MESSAGE TYPE	نوعی پیام را از پایگاه داده حذف می‌کند.
DROP CONTRACT	قراردادی را از پایگاه داده حذف می‌کند.
DROP QUEUE	صفی را از پایگاه داده حذف می‌کند.
DROP ROUTE	مسیری را از پایگاه داده حذف می‌کند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

DROP SERVICE	سرویسی را از پایگاه داده حذف می‌کند.

علاوه بر عبارات T-SQL DDL جدید که برای ایجاد اشیای SQL Server Service Broker جدید استفاده می‌شوند. هم‌چنین گروهی از عبارات T-SQL وجود دارند که با پیام‌ها در یک برنامه SQL Server Service Broker کار می‌کنند. جدول ۲-۶ عبارات T-SQL DML مربوط به SQL Server Service Broker جدید را فهرست کرده است.

(b) جدول ۲-۶ عبارات T-SQL برای پیام‌های SQL Server Service Broker

T-SQL DML	شرح
BEGIN DIALOG	گفتگویی را باز می‌کند.
END CONVERSATION	مکالمه مورد استفاده یک گفتگو را حذف می‌کند.
MOVE CONVERSATION	مکالمه‌ای را به یک گفگوی جدید منتقل می‌کند.
GET SERVICE INSTANCE	ID یک نمونه سرویس را از یک صف بازیابی می‌کند.
RECEIVE	پیامی را از یک صف بازیابی می‌کند.
SEND	تایمری را شروع می‌کند. هنگامی که تایمر منقضی می‌شود.
BEGIN CONVERSATION TIMER	Service Broker یک پیام را در صف قرار می‌دهد.

Section ۴۲،۰۳ برنامه نمونه SQL Server Service Broker

در اولین قسمت این بخش، مرور سطح بالایی از نحوه استفاده از SQL Server Service Broker برای نوشتن برنامه‌های پیام‌رسانی غیرهم‌زمان داشتید. این بخش هم‌اکنون به تشریح عمقی‌تر نحوه استفاده از T-SQL برای ایجاد اشیای SQL Server Service Broker مورد نیاز می‌پردازد و نحوه استفاده از آن‌ها را در یک برنامه ساده نشان می‌دهد. برنامه نمونه یک سیستم رزرواسیون ساده است که یک درخواست رزرواسیون ساده را در یک صف ورودی پذیرفته و آن‌گاه به پیامی در یک صف پاسخ، جواب می‌دهد.

ایجاد اشیای SQL Server Service Broker

کدی که برای ایجاد اشیای SQL Server Service Broker مورد نیاز استفاده می‌شود، در این لیست نشان داده شده است:

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
CREATE MESSAGE TYPE ResRequest
    ENCODING varbinary
CREATE MESSAGE TYPE ResResponse
    ENCODING varbinary

CREATE CONTRACT ResContract
 (ResRequest SENT BY any,
  ResResponse SENT BY any)

CREATE QUEUE ResRequestQueue WITH ACTIVATION
 (STATUS = ON,
  PROCEDURE_NAME = ResRequestProc,
  MAX_QUEUE_READERS = 5,
  EXECUTE AS SELF)

CREATE QUEUE ResResponseQueue

CREATE SERVICE ResRequestService ON QUEUE
  ResRequestQueue(ResContract)

CREATE SERVICE ResResponseService ON QUEUE
  ResResponseQueue(ResContract)
```

اولین مرحله ایجاد برنامه SQL Server Service Broker، ایجاد انواع پیام است که پیام‌هایی را بیان می‌کند که ارسال خواهند شد. دو عبارت اول، دو نوع پیام ساده را ایجاد می‌کند. اولین پارامتر برای نام‌گذاری نوع پیام استفاده می‌شود و کلمه کلیدی ENCODING نشان می‌دهد آیا پیام باینری خواهد بود یا XML. در این مثال، هر دو نوع پیام (ResRequest و ResResponse) به عنوان پیام‌های باینری ایجاد می‌شوند، بدین معنی که آن‌ها هر نوع داده را خواهند پذیرفت.

سپس، قراردادی ایجاد می‌شود. این قرارداد، تمام پیام‌هایی را که می‌توانند با استفاده از یک گفتگوی خاص دریافت شوند، توصیف می‌کند. بخش SENT BY برای طراحی پیام‌های مربوط به قرارداد و محل آمدن این پیام‌ها استفاده می‌شود.

سپس، صف‌ها باید ایجاد شوند. این مثال ایجاد دو صف را نشان می‌دهد: ResRequestQueue و ResResponseQueue. ResRequestQueue از کلمه کلیدی ACTIVATION برای اجرای خودکار رویه ذخیره شده‌ای استفاده می‌کند که محتویات صف را خواهد خواند. این رویه ذخیره شده باید در زمان ایجاد صف وجود داشته باشد، در غیر این صورت، خطایی تولید خواهد شد. رویه ذخیره شده ResRequest در ادامه نشان داده می‌شود. کلمه کلیدی MAX_QUEUE_READERS حداکثر تعداد خوانندگانی را مشخص می‌کند که SQL Server Service Broker به‌طور خودکار فعال خواهد کرد.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

گزینه EXECUTE AS به شما اجازه می‌دهد تا رویه فعال شده‌ای را تحت یک زمینه کاربری متفاوت اجرا کنید.

پس از ایجاد صف‌ها، می‌توانید محتویات آن‌ها را با استفاده از عبارت SELECT نمایش دهید، درست مثل این که صف یک جدول پایگاه داده استاندارد باشد. این خط کد نحوه نمایش محتویات صف Request را نشان می‌دهد:

```
SELECT * FROM ResRequestQueue.
```

درست بعد از ایجاد صف‌ها، آن‌ها خالی هستند. هرچند، اجرای عبارات SELECT در صف، روش خوبی برای بررسی عملکرد برنامه‌های SQL Server Service Broker است که می‌نویسید.

بعد از ایجاد صف‌ها، مرحله بعدی ایجاد یک سرویس است. یک سرویس را با استفاده از عبارت CREATE SERVICE ایجاد کنید. اولین پارامتر نام سرویس است. بخش ON QUEUE صف مربوط به سرویس را مشخص می‌کند و سپس قراردادهایی که مربوط به سرویس هستند، لیست می‌شوند.

اگر یکی از سرویس‌ها در یک سیستم راه‌دور قرار داشته باشد، هم‌چنین باید یک مسیر ایجاد کنید. عبارت CREATE ROUTE اصولاً SQL Server Service Broker را با آدرس سیستمی مهیا می‌کند که سرویس راه‌دور پیدا شده است که نحوه تحویل پیام به سرویس راه‌دور را به SQL Server می‌گوید.

ارسال پیام‌ها به یک صف

بعد از ایجاد اشیای SQL Server Service Broker ضروری، آماده استفاده از آن‌ها در برنامه‌های صف‌بندی خود هستید. لیست کد زیر نشان می‌دهد چگونه می‌توانید پیامی را به صف ResRequestQueue اضافه کنید:

```
DECLARE @ResRequestDialog UNIQUEIDENTIFIER

BEGIN TRANSACTION

BEGIN DIALOG @ResRequestDialog
  FROM SERVICE ResResponseService
  TO SERVICE 'ResRequestService'
  ON CONTRACT ResContract
  WITH LIFETIME = 1000;

SEND ON CONVERSATION @ResRequestDialog
  MESSAGE TYPE ResRequest (N'My Request Message');

SEND ON CONVERSATION @ResRequestDialog
 MESSAGE TYPE ResRequest (N'My Request Message2');
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
COMMIT TRANSACTION
```

در ابتدای این لیست، می‌توانید ببینید که متغیری به نام ResRequestDialog ایجاد شده است؛ این متغیر حاوی شناسه منحصر به فردی است که توسط گفتگوی SQL Server Service Broker نسبت داده خواهد شد. سپس، یک تراکنش شروع می‌شود. در برگرفتن تمام اعمالی که توسط SQL Server Service Broker در یک تراکنش انجام می‌شوند، ایده خوبی است. این امر به شما امکان می‌دهد تا هر تغییراتی را برای صف‌ها commit یا roll back کنید. سپس، عبارت BEGIN DIALOG برای باز کردن یک گفتگوی SQL Server Service Broker جدید استفاده می‌شود. هنگام معرفی یک گفتگو، همیشه باید دو نقطه انتها را مشخص کنید. FROM SERVICE آغازگر پیام‌ها را تعیین می‌کند، در حالی که کلمه کلیدی TO SERVICE نقطه انتهای مقصد را تعیین می‌کند. در این‌جا، ارسال کننده ResResponseService و مقصد ResRequestService نامیده می‌شوند. در حالی که این مثال از نام‌های ساده‌ای استفاده می‌کند، مایکروسافت توصیه می‌کند که از یک نام URL برای تعیین منحصر به فرد اشیای SQL Server Service Broker استفاده کنید. مثلاً برای اطمینان از منحصر به فرد بودن در شبکه، مایکروسافت پیشنهاد می‌کند که از چنین نامی استفاده کنید:

```
[http://MyCompany.com/MyApp/ResResponseService]
```

کلمه کلیدی ON CONTRACT قراردادی را مشخص می‌کند که برای گفتگو استفاده می‌شود. این قرارداد مشخص می‌کند کدام پیام‌ها در این گفتگو قادر به ارسال یا دریافت هستند.

سپس، دو عمل SEND اجرا می‌شود. این عبارات، دو پیام را به سرویس مقصد ارسال می‌کنند که این پیام‌ها را دریافت خواهند کرد و آن‌ها را به صفی اضافه می‌کنند که مربوط به آن سرویس است. بالاخره تراکنش commit می‌شود.

بازیابی پیام‌ها از یک صف

حال که نحوه افزودن پیام به صف را دیده‌اید، مثال بعدی نحوه ایجاد رویه ذخیره شده‌ای را تشریح خواهد کرد که پیام‌ها را از صف خواهند خواند. همان‌گونه که ممکن است از مثال‌های ایجاد شئ SQL Server Service Broker قبلی به خاطر آورید، صف مقصد، ResRequestQueue، با فعال‌سازی ایجاد شد. این بدان معنی است که یک رویه ذخیره شده مرتبط به نام ResRequestProc به‌طور خودکار هنگام رسیدن پیام به صف شروع خواهد شد. می‌توانید کد این رویه ذخیره شده را در این لیست ببینید:

```
CREATE PROC ResRequestProc
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
AS
DECLARE @ResRequestDialog UNIQUEIDENTIFIER
DECLARE @message_type_id int
DECLARE @message_body NVARCHAR(1000)
DECLARE @message NVARCHAR(1000)

while(1=1)
BEGIN

  BEGIN TRANSACTION

    WAITFOR   (RECEIVE top(1)
   @message_type_id = message_type_id,
   @message_body = message_body,
   @ResRequestDialog = conversation_handle
    FROM ResRequestQueue), TIMEOUT 200;

   if (@@ROWCOUNT = 0)
   BEGIN
     COMMIT TRANSACTION
     BREAK
   END

   IF (@message_type_id = 2) -- End dialog message
      BEGIN
         PRINT ' Dialog ended '
       + cast(@ResRequestDialog as nvarchar(40))
      END
   ELSE
      BEGIN
         BEGIN TRANSACTION
            BEGIN DIALOG @ResRequestDialog
            FROM SERVICE ResRequestService
               TO SERVICE 'ResResponseService
               ON CONTRACT ResContract
               WITH LIFETIME = 1000;

            SELECT @message = 'Received:' + @message_body;

             SEND ON CONVERSATION @ResRequestDialog
             MESSAGE TYPE ResResponse (@message);

             PRINT CONVERT(varchar(30), @message)
         COMMIT TRANSACTION
      END CONVERSATION @ResRequestDialog
  END
  COMMIT TRANSACTION
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

END

متغیری که حاوی تعیین هویت گفتگوی پاسخ خواهد بود، در بالای این رویه ذخیره شده معرفی می‌شود و بعد از آن سه متغیر می‌آید که برای اقتباس اطلاعات از صفی که خوانده می‌شود، استفاده خواهند شد. سپس حلقه‌ای برای خواندن تمام ورودی‌ها از صف، مقداردهی اولیه می‌شود. در این حلقه، یک تراکنش شروع شده و عبارت RECEIVE برای دریافت پیام استفاده می‌شود. در این مثال، بخش TOP(1) برای محدود کردن دریافت تنها یک پیام در یک زمان استفاده می‌شود. اگر بخش TOP حذف شود، می‌توانید تمام پیام‌هایی را دریافت کنید که در صف ارایه شده‌اند. عبارت RECEIVE سه متغیر را پر می‌کند. message_type_id نوع پیام را تعیین می‌کند که معمولاً یک پیام تعریف شده کاربر یا یک پیام End Dialog است. متغیر @message_body حاوی محتویات پیام واقعی است، در حالی که متغیر @ResRequestDialog حاوی هندلی است که گفتگوی ارسال شونده را تعیین می‌کند.

سپس، مجموعه نتیجه بررسی می‌شود تا مطمئن شویم که پیام واقعاً دریافت شده است. اگر هیچ ردیفی دریافت نشده باشد، آن‌گاه آخرین تراکنش commit شده و رویه خاتمه می‌یابد. در غیر این صورت، چنان‌چه ردیف‌ها بازیابی شده باشند، پیام typeid بررسی می‌شود تا مشخص شود آیا پیام یک پیام کاربر است یا یک پیام End Dialog. اگر یک پیام کاربر باشد، محتویات پردازش خواهند شد. ابتدا، گفتگویی برای ResResponseService باز می‌شود. این گفتگو برای ارسال پیام اصلاح شده به صف ResResponse استفاده خواهد شد. مجدداً، ResContract مشخص می‌شود که انواع پیام مجاز را محدود می‌کند.

بعد از باز شدن گفتگو، پیام دریافت شده با الحاق رشته "Received:" با محتویات پیامی که دریافت شده است اصلاح شده و سپس عبارت SEND برای ارسال پیام اصلاح شده به ResRequestQueue استفاده می‌شود. بالاخره، مکالمه گفتگو خاتمه یافته و تراکنش commit می‌شود.

Article XLIII. راهبری Service Broker

SQL Server Service Broker تعدادی گزینه پیکربندی را فراهم کرده است که نحوه عملکرد زیرسیستم SQL Server Service Broker را کنترل می‌کند. در این بخش، نگاهی کوتاه به برخی از مهم‌ترین گزینه‌های پیکربندی خواهیم داشت. مطالبی درباره امنیت SQL Server Service Broker می‌آموزید و نگاهی به برخی از دیدگاه‌های سیستم خواهید داشت که SQL Server 2005 فراهم کرده است تا به شما امکان گرفتن اطلاعاتی درباره برنامه‌ها و اشیای SQL Server Service Broker را خواهد داد.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۴۳.۰۱ گزینه‌های پیکربندی سیستم

چندین گزینه پیکربندی سیستم وجود دارند که می‌توانند با استفاده از رویه ذخیره شده سیستمی sp_configure برای تأثیر بر رفتار زیرسیستم SQL Server Service Broker تنظیم شوند. جدول ۳-۶ گزینه‌های sp_configure موجود را برای SQL Server Service Broker فهرست کرده است.

(a) جدول ۳-۶ گزینه‌هایی برای sp_configure

شرح	پارامتر sp_configure
پورت مستمع broker tcp	پورتی که SQL Server Service Broker برای اتصالات شبکه استفاده می‌کند. مقدار پیش‌فرض ۴۰۲۲ است.
حالت تعیین هویت broker tcp	نوعی از تعیین هویت راه‌دور را تنظیم می‌کند که برای اتصالات استفاده خواهند شد. مقدار ۱ به معنی عدم استفاده از تعیین هویت است. مقدار ۲ بدین معنی است که از تعیین هویت پشتیبانی می‌شود. مقدار ۳ به معنی نیاز به تعیین هویت است. مقدار پیش‌فرض ۳ است.
حد اندازه رو به جلوی broker	حداکثر فضای دیسک را برحسب MB برای ذخیره پیام‌های ارسال شونده تنظیم می‌کند. مقدار پیش‌فرض ۱۰ است.
پیام broker ارسال شونده	نوع پیام ارسال شونده مجاز را تنظیم می‌کند. مقدار ۱ به معنی عدم مجاز بودن ارسال است. مقدار ۲ ارسال در حوزه را میسر می‌سازد. مقدار ۳ ارسال خارجی را میسر می‌سازد. مقدار پیش‌فرض ۱ است.

Section ۴۳.۰۲ امنیت گفتگو

هنگامی که گفتگوها ایجاد می‌شوند، آن‌ها می‌توانند به‌طور اختیاری با استفاده از بخش WITH ENCRYPTION ایمن شوند. هنگامی که گفتگویی با استفاده از بخش WITH ENCRYPTION

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

ایجاد می‌شود، یک کلید جلسه ایجاد می‌گردد که برای رمزگذاری پیام‌های ارسال شده با استفاده از گفتگو به کار می‌رود. یک نکته مهم درباره امنیت گفتگو این واقعیت است که این یک شکل امنیتی انتها به انتهاست. به عبارت دیگر، پیام هنگامی رمزگذاری می‌شود که برای اولین بار از یک گفتگو ارسال شده باشد و تا وقتی که پیام به نقطه انتهایی خود نرسد، رمزگشایی نمی‌شود. محتویات پیام هنگامی رمزگذاری می‌شود که پیام در بین هر جهش میانی ارسال شود. برای پیاده‌سازی امنیت گفتگو، SQL Server Service Broker از تعیین هویت مبتنی بر گواهینامه استفاده می‌کند که مدرک کاربر ارسال کننده به همراه پیام ارسال می‌شود. به دلیل طبیعت غیرهمزمان SQL Server Service Broker، اطلاعات امنیتی در هدرهای پیام ذخیره شده و هنگام بازیابی پیام، توسط سرویس دریافت کننده بازیابی می‌شوند. این طراحی به برنامه‌های SQL Server Service Broker امکان می‌دهد تا از نیاز به تولید اتصال برای پیام‌های تعیین هویت دوری کنند.

Section ۴۳،۰۳ دیدگاه سیستمی

SQL Server 2005 چندین دیدگاه سیستمی را مهیا کرده است که به شما امکان بازیابی اطلاعات درباره اشیای SQL Server Service Broker و وضعیت‌های جاری آن‌ها را می‌دهد. جدول ۴-۶ دیدگاه‌های کاتالوگ Service Broker را فهرست می‌کند.

(a) جدول ۴-۶ دیدگاه‌های کاتالوگ Service Broker جدید

دیدگاه سیستمی	شرح
sys.service_message_types	تمام انواع پیام‌هایی را که ایجاد شده‌اند، فهرست می‌کند. انواع پیام‌های سیستم در بالا لیست می‌شوند، در حالی که انواع پیام‌های تعریف شده کاربر در انتهای نمایش فهرست می‌شود.
sys.service_contracts	تمام قراردادهایی را فهرست می‌کند که ایجاد شده‌اند.
sys.service_contract_message_usages	روابط بین قراردادها و انواع پیام را فهرست می‌کند. روابط می‌توانند یک به یک یا یک به چند باشند.
sys.services	سرویس‌های ایجاد شده را فهرست می‌کند.
sys.service_contract_usages	روابط بین قراردادها و سرویس‌ها را فهرست می‌کند. روابط می‌توانند یک به یک یا یک به چند باشند.
sys.service_instances	سرویس‌هایی را فهرست می‌کند که در زمان جاری فعال

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

	هستند.
sys.conversation_endpoints	نقاط انتهایی مکالمه را که هم اینک فعال هستند، فهرست می‌کند.
sys.routes	مسیرهای ایجاد شده را فهرست می‌کند.
sys.remote_service_bindings	روابط سرویس‌ها و کاربرانی را فهرست می‌کند که سرویس را اجرا خواهند کرد.
sys.transmission_queue	تمام پیام‌هایی را که برای ارسال صف‌بندی شده‌اند، فهرست می‌کند.
sys.service_queues	صف‌های ایجاد شده را فهرست می‌کند.

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

فصل هفتم

یکپارچگی XML

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

XML[1] به عنوان یکی از مهم‌ترین فناوری‌های اینترنت در حال گسترش است. ساختار مبتنی بر متن انعطاف‌پذیر XML به آن امکان می‌دهد تا برای آرایه گسترده‌ای از وظایف شبکه‌ای استفاده شود، از جمله انتقال داده/ سند، نمایش صفحه وب و حتی به عنوان انتقالی برای فراخوانی‌های رویه راه‌دور (RPC) درون برنامه از طریق SOAP[2]. XML واقعاً زبان میانجی زبان‌های کامپیوتر شده است.

مایکروسافت ابتدا پشتیبانی XML را به SQL Server 2000 اضافه کرد و این پشتیبانی با بخش FOR XML به عنوان بخشی از عبارت SELECT و تابع OpenXML شروع شد. هنگامی که XML به رشد سریع خود در پذیرش و کاربرد جهانی ادامه داد، مایکروسافت سریعاً عملکردی اضافی را با تولید مجموعه‌ای از نسخه‌های وب فراهم کرد. SQL for XML 1.0 پشتیبانی از UpdateGrams، Templates و BulkLoad را به نسخه پایه SQL Server 2000 اضافه کرد. دو نسخه وب بعدی، SQLXML 2.0 و SQLXML 3.0، باعث بهبود بیشتر SQL Server 2000 با افزودن پشتیبانی از دیدگاه‌های XML و SOAP به علاوه چندین قابلیت جدید دیگر شدند. در حالی که پشتیبانی SQL Server 2000 از XML، نقطه شروع خوبی را برای یکپارچه کردن اسناد XML سلسله مراتبی با داده رابطه‌ای SQL Server فراهم کرد، محدودیت‌هایی نیز داشت، هنگامی که داده XML در یک پایگاه داده SQL Server با استفاده از نوع داده Text یا Image ذخیره می‌شد، بسیار کوچک بود که بتوانید با آن کار کنید. SQL Server 2000 قادر به پرس‌وجوی طبیعی داده سلسله مراتبی که سند XML را تشکیل می‌دهد، بدون استفاده از کد سمت کلاینت یا T-SQL پیچیده نبود.

SQL Server 2005 بر طبق این نقطه شروع با افزودن پشتیبانی از برخی از ویژگی‌های XML جدید ساخته شده است. SQL Server 2005 در یک سطح بالا، سطح جدیدی از حافظه یک شکل را برای XML و داده رابطه‌ای فراهم کرده است. SQL Server 2005 یک نوع داده XML جدید را اضافه کرده است که پشتیبانی از پرس‌وجوهای XML طبیعی و نوع‌گذاری داده قوی را با مرتبط کردن نوع داده XML به یک XSD[3] فراهم می‌کند. علاوه بر این، نگاشت دو جهته را بین داده رابطه‌ای و داده XML فراهم کرده است. پشتیبانی XML به خوبی در موتور پایگاه داده رابطه‌ای SQL Server 2005 یکپارچه شده است، زیرا پشتیبانی از تریگرها را در XML، کپی‌سازی داده XML، بارگذاری حجیم داده XML و پشتیبانی بهبود یافته برای دستیابی داده از طریق SOAP و تعدادی بهبود دیگر را فراهم کرده است. در این فصل، با مهم‌ترین ویژگی‌های XML جدید فراهم شده توسط SQL Server 2005 آشنا خواهید شد.

1- eXtensible Markup Language
2- Simple Object Access Protocol
3- Extensible Schema Definition

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Article XLIV. نوع داده XML طبیعی

بدون شک، مهم‌ترین بهبود مربوط به XML که مایکروسافت به SQL Server 2005 اضافه کرده است، پشتیبانی از نوع داده XML طبیعی است. نوع داده XML که عیناً XML نامیده می‌شود، می‌تواند به عنوان ستونی در یک جدول یا یک متغیر یا پارامتر در یک رویه ذخیره شده استفاده شود. این نوع می‌تواند برای ذخیره داده نوع‌دار و بدون نوع استفاده شود. اگر داده ذخیره شده در یک ستون XML دارای طرح‌واره XSD نباشد، آن‌گاه بدون نوع در نظر گرفته می‌شود. اگر یک طرح‌واره XSD مرتبط وجود داشته باشد، آن‌گاه SQL Server 2005 طرح‌واره را بررسی می‌کند تا مطمئن شود که حافظه داده با تعریف طرح‌واره کامپایل می‌شود. در تمام موارد، SQL Server 2005 داده‌ای را که در نوع داده XML ذخیره می‌شود، بررسی می‌کند تا مطمئن شود سند XML خوش فرم است. اگر داده خوش فرم نباشد، SQL Server 2005 خطایی را به وجود خواهد آورد و داده ذخیره نخواهد شد. نوع داده XML می‌تواند حداکثر 2GB داده را بپذیرد و شبیه نوع داده varbinary(max) ذخیره می‌شود. لیست زیر ایجاد جدولی ساده را نشان می‌دهد که از نوع داده XML جدید برای یکی از ستون‌های خود استفاده می‌کند.

```
CREATE TABLE MyXMLDocs
  (DocID INT PRIMARY KEY IDENTITY,
   MyXmlDoc XML)
```

مهم‌ترین نکته‌ای که در این مثال توجه برانگیز است، تعریف ستون MyXmlDoc است که از نوع داده XML برای مشخص کردن این مسأله استفاده می‌کند که این ستون، داده XML را ذخیره خواهد کرد. می‌توانید داده XML را با استفاده از عبارت T-SQL INSERT استاندارد در یک ستون XML ذخیره کنید. این مثال نحوه پر کردن یک ستون XML را با استفاده از یک عبارت INSERT ساده نشان می‌دهد:

```
INSERT INTO MyXmlDocs Values
('<MyXMLDoc>
   <DocumentID>1</DocumentID>
   <DocumentText>Text</DocumentText>
</MyXMLDoc>')
```

توجه: نکته مهمی که در این‌جا مطرح می‌شود این است که به این دلیل که داده XML بدون نوع است، هر سند XML معتبری می‌تواند در نوع داده XML درج شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۴۴،۰۱ انواع داده XML نوع‌دار قوی

نوع داده XML طبیعی بررسی می‌کند تا مطمئن شود که هر داده‌ای که در یک ستون یا متغیر XML ذخیره می‌شود، یک سند XML معتبر است. هرچند، مایکروسافت نوع داده XML را طوری طراحی کرده است تا بتواند با استفاده از یک طرح‌واره XSD، از تأیید اعتبار سند بهتری پشتیبانی کند. هنگامی که یک طرح‌واره XSD برای یک ستون نوع داده XML تعریف می‌شود، موتور SQL Server بررسی می‌کند تا مطمئن شود که تمام داده‌ای که در ستون XML ذخیره می‌شود، با تعریف آن در طرح‌واره XSD مطابقت دارد.

این لیست یک طرح‌واره XSD ساده را برای سند XML ساده‌ای نشان می‌دهد که در مثال قبل استفاده شد:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" targetNamespace="MyXMLDocSchema"
xmlns="MyXMLDocSchema">
  <xs:element name="MyXMLDoc">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="DocumentID" type="xs:string" />
        <xs:element name="DocumentBody" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

این طرح‌واره داده XSD از فضای MyXMLDocSchema استفاده کرده و یک سند XML را تعریف می‌کند که دارای عنصر مرکبی به نام MyXMLDoc است. عنصر مرکب MyXMLDoc حاوی دو عنصر ساده است. اولین عنصر ساده باید DocumentID و دومی DocumentBody نامیده شوند. هر دو عنصر حاوی داده نوع رشته‌ای XML هستند.

برای ایجاد یک متغیر یا ستون XML نوع‌دار قوی، ابتدا باید طرح‌واره XSD را با SQL Server با استفاده از عبارت CREATE XMLSCHEMA T-SQL DDL رجیستر کنید. لیست زیر نحوه ترکیب کردن عبارت CREATE XML SCHEMA COLLECTION را با MyXMLDocSCHEMA برای رجیستر کردن طرح‌واره با پایگاه داده SQL Server 2005 نشان می‌دهد:

```
CREATE XML SCHEMA COLLECTION MyXMLDocSchema AS
N'<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
elementFormDefault="qualified" targetNamespace="http://MyXMLDocSchema">
 <xs:element name="MyXMLDoc">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="DocumentID" type="xs:string" />
    <xs:element name="DocumentBody" type="xs:string" />
   </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>'
```

عبارت CREATE XML SCHEMA COLLECTION DLL یک آرگومان می‌گیرد که نام کلکسیون است. سپس، بعد از بخش AS، یک طرح‌واره XSD معتبر مورد نیاز است که در گیومه قرار گرفته باشد. اگر طرح‌واره معتبر نباشد، هنگام اجرای عبارت، خطایی صادر خواهد شد. عبارت CREATE XML SCHEMA COLLECTION خاص پایگاه داده است و طرح واره رجیستر شده تنها می‌تواند در پایگاه داده برای آن طرح‌واره رجیستر شده، مورد دستیابی قرار گیرد.

هنگامی که طرح‌واره XML را در SQL Server 2005 رجیستر کردید، می‌توانید ستون‌ها و متغیرهای XML را به آن طرح‌واره مرتبط کنید. انجام این کار اطمینان می‌دهد که هر سند XML ای که در این ستون‌ها یا متغیرها وجود دارد، به تعریف فراهم شده توسط طرح‌واره مربوطه مرتبط می‌شود. این مثال به شما نشان می‌دهد چگونه می‌توانید جدولی را ایجاد کنید که از یک ستون XML نوع‌دار قوی استفاده کند:

```
CREATE TABLE MyXMLDocs
 (DocID INT PRIMARY KEY IDENTITY,
  MyXmlDoc XML(MyXMLDocSchema))
```

در این‌جا می‌توانید ببینید که جدول MyXMLDocs با استفاده از عبارت CREATE TABLE همانند مثال قبل ایجاد شده است. هرچند، در این مورد، ستون MyXMLDoc با استفاده از آرگومانی ایجاد می‌شود که نام تعریف طرح‌واره XSD رجیستر شده را مشخص می‌کند. اگر به لیست قبل مراجعه کنید، می‌توانید ببینید که طرح‌واره با استفاده از نام MyXMLDocSchema رجیستر شده است. بعد از مرتبط شدن ستون MyXMLDoc با طرح‌واره رجیستر شده، هر داده‌ای که در این ستون درج شود، برطبق تعریف طرح‌واره نوع‌دار قوی خواهد شد و هر تلاشی برای درج آن داده که با تعریف طرح‌واره تطابق نداشته باشد، پذیرفته نخواهد شد. این لیست یک عبارت INSERT را تشریح می‌کند که می‌تواند داده را به ستون MyXMLDoc نوع‌دار قوی اضافه کند:

```
INSERT INTO MyXMLDocs Values
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
('<MyXMLDoc xmlns="http://MyXMLDocSchema">
    <DocumentID>1</DocumentID>
    <DocumentBody>"My text"</DocumentBody>
    </MyXMLDoc>')
```

توجه : به دلیل این که این مثال از یک نوع داده XML نوع‌دار استفاده می‌کند، این داده باید با تعریف فراهم شده توسط طرح واره XSD مرتبط تطابق داشته باشد.

در این مورد، سند XML باید به فضای نام XML مرتبط http://MyXMLDocSchema مراجعه کند و باید حاوی عنصر مرکبی به نام MyXMLDoc باشد که به نوبت حاوی عناصر DocumentID و DocumentBody است. موتور SQL Server هر تلاشی برای درج هر سند XML دیگر را در ستون MyXMLDoc نمی‌پذیرد. اگر داده‌ای با طرح‌واره XSD تأمین شده تطابق نداشته باشد، MyXMLDoc پیام خطایی را برمی‌گرداند، شبیه آن‌چه که در این‌جا نشان داده شده است:

```
Msg 6965, Level 16, State 1, Line 1
XML Validation: Invalid content,expected
element(s):MyXMLDocSchema:DocumentID where element 'MyXMLDocSchema:Do'
was
specified
```

توجه : همان‌گونه که ممکن است از رابطه وابسته آن‌ها انتظار داشته باشید، اگر طرح‌واره‌ای را به ستونی در جدول نسبت دهید، آن جدول باید قبل از بهنگام‌رسانی تعریف طرح واره، تغییر کند یا حذف شود.

بازیابی یک طرح‌واره XML رجیستر شده

هنگامی که طرح‌واره‌ای را با استفاده از CREATE XML SCHEMA COLLECTION وارد می‌کنید، اجزای طرح‌واره در فوق داده SQL Server ذخیره می‌شوند. طرح‌واره ذخیره شده می‌تواند با پرس‌وجوی دیدگاه سیستمی Sys.xml_namespaces فهرست شود، همان‌گونه که می‌توانید در این مثال ببینید:

```
SELECT * FROM sys.xml_namespaces
```

این عبارت مجموعه نتیجه‌ای را برخواهد گرداند که تمام طرح‌واره‌های رجیستر شده را در پایگاه داده‌ای نظیر این نشان می‌دهد:

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
xml_collection_id name          xml_namespace_id
------------------------------  ---------------------------------  --------
1                               http://www.w3.org/2001/XMLSchema   1
65540                           http://MyXMLDocSchema              1

(2 row(s) affected)
```

هم‌چنین می‌توانید از تابع XML_SCHEMA_NAMESPACE جدید برای بازیابی طرح‌واره XML استفاده کنید. این پرس‌وجو، طرح‌واره‌ای را از پایگاه داده برای یک فضای نام معین بازیابی می‌کند.

```
SELECT XML_SCHEMA_NAMESPACE(N'dbo',N'MyXMLDocSchema')
```

این عبارت مجموعه نتیجه‌ای را برخواهد گرداند که تمام ستون‌هایی را نشان می‌دهد که از طرح‌واره رجیستر شده استفاده می‌کند، همان‌گونه که می‌توانید در این لیست ببینید:

```
-----------------------------------------------------------------------
<xsd:schema xmlns:xsd=http://www.w3.org/2001/XMLSchema
targetNamespace="http://MyXMLDocSchema" xmlns:t=http://MyXMLDocSchema
elementFormDefault="qualified"><xsd:elementname="MyXMLDoc">
<xsd:complexType><xsd:complexContent><xsd:restriction base="xsd:any

(1 row(s) affected)
```

صفت elementname ستون‌هایی را فهرست می‌کند که از نوع داده XML نوع‌دار استفاده می‌کنند.

Section ۴۴،۰۲ متدهای نوع داده XML

SQL Server 2005 چندین متد تعبیه شده جدید را فراهم کرده است که بسیار شبیه انواع تعریف شده کاربر برای کار کردن با نوع داده XML کار می‌کند. این متدها به شما امکان می‌دهند تا با استفاده از نوع داده XML به‌طور عمقی به محتویات XML ذخیره شده وارد شوید. برای تسهیل کردن یکپارچگی عمقی، نیاز به روشی برای پرس‌وجو و دستکاری داده ذخیره شده با استفاده از نوع داده XML جدید دارید و این که چه چیزی متدهای نوع داده XML را فعال می‌کند.

Exists (XQuery, [node ref]

متد Exists نوع داده XML به شما امکان می‌دهد تا محتویات یک سند XML را برای وجود عناصر یا صفات استفاده کننده از یک عبارت XQuery بررسی کنید (مطالب بیشتر درباره زبان

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

XQuery در بخش بعد ارایه می‌شود). این لیست نحوه استفاده از متد Exists نوع داده XML را نشان می‌دهد:

```
SELECT * FROM MyXMLDocs
WHERE MyXmlDoc.exist('declare namespace xd=http://MyXMLDocSchema
/xd:MyXMLDoc[xd:DocumentID eq "1"]') = 1
```

اولین پارامتر متد XML Exists ضروری است و یک عبارت XQuery را می‌گیرد. دومین پارامتر اختیاری است و مرجع گرهی را در سند XML مشخص می‌کند. در این‌جا، XQuery تساوی عنصر DocumentID را با ۱ بررسی می‌کند. یک فضای نام معرفی شده است، زیرا ستون MyXMLDoc دارای یک طرح‌واره مرتبط است. متد Exists در صورتی می‌تواند مقدار TRUE(1) را برگرداند که عبارات XQuery یک گره را برگردانند و چنان‌چه عبارت یک گره XML را برنگرداند، FALSE(0) را برمی‌گرداند و در صورت null بودن نمونه نوع داده XML، NULL را برمی‌گرداند. می‌توانید نتایج متد XML Exists را در این‌جا ببینید:

```
DocID       MyXmlDoc
----------- -------------------------------------------------------
1           <MyXMLDocxmlns="http://MyXMLDocSchema"><DocumentID>1
</DocumentID> <DocumentBody>"My text"</DocumentBody></MyXMLDoc>

(1 row(s) affected)
```

Modify (XML DML)

همان‌گونه که ممکن است حدس زده باشید، متد Modify نوع داده XML به شما امکان می‌دهد تا یک سند XML ذخیره شده را اصلاح کنید. می‌توانید از متد Modify برای بهنگام‌رسانی کل سند XML یا بخش منتخبی از آن استفاده کنید. می‌توانید مثالی از کاربرد متد Modify را در این لیست ببینید:

```
UPDATE MyXMLDocs
SET MyXMLDoc.modify('declare namespace xd=http://MyXMLDocSchema
  replace value of  (/xd:MyXMLDoc/xd:DocumentBody)[1] with "My New Body"')
WHERE DocID = 1
```

متد Modify نوع داده XML از یک عبارت XML DML به عنوان پارامتر خود استفاده می‌کند. XML DML یک توسعه مایکروسافتی برای زبان XQuery است که اصلاح اسناد XML را ممکن می‌سازد. زبان XQuery از Replace value of عبارات DELETE XML DML و INSERT پشتیبانی می‌کند. در این مثال، با توجه به این که ستون MyXMLDoc XML نوع دار است، عبارت XML DML باید فضای نامی را برای طرح‌واره مشخص کند. سپس، می‌توانید محل فرمان

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Replace value of XML DML مورد استفاده برای جایگزین کردن مقدار عنصر DocumentBody با مقدار جدید "My New Body" برای ردیفی که ستون DocID رابطه‌ای آن با یک برابر است، ببینید. بخش Replace value of باید تنها یک گره را مشخص کند یا ناموفق خواهد بود. بنابراین، اولین گره با استفاده از نماد [1] مشخص می‌شود.

توجه: در حالی که این مثال کارایی عمل جایگزینی را تشریح می‌کند، متد Modify نیز از عملیات حذف و اضافه پشتیبانی می‌کند.

Query (XQuery, [node ref])

متد Query نوع داده XML می‌تواند کل محتویات یک سند XML یا بخش منتخبی از سند XML را بازیابی کند. می‌توانید مثالی از کاربرد متد Query را در این لیست ببینید:

```
SELECT DocID, MyXMLDoc.query('declare namespace xd=http://MyXMLDocSchema
  /xd:MyXMLDoc/xd:DocumentBody') AS Body
FROM MyXMLDocs
```

این عبارت XQuery، مقداری را از عنصر DocumentBody سند XML برمی‌گرداند. مجدداً، فضای نام مشخص شده است، زیرا نوع داده MyXMLDoc دارای یک طرح‌واره مربوطه به نام MyXMLDocSchema است. در این مثال، می‌توانید ببینید که چگونه SQL Server 2005 داده ستون رابطه‌ای را با داده XML یکپارچه کرده است. در اینجا، DocID از یک ستون رابطه‌ای می‌آید، در حالی که عنصر DocumentBody خارج از ستون XML پرس‌وجو می‌شود. این لیست نتایج XQuery را نشان می‌دهد:

```
DocID     Body
------    ---------------------------------------------------------------------------
1         <xd:DocumentBody xmlns:xd="http://MyXMLDocSchema">
                                My New Body</xd:DocumentBody>
2         <xd:DocumentBody xmlns:xd="http://MyXMLDocSchema">
                                "My 2nd text"</xd:DocumentBody>

(2 row(s) affected)
```

Value (XQuery, [node ref])

متد Value اقتباس مقادیر عددی از یک نوع داده XML را ممکن می‌سازد. می‌توانید مثالی از نحوه استفاده از متد Value نوع داده XML را در این لیست ببینید:

```
SELECT MyXMLDoc.value('declare namespace xd=http://MyXMLDocSchema
   (/xd:MyXMLDoc/xd:DocumentID)[1]', 'int') AS ID
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
FROM MyXMLDocs
```

برخلاف متدهای دیگر نوع داده XML، متد XML Value نیاز به دو پارامتر دارد. اولین پارامتر یک عبارت XQuery است و دومین پارامتر، نوع داده SQL را مشخص می‌کند که مقدار عددی برگردانده شونده از متد Value است. این مثال تمام مقادیر موجود در عنصر DocumentID را برمی‌گرداند و آن‌ها را به نوع داده int تبدیل می‌کند، همان‌گونه که در این نتایج نشان داده شده است:

```
ID
-----------
1
2

(2 row(s) affected)
```

Article XLV. پشتیبانی XQuery

در بخش قبل، نحوه استفاده از XQuery در متدهای نوع داده XML جدید بیان شد. XQuery برطبق زبان XPath است که توسط W3C (www.w3c.org) برای پرس‌وجوی داده XML ایجاد شده است. XQuery زبان XPath را با افزودن قابلیت بهنگام‌رسانی داده و پشتیبانی برای تکرار بهتر و مرتب‌سازی نتایج توسعه داده است. در زمان نوشتن این کتاب، زبان XQuery که توسط SQL Server 2005 استفاده می‌شود، یک پیاده‌سازی اولیه برطبق کاری ماهرانه از استاندارد XQuery است که به W3C واگذار شده است، بنابراین تغییر مقداری از جزییات پیاده‌سازی قبل از عرضه رسمی SQL Server 2005 امکان‌پذیر است. شرح زبان XQuery خارج از حیطه این کتاب است، ولی برای جزییات بیشتر درباره استاندارد W3C XQuery، می‌توانید به http://www.w3.org/XML/Query مراجعه کنید.

Article XLVI. ایندکس‌های XML

نوع داده XML از حداکثر 2GB پشتیبانی می‌کند که بسیار بزرگ است. اندازه داده XML و کاربرد آن می‌تواند تأثیر بزرگی بر کارآیی حاصله سیستم داشته باشد، در حالی که داده XML را پرس‌وجو می‌کنید. برای بهبود کارآیی پرس‌وجوهای XML، SQL Server 2005 توانایی ایجاد ایندکس‌ها روی ستون‌هایی را که دارای نوع داده XML هستند، فراهم کرده است.

Section ۴۶.۰۱ ایندکس‌های XML اصلی

برای ایجاد یک ایندکس XML روی یک ستون نوع داده XML، یک کلید اصلی کلاستر شده باید برای جدول وجود داشته باشد. علاوه بر این، اگر نیاز به تغییر کلید اصلی جدول داشته باشید، ابتدا

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

باید ایندکس XML را حذف کنید. یک ایندکس XML تمام عناصر ستون XML را دربرمی‌گیرد و می‌توانید تنها یک ایندکس XML در هر ستون داشته باشید. به دلیل این که ایندکس‌های XML از همان فضای نام ایندکس‌های رابطه‌ای عادی SQL Server استفاده می‌کنند، ایندکس‌های XML نمی‌توانند دارای نامی مشابه یک ایندکس موجود باشند. ایندکس‌های XML می‌توانند تنها روی انواع داده XML در یک جدول ایجاد شوند. آن‌ها نمی‌توانند روی ستون‌هایی در دیدگاه‌ها یا روی متغیرهای نوع داده XML ایجاد شوند. یک XML اصلی شامل یک نمایش قطعه قطعه دایمی از داده در ستون XML است. کدی برای ایجاد یک ایندکس XML اصلی در این لیست نشان داده شده است:

```
CREATE PRIMARY XML INDEX MyXMLDocsIdx ON MyXMLDocs(MyXMLDoc)
```

این مثال، ایجاد ایندکس XML اصلی به نام MyXMLDocIdx را نشان می‌دهد. این ایندکس روی ستون نوع داده XML MyXMLDoc در جدول MyXMLDocs ایجاد می‌شود. درست شبیه ایندکس‌های عادی SQL Server، ایندکس‌های XML می‌توانند با پرس‌وجوی دیدگاه sys.indexes مشاهده شوند.

```
SELECT * FROM sys.indexes WHERE name = 'MyXMLDocsIdx'
```

Section ۴۶،۰۲ ایندکس‌های XML ثانویه

علاوه بر ایندکس اصلی، هم‌چنین می‌توانید ایندکس‌های XML ثانویه را بسازید. ایندکس‌های ثانویه روی یکی از این صفات سند ساخته می‌شوند:

- Path: مسیر سند که برای ساخت ایندکس استفاده می‌شود.
- Value: مقادیر سند که برای ساخت ایندکس استفاده می‌شود.
- Property: خصوصیات اسناد برای ساخت ایندکس استفاده می‌شوند.

ایندکس‌های ثانویه همیشه به همان روش ایندکس XML اصلی پارتیشن‌بندی می‌شوند. این لیست، ایجاد یک ایندکس XML مسیر ثانویه را نشان می‌دهد:

```
CREATE XML INDEX My2ndXMLDocsIdx ON MyXMLDocs(MyXMLDoc)
 USING XML INDEX MyXMLDocsIdx FOR PATH
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Article XLVII. بهبودهای FOR XML

بخش FOR XML ابتدا برای عبارت T-SQL SELECT در SQL Server 2000 معرفی شد. این بخش در SQL Server 2005 بهبود یافته است. برخی از قابلیت‌های جدیدی که در پشتیبانی FOR XML در SQL Server 2005 پیدا می‌شود، شامل پشتیبانی برای نوع داده XML، انواع داده تعریف شده کاربر، نوع داده timestamp و پشتیبانی بهبود یافته برای داده رشته‌ای است. علاوه بر این، بهبودهای FOR XML هم‌چنین شامل پشتیبانی برای یک رهنمود Type جدید، پرس‌وجوهای FOR XML تودرتو و تولید طرح واره XSD درون برنامه‌ای است.

Section ۴۷.۰۱ رهنمود Type

هنگامی که انواع داده XML با استفاده از رهنمود Type بخش‌های FOR XML برگردانده می‌شوند، سریالی نمی‌شوند. در عوض نتایج به عنوان نوع داده XML برگردانده می‌شود. می‌توانید مثالی از کاربرد بخش FOR XML را با رهنمود XML Type ببینید:

```
SELECT DocID, MyXMLDoc FROM MyXMLDocs
  WHERE DocID=1 FOR XML AUTO, TYPE
```

این پرس‌وجو، ستون DocID رابطه‌ای را به همراه ستون نوع داده XML MyXMLDoc برمی‌گرداند. این پرس‌وجو از بخش FOR XML AUTO برای برگرداندن نتایج به عنوان XML استفاده می‌کند. رهنمود Type مشخص می‌کند که نتایج به عنوان یک نوع داده XML برگردانده خواهند شد. می‌توانید نتایج استفاده از رهنمود Type را در اینجا ببینید:

```
<MyXMLDocs DocID="1">
  <MyXMLDoc>
    <MyXMLDoc xmlns="MyXMLDocSchema">
      <DocumentID>1</DocumentID>
      <DocumentBody>My New Body</DocumentBody>
    </MyXMLDoc>
  </MyXMLDoc>
</MyXMLDocs>
```

توجه: رهنمود Type نوع داده XML را به عنوان یک جریان متوالی برمی‌گرداند. من فرمت‌بندی را به لیست قبل اضافه کرده‌ام تا خواناتر شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۴۷.۰۲ پرس‌وجوهای FOR XML تودرتو

SQL Server 2000 محدود به استفاده از بخش FOR XML در سطح بالای یک پرس‌وجو بود زیرا پرس‌وجوها نمی‌توانستند از بخش FOR XML استفاده کنند. SQL Server 2005 توانایی استفاده از پرس‌وجوهای FOR XML تودرتو را اضافه کرده است. پرس‌وجوهای تودرتو برای برگرداندن چندین آیتم مفید هستند که یک رابطه والد-فرزند بین آن‌ها وجود دارد. مثالی از این نوع رابطه ممکن است رکوردهای هدر سفارش و جزییات سفارش باشند؛ مثالی دیگر ممکن است دسته‌ها و زیردسته‌های محصول باشند. می‌توانید مثالی از کاربرد یک بخش FOR XML تودرتو را در این لیست ببینید:

```
SELECT DocID, MyXMLDoc,
 (SELECT MyXMLDoc
  FROM  MyXMLDocs2
  WHERE MyXMLDocs2.DocID = MyXMLDocs.DocID
  FOR XML AUTO, TYPE)
FROM MyXMLDocs Where DocID = 2 FOR XML AUTO, TYPE
```

در این مثال، پرس‌وجوی خارجی در جدول MyXMLDocs با زیرپرس‌وجویی در جدول MyXMLDocs2 ترکیب می‌شود (برای این مثال، یک کپی ساده از جدول MyXMLDocs). مهم‌ترین نکته توجه برانگیز در این لیست، توانایی SQL Server 2005 برای استفاده از بخش FOR XML در زیر پرس‌وجوست. در این مورد، زیرپرس‌وجو از رهنمود Type برای برگرداندن نتایج به عنوان یک نوع داده XML طبیعی استفاده می‌کند. اگر رهنمود Type استفاده نشود، آن‌گاه نتایج به عنوان یک نوع داده nvarchar برگردانده می‌شوند و داده XML دارای موجودیت می‌شود. می‌توانید نتایج پرس‌وجوی FOR XML تودرتو را در این لیست ببینید:

```
<MyXMLDocs DocID="2">
 <MyXMLDoc>
  <MyXMLDoc xmlns="MyXMLDocSchema">
   <DocumentID>1</DocumentID>
   <DocumentBody>&quot;My text&quot;</DocumentBody>
  </MyXMLDoc>
 </MyXMLDoc>
</MyXMLDocs>
```

توجه : فرمت‌بندی را برای خواناتر شدن به لیست قبل اضافه کرده‌ام.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۴۷،۰۳ تولید طرح واره XSD درون برنامه‌ای

ویژگی جدید دیگر در پشتیبانی SQL Server 2005 FOR XML، توانایی تولید یک طرح‌واره XSD با افزودن رهنمود XMLSCHEMA به بخش FOR XML است. می‌توانید مثالی از کاربرد رهنمود XMLSCHEMA جدید را در این لیست ببینید:

```
SELECT MyXMLDoc FROM MyXMLDocs WHERE DocID=1 FOR XML AUTO,
XMLSCHEMA
```

در این مورد، به دلیل این که رهنمود XMLSCHEMA به بخش FOR XML اضافه شده است، این پرس‌وجو طرح‌واره‌ای را تولید کرده و برمی‌گرداند که ستون XML خاصی را به همراه نتیجه XML از ستون منتخب تعریف می‌کند. رهنمود XMLSCHEMA تنها با حالات FOR XML AUTO و FOR XML RAW کار می‌کند و نمی‌تواند در حالت FOR XML EXPLICIT استفاده شود. اگر رهنمود XMLSCHEMA در یک پرس‌وجوی تودرتو استفاده شود، تنها می‌تواند در سطح بالای پرس‌وجو استفاده شود. طرح‌واره XSD که از این پرس‌وجو تولید می‌شود، در این لیست نشان داده شده است:

```
  <xsd:import namespace="http://MyXMLDocSchema" />
  <xsd:element name="MyXMLDocs">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="MyXMLDoc" minOccurs="0">
          <xsd:complexType
sqltypes:xmlSchemaCollection="[tecadb].[dbo].[MyXMLDocSchema]">
            <xsd:complexContent>
              <xsd:restriction base="sqltypes:xml">
                <xsd:sequence>
                  <xsd:any processContents="strict" namespace="http://MyXMLDocSchema" />
                </xsd:sequence>
              </xsd:restriction>
            </xsd:complexContent>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
<MyXMLDocs xmlns="urn:schemas-microsoft-com:sql:SqlRowSet1">
  <MyXMLDoc>
    <MyXMLDoc xmlns="http://MyXMLDocSchema">
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
    <DocumentID>1</DocumentID>
    <DocumentBody>My New Body</DocumentBody>
   </MyXMLDoc>
  </MyXMLDoc>
</MyXMLDocs>
```

رهنمود XMLSCHEMA می‌تواند چندین طرح‌واره را برگرداند، ولی همیشه حداقل دو طرح‌واره را برمی‌گرداند: یک طرح واره برای فضای نام SqlTypes برگردانده می‌شود و طرح‌واره دومی که نتایج پرس‌وجوی FOR XML را بیان می‌کند. در لیست قبل، می‌بینید که شرح طرح‌واره ستون نوع داده XML در <xsl:element="MyXMLDocs"> شروع می‌شود. سپس، نتایج XML می‌توانند در شروع خط با <MyXMLDoc xmls ="urn:schemas-microsoft.com:sql:SqlRowSet1"> ببینید.

توجه : هم‌چنین می‌توانید یک طرح‌واره XDR[1] را با استفاده از رهنمود XMLDATA در ترکیب با بخش FOR XML تولید کنید.

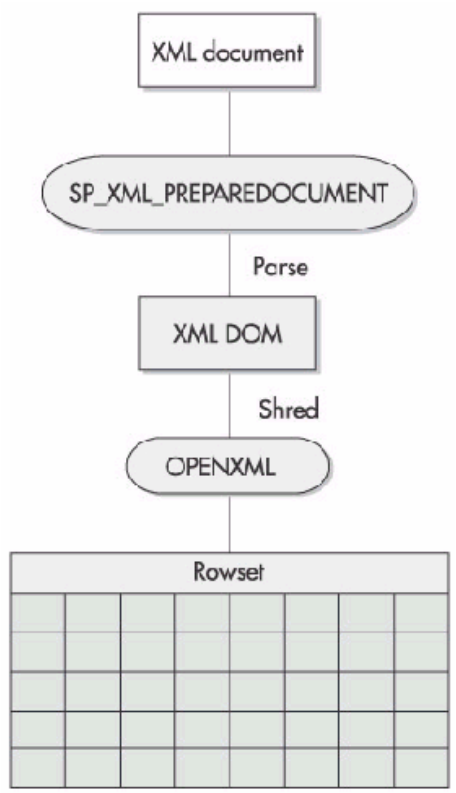
Section ۴۷،۰۴ بهبودهای OPENXML

بخش FOR XML برای بازیابی XML از پایگاه داده SQL Server 2005 عالی است. بخش FOR XML لزوماً یک سند XML را از داده رابطه‌ای ایجاد می‌کند. کلمه کلیدی OPENXML قرینه بخش FOR XML است. تابع OPENXML یک مجموعه ردیف رابطه‌ای را طبق یک سند XML فراهم می‌کند. برای استفاده از عملکرد SQL Server OPENXML، ابتدا باید رویه ذخیره شده sp_xml_preparedocument را فراخوانی کنید که سند XML را با استفاده از DOM[2] XML تجزیه کرده و هندلی را برای OPENXML برمی‌گرداند. سپس OPENXML یک دیدگاه مجموعه ردیف از سند XML تجزیه شده فراهم می‌کند. هنگامی که کار کردن با سند را به پایان رساندید، آن‌گاه رویه ذخیره شده sp_xml_removedocument را فراخوانی کنید تا منابع سیستم مصرف شده توسط OPENXML و XML DOM را آزاد کنید. می‌توانید مروری از این فرآیند را در شکل ۱-۷ ببینید.

1- XML Data Reduced
2- Document Object Model

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

شکل ۱-۷ مروری از OPENXML (b)

در SQL Server 2005، پشتیبانی OPENXML توسعه یافته است تا پشتیبانی از نوع داده XML جدید و نوع داده تعریف شده کاربر جدید را داشته باشیم. مثال زیر نحوه استفاده از OPENXML را در الحاق با یک بخش WITH در الحاق با نوع داده XML جدید نشان می‌دهد:

```
DECLARE @hdocument int
DECLARE @doc varchar(1000)
SET @doc ='<MyXMLDoc>
    <DocumentID>1</DocumentID>
    <DocumentBody>"OPENXML Example"</DocumentBody>
  </MyXMLDoc>'
EXEC sp_xml_preparedocument @hdocument OUTPUT, @doc
SELECT * FROM OPENXML (@hdocument, '/MyXMLDoc', 10)
 WITH (DocumentID  varchar(4),
      DocumentBody varchar(50))
EXEC sp_xml_removedocument @hdocument
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

در بالای این لیست، می‌توانید ببینید که دو متغیر معرفی شده‌اند. متغیر @hdocument برای ذخیره هندل سند XML که توسط رویه ذخیره شده sp_xml_preparedocument برگردانده می‌شود، استفاده خواهد شد، در حالی که متغیر @doc حاوی خود سند XML نمونه است. سپس، رویه ذخیره شده sp_xml_preparedocument اجرا شده و دو متغیر را ارسال می‌کند. رویه ذخیره شده sp_xml_preparedocument از XML DOM برای تجزیه سند XML استفاده کرده و سپس هندلی را برای سند تجزیه شده در متغیر @hdocument برمی‌گرداند. آن‌گاه این هندل سند به کلمه کلیدی OPENXML مورد استفاده در عبارت SELECT ارسال می‌شود.

اولین پارامتر مورد استفاده OPENXML، هندل سند موجود در @hdocument است. دومین پارامتر، یک الگوی XPath است که خاص گره‌ها در سند XML است که مجموعه ردیف رابطه‌ای را خواهد ساخت. سومین پارامتر، نوع نگاشت XML به رابطه‌ای را مشخص می‌کند که انجام خواهد شد. مقدار ۲ نشان می‌دهد که نگاشت عنصرگرا استفاده خواهد شد. یک مقدار ۱ نشان می‌دهد که نگاشت صفت‌گرا انجام می‌شود. بخش WITH فرمت مجموعه ردیفی را که برمی‌گرداند، فراهم می‌کند. در این مثال، بخش WITH مشخص می‌کند که مجموعه ردیف برگردانده شده شامل دو ستون varchar به نام‌های DocumentID و DocumentBody است. در حالی که این مثال نشان می‌دهد نام مجموعه ردیف‌ها با عناصر XML مطابقت دارد، ولی این یک شرط الزامی نیست. بالاخره، رویه ذخیره شده sp_xml_removedocument برای آزاد کردن منابع سیستم اجرا می‌شود.

عبارت SELECT که از ویژگی OPENXML استفاده می‌کند، مجموعه ردیفی را برخواهد گرداند که شامل مقدار عناصر سند XML است. می‌توانید نتایج کاربرد OPENXML را در این لیست ببینید:

```
DocumentID      DocumentBody
----------      ------------------------------------------------
1               "OPENXML Example"

(1 row(s) affected)
```

Article XLVIII. بارگذاری حجیم XML

یکی از اولین مواردی که احتمالاً قصد دارید انجام دهید تا از نوع داده XML جدید بهره ببرید، بارگذاری اسناد XML در ستون‌های XML از دیسک است. ویژگی بارگذاری حجیم XML این کار را بسیار ساده کرده است. این ویژگی یک مکانیزم بسیار سریع برای بارگذاری اسناد XML در ستون‌های SQL Server فراهم می‌کند. می‌توانید مثالی از کاربرد بارگذاری حجیم XML را در این لیست ببینید:

```
INSERT into MyXMLDocs(MyXMLDoc) SELECT * FROM OPENROWSET
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
(Bulk 'c:\temp\MyXMLDoc.xml', SINGLE_CLOB) as x
```

در این مثال، عبارت INSERT برای درج نتایج عبارت SELECT * FROM OPENROWSET در ستون MyXMLDoc در جدول MyXMLDocs استفاده می‌شود. تابع OPENROWSET از تأمین کننده مجموعه ردیف Bulk برای OPENROWSET برای خواندن داده از فایل 'C:\temp\MyXMLDoc.xml' استفاده می‌کند. می‌توانید محتویات فایل MyXMLDoc.xml را در این لیست ببینید:

```
<MyXMLDoc xmlns="http://MyXMLDocSchema">
    <DocumentID>3</DocumentID>
    <DocumentBody>"The Third Body"</DocumentBody>
</MyXMLDoc>
```

اگر این فرمان را از SQL Server Management Studio اجرا کنید، باید به خاطر داشته باشید که این فرمان در سیستم SQL Server اجرا خواهد شد و بنابراین مراجع فایل و مسیر باید در سیستم سرور محلی باشند. آرگومان SINGLE_CLOB مشخص می‌کند که داده این فایل، در یک ردیف تکی درج خواهد شد. اگر آرگومان SINGLE_CLOB را حذف کنید، آن‌گاه داده این فایل می‌تواند در چند ردیف درج شود. به‌طور پیش‌فرض، تأمین کننده Bulk برای تابع OPENROWSET، ردیف‌ها را در کاراکتر Carriage Return برمی‌گرداند که حایل ردیف پیش‌فرض است. به روش دیگر، می‌توانید حایل‌های فیلد و ردیف را با استفاده از آرگومان‌های FIELDTERMINATOR و ROWREMINATOR تابع OPENROWSET مشخص کنید.

Article XLIX. دستیابی HTTP SOAP طبیعی

ویژگی مرتبط XML جدید دیگر در SQL Server 2005، پشتیبانی HTTP SOAP طبیعی است. این ویژگی جدید به SQL Server امکان می‌دهد تا مستقیماً به درخواست‌های HTTP SOAP پاسخ دهد که توسط سرویس‌های وب بدون نیاز به یک سیستم IIS برای عمل کردن به عنوان یک واسطه، صادر شده‌اند. با استفاده از پشتیبانی HTTP SOAP، می‌توانید سرویس‌های وبی ایجاد کنید که قادر به اجرای دسته‌های T-SQL، رویه‌های ذخیره شده و توابع عددی تعریف شده کاربر هستند. برای اطمینان از یک سطح بالای امنیت پیش‌فرض، دستیابی HTTP طبیعی به‌طور پیش‌فرض غیرفعال می‌شود. هرچند می‌توانید پشتیبانی HTTP را برای ایجاد یک نقطه انتهای HTTP برای اولین بار فعال کنید. می‌توانید مثال کدی را برای ایجاد یک نقطه انتهای HTTP در این لیست ببینید:

```
CREATE ENDPOINT MyHTTPEndpoint
STATE = STARTED
```

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com


```
AS HTTP(
  PATH = '/sql',
  AUTHENTICATION = (INTEGRATED ),
  PORTS = ( CLEAR ),
  SITE = 'server'
  )
FOR SOAP (
  WEBMETHOD 'http://tempUri.org/'.'GetProductName'
        (name='AdventureWorks.dbo.GetProductName',
         schema=STANDARD ),
  BATCHES = ENABLED,
  WSDL = DEFAULT,
  DATABASE = 'AdventureWorks',
  NAMESPACE = 'http://AdventureWorks/Products'
  )
```

این مثال، ایجاد یک نقطه انتهای HTTP به نام MyHTTPEndPoint را برای رویه ذخیره شده‌ای به نام GetProductName در پایگاه داده AdvantureWorks نمونه تشریح می‌کند. هنگامی که نقطه انتهای HTTP ایجاد می‌شود، می‌تواند از طریق یک درخواست SOAP صادر شده توسط یک برنامه، مورد دستیابی قرار گیرد. می‌توانید از عبارات DDL ALTER ENDPOINT و DROP ENDPOINT برای مدیریت نقاط انتهای SQL Server HTTP استفاده کنید. نقاط انتهای HTTP جدید هم‌چنین رمزگذاری جریان داده را با استفاده از SSL ممکن می‌سازند. برای کسب اطلاعات بیشتر در مورد پشتیبانی HTTP جدید SQL Server می‌توانید به فصل ۲ مراجعه کنید.

این فرمان نحوه لیست کردن نقاط انتهای HTTP ایجاد شده را نشان می‌دهد:

```
select * from sys.http_endpoints
```

Article L. بهبودهای XML برای Analysis Server

در حالی که اکثر بهبودهای XML در SQL Server 2005 برای موتور پایگاه داده رابطه‌ای پیاده‌سازی شده‌اند، Analysis Services هم‌چنین چندین ویژگی مرتبط به XML جدید مهم را دریافت کرده است که مهم‌ترین آن، XML for Analysis Services است که XMLA نیز نامیده می‌شود. در این بخش، مطالب بیشتری درباره ویژگی XML for Analysis Services جدید می‌آموزید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۵۰.۰۱ XML for Analysis Services

XML for Analysis Services یک API است که دستیابی داده به منابع داده Analysis Services را فراهم می‌کند که در وب قرار دارند. XML for Analysis Services بعد از OLE DB مدلسازی شده است که یک مدل دستیابی داده جهانی را برای منبع داده چند بعدی فراهم می‌کند. هرچند، برخلاف OLE DB مبتنی بر COM، XML for Analysis Services برطبق پروتکل SOAP مبتنی بر XML ساخته شده است. هم‌چنین برخلاف OLE DB که بر طبق مدل کلاینت/ سرور ساخته شده است، XML for Analysis Services برای کاربرد در وب بهینه شده است.

XML for Analysis Services دو متد قابل دستیابی عمومی را فراهم کرده است: متد Discover و متد Execute. متد Discover همان‌گونه که از نام آن برمی‌آید، اطلاعاتی درباره یک منبع داده می‌گیرد. متد Discover می‌تواند اطلاعاتی درباره منابع داده موجود، تأمین کنندگان منبع داده و فوق داده موجود لیست کند. متد Execute به یک برنامه امکان اجرای فرامین را در منابع داده XML for Analysis Services می‌دهد.

Article LI. دیدگاه‌های کاتالوگ XML سیستم

SQL Server 2005 اطلاعاتی درباره XML مورد استفاده در سرور را در تعدادی از دیدگاه‌های جدید سیستم ذخیره می‌کند. جدول ۱-۷ به‌طور خلاصه دیدگاه‌های XML سیستمی جدید را بیان می‌کند.

(a) جدول ۱-۷ دیدگاه‌های کاتالوگ XML SQL Server 2005

دیدگاه کاتالوگ XML	شرح
sys.xml_attributes	این دیدگاه، ردیفی را برای هر صفت XML ذخیره شده فراهم می‌کند.
sys.xml_components	این دیدگاه، ردیفی را برای هر جزء یک طرحواره XML فراهم می‌کند.
sys.xml_component_placements	این دیدگاه، ردیفی را برای هر جایگزین یک جزء XML فراهم می‌کند.
sys.xml_elements	این دیدگاه، ردیفی را برای هر جزء XML که یک عنصر XML است، فراهم می‌کند.
sys.xml_facets	این دیدگاه، ردیفی را برای هر محدودیت یک نوع XML

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

	فراهم می‌کند.
sys.xml_model_groups	این دیدگاه، لیستی از تمام اجزای XML فراهم می‌کند که بخشی از یک Model-Group است.
sys.xml_namespaces	این دیدگاه ردیفی برای هر فضای نام تعریف شده XSD فراهم می‌کند.
sys.xml_types	این دیدگاه ردیفی برای هر جزء XML که یک نوع XML است، فراهم می‌کند.
sys.xml_wildcards	این دیدگاه ردیفی برای هر جانشین صفت یا عنصر XML فراهم می‌کند.
sys.xml_wildcard_namespaces	این دیدگاه ردیفی برای هر فضای نام جانشین XML فراهم می‌کند.

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

فصل هشتم

Reporting Services

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

یکی از مهم‌ترین بهبودها در SQL Server 2005، زیر سیستم Reporting Services جدید است. Reporting Services ابتدا به عنوان یک برنامه افزودنی به SQL Server 2000 معرفی شد و حفره مهمی را در خط تولید SQL Server پر کرد. SQL Server از زمان معرفی، همیشه یکی از ساده‌ترین محیط‌های پایگاه داده رابطه‌ای از لحاظ پیاده‌سازی بوده است و این سهولت کاربرد، آن را برای پیاده‌سازی‌های سطح بخش اداری و یک محیط پایگاه داده برای تجارت‌های کوچک و متوسط محبوب کرده است. هرچند، هنگام نصب SQL Server، اگر بخواهید بازیابی اطلاعات از پایگاه داده را با تولید گزارشات شروع کنید، آن‌گاه باید از آن خارج شده و محصول دیگری را با قابلیت‌های گزارشگیری تعبیه شده برای ارسال آن داده به خروجی به عنوان یک گزارش قابل مشاهده یا چاپی امتحان کنید. SQL Server هیچ ابزار تعبیه شده‌ای نداشت که بتواند گزارشات را تولید کند (متداول‌ترین و مورد انتظارترین خروجی یک سیستم پایگاه داده رابطه‌ای، گزارشات هستند). برای داشتن مقدار گزارشات اولیه خارج از SQL Server، بیشتر شرکت‌ها استفاده از ابزارهای گزارشگیری دسک‌تاپ، نظیر Microsoft Access را شروع کردند، ولی سازمان‌های متوسط و بزرگ محصولات گزارشگیری شخص ثالث قوی‌تر، نظیر Crystal Reports از Business Object پذیرفته شد.

معرفی Reporting Services در SQL Server 2000 همه چیز را تغییر داد. Reporting Services یک راه‌حل گزارشگیری را فراهم کرد که خارج از حیطه قابلیت‌های راه‌حل‌های گزارشگیری ساده نظیر Access است. Reporting Services یک سیستم گزارشگیری جهانی است که تنها توانایی گزارشات طراحی گرافیکی را فراهم نمی‌کند، بلکه هم‌چنین به شما امکان می‌دهد تا این گزارشات را در سراسر جهان با تنوعی از فرمت‌های مختلف از جمله گزارشات HTML مبتنی بر وب، گزارشات کلاینت غنی مبتنی بر Windows و گزارشاتی برای نمایش روی وسایل همراه به صورت ایمن توزیع کنید. SQL Server 2005 Reporting Services به همه‌گیر شدن در استاندارد جدیدی برای گزارشگیری جهانی SQL Server ادامه می‌دهد. در اولین بخش این فصل، مروری بر معماری و اجزای سیستم مورد استفاده SQL Server 2005 Reporting Services داریم. سپس، در بخش دوم این فصل، نگاهی مفصل‌تر خواهیم داشت و نحوه طراحی و توزیع گزارشات را با استفاده از Reporting Services بررسی می‌کنیم.

Article LII. معماری Reporting Services

SQL Server 2005 Reporting Services تنها یک ابزار طراحی گزارش نیست. در عوض، یک محیط گزارشگیری است که نه تنها ایجاد گزارشات را ممکن می‌سازد، بلکه هم‌چنین تعریف گزارشات را ذخیره می‌کند، دستیابی ایمن به گزارشات را فراهم می‌کند، گزارشات را در تنوعی از فرمت‌های

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

خروجی متفاوت نمایش می‌دهد، تحول گزارش را با استفاده از یک مکانیزم تحویل push یا pull زمانبندی می‌کند و توزیع این گزارشات را ممکن می‌سازد.

Reporting Services یک جزء نصب اختیاری است که نیاز به وجود IIS[1] دارد که باید نصب شده باشد. اگر IIS روی سیستم وجود نداشته باشد، گزینه نصب Reporting Services در کادرهای محاوره‌ای نصب SQL Server 2005 ارایه نمی‌شود. در حالی که Reporting Services می‌تواند، ولی سیستم سرور مشابه با موتور داده SQL Server نصب شود، برای مقیاس‌پذیری بهبود یافته، معمولاً بهتر است Reporting Services را روی سرور مجزایی نصب کنید.

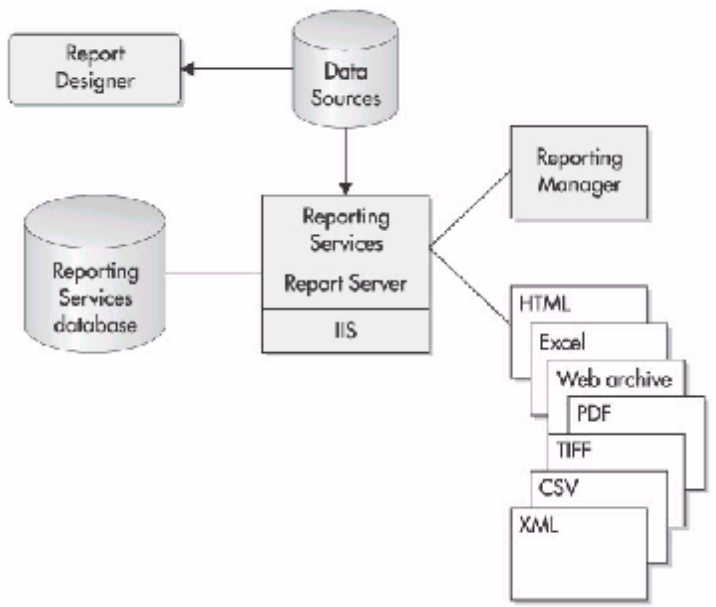
نکته: Reporting Services به عنوان بخشی از Reporting Services تثبیت شده است و نیاز به هیچ ثبت مجزایی برای استفاده در یک سیستم نیست. هرچند، اگر آن را در سیستم مجزایی پیاده‌سازی کنید، نیاز به یک ثبت اضافی دارید.

SQL Server 2005 Reporting Services بیش از یک برنامه تنهاست. Reporting Services یک زیرسیستم مبتنی بر سرور است که برای فعال کردن ایجاد، مدیریت و توزیع گزارشات در سراسر جهان طراحی شده است. Reporting Services می‌تواند برای ایجاد گزارشات جدولی، گرافیکی و ... استفاده شود که می‌تواند با استفاده از یک اتصال مبتنی بر وب مشاهده و مدیریت شود. این گزارشات می‌توانند در یک پایگاه داده Reporting Services و هر منبع پایگاه داده سازگار با ODBC یا OLE DB اجرا شوند. می‌توانید مروری از معماری Reporting Services را در شکل ۱-۸ ببینید.

1- Internet Information Services

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

شکل ۱-۸ مروری بر Reporting Services (b)

همان‌گونه که در شکل ۱-۸ می‌بینید، گزارشات Reporting Services با استفاده از Report Designer ایجاد می‌شوند. Reporting Services Report Designer یک سطح طراحی کاملاً گرافیکی و برنامه‌ای با رابط کاربر است که به شما امکان طراحی محاوره‌ای و امتحان گزارشات را می‌دهد. برخلاف طراح گزارش باندار که بخشی از برنامه Microsoft Access آشناست، Reporting Services Report Designer کاملاً بدون باند است که این مسأله انعطاف‌پذیری زیادی به آن می‌دهد. یک طراح گزارش باندار نواحی خاصی از طراح گزارش را برای کاربردهای از پیش‌تعریف شده محدود می‌کند؛ مثلاً، بخش بالای صفحه همیشه عنوان است، بخش میانی صفحه، بدنه گزارش و بخش پایینی صفحه، برای پاورقی گزارش می‌باشد. در حالی که گزارشات مختلف از این شیوه پیروی می‌کنند، انعطاف‌پذیری برای انجام چیزی غیر از این ندارید. چیدمان بدون باند Reporting Services Report Designer آزادی بیشتری به شما می‌دهد: مثلاً، به سادگی می‌توانید گزارش را در چهار قسمت صفحه طراحی کنید که هر قسمت از داده و فرمت‌بندی خود استفاده کند. یک ظاهر بسیار مفصل‌تر در Reporting Services Report Designer در بخش دوم این فصل ارایه می‌شود.

بعد از طراحی یک گزارش با استفاده از Report Designer، تعاریف گزارش در پایگاه داده ReportServer ذخیره می‌شود. گزارشات Reporting Services در یک فرمت داده مبتنی بر XML جدید به نام RDL ذخیره می‌شود. به‌طور پیش‌فرض، این تعاریف RDL در پایگاه داده

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

SQL Server 2005 ReportServer ذخیره می‌شوند. علاوه بر مشخصه‌های RDL گزارش، پایگاه داده ReportServer هم‌چنین اطلاعاتی درباره امنیت و مقصد یک گزارش ذخیره می‌کند.

Reporting Services Report Server که جزء اصلی در معماری Reporting Services است، یک برنامه ASP.NET است که تحت IIS Windows اجرا می‌شود. مسئولیت اصلی Report Server، پردازش فایل‌های تعریف گزارش ذخیره شده و سپس نمایش گزارشات به فرمتی مناسب برای مکانیزم تحویل مشخص شده است. علاوه بر گزارشات چاپی استاندارد، Reporting Services Report Server می‌تواند این فرمت‌های خروجی مختلف را نمایش دهد:

- HTML
- Excel
- Web archive
- PDF
- TIFF
- CSV
- XML

Reporting Services با استفاده از Report Manager مدیریت می‌شود. Report Manager یک برنامه تحت وب است که به DBA یا راهبر گزارشگیری امکان کنترل امنیت و ویژگی‌های مدیریتی کلی گزارشات ایجاد شده با استفاده از Reporting Services را می‌دهد. این امر شامل مشخص کردن فردی است که می‌تواند یک گزارش را ایجاد کرده و تغییر دهد و فردی که بتواند گزارش معینی را اجرا کند. Report Manager هم‌چنین می‌تواند زمان‌بندی‌های تحویل push و pull را برای گزارشات Reporting Services تنظیم کند.

در بخش بعد، نگاهی عمقی‌تر به عملکرد فراهم شده توسط هر یک از این اجزا می‌اندازیم.

Article LIII. اجزای Reporting Services

سیستم Reporting Services جدید SQL Server 2005 شامل تنوعی مرتبط است. در این بخش، نگاه دقیق‌تری به هر یک از این اجزا داریم، با بخشی شروع می‌کنیم که مشهودتر است: Report Designer

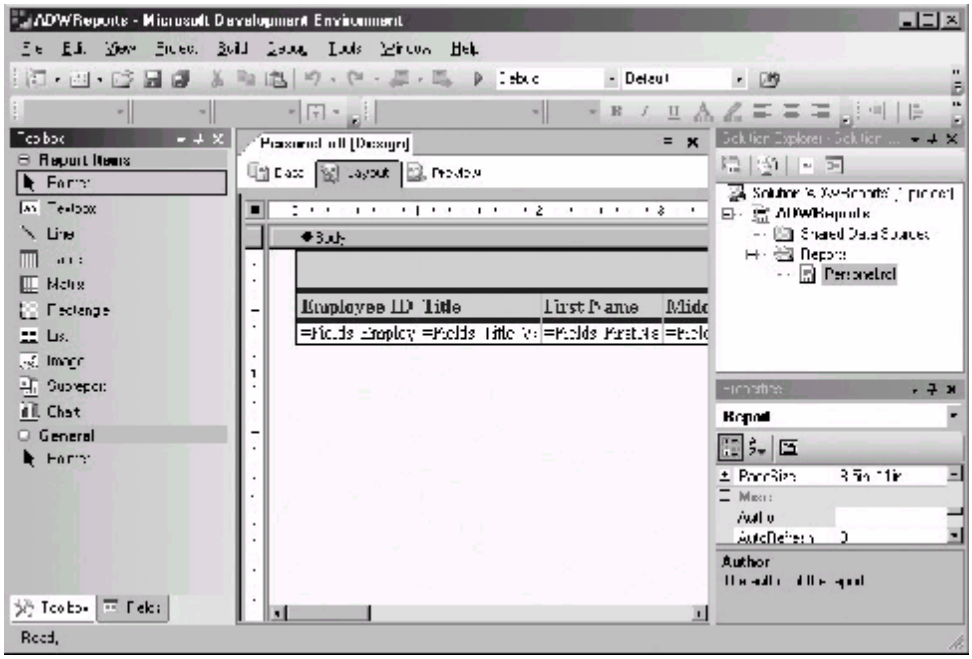
Section ۵۳.۰۱ Report Designer

Reporting Services Report Designer به شما امکان می‌دهد تا گزارشات را به‌طور ویژوال طراحی کنید و توزیع آن‌ها را کنترل نمایید. برخلاف برنامه افزودنی Reporting Services برای SQL Server 2000 که برای طراحی گزارشات نیاز به Visual Studio داشت، SQL Server 2005 Reporting Services، Report Designer را به عنوان بخشی از Business Intelligence Development Studio

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

فراهم کرده است. برای شروع، Report Designer، ابتدا SQL Server 2005 Business Intelligence Studio را باز کرده و سپس گزینه File | New | Project | Report Progect را برای ایجاد یک پروژه Report Services جدید انتخاب کنید. سپس، برای باز کردن Report Designer، گزینه Project | Add Item | Report را انتخاب کنید. می‌توانید مثالی از Reporting Services Report Designer را در شکل ۲-۸ ببینید.



(a)

شکل ۲-۸ Reporting Services Report Designer (b)

جعبه ابزار

پنجره جعبه ابزار در Reporting Services Report Designer در سمت چپ شکل ۲-۸ نشان داده شده است. جعبه ابزار برای کشیدن و انداختن اجزا در سطوح طراح مربوط به آن‌ها استفاده می‌شوند.

کنترل‌های گزارش

جعبه ابزار Report Designer حاوی این کنترل‌هاست:

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

- **Textbox:** کنترل Textbox به شما امکان نمایش داده متنی را روی گزارش می‌دهد. Textbox می‌تواند هر جایی در گزارش قرار گیرد و می‌تواند حاوی داده ستونی، برچسب‌ها و فیلدهای محاسباتی باشد.
- **Line:** کنترل Line به شما امکان رسم یک خط را روی چیدمان گزارش می‌دهد.
- **Table:** Table به شما امکان انقیاد یک جدول به چیدمان گزارش را می‌دهد.
- **Matrix:** می‌توانید از کنترل Matrix برای نمایش مشبکی در چیدمان گزارش استفاده کنید. می‌توانید کنترل Matrix را به مجموعه داده گزارش مقید کنید.
- **Rectangle:** کنترل Rectangle در اصل به عنوان کانتینری برای سایر عناصر گزارش استفاده می‌شود. هم‌چنین می‌تواند به عنوان یک عنصر گرافیکی توسط کنترل Line استفاده شود.
- **List:** کنترل List به شما امکان می‌دهد تا لیستی را در چیدمان گزارش خود قرار دهید. این لیست می‌تواند به فیلدها در مجموعه داده شما مقید شوند.
- **Image:** کنترل Image به شما امکان می‌دهد تا تصاویر باینری را به چیدمان گزارش مقید کنید. از این فرمت‌ها پشتیبانی می‌شود، bmp، jpg، gif و png.
- **Subreport:** کنترل Subreport برای لینک کردن بخشی از گزارش به گزارش تعریف شده قبلی دیگری استفاده می‌شود. Subreport می‌تواند یک گزارش مستقل باشد یا برای اجرا در گزارش دیگری طراحی شود.
- **Chart:** کنترل Chart نموداری را در چیدمان گزارش رسم می‌کند. کنترل Chart می‌تواند به مجموعه داده گزارش مقید شود و از تعداد زیادی از انواع نمودار مختلف پشتیبانی می‌کند، از جمله: ستون‌ها، میله، خط، پای، پخشی، حبابی، ناحیه‌ای، دوناتی، راداری، سهامی و قطبی.

Design Surface

در مرکز صفحه نمایش در شکل ۲-۸، می‌توانید Report Designer Design Surface را ببینید. Report Designer Design Surface محلی است که چیدمان گزارش خود را ایجاد می‌کنید. Design Surface سه برگه را ارایه می‌دهد: Data، Layout و Preview. برای ایجاد یک گزارش، ابتدا باید مجموعه داده‌ای را تعریف کنید و می‌توانید این کار را با استفاده از برگه Data انجام دهید. مجموعه داده با مشخص کردن اتصال پایگاه داده با وارد کردن دستی رشته اتصال مناسب یا با پر کردن اعلان‌های نمایش داده شده در کادر محاوره‌ای Data Link ساخته می‌شود. هنگامی که اتصالی تنظیم می‌شود، آن‌گاه پرس‌وجوی خود را با وارد کردن عبارت SQL یا با استفاده از Query Designer

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

گرافیکی تعریف کنید. Query Designer به شما امکان می‌دهد تا به‌طور ویژوال یک پرس‌وجوی SQL را با کشیدن و انداختن جداول بسازید و سپس ستون‌های مناسب را انتخاب کنید و به‌طور ویژوال هر تلفیق مورد نیاز و معیار انتخاب ردیف را تعریف کنید. هنگامی که پرس‌وجو طراحی شد، یک نمایش سازنده پرس‌وجوی گرافیکی آن پرس‌وجو در برگه Data نشان داده خواهد شد.

برگه Layout محلی است که گزارش خود را طراحی می‌کنید. گزارش را با کشیدن و انداختن آیتم‌ها از جعبه ابزار به پنجره Layout و سپس انتقال و تغییر اندازه آن‌ها طراحی کنید. همان‌گونه که در شکل ۲-۸ می‌بینید، پنجره Layout تمام کنترل‌های Reporting Services را که به گزارش اضافه می‌شوند و انقیاد آن‌ها را نشان می‌دهد. برگه Preview به شما اجازه پیش‌نمایش آن چیزی را می‌دهد که گزارش نمایش داده شده شبیه آن خواهد بود.

هنگامی که برنامه Preview را کلیک می‌کنید، Report Designer گزارش را اجرا خواهد کرد و آن را در برگه Preview نمایش می‌دهد. Preview به شما اجازه نمی‌دهد تا تغییری در ظاهر گزارش بدهید. برای تغییر گزارش، باید از برگه Layout استفاده کنید.

Solution Explorer

می‌توانید Report Designer Solution Explorer را در گوشه سمت راست بالای شکل ۲-۸ ببینید. Solution Explorer یک نمای درختی سلسله مراتبی از فایل‌ها و پروژه‌های متفاوتی که در یک راه‌حل Business Intelligence Developmant Studio هستند، فراهم می‌کند. آیتم بالا در سلسله مرتبه Solution Explorer نام راه‌حل است. در شکل ۲-۸، می‌توانید ببینید که راه‌حل ADWReports نامیده شده است. تحت این راه‌حل، می‌توانید یک یا چند پروژه داشته باشید. در شکل ۲-۸، می‌توانید ببینید که در این مثال، راه‌حل دارای یک پروژه Reporting Services است که reports نامیده می‌شود و تحت آن پروژه گزارشی به نام Personal.rdl است. هر پروژه Reporting Services می‌تواند حاوی چندین گزارش باشد.

Properties

می‌توانید پنجره Report Designer Properties را در گوشه سمت راست پایین شکل ۲-۸ ببینید. پنجره Properties می‌تواند برای تنظیم صفات آیتم‌های چیدمان گزارش در زمان طراحی استفاده شود.

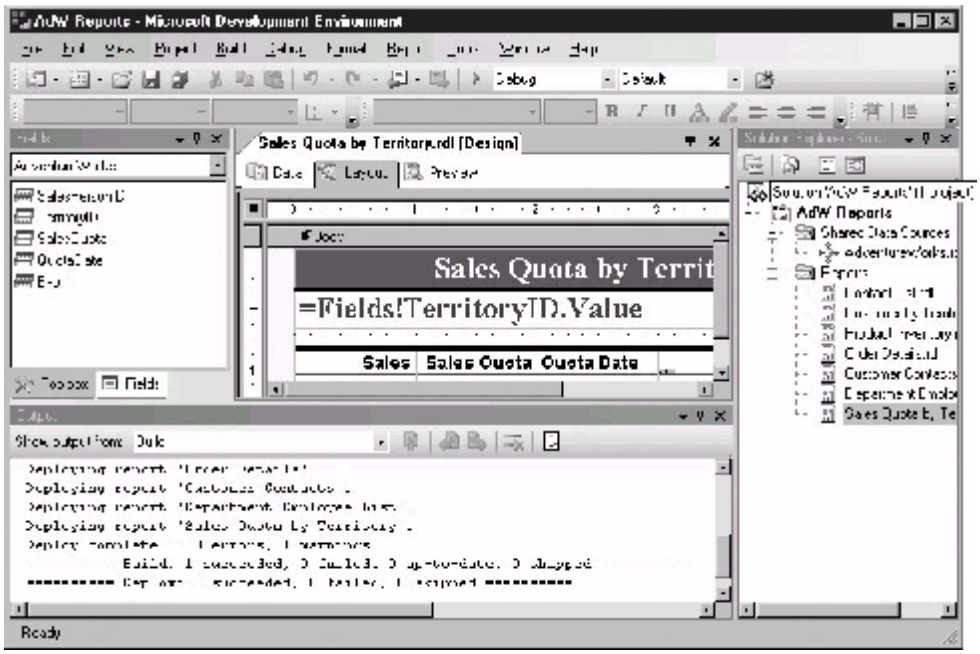
Fields

Report Designer هم‌چنین یک پنجره Fields را فراهم کرده است که محتویات مجموعه داده‌ای را نشان می‌دهد که در حال استفاده است. می‌توانید فیلدهای لیست شده در پنجره Fields را

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

بکشید و در سطح طراحی Report Designer بیندازید تا در گزارش قرار گیرند. می‌توانید پنجره Fields را در گوشه سمت چپ بالای شکل ۳-۸ ببینید.



(c)

شکل ۳-۸ پنجره Report Designer Fields (d)

می‌توانید بین پنجره Fields و جعبه ابزار با کلیک کردن برگه مناسب نشان داده شده در پایین پنجره Fields حرکت کنید.

Output

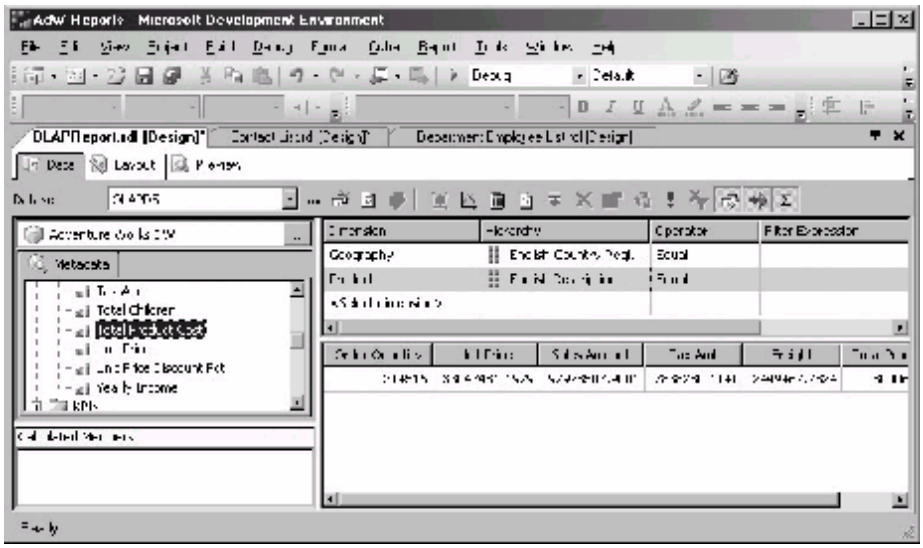
Report Designer همچنین یک پنجره Output را فراهم کرده است که نتایج ساخت و توزیع گزارشات را نشان می‌دهد. بعد از طراحی گزارش، باید ساخته شده و سپس قبل از استفاده، توزیع شود. فرآیند Build یک اسمبلی NET. را ایجاد می‌کند، در حالی که فرآیند Deploy آن اسمبلی را گرفته و آن را در پایگاه داده ReportServer نصب می‌کند. نتیجه این اعمال در پنجره Output نشان داده شده است و شما می‌توانید آن را در پایین شکل ۳-۸ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۵۳،۰۲ OLAP Report Designer

شبیه Reporting Services Report Designer به شما امکان می‌دهد تا گزارشاتی را برای پایگاه‌های داده رابطه‌ای طراحی کنید، OLAP به شما امکان می‌دهد تا گزارشاتی را برای پایگاه‌های داده Analysis Services بسته به نوع منبع داده منتخب خود طراحی کنید، Report Designer به‌طور خودکار به عنوان Report Designer رابطه‌ای یا OLAP Report Designer باز می‌شود. اگر منبع داده‌ای که انتخاب می‌کنید، یک منبع داده ODBC یا Native SQL Client باشد، آن‌گاه Report Designer رابطه‌ای که قبلاً در شکل ۳-۸ نشان داده شد، باز خواهد شد. اگر منبع داده منتخب شما یک منبع داده Microsoft SQL Server Analysis Services باشد، آن‌گاه OLAP Report Designer جدید که در شکل ۴-۸ نشان داده شده است، باز خواهد شد.



(a)

شکل ۴-۸ OLAP Report Designer (b)

OLAP Report Designer برای ساخت گزارشات بر طبق مکعب‌های Analysis Services بهینه می‌شود. به‌جای نشان دادن جداول و ستون‌ها، طراح به شما فوق داده مکعب‌هایی را نشان می‌دهد که می‌توانید برای ساخت گزارشات خود استفاده کنید.

Metadata

در سمت چپ شکل ۴-۸، می‌توانید پنجره OLAP Report Designer Metadata را ببینید. پنجره Metadata تمام صفات مکعب را لیست می‌کند. در بالای لیست، اندازه مکعب‌ها را می‌یابید که

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

بعد از آن‌ها KPIها آمده است و سپس تمام ابعاد مختلف را داریم. می‌توانید فیلدهای لیست شده در پنجره Fields را کشیده و در پنجره‌های Dimesions و Measures بیندازید.

پنجره Dimansions

پنجره Dimansions که در بخش سمت راست بالای شکل ۴-۸ نشان داده شده است، برای مشخص کردن تمام ابعادی استفاده می‌شود که در گزارش استفاده می‌شوند. برای افزودن یک بعد به پنجره Dimansions، بعد مناسب را برای پنجره Metadata کشیده و آن را در پنجره Dimansions بیندازید.

پنجره Measures

پنجره OLAP Report Designer Measures در بخش پایین شکل ۴-۸ نشان داده شده است. معمولاً از پنجره Measures خیلی بیش از پنجره Dimansions استفاده می‌کنید (با کشیدن و انداختن اندازه‌ها از پنجره Metadata در پنجره Measures). OLAP Report Designer به‌طور خودکار هدرها را در سراسر بالای پنجره Measures می‌سازد و بقیه پنجره را با مقادیر داده مناسب از مکعب پر می‌کند.

Section ۵۳،۰۳ Report Server

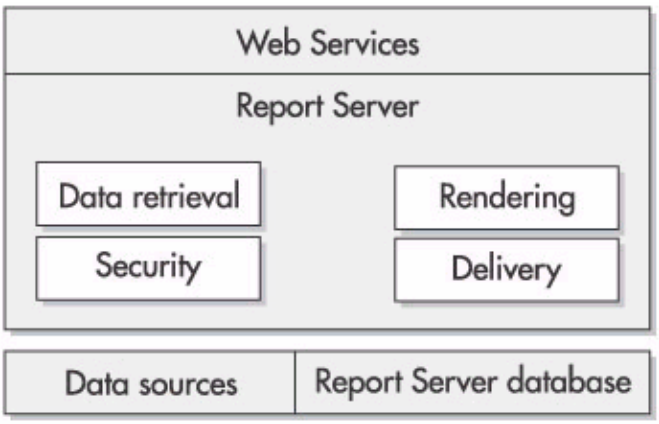
Report Server موتور نمایش و توزیع اصلی Reporting Services است. Reporting Services فایل‌های RDL تولید شده توسط Report Designer را مصرف کرده و گزارش را به فرمت خروجی مناسب نمایش می‌دهد. Report Server علاوه بر نمایش گزارشات، امنیت و توزیع گزارشات را مدیریت می‌کند. شکل ۵-۸ مروری از عملکرد فراهم شده توسط Reporting Services Report Server را نشان می‌دهد.

Report Server یک برنامه مبتنی بر ASP.NET است و یک نقطه انتهای HTTP SOAP را نشان می‌دهد که برای مدیریت و دستیابی سرور استفاده می‌شود. Report Server قلب SQL Server 2005 Reporting Services است و تمام وظایف الزامی تولید و توزیع گزارش را مدیریت می‌کند و هنگامی که کاربری درخواست یک گزارش می‌کند یا گزارشی به یک کاربر نهایی توزیع می‌شود، Report Server صفات امنیتی گزارش را بررسی می‌کند تا اطمینان یابد که کاربر دارای مجوزهایی برای خود گزارش و اشیای پایگاه داده مورد استفاده گزارش است. اگر کاربر دارای مجوزهای مورد نیاز باشد، آن‌گاه Report Server تعریف گزارش را از پایگاه داده ReportServer بازیابی کرده و گزارش را برطبق فرمت مشخص شده در RDL نمایش می‌دهد. هنگام نمایش گزارش، Report Server به تمام

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

منابع داده مورد نیاز دستیابی دارد، داده را بازیابی کرده و گزارش را می‌سازد. هنگامی که گزارشی ایجاد شده باشد، Report Server توزیع گزارش را به تمام مقاصد تحویل از پیش‌تعریف شده مدیریت می‌کند.



(a)

(b) **شکل ۵-۸ مروری بر Report Server**

Report Server نتایج بازیابی شده را به فرمت واسطه‌ای برای یک مقدار زمان از پیش‌تعریف شده بازیابی می‌کند. این کش کردن، یک ویژگی عالی برای مقیاس‌پذیری است، زیرا درخواست‌های تکراری را برای پردازش سریع گزارش ممکن می‌سازد. هنگامی که گزارشات کش می‌شوند، تمام بازیابی داده مورد نیاز و مراحل نمایش تکمیل شده و Report Server باید گزارش کش شده را به کاربر نهایی توزیع می‌کند.

تحویل گزارش

یکی از نکات کلیدی در ایجاد راه‌حل گزارشگیری، تعریف نحوه تحویل گزارشات به کاربر نهایی است. Report Server مسئول تحویل گزارشات و پشتیبانی از گزارشات طبق تقاضا به شیوه Pull و شیوه Push و گزینه‌های تحویل مبتنی بر اشتراک است.

- **تحویل طبق تقاضا:** Reporting Services دو نوع تحویل گزارش طبق تقاضا را فراهم کرده است: دستیابی گزارش مبتنی بر URL و دستیابی گزارش مبتنی بر سرویس‌های وب. با دستیابی URL، کاربر نهایی مستقیماً یک URL را در مرورگر وارد کرده یا روی لینکی کلیک می‌کند که به Report Server دستیابی دارد. هر پارامتر مورد نیاز به عنوان بخشی از URL ارسال می‌شود. Report Server گزارش را به فرمت HTML در مرورگر در

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

کاربر نمایش خواهد داد. کاربران نهایی هم‌چنین می‌توانند با صادر کردن درخواست‌های SOAP برای Reporting Services Report Server، به گزارشات دستیابی داشته باشند. در این روش، کاربر معمولاً گزینه‌ای را از برنامه خود انتخاب می‌کند که واقعاً درخواست SOAP را واگذار می‌نماید. یک مزیت این روش این است که SOAP روش‌های اکتشافی را فراهم می‌کند که به برنامه امکان کشف پویای هر پارامتر گزارش مورد نیاز را می‌دهد. همان‌گونه که در بخش بعدی خواهید دید، دستیابی Reporting Services SOAP تنها محدود به تحویل گزارشات نیست. هم‌چنین دستیابی کاملی به توابع مدیریتی Reporting Services فراهم می‌کند.

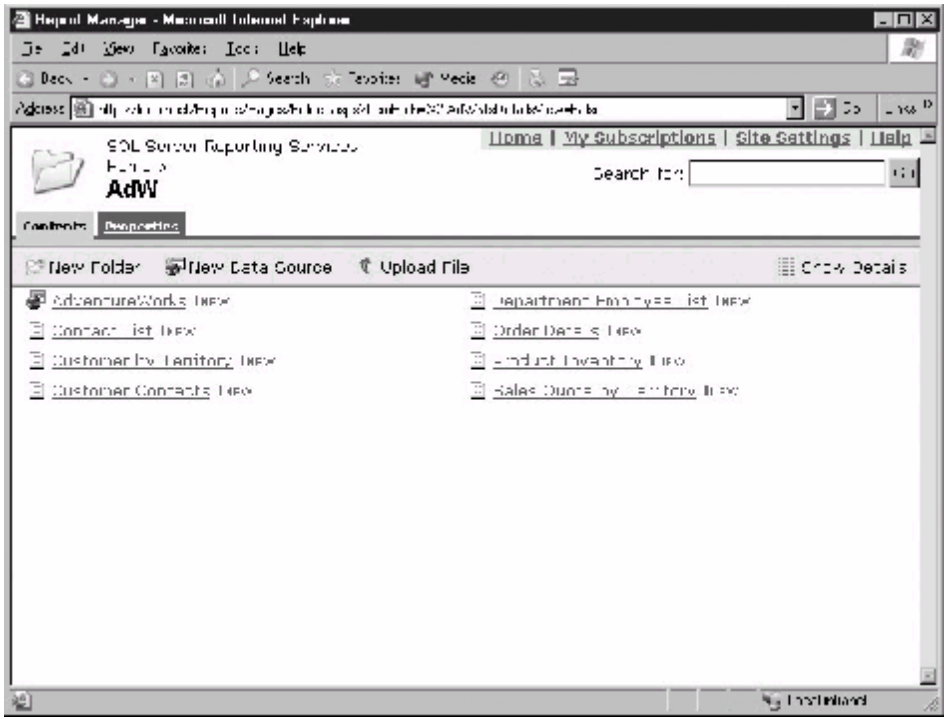
- **تحویل اشتراکی:** علاوه بر تحویل گزارش شیوه pull، Reporting Services هم‌چنین از تحویل اشتراکی شیوه push پشتیبانی می‌کند. در مقایسه با تحویل گزارش pull که کاربر تولید گزارش را با تحویل گزارش شیوه push مبتنی بر اشتراک شروع می‌کند، موتور Report Server گزارشات را بر طبق یک زمان‌بندی از پیش‌تعریف شده به کاربران نهایی تحویل می‌دهد. اشتراک‌ها می‌توانند بر طبق زمان‌ها باشند یا می‌توانند مبتنی بر داده باشند. برای نمونه، می‌توانید Report Server را برای تحویل یک گزارش معین یا مجموعه‌ای از گزارشات در انتهای روز، هفته یا ماه پیکربندی کنید. برای تحویل گزارشات بر طبق یک زمان‌بندی معین، Reporting Services از SQL Agent استفاده می‌کند. اشتراک‌های مبتنی بر داده متفاوت از اشتراک عادی هستند که در آن، اشتراک‌های مبتنی بر داده، اطلاعات اشتراک را در زمان اجرا به دست می‌آورند. اشتراک‌های مبتنی بر داده برای مدیریت موقعیت‌هایی طراحی شده‌اند که دریافت کنندگان گزارش بتوانند بین اجراهای مختلف گزارش تغییر دهند. مکانیزم‌های تحویل طبق تقاضا و مبتنی بر اشتراک، هر دو گزینه‌های خروجی گزارش مشابهی را به اشتراک می‌گذارند.

Section ۵۳،۰۴ Report Manager

Report Manager، همان‌گونه که از نامش حدس زده می‌شود، ابزار اصلی برای مدیریت راه‌حل‌های گزارشگیری Reporting Services است. Report Manager یک برنامه مبتنی بر وب ASP.NET است و با اشاره کردن مرورگر وب به http://<servername>/Reports مورد دستیابی قرار می‌گیرد. می‌توانید Report Manager را در شکل ۶-۸ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

(b)

شکل ۶-۸ Reporting Services Report Manager

اگر از Sharepoint استفاده کرده‌اید، بنابراین شکی نیست که به شباهت آشکار بین Reporting Services Report Manager و یک سایت Sharepoint توجه کرده‌اید. هر دو اشتراک، ظاهر و احساس کاملاً شبیه به هم دارند. Report Manager به شما امکان مشاهده و مدیریت تمام گزارشاتی را می‌دهد که برای Report Server توزیع کرده‌اید. می‌توانید رشته‌های اتصال و منابع داده مورد استفاده گزارشات را ویرایش کرده و خصوصیات مختلف گزارش را اصلاح کنید. علاوه بر این، Report Manager به راهبر Reporting Services امکان تنظیم امنیت و اشتراک‌ها را برای گزارشاتی می‌دهد که می‌توانند با استفاده از Reporting Services مورد دستیابی قرار گیرند.

Section ۵۳،۰۵ کلاینت گزارشگیری کاربر نهایی Report Builder

برای یک تجربه گزارشگیری کاربر نهایی بهبود یافته، Reporting Services هم‌چنین شامل Report Builder است که بر طبق فناوری ActiveViews است. مایکروسافت فناوری Report Builder را در سال ۲۰۰۴ از شرکت ActiveViews گرفت. وجود Report Builder یک گزارش مبتنی بر وب را به SQL Server 2005 می‌دهد که یک ابزار پرس‌وجوی کاربر نهایی را تألیف می‌کند. Report Builder

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

مستقیماً مشکل درخواست‌های گزارش کاربر خاتمه نیافته را با فراهم کردن ابزاری که به کاربران نهایی امکان تألیف گزارشات خاص آن‌ها را می‌دهد، حل کرده است. Report Builder برطبق چارچوب کاری NET. مایکروسافت است و از ابتدا طراحی شده است تا به‌طور کامل با Reporting Services یکپارچه شود. استفاده از ابزار گزارشگیری کاربر نهایی Report Builder بسیار آسان است؛ جزییات گزارشات می‌توانند مستقیماً در پنجره Report Builder مشاهده شوند، هنگامی که گزارش ایجاد می‌شود. بعد از تألیف گزارشات Report Builder، می‌توانند در یک مرورگر شبیه هر گزارش Reporting Services دیگر توزیع شوند.

Article LIV. تألیف گزارش

در اولین بخش این فصل، مطالبی درباره اجزای مختلفی آموختید که SQL Server 2005 Reporting Services را تشکیل می‌دهند. در نیمه دوم این فصل، نگاهی مفصل‌تر به مراحل مورد نیاز برای طراحی و توزیع یک گزارش ساده می‌اندازیم.

Section ۵۴.۰۱ مراحل ساخت

فرآیند ساخت یک برنامه Reporting Services با استفاده از Report Designer برای تعریف چیدمان و منابع داده گزارش شروع می‌شود. سپس، باید گزارش را برای Report Server بسازید و توزیع کنید. بالاخره، باید گزارش را با تعبیه کردن گزارش در یک برنامه یا افزودن اشتراک‌ها به گزارش، در دسترس کاربران نهایی قرار دهید.

طراحی راه‌حل گزارشگیری

در ایجاد راه‌حل‌های گزارشگیری با استفاده از Reporting Services، ابتدا مجموعه داده‌ای را انتخاب کنید که داده‌ای را تعریف کند که در گزارش استفاده خواهد شد و سپس فیلدهای داده مجزا را روی گزارش بچینید. برای مدیریت گزارشات سهامی که به یک شکل جدولی یا ماتریسی ارایه می‌شوند، مایکروسافت یک Report Design Wizard را فراهم کرده است که شما را در فرآیند ایجاد یک منبع داده و طراحی گزارش به صورت گام به گام هدایت می‌کند.

ساخت و توزیع راه‌حل گزارشگیری

هنگامی که گزارشی طراحی شد، باید گزارش را ساخته و سپس آن را به Report Server توزیع کنید. ساخت گزارش موجب ایجاد یک اسمبلی NET. می‌شود که گزارش را اجرا خواهد کرد. توزیع

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

گزارش، لزوماً این گزارش را گرفته و آن را در Reporting Services Report Server کپی می‌کند. در حالی که می‌توانید این کار را به صورت دستی انجام دهید، Report Designer دارای گزینه‌های تعبیه شده‌ای برای ساخت و توزیع گزارشات برای Report Server است.

در دسترس ساختن گزارش برای کاربر نهایی

بعد از توزیع گزارش به Report Server، سپس می‌توانید گزارش را از طریق چندین مکانیزم مختلف، در دسترس کاربر نهایی قرار دهید. می‌توانید دستیابی به گزارشات را با تعبیه آن‌ها در یک برنامه، از طریق URLها یا با ایجاد اشتراکی که گزارش را به کاربر نهایی تحویل می‌دهد، میسر سازید. اشتراک گزارشات می‌تواند برای تحویل در زمان خاصی تنظیم شود، یا می‌توانند مبتنی بر داده باشند.

حال که مروری بر فرآیند ساخت داشتید، بخش بعد شما را از طریق مراحل ساخت و توزیع یک گزارش ساده با استفاده از SQL Server 2005 Reporting Services هدایت می‌کند.

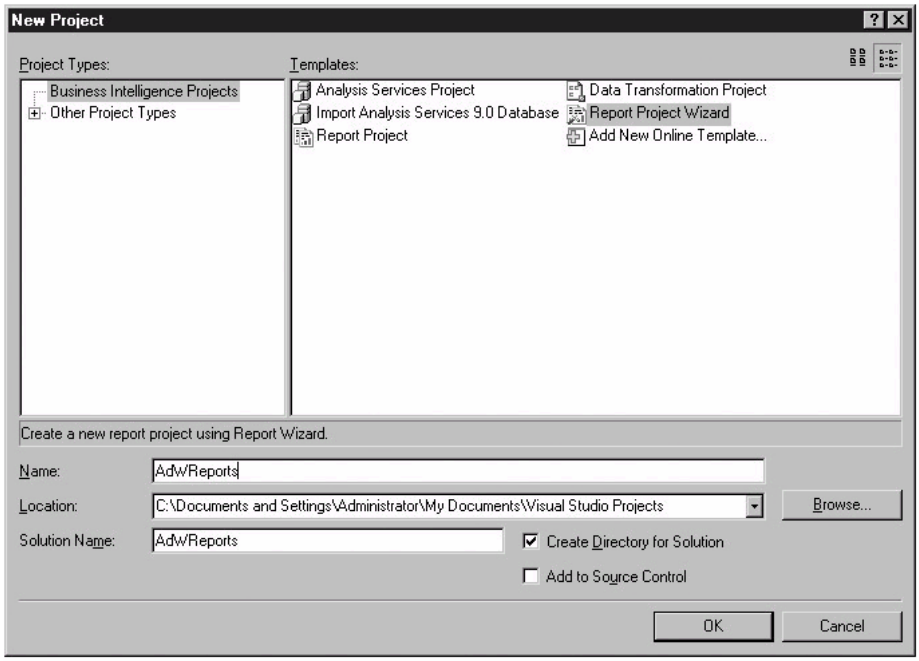
Section ۵۴،۰۲ ایجاد یک گزارش Reporting Services

می‌توانید استفاده از Report Designer را با شروع Report Wizard و استفاده از آن برای ایجاد گزارش اولیه خود یا با تنظیم شروع با یک سطح طراحی خالی و سپس افزودن عناصر تعریف گزارش خاص خود آغاز کنید. در هر یک از این موارد، تعریف یک مجموعه داده اولین کاری است که برای ایجاد یک گزارش باید انجام دهید. در این مثال، نحوه ساخت سریع یک گزارش را با استفاده از Report Wizard به شما نشان می‌دهد.

برای ساخت یک برنامه Reporting Services، ابتدا Business Intelligence را باز کرده و سپس File | New | Project را برای نمایش کادر محاوره‌ای New Project انتخاب کنید که در شکل ۷-۸ نشان داده شده است.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

شکل ۷-۸ ایجاد گزارشی جدید: New Project (b)

برای ایجاد یک گزارش Reporting Services جدید با استفاده از Report Wizard، ابتدا گزینه Business Intelligence Projects را از لیست Project Types انتخاب کنید. سپس، در لیست Templates که در سمت راست این صفحه نشان داده شده است، گزینه Report Project Wizard را انتخاب کنید. سپس کادرهای پایین این کادر محاوره‌ای را پر کنید. در کادر متنی Name، نام پروژه جاری را وارد کنید. کادر Location محل فایل‌های منبع پروژه گزارش را مشخص کنید. کادر Solution Name به شما اجازه می‌دهد تا راه‌حل Reporting Services را نام‌گذاری کنید. در این‌جا، می‌توانید ببینید که مقدار AdWReports به عنوان نام پروژه و راه‌حل استفاده می‌شود. کلیک کردن OK موجب شروع کادر محاوره‌ای Report Wizard Welcome می‌شود که در شکل ۸-۸ نشان داده شده است.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

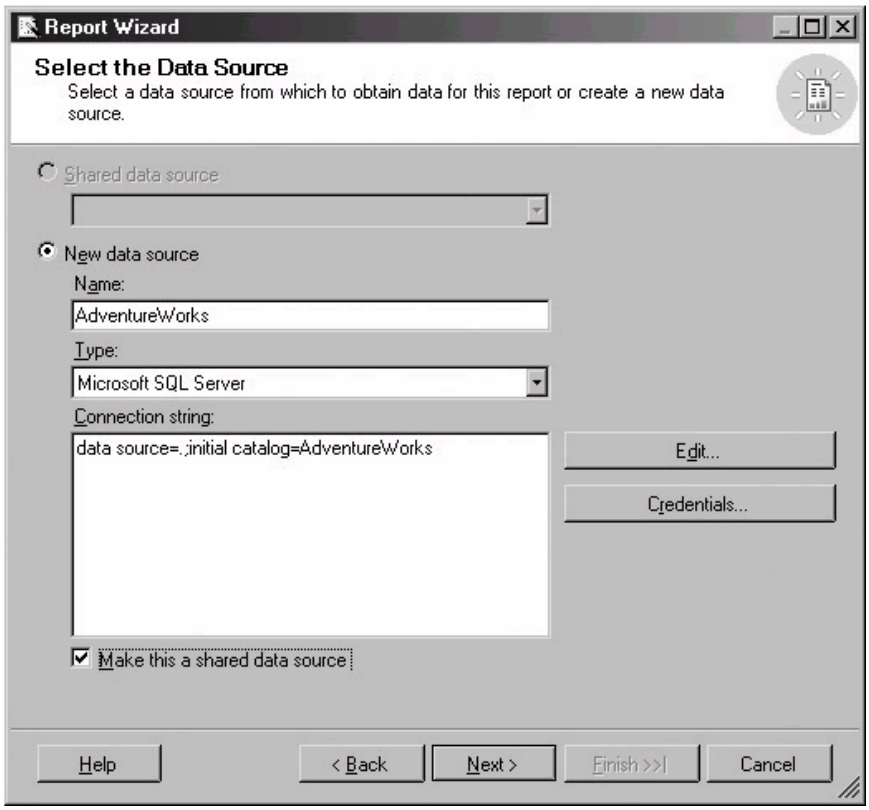
(c)

شکل ۸-۸ کادر محاوره‌ای Report Wizard Welcome (d)

کادر محاوره‌ای Report Wizard Welcome، مروری از مراحلی را به شما می‌دهد که Report Wizard در طی ایجاد پروژه طی می‌کند. ابتدا منبع داده‌ای را انتخاب کنید، سپس پرس‌وجویی را طراحی می‌کنید، سپس نوع گزارش مورد نظر خود را انتخاب کنید و بالاخره فرمت‌بندی را برای گزارش مشخص کنید. کلیک کردن Next موجب نمایش کادر محاوره‌ای Select The Data Source می‌شود که می‌توانید آن را در شکل ۹-۸ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(e)

شکل ۹-۸ Select the Data Source (f)

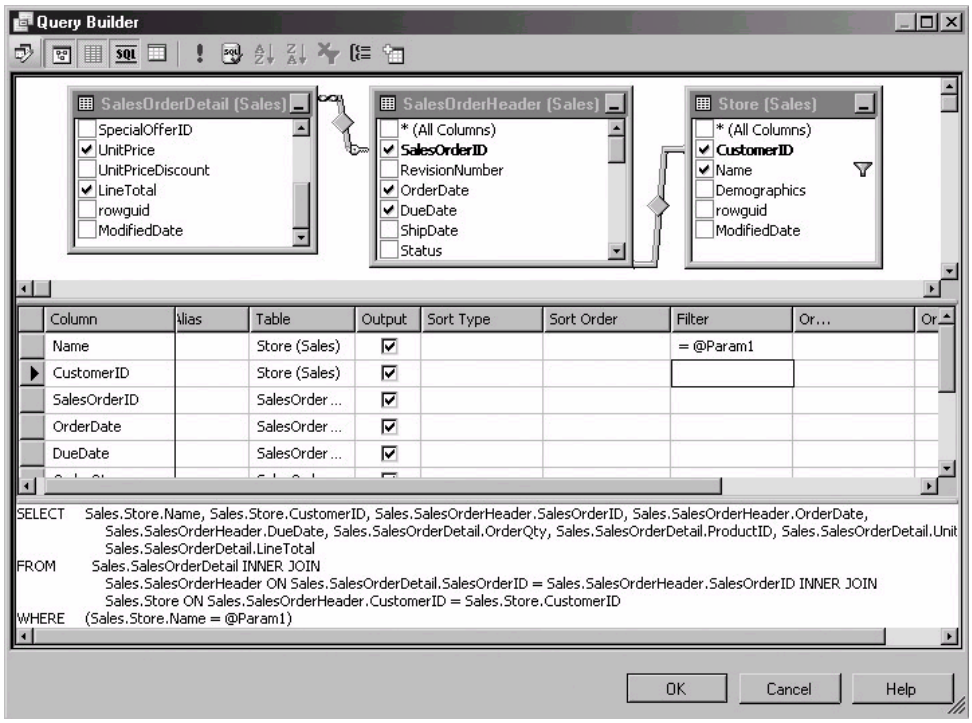
کادر محاوره‌ای Select the Data Source به شما اجازه تعریف کردن اتصال به پایگاه داده را می‌دهد. برای ایجاد Data Source، ابتدا نامی به آن بدهید. نام می‌تواند هر چیزی باشد. این نام برای تعیین منبع داده به کار می‌رود. سپس، از کادر بازشوی Type برای انتخاب نوع سیستم پایگاه داده مورد استفاده منبع داده استفاده کنید. مقدار پیش‌فرض، Microsoft SQL Server است، ولی هم‌چنین می‌توانید OLE DB، Microsoct SQL Server Analysis Services، Oracle یا ODBC باشد. سپس، در کادر Connection String، رشته اتصالی را وارد کنید که برای اتصال به پایگاه داده مقصد مورد نیاز است. اگر با مقادیر رشته اتصال آشنا نیستید، می‌توانید Edit را برای نمایش کادر محاوره‌ای Data Link کلیک کنید که شما را در مراحل ایجاد Data Source راهنمایی می‌کند. سپس، با استفاده از کادر انتخابی که در پایین صفحه نمایش است و به شما اجازه می‌دهد تا منبع داده را با گزارشات دیگر به اشتراک گذارید، آن را یک منبع داده مشترک کنید. می‌توانید یک منبع داده مشترک هم ایجاد کنید

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

که می‌تواند گزارشات مختلف مورد استفاده قرار گیرد یا منبع داده‌ای را ایجاد کنید که تنها توسط گزارشی استفاده خواهد شد که به‌طور صحیح ایجاد کرده‌اید. اگر قصد دارید چندین گزارش را ایجاد کنید که همگی از پایگاه داده یکسانی می‌آیند، ایجاد یک منبع داده مشترک، ایده خوبی است، زیرا می‌تواند به راحتی توسط تمام گزارشات در راه‌حل شما استفاده شده و آن را برای ایجاد یک منبع داده منحصر به فرد برای هر گزارش غیرضروری می‌کند. کلیک کردن Next موجب نمایش کادر محاوره‌ای Design the Query می‌شود.

از کادر محاوره‌ای Design the Query، می‌توانید به‌طور دستی یک عبارت SQL را وارد کنید که مجموعه داده مورد استفاده گزارش را تعریف خواهد کرد یا در غیر این صورت می‌توانید دکمه Query Builder را برای نمایش Query Builder نشان داده شده در شکل ۱۰-۸ کلیک کنید.



(g)

شکل ۱۰-۸ Query Builder (h)

Query Builder یک ابزار طراحی پرس‌وجوی محاوره‌ای است که به شما امکان ساخت پرس‌وجوهای SQL را بدون نیاز به یک کارشناس SQL می‌دهد. هرچند، برای کاربرد مؤثر ابزار Query Design باز هم باید دانش پایه خوبی از طراحی و الگوی پایگاه داده داشته باشید. می‌توانید جداولی را

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

از پایگاه داده خود با کلیک راست روی قسمت بالای Query Builder و سپس انتخاب Add Table از منوی بازشو برای نمایش کادر محاوره‌ای Add Tables انتخاب کنید. می‌توانید یک یا چند جدول را انتخاب کنید (چندین جدول با نگه داشتن کلید CTRL و کلیک کردن روی جدول مناسب، انتخاب می‌شوند). Query Builder به‌طور خودکار روابط بین جداول را برطبق مواردی نظیر نام ستون‌ها و انواع داده تشخیص داده و لینک‌های بین جداول را به‌طور ویژوال با نشان دادن روابط رسم خواهد کرد.

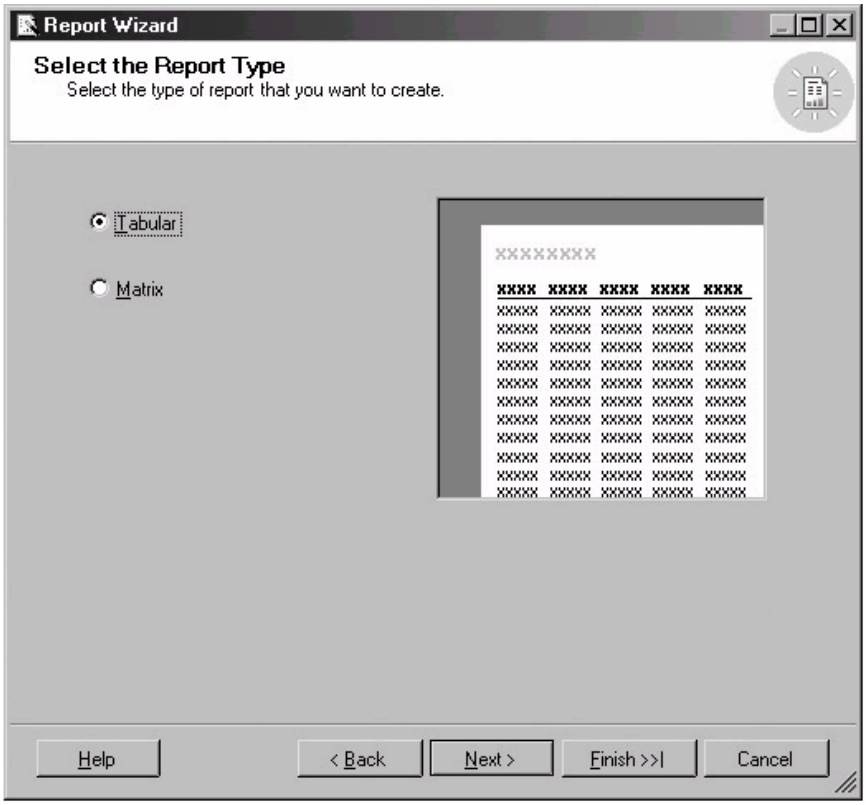
بعد از انتخاب جداول، ستون‌های مناسب را از هر جدول با قرار دادن یک علامت تأیید در کادر انتخاب قبل از نام ستون، انتخاب کنید. همان‌گونه که ممکن است حدس زده باشید، تأیید ورودی (All Columns) *، به‌طور خودکار تمام ستون‌ها را از جدول انتخاب می‌کند. هنگامی که به‌طور محاوره‌ای جداول و ستون‌ها را انتخاب کرده و روابط بین جداول را تعریف می‌کنید، Query Designer به‌طور خودکار عبارت SQL را می‌سازد و می‌توانید آن را در پایین شکل ۱۰-۸ ببینید.

چشم‌پوشی کردن از این واقعیت که Query Builder هم‌چنین می‌تواند برای ساخت پرس‌وجوهای پارامتری استفاده شود، آسان است که کاربر نهایی مقداری را برای پرس‌وجو در زمان اجرا تأمین می‌کند. برای ساخت یک پرس‌وجوی پارامتری با استفاده از Query Builder، یک علامت سؤال را در ستون Filterای تایپ کنید که در ردیفی از نام ستون پایگاه داده است که باید با یک پارامتر استفاده شود. Query Builder به‌طور خودکار کاراکتر علامت سؤال را به مقدار @Param= تبدیل می‌کند و می‌توانید آن را در وسط شکل ۱۰-۸ ببینید.

می‌توانید پرس‌وجو را با کلیک کردن آیکن تعجب (!) در نوار ابزار امتحان کنید. بعد از تکمیل طراحی پرس‌وجو، می‌توانید پرس‌وجو را ذخیره کرده و با کلیک کردن OK ادامه دهید. عبارت SQL ای که توسط Query Builder ایجاد شده بود، در کادر محاوره‌ای Design The Query نوشته خواهد شد. کلیک کردن Next موجب نمایش کادر محاوره‌ای Select The Report Type می‌شود که می‌توانید آن را در شکل ۱۱-۸ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(i)

شکل ۱۱-۸ Select The Report Type (j)

در حالی که Report Designer به شما اجازه می‌دهد تا مقدار شگفت‌انگیزی انعطاف‌پذیری را در طراحی گزارشات داشته باشید، Report Wizard در نوع گزارشاتی که برای شما خواهد ساخت، محدودتر است. Report Wizard یکی از دو نوع گزارش متفاوت را تولید خواهد کرد: یک گزارش شیوه جدولی یا یک گزارش شیوه ماتریسی.

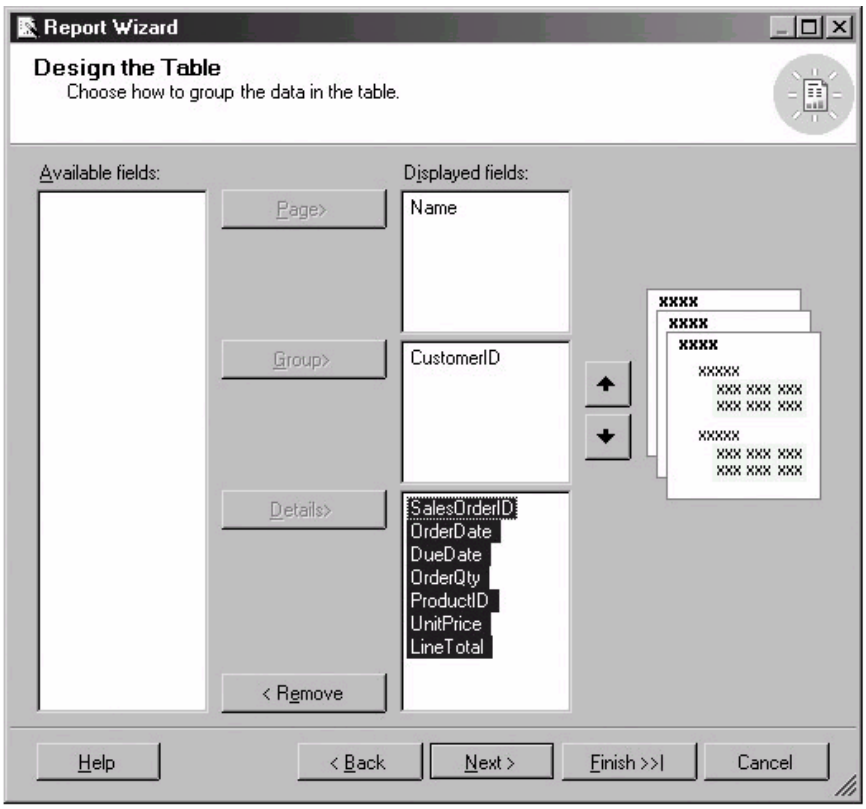
تـــوجـه : در حالی که شیوه‌های گزارشات محدود هستند، Report Wizard نقطه شروع خوبی برای ساخت یک گزارش پایه است که بعداً می‌توانید آن را در Report Designer اختصاصی کنید.

گزارش جدولی از چیدمان طراحی متداول شما پیروی می‌کند که هدرها در بالای صفحه و اطلاعات تفصیلی در زیر در بدنه گزارش لیست شده‌اند. شیوه گزارش ماتریسی یک گزارش شیوه

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

crosstab را ارایه می‌کند که هدرها در سراسر بالای صفحه و پایین سمت چپ صفحه قرار دارند. در شکل ۱۰-۸، می‌توانید ببینید که شیوه جدولی گزارش انتخاب شده است. اگر در پایین این کادر محاوره‌ای به دقت نگاه کنید، دکمه Finish را خواهید دید. کلیک کردن Finish به شما اجازه می‌دهد تا به سرعت فرمت‌بندی گزارش را با انتخاب تمام مقادیر پیش‌فرض کامل کنید. کلیک کردن Next موجب نمایش کادر محاوره‌ای Design The Table می‌شود که در شکل ۱۲-۸ نشان داده شده است.



(k)

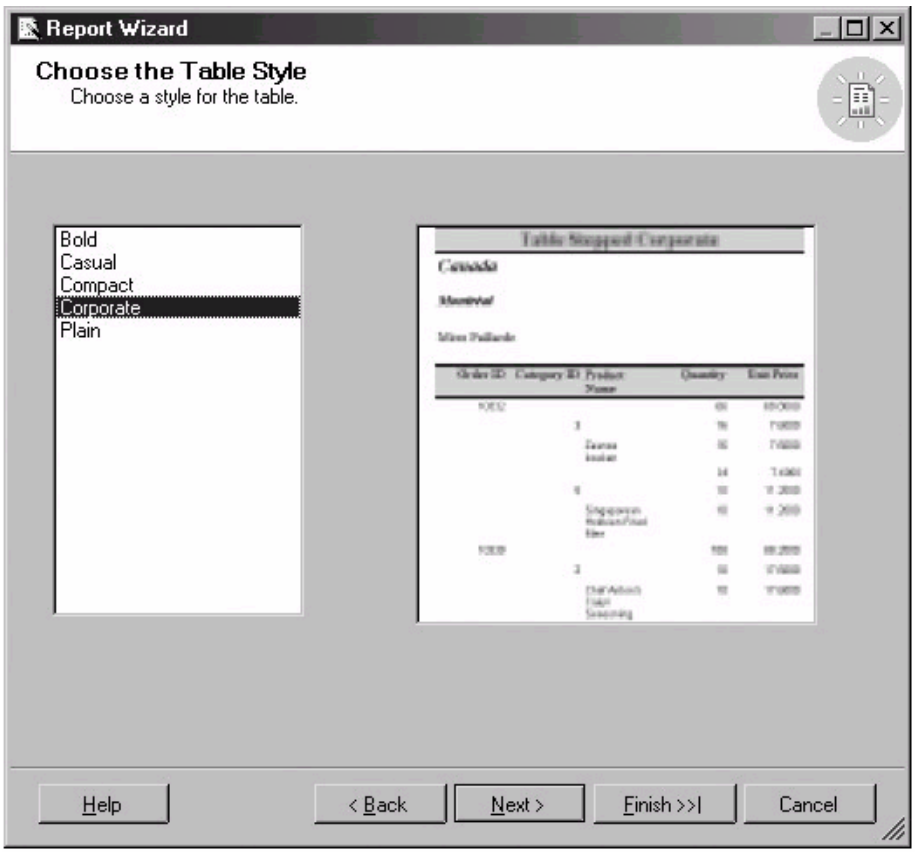
شکل ۱۲-۸ طراحی جدول (l)

در ابتدا، تمام ستون‌های موجود از پرس‌وجو در لیست Available Fields نشان داده می‌شوند که می‌توانید آن را در سمت چپ شکل ۱۲-۸ ببینید. از آن لیست فیلدهای موجود، می‌توانید فیلدها را به‌طور انتخابی به ناحیه‌ای از گزارش بکشید که می‌خواهید ظاهر شوند. اگر بخواهید فیلدی در هدر صفحه نمایش داده شود، آن را به بخش Page بکشید که در سمت راست بالای صفحه نمایش نشان

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

داده شده است. اگر بخواهید فیلدی برای جمع کل گروه‌ها استفاده شود، آن را به بخش Group بکشید. و چنانچه بخواهید فیلد یک فیلد داده باشد، آن را بخش Details بکشید. در شکل ۱۲-۸، می‌توانید ببینید که فیلد Name به عنوان هدر صفحه استفاده می‌شود؛ فیلد CustomerID به عنوان هدر گروه استفاده می‌گردد و فیلدهای SalesOrderID، OrderDate، DueDate، OrderQty، ProductID، UnitPrice و LineTotal در ناحیه جزییات استفاده می‌شوند. هنگامی که فیلدهای مورد استفاده در گزارش را طراحی کردید، کلیک کردن Next موجب نمایش کادر محاوره‌ای Report Wizard بعدی می‌شود که در شکل ۱۳-۸ نشان داده شده است.



(m)

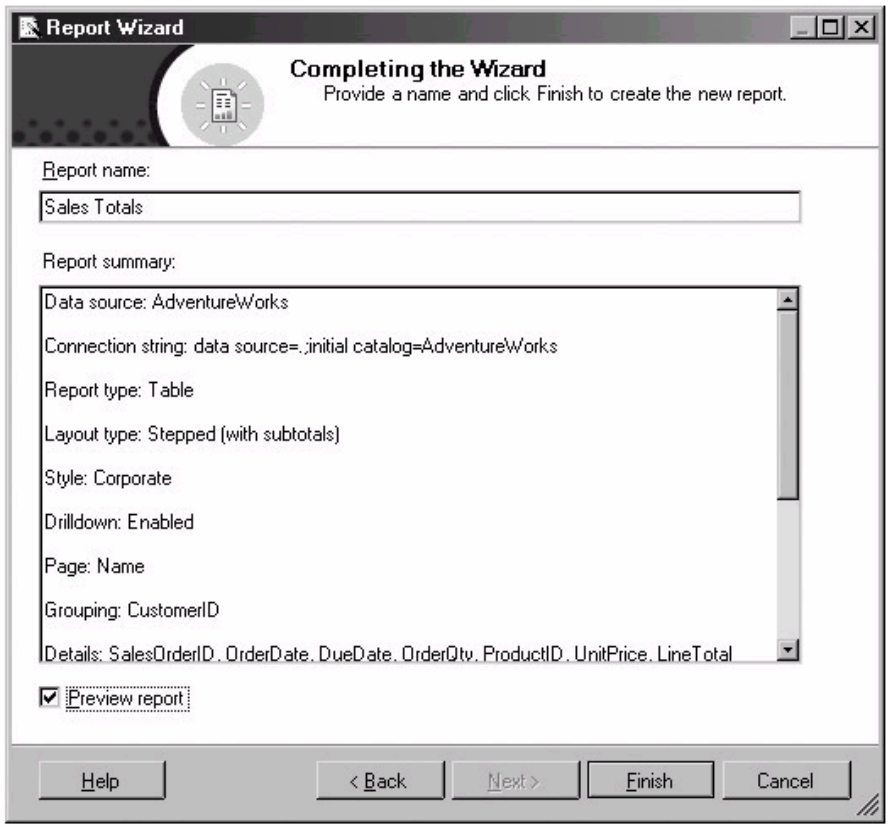
شکل ۱۳-۸ Choose The Table Style (n)

کادر محاوره‌ای Choose The Table Style به شما اجازه می‌دهد تا طراحی کلی گزارشی را که توسط Reporting Services تولید خواهد شد، انتخاب کنید. شیوه‌های جدولی متفاوت مشابه هستند،

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

ولی هر شیوه از الگوی رنگ متفاوت و فرمت‌بندی کاملاً مختلفی استفاده می‌کند. در شکل ۱۳-۸، می‌توانید ببینید که شیوه Corporate انتخاب شده است. کادر محاوره‌ای Choose The Table Style کادر محاوره‌ای ایجاد گزارش نهایی است که توسط Report Wizard نمایش داده می‌شود. کلیک کردن Next موجب نمایش صفحه Completing The Wizard می‌شود که در شکل ۱۴-۸ نشان داده شده است.



(o)

شکل ۱۴-۸ Completing The Wizard (p)

کادر محاوره‌ای Completing The Wizard به شما اجازه می‌دهد تا تمام انتخاب‌هایی را که در کادرهای محاوره‌ای ویزارد قبلی صورت گرفته‌اند، مرور کنید. در این‌جا، می‌توانید از دکمه Back برای برگشت به صفحه قبل و انجام تصحیحاتی در مشخصه‌های گزارش استفاده کنید یا می‌توانید Finish را برای تولید گزارش کلیک کنید. تأیید کادر انتخاب Preview موجب نمایش گزارش برای شما برای

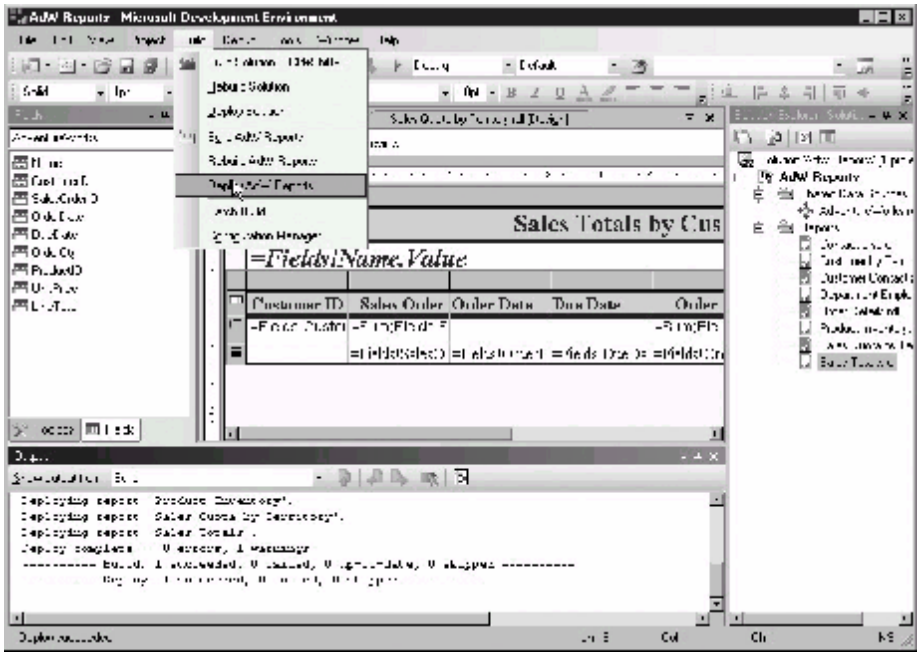
www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

مشاهده در پنجره Report Designer Preview می‌شود. بعد از تولید، گزارش به راه‌حل Reporting Services نشان داده شده در Business Intelligence Design Surface اضافه می‌شود.

Section ۵۴،۰۳ توزیع یک گزارش Reporting Services

بعد از ایجاد گزارش، مرحله بعدی در ایجاد یک برنامه Reporting Services، ساخت گزارش و توزیع آن به Report Server است. ساخت گزارش موجب ایجاد یک اسمبلی NET DLL. می‌شود و توزیع گزارش موجب کپی شدن آن اسمبلی در Reporting Services Report Server می‌شود. می‌توانید راه‌حل‌های گزارشگیری را از Report Designer با انتخاب گزینه Build/Deploy Reports توزیع کنید که در شکل ۱۵-۸ مشاهده می‌کنید.



(a)

(b)

شکل ۱۵-۸ توزیع راه‌حل Reporting Services

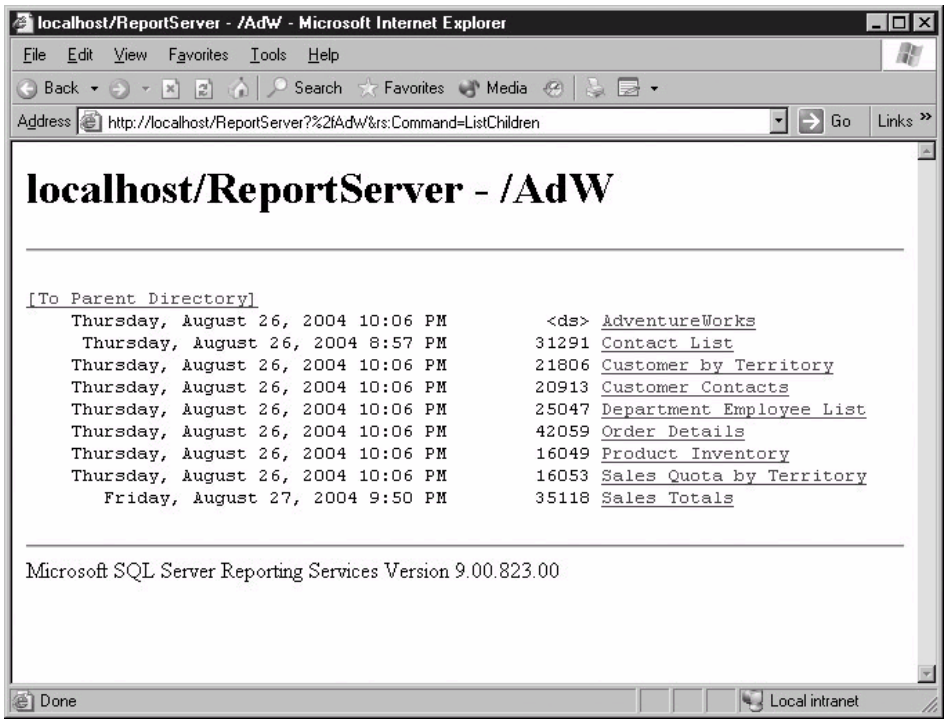
اگر یکی از گزینه‌های توزیع را انتخاب کنید و گزارش را ذخیره کرده باشید، Report Designer به‌طور خودکار گزارش را می‌سازد، قبل از این که آن را توزیع کنید. خروجی فرآیندهای ساخت و توزیع در پنجره Output نشان داده شده است که می‌توانید در پایین شکل ۱۵-۸ ببینید. هر خطا یا مشکلی در این پنجره لیست می‌شود. ضمناً، اگر توزیع گزارش موفق باشد، آن‌گاه پیام موفقیتی در لیست Output لیست می‌شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۵۴،۰۴ اجرای یک گزارش Reporting Services

گزارشات Reporting Services می‌توانند با دستیابی به URL آن‌ها یا با تعبیه آن‌ها در برنامه‌ها اجرا شوند. می‌توانید با اشاره مرورگر خود به http://<servername>/ReportServer URL، به گزارشات Reporting Services خود دستیابی داشته و آن‌ها را اجرا کنید، که تمام گزارشات Reporting Services در آنجا لیست شده‌اند. شکل ۱۶-۸ صفحه وب ReportServer را نشان می‌دهد.



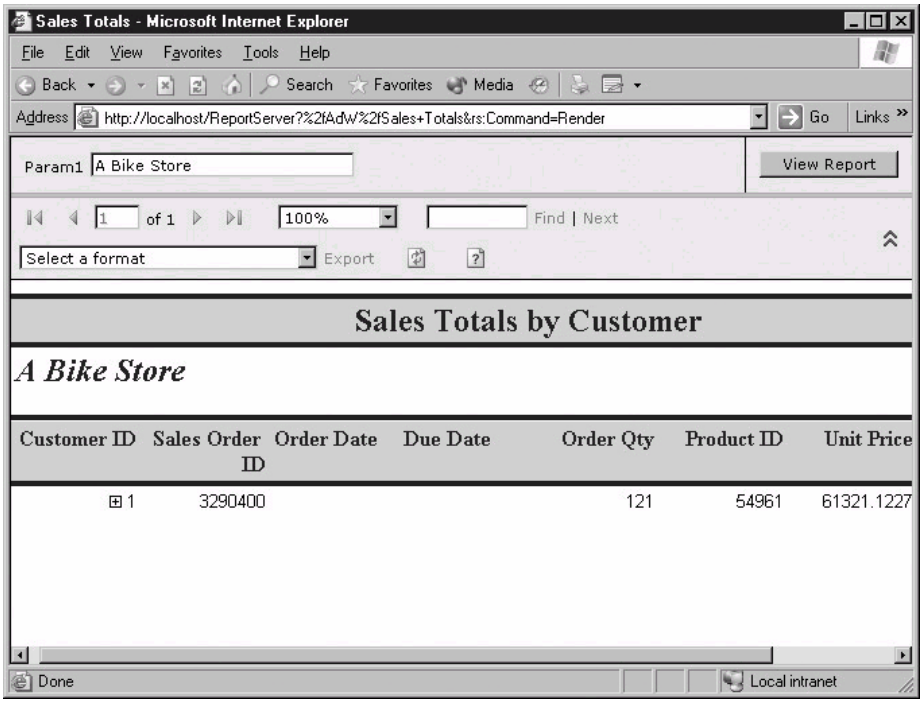
(a)

شکل ۱۶-۸ دستیابی به گزارشات Reporting Services از یک URL (b)

ReportServer URL تمام گزارشاتی را که به Report Server توزیع شده‌اند، لیست می‌کند. هر راه‌حل متفاوت در زیر فهرست خاص خودش ذخیره می‌شود. برای امتحان گزارشاتی که توزیع شده‌اند، روی لینک کلیک کنید تا Report Server گزارش را در مرورگر نمایش دهد. شکل ۱۷-۸ گزارش نمونه‌ای را در مرورگر نشان می‌دهد.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(c)

شکل ۱۷-۸ اجرای گزارشات Reporting Services (d)

گزارشی که در مرورگر نمایش داده می‌شود، از فرمتی پیروی می‌کند که در فاز طراحی گزارش تنظیم شده است و با توجه به این که این گزارش از پارامترها استفاده می‌کند، فیلد param1 در بالای صفحه نمایش نشان داده می‌شود. کاربر نهایی باید مقداری را در این فیلد وارد کند و سپس دکمه View Report را برای نمایش گزارش کلیک کنید. در شکل ۱۷-۸، می‌توانید ببینید که مقدار A Bike Store برای پارامتری قابل جایگزینی وارد شده است.

علاوه بر این، به دلیل این که این گزارش با استفاده از گزینه‌ای عمقی تولید شده است، یک علامت بعلاوه در جلوی خط جزییات نشان داده شده در گزارش نمایش داده می‌شود. کلیک کردن علامت بعلاوه (+) موجب نمایش خطوط جزییات ردیفی می‌شود که به آن خط خلاصه سطح بالا منتهی می‌شود. در این مثال، کلیک کردن علامت بعلاوه (+) موجب باز شدن نمایش می‌شود، همان‌گونه که در شکل ۱۸-۸ نشان داده شده است.

در شکل ۱۸-۸، می‌توانید جزییاتی را ببینید که بر خط خلاصه نشان داده شده در شکل قبل منتهی می‌شوند. کلیک کردن علامت تفریق (-) موجب کاهش نمایش جزییات و نشان دادن خط خلاصه می‌شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

اجرای مستقیم گزارشات از Reporting Services URL برای امتحان بسیار عالی است، ولی هنگامی که برنامه شما زنده است، باید URL گزارش را در برنامه خود تعبیه کنید یا از طریق فراخوانی سرویس‌های وب به Report Server دستیابی داشته باشید.

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

فصل نهم

Integration Services

DTS[1] یکی از مفیدترین و محبوب‌ترین برنامه‌های SQL Server از زمان معرفی آن در SQL Server 7 بوده است. DTS در اصل ابزاری است که برای انجام اقتباس داده، تبدیل و بارگذاری (ETL) برای انبارهای داده OLAP Services استفاده می‌شود. هرچند، مایکروسافت سریعاً مفید بودن آن را تشخیص داد و DTS را ابزار اصلی برای وارد و صادر کردن داده از پایگاه‌های داده رابطه‌ای SQL Server و یک ابزار BI ETL کرد. علاوه بر سهولت استفاده از ابزار bcp[2] خط فرمان قدیمی که جایگزین

1- Data Transformation Services
2- Bulk Copy Program

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

شده است، DTS گام بزرگی ماورای یک برنامه انتقال داده است که این کار با فراهم کردن توانایی انتقال داده هنگام انتقال‌های داده ساده طراحی شده است و یک طراحی گرافیکی را ارایه می‌کند که برای عملیات انتقال داده پیچیده‌تر و تبدیل است. نگارش‌های DTS از SQL Server 7 و 2000، از ۱۰۰ درصد OLE DB برای پایگاه داده منبع و مقصد پشتیبانی می‌کنند. این بدان معنی است که هرچند DTS بخشی از SQL Server بود، واقعاً می‌توانست برای انتقال داده بین هر دو منبع داده OLE DB استفاده شود، بدون این که نیاز باشد SQL Server منبع یا مقصد داده باشد. برای نمونه، علاوه بر توانایی وارد و صادر کردن داده از پایگاه‌های داده SQL Server، DTS هم‌چنین می‌توانست برای انتقال داده بین سایر سیستم‌های پایگاه داده از قبیل Access، Oracle و DB2 بدون دخالت SQL Server استفاده شود. این نوع انعطاف‌پذیری موجب قوی‌تر و مفیدتر شدن DTS به عنوان ابزار انتقال داده می‌شود.

هرچند، DTS در حد امکان مفید و جالب است، باز هم محدودیت‌های نسبتاً مهمی دارد. DTS برای این که آماده عرضه جهانی شود، نیاز به مقیاس‌پذیری بهتری دارد. علاوه بر این، بسته‌های DTS به سادگی بین سیستم‌ها قابل حمل نیستند. به عبارت دیگر، یک بسته DTS که برای انجام انتقالی از سیستم SQL Server A طراحی شده است، به سادگی نمی‌تواند برای انجام همان انتقال از سیستم B مجدداً استفاده شود. علاوه بر این محدودیت‌های توزیع، نگارش‌های قبلی DTS فاقد سیستم Login و مدیریت خطای قوی بوده و قابلیت مدیریت محدودتری داشتند.

در SQL Server 2005، مایکروسافت DTS را اصلاح کرده و آن را از زمینه بازنویسی کرده است. طبق تمام طبیعت جدید آن، مایکروسافت نام DTS را به Integration Services تغییر داده است. هدف مایکروسافت از SQL Server 2005 Integration Services، ساخت آن به عنوان یک محیط ETL جهانی برای Windows برابر با هر محصول BI ETL سطح جهانی مستقل است. برای این مسائل، مایکروسافت Integration Services کاملاً جدید را با استفاده از کد NET. مدیریت شده نوشته است و پایه مستحکم‌تری به آن بخشیده است. در این فرآیند، مایکروسافت Integration Services را کاملاً از اول طراحی کرده است، معماری کاملاً جدیدی به آن داده و پشتیبانی بهتری برای قابلیت برنامه‌نویسی و کارآیی زمان اجرای بهبود یافته فراهم کرده است. Integration Services جدید، دارای ویژگی‌هایی نظیر یک طراح گرافیکی جدید و یک انتخاب بهبود یافته مهم از وظایف انتقال داده و تبدیلات است. Integration Services شبیه DTS قدیمی، از ۱۰۰ درصد منبع و مقصد پشتیبانی می‌کند، بدین معنی که می‌تواند به‌طور مستقل به هر منبع داده منبع و مقصد متصل شود، بدون نیاز به این که منبع داده، یک سیستم SQL Server باشد. در این فصل، مطالبی درباره این ویژگی‌های جدید در SQL Server 2005 Integration Services می‌آموزید. ابتدا، نگاهی به درون معماری Integration Services جدید خواهید داشت و سپس توری راهنما از اجزا و ابزارهای Integration Services جدید داریم.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

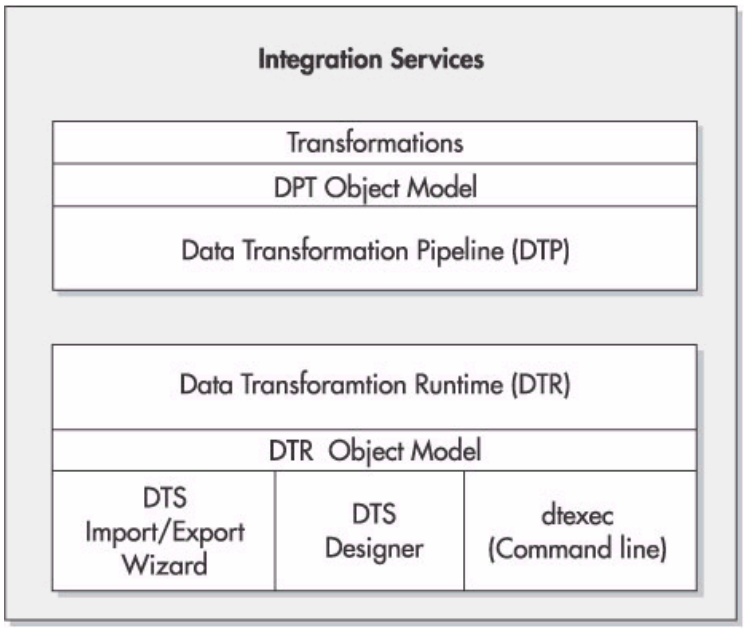
Article LV. معماری Integration Services جدید

معماری Integration Services جدید به دو بخش اصلی تقسیم می‌شود: DTP[1] و DTR[2]. مایکروسافت Integration Services را به دو بخش اصلی تقسیم کرده است، به ویژه به این دلیل که حایل واضحی را بین جریان داده و جریان کاری به وجود آورد. در نگارش‌های قبلی DTS، موتور جریان داده قوی‌تر از قابلیت‌های جریان کاری بود. این تقسیم‌بندی جدید لزوماً بخش جریان کاری Integration Services را یک جزء کلاس بالا در همان سطح جزء جریان داده ساخته است. DTP جدید لزوماً جایگزین Data Pump DTS قدیمی شده است که در نگارش‌های SQL Server 7 و 2000 از DTS استفاده می‌شوند. کار اصلی آن، مدیریت جریان داده بین منبع و مقصد است. DTR اصولاً یک محیط اجرای کاری است که جریان کاری مورد استفاده در یک بسته Integration Services را کنترل می‌کند. هر یک از این اجزا با استفاده از DLL خاص خود و مدل شیئی مجزای خود پیاده‌سازی می‌شوند. در شکل ۱-۹، می‌توانید مروری از معماری Integration Services جدید را ببینید.

1- Data Transformation Pipeline
2- Data Transformation Runtime

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

(b)

شکل ۱-۹ معماری Integration Services

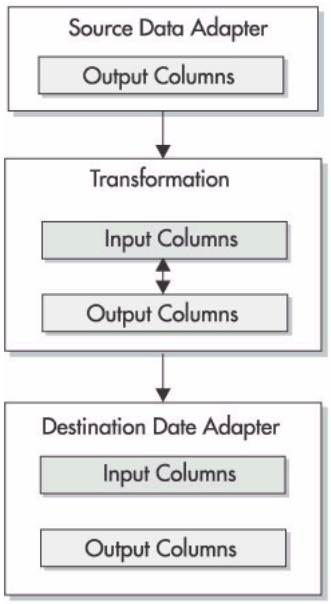
DTP و Integration Services DTR جدید را در بخش‌های بعد به‌طور مفصل تشریح می‌کنیم. برای کسب اطلاعات بیشتر درباره این مجموعه ابزار Integration Services جدید، در ادامه فصل مطالبی ارایه می‌شود.

Section ۵۵،۰۲ DTP

DTP به تبدیلات و جریان داده‌ای توجه دارد که هنگام انتقال ردیف‌ها بین منبع و مقصد داده انجام می‌شوند. DTP از آداپتورهای داده برای اتصال به منابع و مقصد استفاده می‌کند. همان‌گونه که در شکل ۱-۹ می‌بینید، موتور DTP با استفاده از مدل شیئی DTP Pipeline مورد دستیابی قرار می‌گیرد. این مدل شیئی، APIای است که توسط تبدیلات تعبیه شده و تأمین شده توسط مایکروسافت و تبدیلات اختصاصی ایجاد شده کاربر استفاده می‌شوند. تبدیلات داده، ردیف را هنگامی انتقال داده و دستکاری می‌کنند که داده از ستون‌های منبع به ستون‌های مقصد منتقل می‌شوند. می‌توانید در شکل ۲-۹، معماری DTP جدید را به‌طور مفصل‌تر بررسی کنید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

(b)

شکل ۲-۹ اجزای DTP

DTP از آداپتورهای داده برای اتصال نقاط انتهایی منبع و مقصد داده استفاده می‌کند. همان‌گونه که از نام آن‌ها مشخص است، آداپتورهای داده منبع داده متصل شده و ورودی برای بسته‌های Integration Services را فراهم می‌کنند. آداپتورهای داده مقصد، به مقصد داده متصل شده و داده را به خروجی ارسال می‌کنند. SQL Server 2005 تعدادی آداپتور داده منبع و مقصد را فراهم کرده است. SQL Server 2005 Integration Services همراه با تعدادی آداپتور منبع و مقصد تعبیه شده است، از جمله آداپتورهایی برای SQL Server، فایل‌های flat و سایر منابع داده سازگار با OLE DB. برای کسب اطلاعات بیشتر درباره آداپتورهای داده خاصی که پشتیبانی می‌شوند، به بخش بعدی این فصل مراجعه کنید.

در حالی که کار آداپتورهای داده، ساخت اتصالاتی به نقاط انتهایی منبع و مقصد داده است، کار تبدیلات Integration Services انتقال و احتمالاً دستکاری داده هنگام انتقال بین نقاط انتهایی منبع و مقصد است. تبدیل Integration Services می‌تواند به سادگی یک نگاشت یک به یک بین ستون‌های منبع و مقصد، ایجاد ستون‌های مقصد جدید با استفاده از نگاشت‌های یک به چند یا محاسبه ستون‌های مشتق شده را انجام دهد. SQL Server 2005 Integration Services همراه با تعدادی

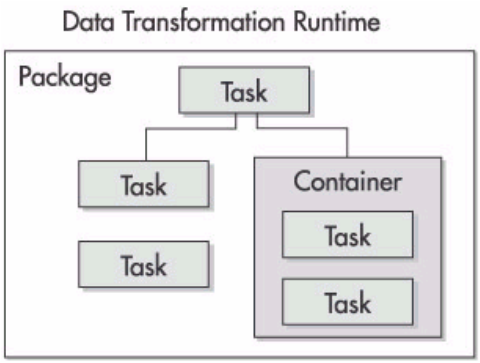
www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

تبدیلات تعبیه شده واقعی است که در ادامه این فصل بیان می‌گردند. علاوه بر این تبدیلات تعبیه شده، می‌توانید تبدیلات اختصاصی خاص خود را با بهره بردن از API مدل شیئی DTP بسازید.

Section ۵۵،۰۳ DTR

DTR شامل موتور DTR و اجزای DTE است. اجزای DTR اشیایی هستند که به شما امکان می‌دهند تا بر اجرای Integration Services نظارت داشته باشید. اجزای DTR برای ساخت جریانات کاری استفاده می‌شوند، کانتینرهایی که عملیات ساخت یافته را فراهم می‌کنند، وظایفی که انتقال داده را و عملکرد تبدیل را فراهم می‌کنند و الزاماتی که توالی یک جریان کاری را در یک بسته کنترل می‌کنند. می‌توانید مروری از معماری DTR جدید را در شکل ۳-۹ ببینید.



(a)

شکل ۳-۹ مروری بر DTR (b)

اجزای DTR اصلی، کانتینرها و وظایف هستند. وظایف کلکسیون‌هایی از اجزای DTR هستند؛ هر وظیفه مرکب از منابع و مقاصد داده و تبدیلات داده هستند. کانتینرها برای سازمان‌دهی وظایف مرتبط استفاده می‌شوند. این کانتینرها و وظایف برای تشکیل بسته‌ها، با یکدیگر گروه‌بندی می‌شوند. بسته[1] Integration Services واحدی فیزیکی است که تمام اعمالی را که در یک عمل انتقال معین انجام خواهند شد، با یکدیگر گروه‌بندی می‌کند. بسته‌ها توسط DTR برای انجام انتقال‌های داده اجرا می‌شوند. بسته‌های Integration Services می‌توانند به سادگی مجدداً اجرا شوند یا حتی به سیستمی متفاوت منتقل شده و به‌طور مستقل اجرا شوند. برای کسب اطلاعات بیشتر درباره اجزای خاصی که DTR را تشکیل می‌دهند، به ادامه فصل مراجعه کنید.

هدف اصلی موتور DTR کنترل اجرای بسته‌های Integration Services است. DTR جریان کاری وظایف موجود در یک بسته Integration Services را کنترل می‌کند. علاوه بر این، موتور DTR

1- Package

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

چیدمان بسته را ذخیره می‌کند؛ بسته‌ها را اجرا کرده و سرویس‌های اشکال‌زدایی، login کردن و مدیریت رویداد را فراهم می‌کند. موتور DTR هم‌چنین به شما امکان مدیریت اتصالات و دستیابی به متغیرهای بسته Integration Services را می‌دهد.

DTR با استفاده از چارچوب کاری شیئی DTR مورد دستیابی قرار می‌گیرد. چارچوب کاری شیئی زمان اجرای DTR یک API است که از Integration Services Import/Export Wizard و Integration Services Designer علاوه بر ابزار dtexec خط فرمان پشتیبانی می‌کند. Import/Export Wizard و Designer برای ایجاد بسته‌ها استفاده می‌شوند. موتور DTR که به C++ نوشته شده است و API آن با استفاده از یک C++ API طبیعی و یک اسمبلی NET. ارایه می‌شوند که به آن امکان می‌دهد از طریق برنامه‌های NET. مدیریت شده مورد دستیابی قرار گیرد. برنامه‌هایی که از مدل شیئی DTR استفاده می‌کنند می‌توانند ایجاد و اجرای بسته‌های Integration Services را خودکار کنند.

Article LVI. اجزای بسته Integration Services

بسته‌های Integration Services، کلکسیون‌هایی از اجزای DTR و DTP را سازمان‌دهی می‌کنند. بسته واحد اجرایی برای عمل انتقال Integration Services است. به عبارت دیگر، برای استفاده از Integration Services برای انجام انتقال‌های داده و عملیات ETL، باید بسته‌ای را ایجاد کنید که حاوی تمام اجزای DTP باشد که منبع و مقصد را برای داده و تبدیلاتی که انجام خواهند شد تعریف می‌کنند. هنگامی که بسته‌ای ایجاد می‌شود، بسته را برای انجام انتقال داده اجرا می‌کنید. می‌توانید مروری از یک بسته Integration Services نمونه را در شکل ۴-۹ ببینید.

بسته‌های Integration Services می‌توانند با استفاده از مجموعه‌ای از ابزارهای فراهم شده توسط SQL Server 2005 ایجاد شوند یا می‌توانند با برنامه‌نویسی با استفاده از DTR API ایجاد گردند. بسته‌های Integration Services می‌توانند در پایگاه داده SQL Server msdb به عنوان فایل‌های XML که در سیستم‌های فایل قرار می‌گیرند، ذخیره شوند. می‌توانید پیش‌نمایشی از ابزارهای طراحی Integration Services را در ادامه فصل ببینید.

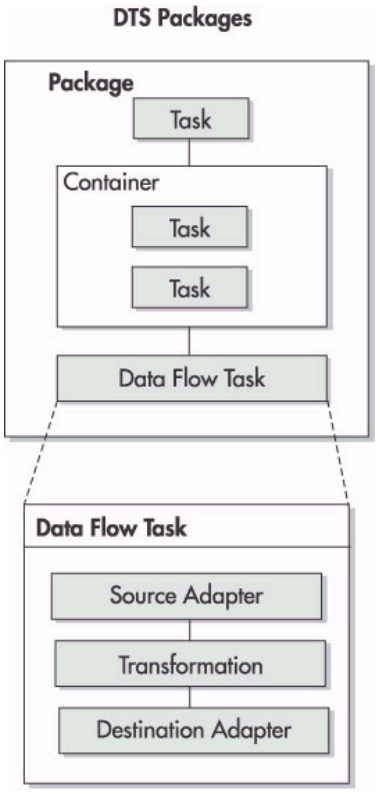
Section ۵۶،۰۱. ویژگی‌های بسته Integration Services

عجیب نیست که معماری Integration Services جدید، تعدادی قابلیت جدید مهم را به Integration Services اضافه کرده است که آن را یک ابزار تبدیل داده جهانی قوی‌تر کرده است.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

سپس، برخی از بهبودهای مهمی را بررسی می‌کنیم که این نگارش جدید از Integration Services برای جدول آورده است.



(a) (b)

شکل ۴-۹ مروری بر بسته Integration Services

تنظیم خصوصیات زمان اجرای بسته‌های Integration Services

یکی از محدودیت‌های عمده نگارش‌های Integration Services این واقعیت است که بسته‌ها لزوماً به یک منبع و مقصد معین مرتبط هستند. این بدان معنی است که داشتن یک بسته معین و استفاده مجدد آسان از آن در سروری متفاوت واقعاً مشکل بود، حتی اگر آن سرور از همان پایگاه‌های داده پشتیبانی کند. در حالی که استفاده از متغیرهای سراسری و سایر workaroundهای پیچیده برای رفع این محدودیت امکان‌پذیر بود، این workaroundها راه‌حل‌هایی قوی برای این مشکل نیستند. SQL Server 2005 Integration Services جدید به یک بسته امکان مدیریت چندین نقطه انتهای

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

منبع و مقصد را می‌دهد. می‌توانید از این ویژگی برای تنظیم خصوصیات بسته در زمان اجرا از فایل‌های پارامترها، Registry یا اسناد XML استفاده کنید.

Login کردن

Integration Services اصلی هم‌چنین از قابلیت‌های login تعبیه شده پشتیبانی می‌کند. عمل login بسته‌ها، به ویژه برای نظارت و عیب‌یابی مفید است. در حالی که می‌توانید آن‌ها را شخصاً اضافه کنید، این مسأله نیاز به کار برنامه‌نویسی در بخش شما دارد. SQL Server 2005 Integration Services جدید، گزینه‌های login کردن تعبیه شده را برای بسته‌ها، وظایف و تبدیلات را فراهم کرده است.

شروع مجدد و نقطه بررسی

ویژگی جدید مهم دیگر در SQL Server 2005 Integration Services، پشتیبانی برای نقاط بررسی و شروع مجدد در بسته‌های Integration Services است. این ویژگی جدید امکان می‌دهد نقاط بررسی متفاوت به مراحل مختلف در بسته‌های Integration Services پیچیده مرتبط شوند. بنابراین اگر بسته‌ای که از نقاط بررسی استفاده می‌کنند، ناموفق باشد، کل بسته نباید مجدداً از ابتدا اجرا شود. در عوض، این بسته می‌تواند در ابتدای نقطه بررسی ناموفق مجدداً شروع شود. برای بسته‌های با اجرای طولانی، این امر می‌تواند در زمان صرفه‌جویی زیادی کند، زیرا کل بسته نباید مجدداً پردازش شود. در عوض، بسته Integration Services می‌تواند از محلی که پردازش رها شده است، ادامه دهد. هنگامی که این مسأله با مدیر رویداد ترکیب می‌شود، می‌تواند ابزار مفیدی برای مدیریت خودکار کارهای Integration Services شما باشد.

متغیرها

پشتیبانی از متغیرها، ویژگی جدید دیگری در SQL Server 2005 Integration Services است. در SQL Server 7 و SQL Server 2000، بسته‌های DTS از متغیرهایی برای استفاده در اسکریپت‌های تبدیل پشتیبانی می‌کند، ولی از متغیرهای حوزه بسته پشتیبانی نمی‌کند. این فقدان متغیرهای سراسری، آن را برای استفاده مجدد از بسته‌های DTS در چندین پایگاه داده و سرورهای مختلف مشکل می‌کرد. برای اطمینان، راه‌حل‌هایی وجود داشت، ولی با توجه به این که بسته‌های DTS برای

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

استفاده شدن در این الگو طراحی نشده‌اند، این راه‌حل‌ها قدری پیچیده بوده و چندان قوی نیستند. در SQL Server 2005، طراحی Integration Services جدید، پشتیبانی از بسته، وظیفه و متغیرهای سطح رویداد را فراهم کرده است، انعطاف‌پذیری بسته‌های Integration Services را افزایش داده و هم‌چنین موجب تسهیل زیاد بسته‌های Integration Services شده است. همان‌گونه که از نام آن‌ها مشخص است، متغیرهای حوزه بسته می‌توانند با تمام اجزا در بسته Integration Services دیده شوند. متغیرهای حوزه وظیفه می‌توانند توسط اجزای آن وظیفه خاص دیده شده و مورد دستیابی قرار گیرند، ولی نه با هر یک از اجزای دیگری که بخشی از بسته Integration Services هستند. یک کاربرد نمونه از یک متغیر Integration Services حوزه بسته ممکن است حاوی نام سیستم SQL Server جاری باشد، به بسته اجازه می‌دهد تا در سیستم دیگری با جایگزین کردن آن نام سرور سیستم در متغیرهای Integration Services در زمان اجرا به سادگی اجرا شوند.

متغیرهای سیستم

تمام بسته‌های Integration Services دارای تعدادی متغیر سیستمی تعبیه شده هستند که جنبه‌های مختلف یک بسته Integration Services را نشان می‌دهند. جدول ۱-۹ متغیرهای سیستمی Integration Services جدید را فهرست کرده است.

(c) جدول ۱-۹ متغیرهای سیستمی Integration Services

متغیر سیستمی	شرح
BreakpointTargetDescription	حاوی شرحی از نقطه شکست است.
Cancel	نشان می‌دهد که اجرا باید لغو شود.
CountDone	حاوی شمارنده پیشرفت انتقال است.
CreationDate	حاوی تاریخ ایجاد بسته است.
CreatorComputerName	حاوی نام کامپیوتر مورد استفاده برای ایجاد بسته است.
CreatorName	حاوی نام کاربری است که بسته را ایجاد کرده است.
CustomEventDescription	یک رویداد اختصاصی را توصیف می‌کند.
CustomEventInfo	حاوی اطلاعات رویداد اختصاصی است.
CustomEventName	حاوی یک نام رویداد اختصاصی است.
CustomEventValue	حاوی مقدار رویداد اختصاصی است.
CustonEventGUID	حاوی GUID رویداد اختصاصی است.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

ErrorCode	حاوی یک کد خطاست.
ErrorDescription	حاوی یک شرح خطاست.
ExecutionInstanceGuid	حاوی یک GUID است که نمونه بسته جاری را تعیین می‌کند.
ExecutionStatus	حاوی وضعیت اجرای بسته است.
LocaleId	حاوی کد تعیین هویت Locale است.
MachineName	حاوی نام ماشین جاری است.
MaxCount	حاوی حداکثر تعداد آیتم‌های منتقل شونده است.
PackageId	حاوی تعیین هویت بسته جاری است.
PackageName	حاوی نام بسته جاری است.
PercentComplete	حاوی وضعیت پیشرفت انتقال است.
ProgressCountHigh	حاوی حد بالای پیشرفت انتقال است.
ProgressCountLow	حاوی حد پایین پیشرفت انتقال است.
ProgressDescription	حاوی شرح پیشرفت انتقال است.
ProgressEvent	حاوی تعیین هویت رویداد پیشرفت انتقال است.
Propagate	نشان می‌دهد آیا یک رویداد می‌تواند منتشر شود.
SourceDescription	حاوی یک شرح منبع رویداد است.
SourceID	حاوی یک تعیین هویت منبع رویداد است.
SourceName	حاوی نام یک منبع رویداد است.
StartTime	حاوی زمان شروع اجرای بسته است.
TaskID	حاوی یک تعیین هویت وظیفه است.
TaskName	حاوی نام یک وظیفه است.
UserName	حاوی نام کاربر اجرا کننده بسته است.
VersionBuild	حاوی شماره ساخت بسته است.
VersionComment	حاوی شرح بسته است.
VersionGUID	حاوی GUID بسته است.
VersionMajor	حاوی شماره نگارش اصلی بسته است.
VersionMinor	حاوی شماره نگارش فرعی بسته است.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

کنترل جدید پیچیده

SQL Server 2005 Integration Services هم‌چنین از کنترل جریان پیچیده پشتیبانی می‌کند. در هر بسته Integration Services، می‌توانید مسیری را مشخص کنید که چنانچه یک عمل معین موفق یا ناموفق باشد، انجام خواهد شد. مثلاً، اگر عملی موفق باشد، می‌توانید بسته Integration Services خود را برای رفتن به عمل بعدی تنظیم کنید. در غیر این صورت، اگر عملی ناموفق باشد، می‌توانید عمل دیگری را انجام دهید. اطلاعات بیشتر درباره ویژگی جریان خطایی جدید در این بخش ارایه می‌شود هم‌چنین می‌توانید جریان کنترل را در روشی تنظیم کنید که چندین وظیفه می‌توانند به‌طور موازی اجرا شوند یا می‌توانید وظایف را مجبور به اجرای ترتیبی کنید که این کار با مشخص کردن این وظیفه بعدی اجرا نخواهد شد تا وقتی که وظیفه جاری تکمیل شود. هم‌چنین می‌توانید از یک ساختار Integration Services جدید به نام کانتینر[1] برای گروه‌بندی وظایف Integration Services مرتبط با یکدیگر استفاده شود. هم‌چنین می‌توانید دارای متغیرها و جریان کنترل داخلی خاص خود باشند. هم‌چنین چندین ساختار حلقه وجود دارد که به شما امکان تنظیم اعمال تکراری را می‌دهند. یک کانتینر For Each Loop وجود دارد که قادر به تکرار گروهی از اشیاست و عملی را روی این اشیا در هر تکرار انجام می‌دهد. علاوه بر این، یک کانتینر For Loop وجود دارد که می‌تواند یک عبارت را ارزیابی کند و به‌طور شرطی اعمال تکراری را انجام دهد.

جریان‌های خطا

یکی از ویژگی‌های جدید مهم SQL Server 2005 Integration Services، توانایی آن برای پشتیبانی از جریان‌های خطاست. ویژگی Error Flows جدید، لزوماً به شما امکان افزودن مدیریت خطا را به بسته‌های Integration Services شما می‌دهد. با این ویژگی جریان‌های خطای جدید، هنگامی که یک Integration Services، ردیف‌های حاوی شروط خطا را تبدیل می‌کند، به‌جای متوقف کردن فرآیند با یک خطا، Integration Services می‌تواند ردیف مشکل‌دار را بر طبق جریان خطایی که تنظیم کرده است، مسیریابی کند. مثلاً، جریان خطا ممکن است نشان دهد که ردیفی به یک فایل log نوشته شده است یا هم‌چنین می‌تواند بسته را به وظیفه‌ای هدایت کند که روال‌های خطای بهتری را انجام دهد که حتی می‌تواند داده را دستکاری کرده و ردیف را برای پردازش مجدد در pipeline قرار دهد.

حالت فوری و حالت پروژه

1- Container

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

شبیه روش کار SQL Server 2000 DTS که DTS دارای یک رابط ویزارد بود که در اصل برای اجرای انتقال‌های داده ویژه طراحی شده بود و یک DTS Designer بود که برای ساخت بسته‌های DTS پیچیده‌تر استفاده می‌شد، SQL Server 2005 Integration Services از یک حالت فوری[1] و یک حالت پروژه[2] پشتیبانی می‌کند. برای استفاده از Integration Services در حالت فوری، می‌توانید Integration Services Import/Export Wizard را از منو اجرا کنید. Integration Services Import/Export Wizard می‌تواند برای ساخت، اجرا و ذخیره اختیاری بسته‌های Integration Services که انتقال‌های ساده را انجام می‌دهند، استفاده شود. Integration Services Import/Export Wizard جدید SQL Server 2005 بسیار شبیه نگارش موجود در SQL Server 2000 بوده و عمل می‌کند.

در حالی که حالت آنی برای عملیات انتقال داده یک زمانه سریع مفید است، حالت پروژه برای ساخت بسته‌های Integration Services مهم‌تر با استفاده از Business Intelligence Development Studio مفید است. Business Intelligence Development Studio حاوی یک Integration Services Designer کاملاً جدید است که از یک مجموعه کاملاً جدید از Data Flow، Control Flow و Event handlers پشتیبانی می‌کند که می‌توانند برای ساخت بسته‌های Integration Services به کار روند. Integration Services Designer جدید هم‌چنین پشتیبانی کاملی را برای اشکال‌زدایی بسته‌های Integration Services فراهم کرده است. می‌توانید Integration Services Designer جدید را در ادامه فصل به‌طور مفصل ببینید.

امضای دیجیتال بسته‌های Integration Services

با استفاده از متدی شبیه ویژگی امضای دیجیتال موجود که در برنامه‌های Microsoft .NET در دسترس قرار گرفته است، بسته‌های Integration Services هم اینک می‌توانند به خوبی امضا شوند. بسته‌ها می‌توانند در طی فرآیند طراحی با استفاده از Integration Services Designer به صورت دیجیتال امضا شوند. هنگامی که بسته‌ای به‌طور دیجیتالی امضا شده باشد، آن بسته فقط خواندنی است و دیگر نمی‌تواند اصلاح شود.

1- Immediate Mode
2- Project Mode

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۵۶،۰۲ آداپتورهای داده

DTP از آداپتورهای داده برای اتصال نقاط انتهایی منبع و مقصد داده استفاده می‌کند. همان‌گونه که از نام آن‌ها مشخص است، آداپتورهای داده منبع به منبع داده متصل شده و ورودی برای بسته‌های Integration Services فراهم می‌کنند. آداپتورهای داده به مقصد داده متصل شده و داده را به خروجی ارسال می‌کنند. SQL Server 2005 تعدادی آداپتور داده منبع و مقصد فراهم می‌کند. جدول ۲-۹ مجموعه‌ای از آداپتورهای داده تعبیه شده را فهرست می‌کند که در SQL Server 2005 تأمین شده‌اند.

جدول ۲-۹ آداپتورهای داده تعبیه شده (a)

آداپتور داده	شرح
Flat File Destination Adapter	یک آداپتور سیستم فایل که داده متنی حایل‌دار را به یک فایل می‌نویسد.
Flat File Source Adapter	یک آداپتور سیستم فایل که داده متنی حایل‌دار را از یک فایل می‌خواند.
OLE DB Destination Adapter	یک تأمین کننده OLE DB که داده را به یک مصرف کننده OLE می‌نویسد.
OLE DB Source Adapter	یک مصرف کننده OLE DB که داده را از یک تأمین کننده OLE DB می‌خواند.
Raw File Destination Adapter	یک آداپتور سیستم فایل که داده را به یک فایل می‌نویسد.
Raw File Source Adapter	یک آداپتور سیستم فایل که داده را از یک فایل می‌خواند.
SQL Server Destination Adapter	یک آداپتور SQL Server که برای نوشتن داده به یک جدول یا دیدگاه می‌نویسد.
Web Service Source Adapter	یک آداپتور Web Service که داده را از یک سرویس وب XML می‌خواند.

Section ۵۶،۰۳ کانتینرها

کانتینرها ساختار جدیدی هستند که مایکروسافت به SQL Server 2005 Integration Services اضافه کرده است. هدف اصلی از کانتینرهای Integration Services، افزودن ساختار و کنترل جریان به بسته‌های Integration Services است. کانتینرها وظایف مرتبط را با یکدیگر گروه‌بندی کرده

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

و طراحی شده است تا برای اجرای وظایف تکراری یا فراهم کردن بُردی برای متغیرها استفاده شوند. SQL Server 2005 Integration Services از انواع کانتینرهای نشان داده شده در جدول ۳-۹ پشتیبانی می‌کند.

جدول ۳-۹ انواع کانتینر Integration Services (a)

کانتینر	شرح
Package Container	کلکسیونی از عناصر بسته
Foreach Loop Container	جریان کنترل تکرار را در یک بسته فراهم می‌کند.
For Loop Container	پشتیبانی از اعمال تکراری را در یک بسته فراهم می‌کند.
Sequence Container	کانتینرها و وظایف مرتبط را در بسته گروه‌بندی می‌کند.
TaskHost Container	سرویس‌هایی را برای یک وظیفه فراهم می‌کند.
Container Properties	مقادیری را نگهداری می‌کند که برای کانتینر مشترک هستند.
Container Collections	کلکسیونی از کانتینرها

Section ۵۶،۰۴ وظایف

وظایف Integration Services اساسی‌ترین عناصر بسته Integration Services هستند. وظیفه Integration Services لزوماً عملی را تعریف می‌کند که انجام خواهد شد. این اعمال در زمینه کپی کردن فایل‌ها، اجرای عبارات T-SQL و اجرای اسکریپت‌ها برای انجام انتقال‌های FTP و اجرای مدل‌های data mining هستند. چندین وظیفه مرتبط می‌توانند در کانتینرها گروه‌بندی شوند. جدول ۴-۹ وظایفی را نشان می‌دهد که در SQL Server 2005 Integration Services وجود دارند.

جدول ۴-۹ وظایف Integration Services (a)

وظیفه	شرح
ActiveX Script Task	یک اسکریپت ActiveX را اجرا می‌کند که عمل خاصی را انجام می‌دهد.
Analysis Services Execute DDL Task	عبارات T-SQL DDL را اجرا می‌کند.
Analysis Services Processing Task	اشیای Analysis Services را پردازش می‌کند.
Bulk Insert Task	داده را از یک فایل متنی در یک جدول درج می‌کند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Data Flow Task	داده را بین منابع داده کپی کرده و منتقل می‌کند.
Data Mining Query Task	پرس‌وجوهای data mining را اجرا می‌کند.
Execute Package Task	بسته‌های دیگر را اجرا می‌کند.
Execute Process Task	یک برنامه یا اسکریپت را اجرا می‌کند.
Execute SQL Task	عبارات T-SQL را اجرا می‌کند.
File System Task	اعمالی را در سیستم فایل اجرا می‌کند.
File Transfer Protocol Task	انتقال‌های داده FTP را اجرا می‌کند.
Message Queue Task	پیام‌ها را از پرس‌وجوهای داده MSMQ ارسال و دریافت می‌کند.
Script Task	اسکریپت‌های نوشته شده در VB.NET را با استفاده از محیط Microsoft Visual Studio for Applications اجرا می‌کند.
Send Mail Task	یک پیام e-mail را ارسال می‌کند.
XML Task	به داده‌ها در اسناد XML دستیابی دارد.

Section ۵۶،۰۵ تبدیلات

تبدیلات Integration Services آن‌چه را که هنگام انتقال داده از آداپتور داده منبع به آداپتور داده مقصد بر داده رخ می‌دهد، کنترل می‌کند. SQL Server 2005 از تعدادی تبدیلات تعبیه شده و تبدیلات اختصاصی تعریف شده کاربر پشتیبانی می‌کند. می‌توانید تبدیلات اختصاصی را با استفاده از API فراهم شده توسط مدل شیئی DTP ایجاد کنید. SQL Server 2005 Integration Services لیست جامعی از تبدیلات استاندارد تعبیه شده را فراهم کرده است که در جدول ۵-۹ نشان داده شده‌اند.

جدول ۵-۹ تبدیلات استاندارد تعبیه شده (a)

تبدیل	شرح
Aggregate Transformation	انبوهه‌ها را انجام می‌دهد.
Allocation Transformation	مقدار یک ستون ورودی را در بین چندین ستون خروجی گسترش می‌دهد.
Character Map Transformation	توابع رشته‌ای را بر داده کاراکتری اعمال می‌کند.
Conditional Split Transformation	داده را ارزیابی کرده و آن را برای خروجی‌های مختلف مسیریابی

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

	می‌کند.
Copy/Map Transformation	ستون‌های خروجی جدید را با کپی کردن ستون‌های ورودی ایجاد می‌کند.
Data Mining Model Accuracy Transformation	دقت مدل‌های data mining را محاسبه می‌کند.
Data Mining Model Training Transformation	مدل‌های data mining را آموزش می‌دهد.
Data Mining Query Transformation	پرس‌وجوهای گزاره‌ای data mining را اجرا می‌کند.
Data Conversion Transformation	نوع داده یک ستون ورودی را به نوع داده خروجی متفاوتی تبدیل می‌کند.
Derived Column Transformation	یک ستون خروجی را از نتایج عبارات ایجاد می‌کند.
Dimension Processing Transformation	ابعاد مکعب OLAP را پردازش می‌کند.
File Extractor Transformation	داده را از یک جریان داده بسته خوانده و آن داده را در فایل می‌نویسد.
File Injector Transformation	داده را از یک فایل خوانده و آن داده را به یک جریان داده بسته اضافه می‌کند.
Fuzzy Grouping Transformation	مقادیر را در داده ستون ورودی استاندارد می‌کند.
Fuzzy Lookup Transformation	مقادیر را در یک جدول مرجع با استفاده از تطابق فازی جستجو می‌کند.
Logged Lineage Transformation	اطلاعات محیطی را برای جریان داده بسته فراهم می‌کند.
Lookup Transformation	مقادیر یک جدول مرجع را با استفاده از تطابق دقیقی جستجو می‌کند.
Merge Transformation	دو مجموعه داده ذخیره شده را ادغام می‌کند.
Merge Join Transformation	دو مجموعه داده را با استفاده از یک تلفیق FULL، LEFT یا INNER تلفیق می‌کند.
Multicast Transformation	داده ورودی را به خروجی‌های مختلف توزیع می‌کند.
Partition Processing Transformation	پارتیشن‌های OLAP را پردازش می‌کند.
Pivot Transformation	داده ورودی را برطبق یک مقدار ستون ورودی محوری می‌کند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Row Count Transformation	ردیف‌های ورودی را شمارده و تعداد را در یک متغیر ذخیره می‌کند.
Sampling Transformation	یک نمونه‌سازی ارایه‌ای از مجموعه داده ورودی ایجاد می‌کند.
Script Transformation	یک اسکریپت را برای انجام ورودی اجرا می‌کند.
Slowly Changing Dimension Transformation	بهنگام‌رسانی و درج ردیف‌ها را در ابعاد OLAP هماهنگ می‌کند.
Sort Transformation	داده ورودی را مرتب کرده و داده ذخیره شده در خروجی تبدیل را کپی می‌کند.
Surrogate Key Transformation	خصوصیات اختصاصی اضافی را برای بسته Integration Services فراهم می‌کند.
Union All Transformation	چند مجموعه داده را ادغام می‌کند.
UnPivot Transformation	داده ورودی را برطبق یک مقدار ستون ورودی از حالت محوری خارج می‌کند.

Section ۵۶،۰۶ مدیریت خطا

توانایی به وجود آوردن و مدیریت رویدادها، ویژگی جدید دیگری در SQL Server 2005 Integration Services است. مدیریت رویداد به بسته‌های Integration Services امکان پاسخ دادن به رویدادهایی را می‌دهد که در زمان اجرا توسط کانتینرها و وظایف به وجود می‌آورند. رویدادها می‌توانند توسط عناصر بسته Integration Services برای مطلع کردن تعدادی از حالات مختلف فعال شوند، از جمله شروط خطا، هنگامی که وظیفه‌ای شروع شده باشد، هنگامی که وظیفه‌ای تکمیل شده باشد یا تغییر در وضعیت متغیر رخ داده باشد. جدول ۶-۹ مدیران رویداد Integration Services را فهرست می‌کند.

(a) جدول ۶-۹ مدیران رویداد Integration Services

مدیر رویداد	شرح
OnCustomerEvent	توسط یک وظیفه یا بسته طبق تقاضا به وجود می‌آید.
OnError	توسط یک وظیفه یا کانتینر در یک خطا به وجود می‌آید.
OnExecStatusChanged	توسط یک وظیفه یا کانتینر هنگامی به وجود می‌آید که وضعیت اجرای آن تغییر کند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

OnPostExecute	توسط یک وظیفه یا کانیتنر بعد از اجرای آن به وجود می‌آید.
OnPostValidate	توسط یک وظیفه بعد از تأیید اعتبار آن به وجود می‌آید.
OnPreExecute	توسط یک وظیفه یا کانیتنر قبل از اجرای آن به وجود می‌آید.
OnPreValidate	توسط یک وظیفه قبل از تأیید اعتبار آن به وجود می‌آید.
OnProgress	توسط یک وظیفه یا کانیتنر هنگامی به وجود می‌آید که معیار پیشرفت خاصی برآورده شده باشد.
OnQueryCancel	توسط یک وظیفه یا کانتینر برای تعیین این مسأله به وجود می‌آید که آیا آن [OK] باید اجرا را متوقف کند.
OnTaskFailed	توسط یک وظیفه هنگامی به وجود می‌آید که ناموفق باشد.
OnVariableValueChanged	به‌طور اختیاری توسط یک متغیر هنگامی به وجود می‌آید که مقدار آن تغییر کند.
OnWarning	توسط یک وظیفه یا کانتینر هنگامی به وجود می‌آید که هشداری تولید شده باشد.

Section ۵۶،۰۷ تأمین‌کنندگان Log

تأمین‌کنندگان Log ویژگی کاملاً جدید دیگری است که مایکروسافت به بسته‌های Integration Services اضافه کرده است. همان‌گونه که از نام آن‌ها حدس زده می‌شود، تأمین‌کنندگان Integration Services Log به شما امکان افزودن امکان ثبت به بسته‌ها، کانتینرها و وظایف را می‌دهند. Logging برای ثبت اطلاعات خطا یا سایر اطلاعات وضعیت یا زمان اجرای مهم استفاده می‌شود. جدول ۷-۹ تأمین‌کنندگان Log را فهرست کرده است که مایکروسافت همراه با SQL Server 2005 عرضه می‌کند.

جدول ۷-۹ تأمین‌کنندگان Log (a)

تأمین‌کننده log	شرح
تأمین کننده log Text File	ورودی‌های log را در فایل‌های متنی ASCII با استفاده از یک فرمت مقدار مجزا شده با کاما (CSV) با استفاده از یک توسعه فایل پیش‌فرض log. می‌نویسد.
تأمین کننده log SQL Profiler	داده ردیابی SQL را با استفاده از یک توسعه فایل پیش‌فرض trc. در یک فایل ردیابی می‌نویسد.
تأمین کننده log SQL Server	ورودی‌های log را در جدول sysdtslog90 در یک پایگاه داده SQL Server

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

	می‌نویسد.
تأمین کننده log Windows Event	ورودی‌های را در Windows Application می‌نویسد.
تأمین کننده log XML File	فایل‌های log را با استفاده از توسعه فایل پیش‌فرض xml. در یک فایل XML می‌نویسد.

Article LVII. ابزارهای Integration Services

در نتیجه این معماری کاملاً جدید، مجموعه ابزار Integration Services کاملاً اصلاح شده و ظاهر کاملاً جدیدی به خود گرفته است. در حالی که شباهت‌هایی در برخی از ابزارهای ساده‌تر وجود دارد (مثلاً، Integration Services Import/Export Wizard که برای انجام عملیات انتقال داده ویژه استفاده می‌شود)، ابزارهای مهم‌تری نظیر Integration Services Designer کاملاً متفاوت هستند. این که شما یک کارشناس DTS یا یک کاربر مبتدی آن باشید، SQL Server 2005 Integration Services نیازمند این است که شما نحوه استفاده از این ابزارهای جدید را از زمینه بیاموزید. در این بخش، مروری از ابزارها و برنامه‌های سودمند جدید Integration Services خواهیم داشت که همراه با SQL Server 2005 هستند.

مجموعه ابزار SQL Server 2005 Integration Services به دو ناحیه اصلی تقسیم می‌شود: ابزارهایی که برای ایجاد و اجرای بسته‌های Integration Services استفاده می‌شوند و ابزارهایی که برای کار با بسته‌های DTS موجود استفاده می‌گردند. ابزارهای Integration Services Designer شامل Integration Services Import/Export Wizard اصلی و Designer پیشرفته‌تر هستند. همان‌گونه که در ادامه خواهید دید، Integration Services Designer جداسازی معماری جریان کنترل و جریان داده را با فراهم کردن ویراستارهای مجزا برای طراحی جریان کنترل و جریان داده بسته منعکس می‌کند. Integration Services Designer یک ارایه گرافیکی از جریان کاری فراهم می‌کند که در داخل یک بسته Integration Services رخ می‌دهد. این Integration Services Designer جدید، ویژگی‌های سطح جهانی از قبیل کنترل منبع را برای تسهیل محیط‌های چند برنامه‌نویسه فراهم کرده است. هم‌چنین ابزارهای توزیع و اشکال‌زدایی بسته تعبیه شده وجود دارند، از جمله توانایی تنظیم و کنترل نقاط شکست در یک بسته. هم‌چنین می‌توانید از Integration Services Designer برای کنترل اجرای یک بسته استفاده کند (پیشرفت کامل آن و نتیجه تبدیلات و وظایف مجزا). ابزارهای بسته Integration Services شامل یک Package Migration Wizard هستند که برای انتقال بسته‌های SQL Server 7 و SQL Server 2000 به فرمت SQL Server 2005 جدید طراحی شده است. هم‌چنین یک Integration Services Package Installer Wizard وجود دارد که برنامه‌های نصبی را برای

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

بسته‌های Integration Services ایجاد می‌کند و توزیع این بسته‌ها را به سیستم‌های راه‌دور آسان‌تر می‌کند. علاوه بر این، یک برنامه سودمند Integration Services Package Excution وجود دارد که به شما امکان اجرای بسته‌های Integration Services را از خط فرمان می‌دهد؛ این یک ابزار مفید برای مشارکت انتقال Integration Services به عنوان بخشی از اسکریپت‌های مدیریتی شماست.

Section ۵۷.۰۱ ابزارهای طراحی Integration Services

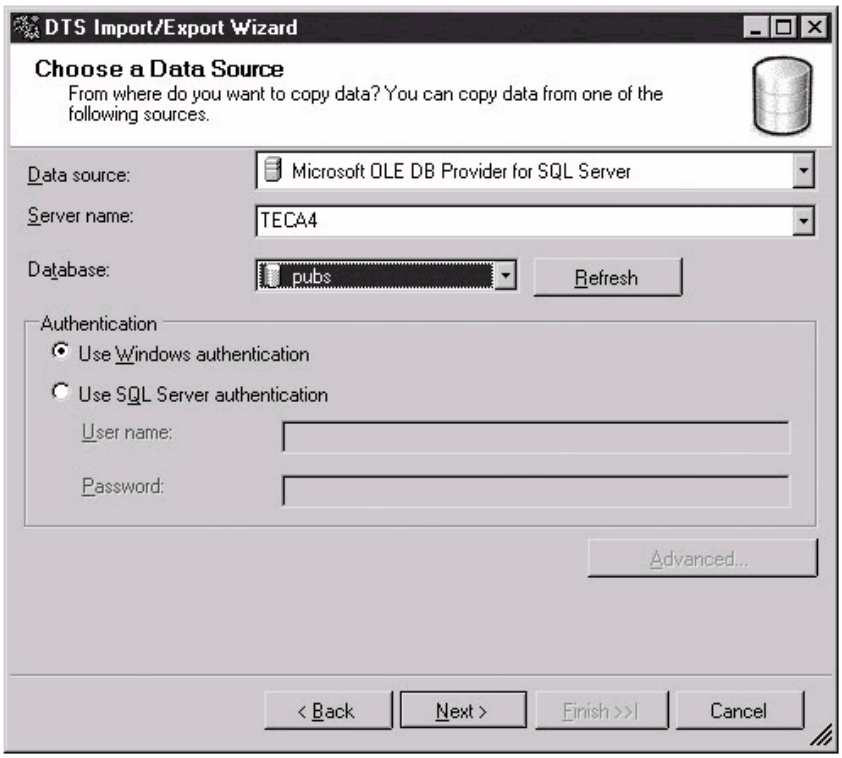
در این‌جا نگاه دقیق‌تری به ابزارهای طراحی Integration Services جدید می‌اندازیم. ابتدا، Integration Services Import/Export Wizard را خواهید دید. سپس، نگاهی تفصیلی به Integration Services Designer جدید خواهید داشت.

Integration Services Import/Export Wizard

Integration Services Import/Export Wizard اولین نقطه ورودی SQL Server 2005 در Integration Services جدید است. شبیه مشابه‌های قبلی آن در SQL Server 7 و SQL Server 2000، SQL Server 2005 Integration Services Import/Export Wizard مجموعه‌ای از کادرهای محاوره‌ای را فراهم کرده است که شما را از طریق فرآیند انتخاب منبع داده، مقصد و اشیایی که منتقل خواهند شد، هدایت می‌کند. Integration Services Import/Export Wizard هم‌چنین به شما اجازه می‌دهد تا به‌طور اختیاری بسته Integration Services را ذخیره و اجرا کنید. می‌توانید Integration Services Import/Export Wizard را با انتخاب گزینه Integration Services Import/Export از منوی SQL Server یا با وارد کردن dtswizard در خط فرمان اجرا کنید. ذخیره کردن بسته‌های تولید شده با Integration Services Import/Export Wizard و آن‌گاه ویرایش آن‌ها در Integration Services Designer، روشی عالی برای یادگیری مطالب بیشتر درباره Integration Services است، بخصوص اگر با Integration Services شروع کرده باشید یا اگر از یکی از نگارش‌های قبلی به SQL Server 2005 Integration Services جدید منتقل شده باشید. می‌توانید Integration Services Import/Export Wizard را در شکل ۵-۹ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

(b)

شکل ۵-۹ Data Source –Integration Services Import/Export Wizard Selection

Integration Services Import/Export Wizard ابتدا شما را از طریق فرآیند انتخاب یک منبع داده هدایت می‌کند. در بازشوی Data Source، تأمین کننده‌ای انتخاب کنید که می‌خواهید استفاده کنید. بسته به تأمین کننده منتخب شما، گزینه‌هایی برای بقیه صفحه نمایش تغییر می‌کنند. اگر Microsoft OLE DB Provider for SQL Server را انتخاب کنید، صفحه‌ای شبیه شکل ۵-۹ را خواهید دید که سپس سروری را انتخاب می‌کنید که باید به پایگاه داده مرتبط متصل شده و نوع تعیین هویتی که باید استفاده شود. کلیک کردن Next شما را از طریق کادرهای محاوره‌ای ویزارد بعدی هدایت می‌کند. کادر محاوره‌ای بعدی به شما اجازه انتخاب مقصد داده را می‌دهد که لزوماً با کادر محاوره‌ای منبع داده یکسان است، به جز این که محل انتقال داده را تعریف می‌کند.

توجه : در حالی که SQL Server 2005 Integration Services Wizard همان عملکرد پایه Integration Services Import/Export Wizard را فراهم می‌کند که در SQL Server 7 و SQL Server

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

2000 وجود داشتند، نگارش جدید فاقد توانایی انجام نگاشت‌های داده اختصاصی و توانایی وارد کردن اسکریپت‌های تبدیل داده اختصاصی است. برای استفاده از این قابلیت‌های پیشرفته‌تر، باید از Integration Services Designer استفاده کنید.

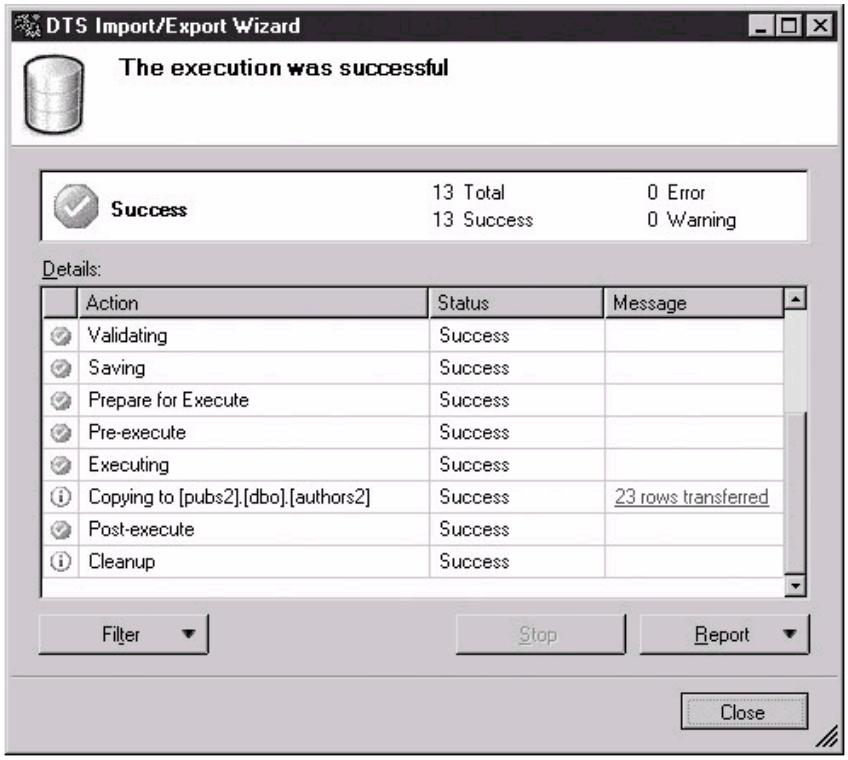
بعد از انتخاب منبع و مقصد داده، این ویزارد به شما در مورد ذخیره اختیاری و اجرای بسته Integration Services به شما اعلان می‌کند. هنگامی که هر وظیفه در بسته اجرا می‌شود، پنجره انتقال به‌طور خودکار بهنگام شده و پیشرفت انتقال بسته Integration Services را نشان می‌دهد. هنگامی که بسته Integration Services به‌طور موفق اجرا می‌شود، Integration Services Import/Export Wizard کادر محاوره‌ای را نمایش خواهد داد که در شکل ۶-۹ نشان داده شده است.

Section ۵۷.۰۲ Integration Services Designer

در حالی که Integration Services Import/Export Wizard برای وظایف انتقال‌های ویژه ساده مفید است، اقتباس، تبدیل و بارگذاری(ETL) نیازی به تأکید و توانی بیشتر از آن چیزی دارند که ویزارد Integration Services ارایه می‌دهد. وظایف ETL، طبق طبیعتشان بیش از انتقال‌های داده ساده از یک مقصد دیگر هستند. در عوض، آن‌ها اغلب داده را از چندین منبع ترکیب کرده، داده را دستکاری نموده، مقادیر را به ستون‌های جدید نگاشت کرده، ستون‌ها را از مقادیر محاسباتی ایجاد نموده و انواع وظایف پاکسازی و تصحیح داده را فراهم می‌کنند. این قابلیت‌های پر تقاضاتر خارج از حوزه Integration Services Import/Export Wizard ساده هستند. اینجا محلی است که Integration Services Designer مجموعه‌ای از ابزارهای گرافیکی است که می‌توانید از آن برای ایجاد راه‌حل‌های تبدیل و انتقال داده قوی استفاده نمایید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)
(b)

شکل ۹-۶ Package Execution - Integration Services Import/Export Wizard

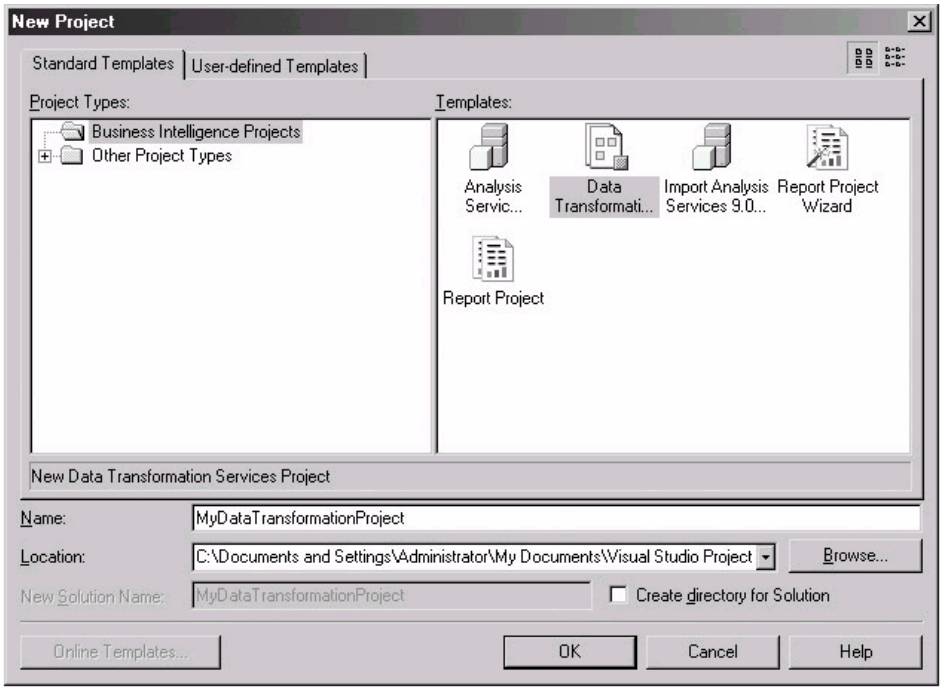
Integration Services Designer از Business Intelligence Development Studio با انتخاب File | New | Project برای باز کردن کادر محاوره‌ای New Project شروع می‌شود. سپس برای ایجاد یک پروژه Integration Services جدید، Business Intelligence Projects را از لیست Project Types انتخاب کرده و سپس Data Transformation Project را از لیست قالب‌ها برگزینید، همان‌گونه که در شکل ۹-۷ نشان داده شده است.

توجه : در مورد این واقعیت که Integration Services Designer از Business Intelligence Development Studio شروع می‌شود، تعجب نکنید. این مسأله تنها محدود به پروژه‌های Analysis Services نیست. Integration Services Designer و پروژه‌های ساخته شده در Business Intelligence Development Studio کاملاً قادر به کار کردن با انواع داده رابطه‌ای و غیره هستند و محدود به داده Analysis Service نیستند.

---

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(c)

شکل ۷-۹ باز کردن یک پروژه تبدیل داده (d)

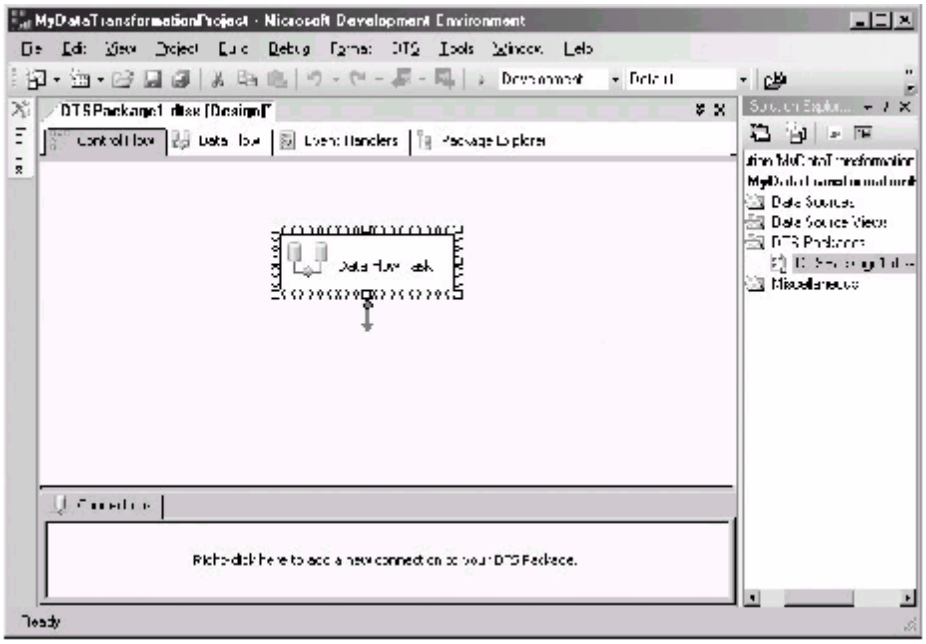
پس از ایجاد پروژه، می‌توانید Integration Services Designer را با کلیک راست روی Packages در پنجره Solution Explorer که در بخش سمت راست صفحه نمایش نشان داده شده است، باز کنید. سپس گزینه New Package را برای شروع Integration Services Designer انتخاب کنید. هنگامی که Integration Services Designer در ابتدا شروع می‌شود، یک سطح خالی ارایه می‌شود که چندان شبیه Integration Services Designer قبل نیست، بنابراین شروع می‌تواند کمی چالش‌آمیز باشد.

Control Flow Designer جدید

در حالی که می‌توانید از Integration Services Designer به دو روش مختلف استفاده کنید، ساده‌ترین روش احتمالاً شروع با برگه Control Flow و سپس باز کردن جعبه ابزار Control Flow است. بعد از نمایش جعبه ابزار، روی وظیفه Data Flow کلیک کرده و آن را به سطح طراحی بکشید. سطح Integration Services Designer Control Flow همان‌گونه که در شکل ۸-۹ نشان داده شده است، ظاهر می‌شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(e)

شکل ۸-۹ Control Flow Designer (f)

در اینجا، بسته Integration Services بسیار ساده است، زیرا شامل یک وظیفه Data Flow است. بدیهی است که می‌توانید با افزودن وظایف اضافی از جعبه ابزار Control Flow و حتی سازمان‌دهی چندین وظیفه مرتبط در کانتینرها، این بسته را بسیار پیچیده‌تر کنید.

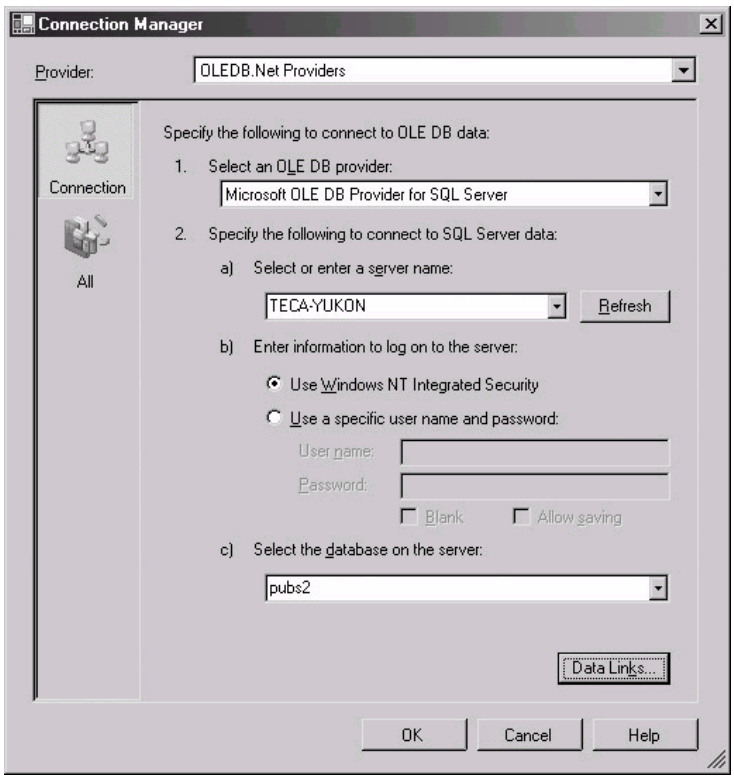
Data Flow Designer جدید

در این لحظه، بسته می‌داند که قصد دارد یک عمل انتقال داده را انجام دهد، ولی نمی‌داند که چه چیزی را انتقال دهد یا این که داده از کجا ارسال می‌شود. بعد از افزودن وظیفه Data Flow، مرحله بعد تعریف جریان‌های داده واقعی است. برای تعریف جریان‌های داده، روی وظیفه Data Flow دابل کلیک کنید تا برگه Data Flow نمایش داده شود. در ابتدا، برگه Data Flow خالی خواهد بود. برای افزودن اتصالات منبع و مقصد داده، جعبه ابزار Data Flow را در سمت چپ IDE باز کرده و یک آیتم جریان داده OLE DB Source و یک آیتم جریان داده OLE DB Destionation را کشیده و در سطح Data Flow Designer بیندازید. علاوه بر این، می‌توانید یک Flat File Destination را اضافه کنید که می‌تواند برای ارسال هر ردیف خطا به خروجی یک فایل ASCII log استفاده شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

حال که بیان شد که Integration Services داده را از یک منبع OLE DB به مقصد OLE DB دیگری منتقل می‌کند، ولی باز هم باید اتصالات منبع و مقصد را تعریف کنید. علاوه بر این، باید اتصالی را برای جریان خطای Flat File تنظیم کنید. اساساً هر منبع یا مقصد جریان داده نیاز به یک اتصال همراه دارد که لزوماً جزییات خاص را درباره نقطه انتهای اتصال فراهم می‌کند. به دلیل این که این مثال از سه آیتم Data Flow استفاده می‌کند، سه اتصال متفاوت مورد نیاز است. دو اتصال OLE DB و یک اتصال Flat File. برای تعریف این اتصالات، می‌توانید در پنجره Connection پایین Integration Services Designer نشان داده شده است، کلیک راست کنید که یک منوی بازشو را نمایش می‌دهد که به شما امکان انتخاب نوع اتصال مورد نظر برای ایجاد را می‌دهد. انتخاب نوع اتصال از منوی بازشو موجب نمایش Connection Manager می‌شود. در شکل ۹-۹، می‌توانید ببینید که برای ایجاد اتصال OLE DB جدید استفاده می‌شود.



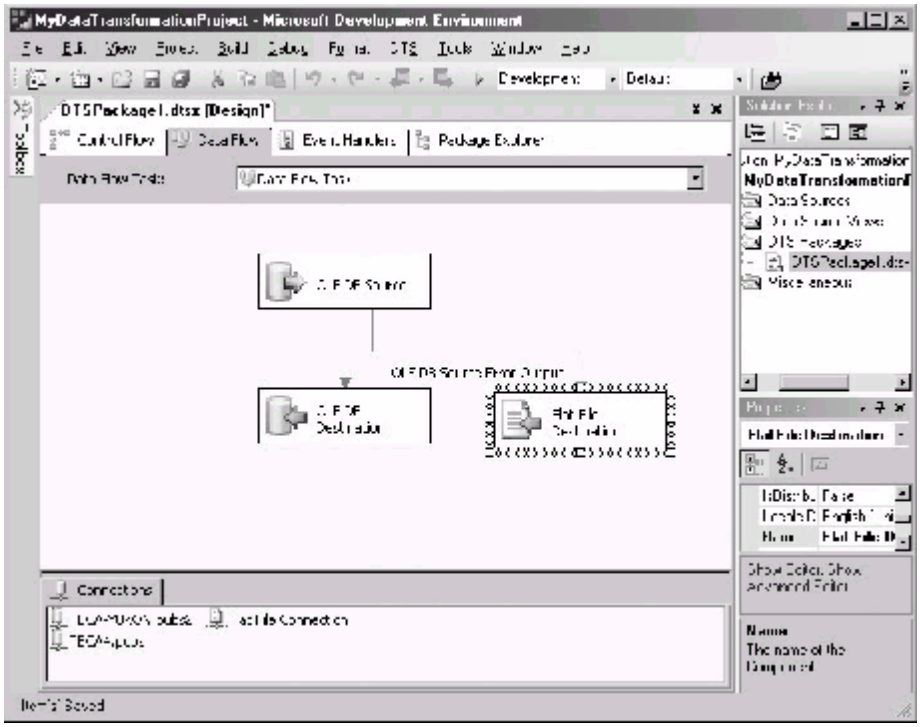
(g)

شکل ۹-۹ Connection Manager (h)

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

پس از ایجاد اتصالات، باید هر اتصال را به آیتم جریان داده مناسبی مرتبط کنید. برای لینک کردن اتصال به یک آیتم جریان داده، روی آیتم جریان داده کلیک راست کنید، گزینه Edit یا Advanced Edit را از منوی زمینه انتخاب کرده و سپس اتصالی را از لیست بازشوی نمایش داده شده در کادر محاوره‌ای Edit برگزینید. هنگام تعریف هر اتصال، نگاشت ستون و سایر تبدیلاتی را که انجام خواهند شد، مشخص کنید. پس از تکمیل این پیکربندی، برگه Integration Services Designer Data Flow شبیه شکل ۱۰-۹ ظاهر می‌شود.



(i)

شکل ۱۰-۹ Data Flow Designer (j)

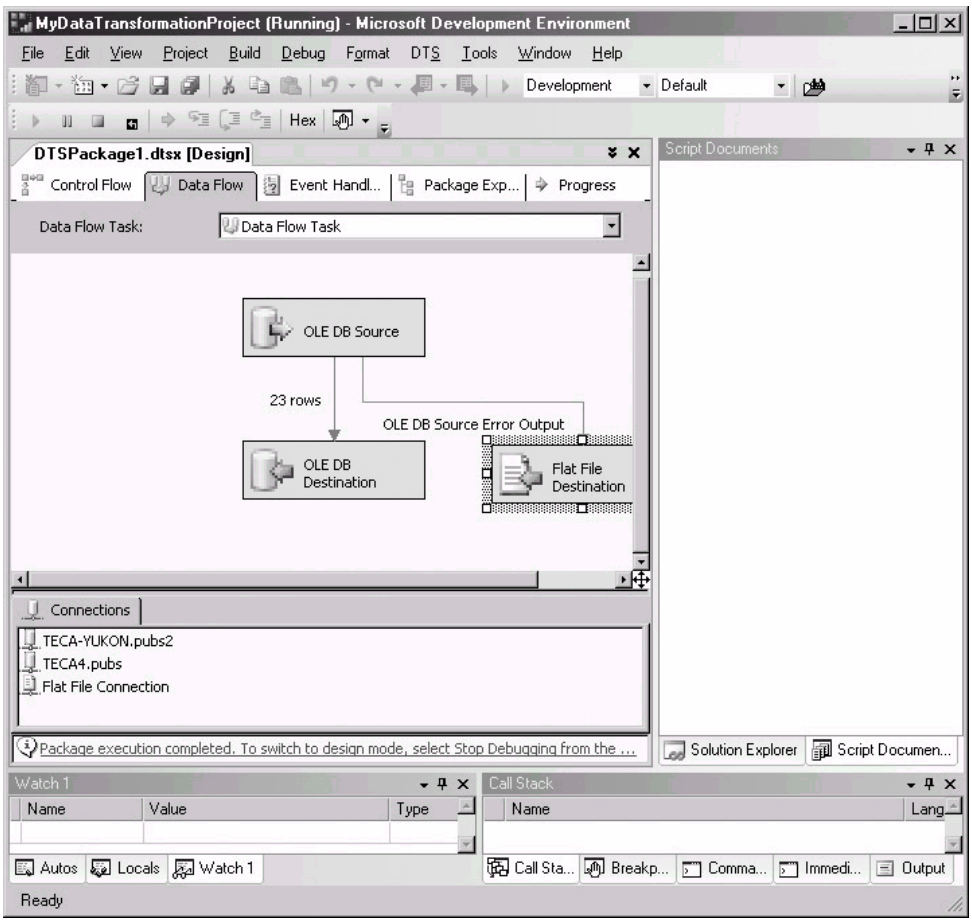
در وسط برگه Data Flow Designer می‌توانید منبع OLE DB، مقصد OLE DB و آیتم‌های جریان داده Flat File Destination را ببینید. یک جریان داده فلشی سبز مستقیم، منبع OLE DB را به مقصد OLE DB متصل کرده و یک جریان خطای فلش قرمز خمیده، منبع OLE DB را به مقصد Flat File متصل می‌کند. اتصالات برای هر یک از این آیتم‌ها در پنجره Connections در پایین صفحه نمایش نشان داده شده‌اند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

ارایه گرافیکی اجرای بسته

هنگامی که تمام اتصالات و جریانات داده تعریف شدند، بسته می‌تواند با کلیک کردن فلش سبز رنگ در نوار ابزار یا با انتخاب گزینه Start از منوی Debug اجرا شود. قبل از اجرای بسته، می‌توانید به‌طور اختیاری نقاط شکست را با انتخاب یک آیتم به همراه گزینه New Breakpoint از منوی Debug در بسته تعریف کنید. هنگامی که بسته اجرا شد، پنجره‌های Watch و Call Stack به‌طور خودکار باز شده و Data Flow Item در حال اجرا به رنگ سبز تغییر می‌کند. می‌توانید نتیجه اجرای بسته نمونه را در شکل ۱۱-۹ ببینید.



(k)

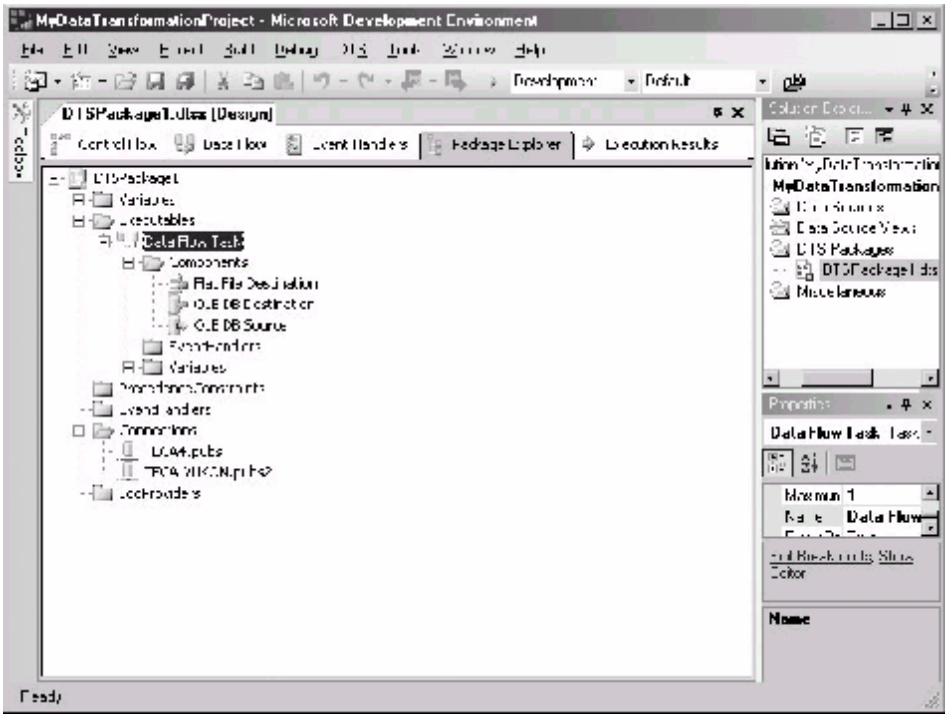
شکل ۱۱-۹ اجرای بسته در Designer (l)

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Package Explorer جدید

ویژگی جدید دیگر در Integration Services Designer، Package Explorer است. Package Explorer یک نمای درختی سلسله مراتبی از Integration Services Package فراهم می‌کند که در طراح نمایش داده می‌شود. Package Explorer در شکل ۱۲-۹ نشان داده شده است.



(m)

شکل ۱۲-۹ Package Explorer (n)

در سطح بالا، می‌توانید نام بسته را ببینید و هر یک از اجزای بسته در زیربسته لیست شده‌اند. کلیک کردن علامت بعلاوه در جلوی هر جزء، موجب نمایش آیتم‌های واقعی می‌شود. توجه به این نکته حایز اهمیت است که Package Explorer تنها ابزاری برای مشاهده نیست. از نمای Explorer، می‌توانید آیتم‌ها را حذف کرده و خصوصیات آن‌ها را ویرایش کنید. هرچند، نمی‌توانید آیتم‌های جدیدی را اضافه کنید. این کار تنها می‌تواند در سطح طراح انجام شود.

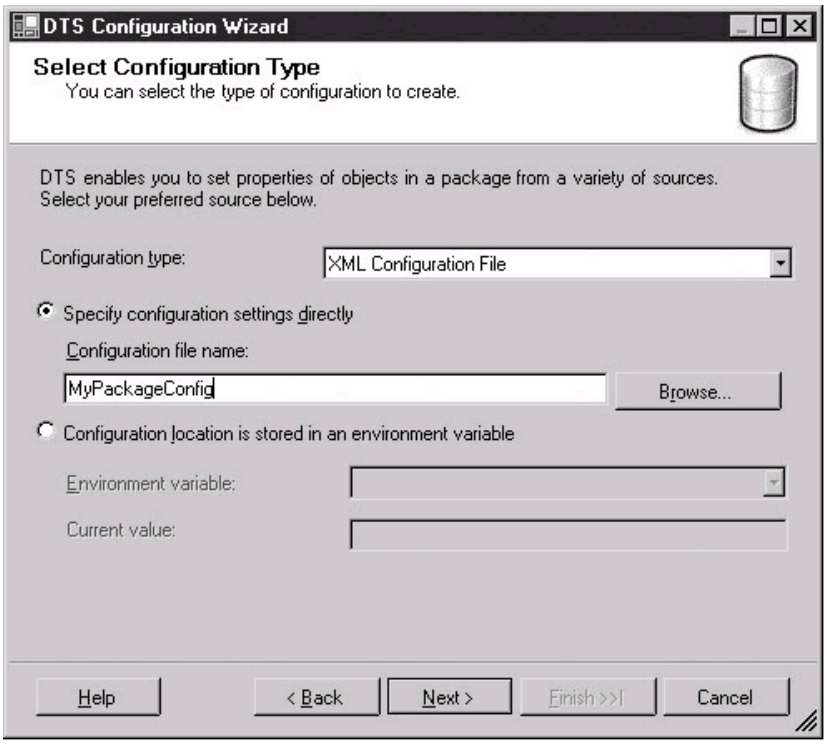
پیکربندی

ویژگی پیکربندی‌های Integration Services جدید برای ساده‌تر شدن توزیع بسته‌های Integration Services طراحی شد. ویژگی پیکربندی‌های جدید به شما امکان می‌دهد تا به‌طور پویا

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

یک بسته را برای اجرا در محیطی متفاوت بهنگام کنید. مثلاً، اتصالات نیاز به رشته‌های اتصال دارند و این رشته‌های اتصال، اغلب تنها برای محیط معینی مناسب هستند. پیکربندی‌ها به شما امکان می‌دهند تا این انواع مقادیر خاص سایت/ سرور را هنگام توزیع بسته، به‌طور پویا بهنگام کنید. می‌توانید پیکربندی‌ها را با استفاده از Configuration Wizard ایجاد کنید. می‌توانید Configuration Wizard را با استفاده از گزینه منوی Integration Services | Configurations برای نمایش کادر محاوره‌ای Package Configuration Organization شروع کنید، که می‌توانید روی Add برای شروع Configuration Wizard کلیک کنید. می‌توانید Configuration Wizard را در شکل ۱۳-۹ ببینید.



(o)

شکل ۱۳-۹ Configuration Wizard (p)

هرچند این بخش قادر به تماس با تنها بخشی از قابلیت موجود در Integration Services جدید SQL Server 2005 است، خوشبختانه این بخش احساسی برای نحوه کار Integration Services Designer به شما می‌دهد.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Article LVIII. ابزارهای بسته‌بندی Integration Services

آخرین بخش این فصل، برخی از سایر برنامه‌های سودمند مدیریت بسته را به شما معرفی خواهند کرد که توسط SQL Server 2005 فراهم شده‌اند. یک Package Migration Wizard برای کمک به شما در تبدیل بسته‌های Integration Services موجود به فرمت بسته جدید SQL Server 2005 فراهم شده است. هم‌چنین یک برنامه Packege Management وجود دارد که به شما امکان کار کردن با بسته‌های Integration Services موجود را می‌دهد. علاوه بر این، یک ابزار خط فرمان جدید به شما امکان اجرای بسته‌ها را از خط فرمان می‌دهد.

Section ۵۸.۰۱ Package Migration Wizard

تغییر معماری کامل و سازگاری با NET Framework. را در نظر بگیرید و نباید تعجب کنید که بسته‌های DTS که برای SQL Server 7 و SQL Server 2000 ساخته شده‌اند، باید قبل از امکان اصلاح به فرمت جدید منتقل شوند.

توجه : بسته‌های DTS موجود که برای SQL Server 2000 ساخته می‌شوند، می‌توانند در SQL Server 2005 اجرا شوند. آن‌ها هم‌چنین می‌توانند در راه‌حل‌های بسته SQL Server 2005 Integration Services جدید قرار گیرند. آن‌ها نمی‌توانند اصلاح شوند، مگر اینکه ابتدا به فرمت جدید منتقل شوند.

SQL Server 2005 Package Migration Wizard بسته‌های Integration Services موجود را به فرمت بسته SQL Server 2005 Integration Services منتقل می‌کند. Package Migration Wizard تلاش می‌کند تا عناصر بسته Integration Services موجود نظیر وظایف، تقدم، الزامات و متغیرها را گرفته و آن‌ها را به آیتم‌های بسته SQL Server 2005 معادل تبدیل کند. Package Migration Wizard می‌تواند اجزای بسته DTS تأمین شده مایکروسافت را مدیریت کند، ولی نمی‌تواند وظایف اختصاصی را تبدیل کند. وظایف اختصاصی ساختار قدیمی خود آن‌ها را نگهداری کرده و به عنوان زیربسته‌ای کپسوله می‌کنند که به بسته منتقل شده لینک می‌شود. اسکریپت‌ها می‌توانند بحث مسأله‌دار دیگری برای Package Migration Wizard باشند. در حالی که بیشتر اسکریپت‌های تبدیل ActiveX می‌توانند به وظیفه ActiceX Script جدید تبدیل شوند، اسکریپت‌های موجودی که به مدل شیئی DTS قدیمی مراجعه می‌کنند، نمی‌توانند تبدیل شوند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۵۸،۰۲ برنامه سودمند Integration Services Packege Management

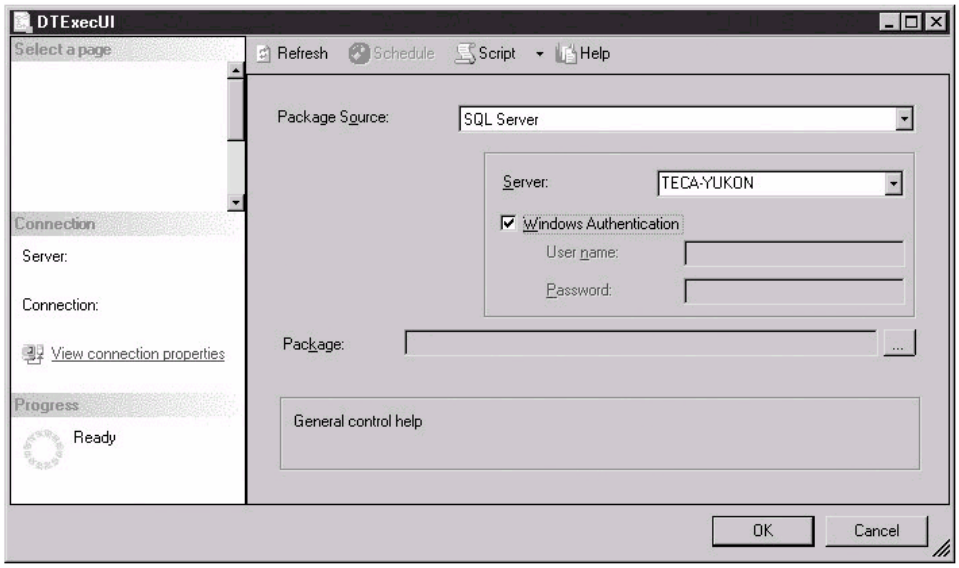
ابزار خط فرمان دیگری که توسط SQL Server 2005 Integration Services فراهم شده است، برنامه سودمند Packege Management است. می‌توانید برنامه Packege Management را با وارد کردن dtutil در خط فرمان اجرا کنید. ابزار dtutil به شما امکان دستیابی به بسته‌هایی را می‌دهد که در پایگاه داده SQL Server msdb ذخیره می‌شوند. هم‌چنین می‌تواند برای کپی، حذف و امضای بسته‌های موجود استفاده شود.

Section ۵۸،۰۳ برنامه‌های سودمند Integration Services Package Execution

دو برنامه اضافی می‌توانند برای اجرای بسته‌ها استفاده شوند: dtexec و dtexecui. همان‌گونه که از نام آن‌ها حدس زده می‌شود، ابزار dtexec از خط فرمان اجرا می‌شود، در حالی که ابزار dtexecui یک رابط کاربر گرافیکی را نمایش می‌دهد که به شما امکان بارگذاری و اجرای بسته‌های Integration Services را می‌دهد. یک جنبه قابل توجه ابزار dtexec این واقعیت است که برای امنیت بهبود یافته، می‌تواند با استفاده از آرگومان‌های رمزگذاری شده اجرا شود. ابزار dtexecui می‌تواند برای تولید یک فایل متنی استفاده شود که حاوی فرامین و پارامترهای مورد نیاز برای اجرای ابزار خط فرمان dtexec هستند (از جمله گزینه‌ای برای تولید آرگومان‌های رمزگذاری شده). می‌توانید ابزار dtexecui را در شکل ۱۴-۹ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

شکل ۱۴-۹ ابزار dtexecui (b)

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

فصل دهم

Analysis Services

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

از زمان عرضه SQL Server 7 و سرویس‌های OLAP، SQL Server در بازار هوش تجاری (BI) پیشرو شده است. SQL Server 7 و نگارش بعدی آن SQL Server 2000، BI را از یک فناوری مناسب در جریان اصلی امور کامپیوتری ایجاد کردند. هم اینک BI یکی از مهیج‌ترین بخش‌های بازار در دنیای IT است. یکی از دلایل اصلی تحت رشد سریع BI این واقعیت است که به یک سازمان امکان گرفتن اطلاعات بیشتر و مقداری از منابع اطلاعاتی نسبت به قبل می‌دهد. این یک فاکتور حیاتی نسبت به چند سال قبل است، زیرا بیشتر شرکت‌ها در سال‌های اخیر، IT را تجربه کرده‌اند. انجام بیشتر امور با این منابع موجب شده است که هم‌اکنون موجب می‌شود IT ارزش راهبردی زیادی برای سازمان‌های تجاری د اشته باشد.

BI به یک حرفه اجازه می‌دهد تا اطلاعات با معنی‌تری را خارج از داده عملیاتی دریافت کند که هم‌اکنون در خط برنامه‌های تجاری استفاده می‌شود. با گرفتن داده عملیاتی از ورودی سفارش آن، حمل و نقل و برنامه‌های فروش، OLAP، انبار کردن داده‌ها و فناوری‌های data mining به مشاغل اجازه می‌دهند تا بخش‌های مهم اطلاعات را به روش‌هایی که با تکنیک‌های دستیابی داده رابطه‌ای خالص امکان‌پذیر نیست، پر و انبوهه کنند. نتیجه نهایی، اطلاعات با معنی‌تری درباره حرفه شما و فاکتورهای مهمی را که بر آن تأثیر می‌گذارند، در برنمی‌گیرد. هیچ چیزی مجانی نیست و BI نیز از این قاعده مستثنی نیست. جاده به سمت BI آزاد نیست (مسائل پرسنلی و آموزشی و مسائل فناوری باید سامان یابند)، ولی در طولانی مدت، هزینه‌ها ممکن است تعدیل شوند. توانایی داشتن اطلاعات با معنی‌تر غیر از داده عملیاتی می‌تواند به تبدیل IT به یک مزیت راهبردی هسته‌ای برای حرفه شما کمک کند.

در سال قبل، واژه BI ماورای تفسیر سنتی آن رشد کرده است که BI لزوماً مترادف با OLAP است. هم‌اکنون BI گسترش یافته است تا تمام فناوری‌هایی را در برگیرد که به حرفه امکان فراهم کردن اطلاعات تصمیم‌گیری را می‌دهند. در SQL Server 2005، مایکروسافت این تعریف گسترده از BI را پذیرفته است. BI بیشتر از OLAP و تحلیل، شامل ابزارهای اطلاعاتی دیگر از قبیل Integration Services و Reporting Services است. ویژگی‌های جدید در Integration Services و Reporting Services در فصل‌های ۸ و ۹ به‌طور مفصل بررسی شدند. در این فصل مروری بر ویژگی‌های Analysis Services جدید SQL Server 2005 خواهیم داشت.

در این نسخه از SQL Server، تأکید زیادی روی BI شده است و ویژگی‌های جدید زیادی وجود دارند که تشریح تمام آن‌ها نیاز به تخصیص یک کتاب کامل دارد. در این فصل، به مهم‌ترین

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

ویژگی‌های جدید در SQL Server 2005 Analysis Services اشاره خواهیم کرد. اولین بخش این فصل، مروری مختصر از OLAP و نقش Analysis Services فراهم می‌کند. سپس، مهم‌ترین ویژگی‌های جدید موجود در Analysis Services Engine را خواهید دید و هم‌چنین مقدمه‌ای بر Unified Dimensional Model جدید مایکروسافت خواهید داشت. بخش بعد، ویژگی‌های جدید برنامه‌نویسی و مدیریت Analysis Services را در برمی‌گیرد. بالاخره، این فصل الگوریتم‌های data mining جدید را ارایه خواهد کرد که مایکروسافت به Analysis Services اضافه کرده است.

Article LIX. مروری بر Analysis Services

Analysis Services و سرویس‌های OLAP مقدم آن در اصل به عنوان یک راه‌حل گزارشگیری برای داده موجود در یک انبار داده یا data mart طراحی شده‌اند. اطلاعات ذخیره شده در یک انبار داده، معمولاً داده رابطه‌ای نیست، بلکه داده خلاصه‌ای است که معمولاً از یک حافظه داده رابطه‌ای مشتق شده‌اند. این الگوهای انبار داده شبیه دانه‌های برف یا ستاره ذخیره می‌شوند. داده در یک انبار داده یا data mart با استفاده از فناوری OLAP پردازش می‌شود. برخلاف فناوری رابطه‌ای که نتایج را با خواندن و تلفیق داده هنگام صدور پرس‌وجو به دست می‌آورد، OLAP برای حرکت در داده خلاصه برای برگرداندن سریع نتایج بهینه می‌شود. کارآیی پرس‌وجوی بهبود یافته دلیلی تحت این تحریک OLAP است. مثلاً، برای مطرح کردن جمع‌های فروش محلی، ناحیه‌ای، ملی و جهانی برای یک سازمان معین، یک پرس‌وجوی رابطه‌ای ممکن است صدها هزار یا حتی میلیون‌ها ردیف را پردازش کند (فرآیندی که ممکن است در سریع‌ترین سیستم‌ها هم بسیار طولانی باشد). در مقایسه، OLAP به دلیل این که در اصل با اطلاعات خلاصه کار می‌کند، ممکن است نیاز به خواندن تنها دو یا سه محل داده برای ذکر پاسخ مشابه داشته باشد. بدیهی است، این امر موجب کارآیی قابل توجهی می‌شود. کارآیی سریع‌تر OLAP، پرس‌وجو و پردازش داده‌ای را ممکن می‌سازد که با استفاده از ابزارهای متداول دستیابی داده رابطه‌ای امکان‌پذیر نبود.

به جای کار کردن با مجموعه‌هایی از جداول مرتبط، فناوری‌های OLAP با مکعب‌ها[1] کار می‌کنند که شامل ابعاد و اندازه‌ها هستند. بُعد[2] یک دسته توصیفی است. مثلاً، یک بُعد ممکن است یک مکان جغرافیایی یا یک نوع محصول باشد. اندازه[3] یک مقدار کمی نظیر فروش برحسب دلار، مقدار

1- Cubes
2- Dimension
3- Measure

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

موجودی یا هزینه‌های کل باشد. انبوهه‌ها که از منبع داده اصلی مشتق می‌شوند، در هر سلول مکعب ذخیره می‌گردند. این روش سازمان‌دهی داده، موجب ساده شدن فیلتر کردن داده و سریع و کارآمد شدن پرس‌وجوهای تکراری می‌شود. هرچند، ترفندی نیز وجود دارد. در حالی که انبوهه‌های OLAP کلیدی برای حصول‌پذیری کارآیی پرس‌وجو در پرس‌وجوهای انبار داده هستند، هزینه ذخیره داده انبوهه حافظه دیسک است. در واقع، تعداد انبوهه‌ها به آسانی ممکن است از تعداد ردیف‌های جزیی اصلی تجاوز کند. علاوه بر این، برنامه‌های خط تجاری معمولاً داده خود را در پایگاه‌های داده OLAP ذخیره می‌کنند. در عوض، داده برای بارگذاری یک انبار داده معمولاً از پایگاه‌های داده رابطه‌ای اقتباس شده و برای پایگاه داده OLAP در فرآیندی به نام اقتباس، تبدیل و بارگذاری (ETL) بارگذاری می‌شوند.

Section ۵۹.۰۱ انواع حافظه OLAP

سه روش اصلی برای ذخیره داده بُعدی مورد استفاده در انبار داده به کار می‌روند: OLAP چند بُعدی (MPLAP)، OLAP رابطه‌ای (ROLAP) و OLAP هیبریدی (HOLAP). هر یک از این روش‌ها دارای الزمات حافظه داده مشخص خود و سرعت بازیابی داده هستند. SQL Server 2005 از تمام این روش‌ها پشتیبانی می‌کند.

MOLAP

OLAP چند بُعدی (MOLAP)، بُعد و داده واقعی را در یک حافظه داده دایمی با استفاده از ایندکس‌های فشرده شده ذخیره می‌کند. انبوهه‌ها برای تسهیل دستیابی داده سریع ذخیره می‌شوند. موتورهای پرس‌وجوی MOLAP معمولاً برای فرمت حافظه مورد استفاده حافظه داده MOLAP اختصاصی و بهینه شده هستند. MOLAP پردازش پرس‌وجویی سریع‌تر از ROLAP را ارایه می‌دهند و معمولاً نیاز به حافظه کمتری دارند. هرچند، به خوبی مقیاس‌بندی نمی‌شوند و نیاز به پایگاه داده مجزایی برای حافظه دارند.

ROLAP

OLAP رابطه‌ای (ROLAP) انبوهه‌ها را در جداول پایگاه داده رابطه‌ای ذخیره می‌کند. کاربرد ROLAP از پایگاه‌های داده رابطه‌ای به آن اجازه می‌دهد تا از منابع پایگاه داده موجود بهره ببرد، به علاوه این که به برنامه‌های ROLAP اجازه می‌دهد تا به خوبی مقیاس‌گذاری شوند. هرچند کاربرد

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

جداول ROLAP برای ذخیره انبوهه‌ها معمولاً نیاز به حافظه دیسک بیشتر از MOLAP دارد و معمولاً چندان سریع نیست.

HOLAP

OLAP هیبریدی (HOLAP)، همان‌گونه که از نامش حدس زده می‌شود، تقاطعی بین MOLAP و ROLAP است. HOLAP مثل ROLAP داده اصلی ذخیره شده در پایگاه داده منبع را رها می‌کند. HOLAP مثل MOLAP انبوهه‌ها را در یک حافظه ذخیره دایمی که مجزا از پایگاه داده رابطه‌ای اصلی است، ذخیره می‌کند. این ترکیب به HOLAP اجازه می‌دهد تا مزایای هر MOLAP و ROLAP را ارایه دهد. هرچند برخلاف MOLAP و ROLAP که از استانداردهایی با تعریف خوب پیروی می‌کنند، HOLAP هیچ پیاده‌سازی یک شکلی ندارد. با درک پایه‌ای از OLAP، اجازه دهید نگاهی به برخی از بهبودهای جدید در SQL Server 2005 Analysis Services داشته باشیم.

Article LX. بهبودهای موتور Analysis Services

یکی از نواحی اصلی بهبودها در Analysis Services، در خود موتور Analysis Services بوده است. بیشتر این بهبودها، محدودیت‌هایی را که در نگارش SQL Server 2000 از Analysis Services وجود داشتند، برطرف کرده‌اند و سایرین، محصول را به سمت نواحی عملکرد کاملاً جدید سوق می‌دهند.

Section ۶۰،۰۱. پشتیبانی چند نمونه‌ای

در SQL Server 2000، Analysis Services پشتیبانی چند نمونه‌ای را فراهم نمی‌کرد، حتی اگر موتور پایگاه داده رابطه‌ای قادر به پشتیبانی تا ۱۶ نمونه بود. پشتیبانی چند نمونه‌ای، بخصوص برای تأمین کننده سرویس برنامه (ASP) هنگامی مفید است که چندین مشتری، هر یک نمونه‌های پایگاه داده خود را نگهداری کنند که همگی در یک سرور مشترک قرار دارند.

نگارش‌های قبلی Analysis Services واقعاً نمی‌توانستند در این موقعیت‌ها توزیع شوند. در SQL Server 2005، Analysis Services هم اینک پشتیبانی تا ۵۰ نمونه در هر سرور را فراهم کرده است. نمونه‌های SQL Server 2005 Analysis Services هم‌چنین می‌توانند برای اجرای پهلو به پهلو با نگارش‌های قبلی Analysis Services تنظیم شوند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۶۰،۰۲ پشتیبانی از کلاسترینگ Failover

SQL Server 2005 پشتیبانی خود را برای کلاسترینگ failover برای Analysis Services توسعه داده است. SQL Server 2000 Analysis Services از کلاسترینگ failover برای Analysis Services پشتیبانی نمی‌کرد. فرآیند نصب SQL Server 2005 Analysis Services مبتنی بر کلاستر بوده و می‌تواند بدون عیب و نقص Analysis Services را در گره‌های کلاستر نصب کند. در SQL Server 2005، کلاسترینگ failover هم اینک مبتنی بر Analysis Services است و SQL Server Agent و Notification Services کلاسترینگ failover را یک راه‌حل در دسترس بودن سطح سرور کامل کرده‌اند.

Section ۶۰،۰۳ یکپارچگی با .NET Framework

درست شبیه موتور پایگاه داده رابطه‌ای SQL Server 2005 که با .NET Framework یکپارچه است، Analysis Services نیز یکپارچگی .NET را فراهم کرده است. یکپارچگی Analysis Services .NET جدید، پشتیبانی کامل برای XML و SOAP[1] و پشتیبانی از ایجاد رویه‌های ذخیره شده و تریگرها را در زبان‌های .NET از قبیل Visual Basic و #C فراهم کرده است.

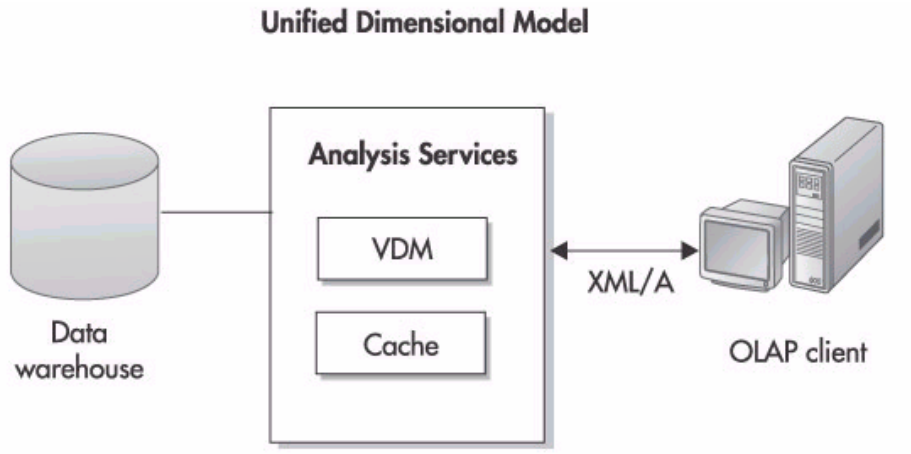
Section ۶۰،۰۴ Unified Dimensional Model

یکی از مهم‌ترین تغییرات Analysis Services در SQL Server 2005، معرفی مدل بُعدی یک شکل جدید است. UDM که می‌تواند به عنوان مرحله تحویل بعدی در پردازش OLAP ماورای مکعب‌ها به حساب آید، یک مدل گزارشگیری یک شکل را با ترکیب بهترین OLAP و گزارشگیری رابطه‌ای فراهم می‌کند. در فناوری‌های قبلی، برخی گزارشات رابطه‌ای از قبیل تولید سفارشات و صورتحساب‌ها با استفاده از ابزارهای OLAP پشتیبانی می‌شد و نمی‌توانست به خوبی با گزارشگیری رابطه‌ای سازگار شود. UDM یک زمینه عادی را فراهم می‌کند که می‌تواند هر دوی این انواع کاملاً متفاوت از نیازمندی‌ها را مدیریت کند. می‌توانید مروری سطح بالا از رابطه بین برنامه‌ها و UDM را در شکل ۱-۱۰ ببینید.

1- Simple Object Access Protocol

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

شکل ۱-۱۰ مدل بُعدی یک شکل (b)

در SQL Server 2005، مکعب لزوماً نمایش خارجی UDM است. در حالی که یک مکعب هنوز برای برنامه گزارشگیری تحت پوشش ارایه می‌شود، مکانیزم دستیابی داده کاملاً متفاوت است. UDM حاوی فوق داده‌ای است که قابلیت‌هایی را ممکن می‌سازد که توسط مکعب‌های MOLAP، ROLAP یا HOLAP که در SQL Server 2000 Analysis Services ارایه شده بودند، پردازش نمی‌شوند. در SQL Server 2005، برنامه‌های OLAP با استفاده از XMLA و پرس‌وجوی UDM متصل می‌شوند که می‌توانند مستقیماً روی هر دو منبع داده تحلیلی و رابطه‌ای ساخته شوند.

کش کردن کنشگرا

UDM می‌تواند به‌طور خودکار داده را کش کند، دستیابی داده شیوه MOLAP بسیار سریع را فراهم کند، بدون این که نیاز به تعریف صریح حافظه MOLAP باشد. با استفاده از کنترل‌های شیوه اسلایدر، می‌توانید تأخیر و حداکثر عمر داده در کش را کنترل کنید. یک تنظیم تأخیر صفر بدین معنی است که تمام داده‌ها به صورت داده MOLAP کش خواهند شد. اسلایدر lafttime داده، مدت زمان فعال بودن داده در کش را کنترل می‌کند. می‌توانید دوره‌های حیاتی فواصل متغیر از قبیل روزانه، هفتگی یا ماهانه و زمان انقضای فاصله در صورت پاک شدن کش، مشخص کنید. تحت این پوشش‌ها، کش از یک ساختار دیسکی استفاده می‌کند که شبیه یک مکعب MOLAP است.

کش کردن کنشگرا واقعاً نقاط تلاش توزیع مکعبی را سامان می‌بخشند که در SQL Server 2000 Analysis Services ارایه می‌شوند. در SQL Server 2000 Analysis Services، مکعبی باید پردازش شود (با داده پر شود) قبل از این که بتواند توزیع شود. برای مجموعه داده‌های بزرگ، این

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

میزان پردازش می‌تواند طولانی باشد. کش کردن کنشگرا، این مشکل را با دادن امکان توزیع مکعب‌ها قبل از پردازش آن‌ها حل کرده است. مکعب به‌طور خودکار هنگام صورت گرفتن درخواست‌هایی برای داده پر می‌شود.

Section ۶۰،۰۵ پشتیبانی از تریگر

الحاق پشتیبانی از تریگر در SQL Server 2005 Analysis Services، بهبود قابل پیش‌بینی دیگر است. تریگرهای Analysis Services شبیه همتاهای پایگاه داده رابطه‌ای می‌توانند هنگام رخ دادن عمل پایگاه داده خاصی، رویه‌های ذخیره شده را فعال کنند. تریگرهای Analysis Services به‌طور هم‌زمان اجرا می‌شوند، بدین معنی که این کار که تریگر را فعال می‌کند، تا وقتی که رویه ذخیره شده تریگر شده‌ای اجرا شود، بلوکه می‌شود.

Section ۶۰،۰۶ پشتیبانی ردیابی

بهبود مهم جدید دیگر در SQL Server 2005 Analysis Services، پشتیبانی از ردیابی است. رویدادهای ردیابی غیر هم‌زمان بوده و برای کنترل مسائل عیب‌یابی و کارآیی سیستم استفاده می‌شوند.

Section ۶۰،۰۷ پشتیبانی از اسکریپت‌نویسی

SQL Server 2005 Analysis Services هم اینک از ایجاد اشیا و پایگاه‌های داده Analysis Services از طریق اسکریپت‌نویسی پشتیبانی می‌کند. SQL Server 2005 زبان تعریف شئ (ODL) مبتنی بر XML جدید را فراهم کرده است که می‌تواند برای ایجاد، اصلاح و حذف اشیای پایگاه داده Analysis Services استفاده شود. ODL هم‌چنین می‌تواند چنین اعمال سروری از قبیل پردازش مکعب و مقایسه نگارش‌های پایگاه داده را مقداردهی کند.

Section ۶۰،۰۸ بهبودهای محلی‌سازی

مهم دیگر در موتور Analysis Services، پشتیبانی محلی‌سازی بهبود یافته است. موتور SQL Server 2005 Analysis Services قادر به ذخیره اطلاعات شیئی به چندین زبان است. این مسأله به برنامه‌های Analysis Services اجازه می‌دهد تا فوق داده مکعبی و داده تجاری را به زبان طبیعی کاربر نهایی نمایش دهد. موتور Analysis Services هم‌چنین از تنظیمات زبان پیش‌فرض برای برنامه‌های

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

کلاینت پشتیبانی می‌کند. این امر به برنامه کلاینت چند زمانی امکان استفاده خودکار از زبان مناسب را برای نمایش اطلاعات شئ می‌دهد.

Section۶۰.۰۹ پشتیبانی ردیف جدول واقعیت بی‌مرجع

محدودیت دیگر SQL Server 2000 Analysis Services که SQL Server 2005 آن را سامان بخشیده است، مسأله ردیف‌های بی‌مرجع است. SQL Server 2000 Analysis Services از ردیف‌هایی که دارای یک عضو تعریف نشده برای یک بُعد بودند، صرف‌نظر می‌کرد. این امر هنگام مقایسه با داده‌ای از منبع داده، جمع کل مکعب‌ها را اشتباه به دست می‌آورد. Analysis Services در SQL Server 2005 به شما اجازه می‌دهد تا نحوه مدیریت ردیف‌های جدول واقعیت را که دارای اطلاعات بُعد نیستند، توسط موتور Analysis Services مشخص کنید. می‌توانید ادامه کار را با صرف‌نظر از اطلاعات ناموجود یا تقویت Analysis Services برای ایجاد یک عضو بُعد ناشناخته برای یک ردیف جدول واقعیت که بدون اطلاعات بُعد است، انتخاب کنید.

Article LXI. بهبودهای مدیریتی Analysis Services

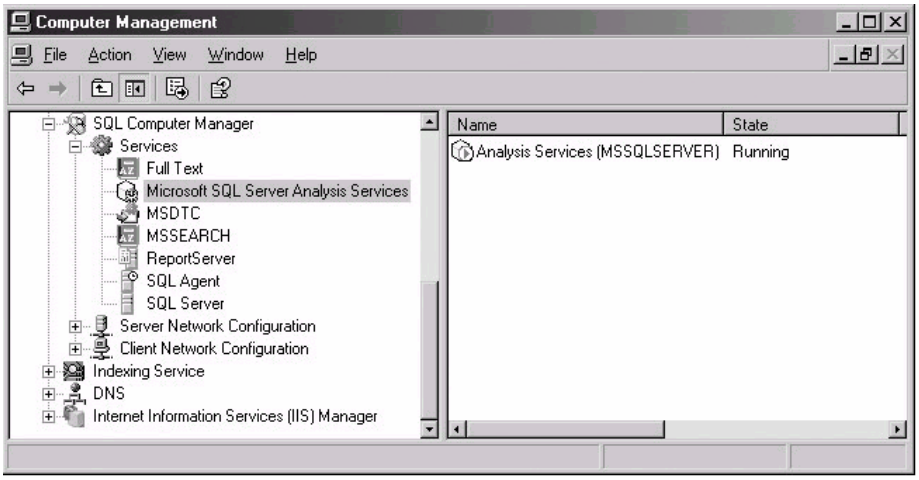
ابزارهای مدیریتی برای Analysis Services در SQL Server 2005 کاملاً تغییر کرده‌اند. در SQL Server 2000، Analysis Services با استفاده از Analysis Manager مدیریت می‌شد. در SQL Server 2005، Analysis Manager قدیمی از بین رفته است. این ابزار با SQL Server Computer Manager و SQL Server Management Studio جایگزین شده است.

Section۶۱.۰۱ SQL Server Computer Manager

SQL Server Computer Manager ابزاری برای استفاده جهت شروع و توقف سرویس Analysis Services است. با کلیک راست روی My Computer و سپس انتخاب گزینه Manage از منوی بازشو، به Computer Manager دستیابی دارید. برای شروع و متوقف کردن Analysis Services، گره Services and Applications را باز کرده و سپس گره SQL Computer Manager را باز کنید. نمایشی شبیه شکل ۲-۱۰ خواهیم داشت.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

شکل ۲-۱۰ SQL Server Computer Manager (b)

برای متوقف کردن، شروع و مکث دادن به سرویس Analysis Services، روی ورودی سرویس Analysis Services (MSSQL SERVER) که در پنجره سمت راست نشان داده شده است، کلیک راست کنید.

Section۲.۰۱.۶ SQL Server Management Studio

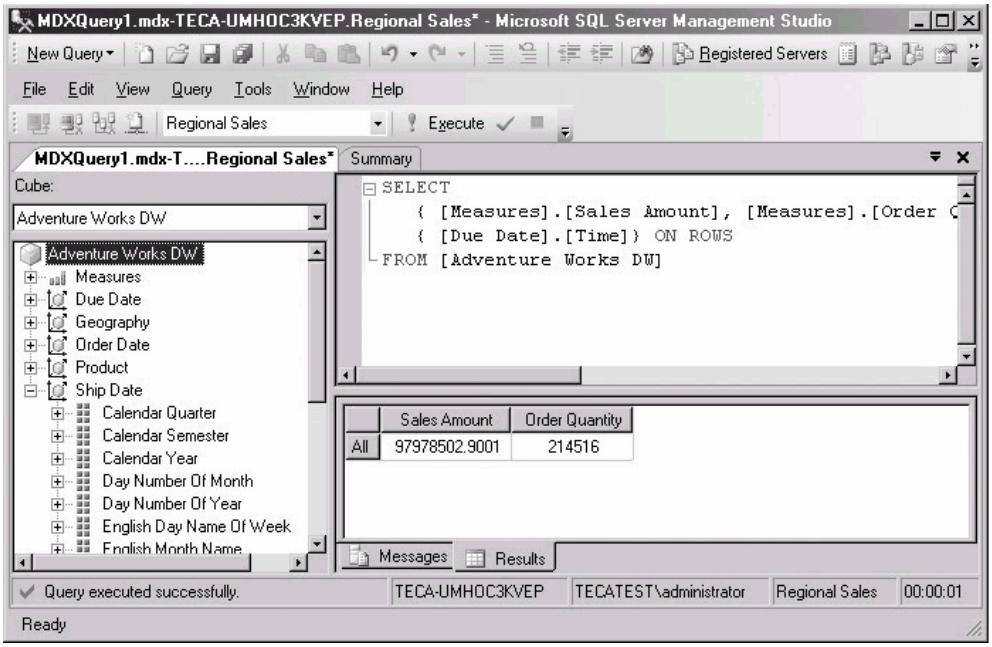
SQL Server Management Studio ابزار مدیریتی اصلی برای SQL Server و Analysis Services است. می‌توانید از SQL Server Management Studio برای انجام تعدادی از اعمال مدیریتی متفاوت استفاده کنید. این ابزار می‌تواند پایگاه‌های داده SQL Server را ایجاد کند، هم‌چنین می‌تواند پایگاه‌های داده Analysis Services را اسکریپت‌نویسی و توزیع کند. می‌توانید از SQL Server Management Studio برای تنظیم مجوزهایی برای اشیای Analysis Services برای کنترل دستیابی کاربر نهایی برای Analysis Services استفاده کنید.

Multidimentional Expression (MDX) Query Editor

علاوه بر این اعمال مدیریتی، SQL Server Management Studio دارای Multidimentional Expression (MDX) Editor است که می‌توانید برای نوشتن و اجرای پرس‌وجوهای ویژه و ساخت اشیای Analysis Services با استفاده از اسکریپت‌ها، از آن استفاده کنید. می‌توانید ویراستار MDX جدید را در شکل ۳-۱۰ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

(b)

شکل ۳-۱۰ SQL Server Management Studio MDX Editor

همان‌طور که در شکل ۳-۱۰ می‌بینید، ویراستار MDX جدید SQL Server 2005، به‌طور کامل از کلمات کلیدی کد رنگی و یک پنجره خروجی حاوی نتایج پرس‌وجوی MDX پشتیبانی می‌کند. هم‌چنین یک مرورگر فوق داده مکعبی یکپارچه وجود دارد که می‌توانید آن را در سمت چپ شکل ببینید.

Section ۶۱،۰۳ امنیت

امنیت دغدغه بزرگی برای مایکروسافت در دو سال گذشته بوده است و Analysis Services در این زمینه استثنا نیست. Analysis Services for SQL Server 2005 دارای بیش از ۱۰۰ بهبود امنیتی است که برخی از مهم‌ترین آن‌ها در بخش بعد لیست شده‌اند.

امنیت پیش‌فرض

Analysis Services به‌طور پیش‌فرض طوری طراحی شده است که ایمن باشد. ابتدا، سرویس برای اجرا با حداقل امتیازات نصب می‌شود، در صورتی که سیستم پیچیده باشد، عرضه شرکت شما کاهش می‌یابد. سپس، هنگامی که محصول برای اولین بار نصب می‌شود، تمام گزینه‌های امنیتی به‌طور

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

پیش‌فرض فعال می‌شوند. ضمناً، تمام ویژگی‌هایی که ممکن است سیستم را در معرض خطرات قرار دهند، به‌طور پیش‌فرض غیرفعال می‌شوند. این موارد عبارتند از:

- غیرفعال کردن دستیابی HTTP
- غیرفعال کردن اتصالات بی‌نام
- غیرفعال کردن رویه‌های ذخیره شده
- غیرفعال کردن پرس‌وجوهای با مجموعه ردیف باز

اگر آن‌ها مورد نظر باشند، تمام این ویژگی‌ها به سادگی می‌توانند توسط راهبر فعال شوند، ولی آن‌ها غیرفعال می‌شوند.

رمزگذاری

رمزگذاری نیز برای بهبود امنیت در Analysis Services for SQL Server 2005 استفاده می‌شود. رمزگذاری هم اینک در چندین ناحیه ارایه می‌شود: اولین ناحیه، کانال ارتباطات بین برنامه کلاینت و سرور که می‌تواند رمزگذاری شود و سرور می‌تواند برای دستیابی تنها به اتصالات رمزگذاری شده پیکربندی شود. مکعب‌های محلی رمزگذاری می‌شوند، به علاوه این که فایل‌های پشتیبان Analysis Services نیز می‌توانند به‌طور اختیاری رمزگذاری شوند.

امتیازات راهبری سنجیده

معرفی امتیازات راهبری سنجیده، بهبود دیگر مربوط به امنیت Analysis Services در SQL Server 2005 است. در SQL Server 2000، نیاز به عضوی از گروه OLAP Administrators برای اعمال تغییرات در پایگاه داده و پیکربندی Analysis Services دارید. امتیازات راهبری جدید در SQL Server 2005 Analysis Services به شما امکان ایجاد راهبران متفاوت برای هر پایگاه داده را می‌دهند. مجوزهای جدیدی برای خواندن فوق داده و مجوز دیگری وجود دارند که به یک حساب امکان پردازش مکعب‌ها را می‌دهند.

Section ۶۱.۰۴ بهبودهای پشتیبان‌گیری و بازیابی

پشتیبان‌گیری و بازیابی Analysis Services هم‌چنین در SQL Server 2005 بهبود یافته‌اند. بهبودهای پشتیبان‌گیری جدید شامل حذف حد پشتیبان‌گیری 2GB، توانایی فشرده‌سازی و رمزگذاری پشتیبان‌گیری و توانایی بازیابی آسان یک پشتیبان برای یک نمونه متفاوت از Analysis Services هستند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

حذف حد پشتیبان‌گیری 2GB

پشتیبان‌های SQL Server 2000 Analysis Services محدود به فایل‌های پشتیبان کمتر از 2GB بود. این حد در SQL Server 2005 افزایش یافته است و پشتیبان‌های SQL Server 2005 از فایل‌های تا حد NTFS، 16TB پشتیبانی می‌کنند.

توانایی فشرده‌سازی و رمزگذاری پشتیبان

پشتیبان Analysis Services هم اینک قادر به فشرده‌سازی و رمزگذاری فایل‌هایی است که پشتیبان‌گیری می‌شوند. این گزینه‌ها، مقداری زمان را به پنجره مورد نیاز برای انجام پشتیبان‌گیری اضافه خواهند کرد، ولی این امر موجب کوچک‌تر و ایمن‌تر شدن پشتیبان می‌شود. هم‌چنین گزینه صرف‌نظر از پشتیبان‌گیری اطلاعات امنیتی را دارید.

توانایی پشتیبان‌گیری و بازیابی آسان برای نمونه متفاوت

SQL Server 2005 Analysis Services از توانایی پشتیبان‌گیری پایگاه‌های داده Analysis Services پشتیبانی می‌کند و هم‌چنین می‌توانید از پایگاه‌های داده SQL Server رابطه‌ای پشتیبان‌گیری کرده و از آن‌ها برای نمونه متفاوتی از Analysis Services بازیابی کنید. این امر موجب انتقال سریع پایگاه‌های داده Analysis Services بین سرورها می‌شود.

Article LXII. بهبودهای برنامه‌نویسی

Analysis Services تعدادی از بهبودهای برنامه‌نویسی را در SQL Server 2005 هموار کرده است. Business Intelligence Development Studio جایگزین Analysis Manager به عنوان ابزار برنامه‌نویسی اصلی شده است. علاوه بر این، Analysis Services هم اینک با استفاده از یک پروتکل کاملاً جدید XMLA مورد دستیابی قرار می‌گیرد و دو چارچوب کاری شیئی NET. جدید وجود دارد: یکی برای مدیریت (AMO)[1] و دیگری برای نوشتن برنامه‌ها (ADO MD.NET).

Section ۶۲.۰۱ Business Intelligence Development Studio

Business Intelligence Development Studio جدید، ابزار اصلی برای نوشتن برنامه‌های Analysis Services است. در SQL Server 2005 یک ترسیم واضح بین Analysis Services

1- Analysis Management Objects

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Management و برنامه‌نویسی پایگاه‌های داده Analysis Services وجود دارد. در حالی که SQL Server Management Studio برای مدیریت Analysis Services استفاده می‌شود، Business Intelligence Development Studio برای نوشتن راه‌حل‌های Analysis Services استفاده می‌گردد. Business Intelligence Development Studio شبیه SQL Server Management Studio، برطبق پوسته Visual Studio است.

حالات Online و Offline

برخلاف نگارش قبلی Analysis Manager که همیشه در یک حالت متصل کار می‌کند، Business Intelligence Development Studio جدید در حالات Online و Offline کار می‌کند. به‌طور پیش‌فرض، Business Intelligence Development Studio در یک حالت Offline کار می‌کند. در این حالت، Business Intelligence Development Studio به سرور Analysis Services متصل نیست و تمام تغییرات و اشیایی که تعریف می‌کنید، در محیط برنامه‌نویسی هستند، تا وقتی که بخواهید پروژه را توزیع کنید. هنگامی که پروژه‌ای توزیع می‌شود، Business Intelligence Development Studio یک اسکریپت توزیع AMO را ایجاد و اجرا می‌کند. می‌توانید پیشرفت این اسکریپت را در پنجره خروجی ردیابی کنید که در پایین پنجره Business Intelligence Development Studio نشان داده می‌شود.

در مقایسه، حالت Online بسیار شبیه Analysis Manager قدیمی کار می‌کند که تغییراتی را ذخیره می‌کند که در Business Intelligence Development Studio بلافاصله اعمال می‌کنید و موجب بهنگام شدن پایگاه داده Analysis Services در سرور می‌شود. می‌توانید با انتخاب گزینه File | Connect To Analysis Services Database در منوی Business Intelligence Development Studio، از حالت Offline به حالت Online بروید.

منابع داده و دیدگاه‌های Data Source

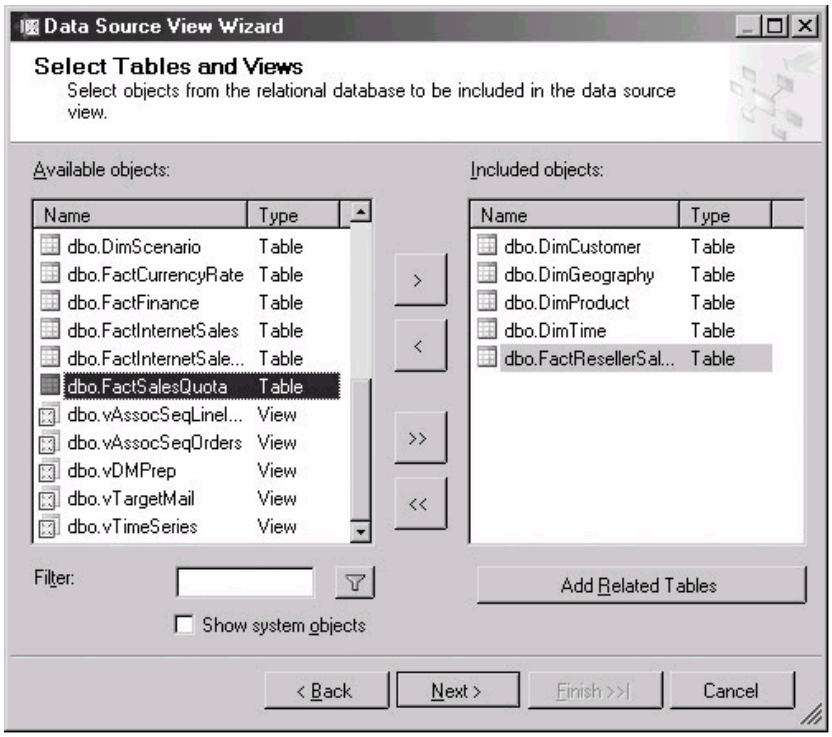
اولین مرحله در ایجاد پروژه‌های SQL Server 2005 Analysis Services، انتخاب منبع داده و ایجاد یک دیدگاه منبع داده است. منبع داده Analysis Services، بسیار شبیه یک منبع داده رابطه‌ای لزوماً سرور و پایگاه داده را تعریف می‌کند که داده از آن‌جا ناشی می‌شود و اطلاعات تعیین هویت را کپسوله می‌کنند. یک منبع داده با کلیک راست روی گره Data Source در پنجره Solution Explorer و سپس انتخاب گزینه New Data Source برای شروع ویزارد Data Source تعریف می‌شود که شما را

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

از طریق فرآیند انتخاب سرور و پایگاه داده مناسب راهنمایی می‌کند. Analysis Services از اتصالات پایگاه داده به SQL Server، Oracle، DB2 و پایگاه‌های داده Teradata پشتیبانی می‌کند.

بعد از ایجاد منبع داده، یک دیدگاه منبع داده را برای تعریف جداول واقعیت و بُعد تعریف کنید که باید برای انقیاد مکعب خود استفاده کنید. برای ایجاد یک دیدگاه منبع داده جدید، روی گره Data Source View در پنجره Solution Explorer کلیک راست کرده و سپس گزینه New Data Source View را انتخاب کنید. این امر موجب شروع Data Source View Wizard می‌شود. اولین مرحله در این ویزارد به شما اجازه انتخاب منبع داده مناسب را می‌دهد. بعد از انتخاب منبع داده، سپس جداول واقعیت و بُعد را با استفاده از کادر محاوره‌ای Select Tables And Views انتخاب کنید که در شکل ۴-۱۰ نشان داده شده است.



(a)

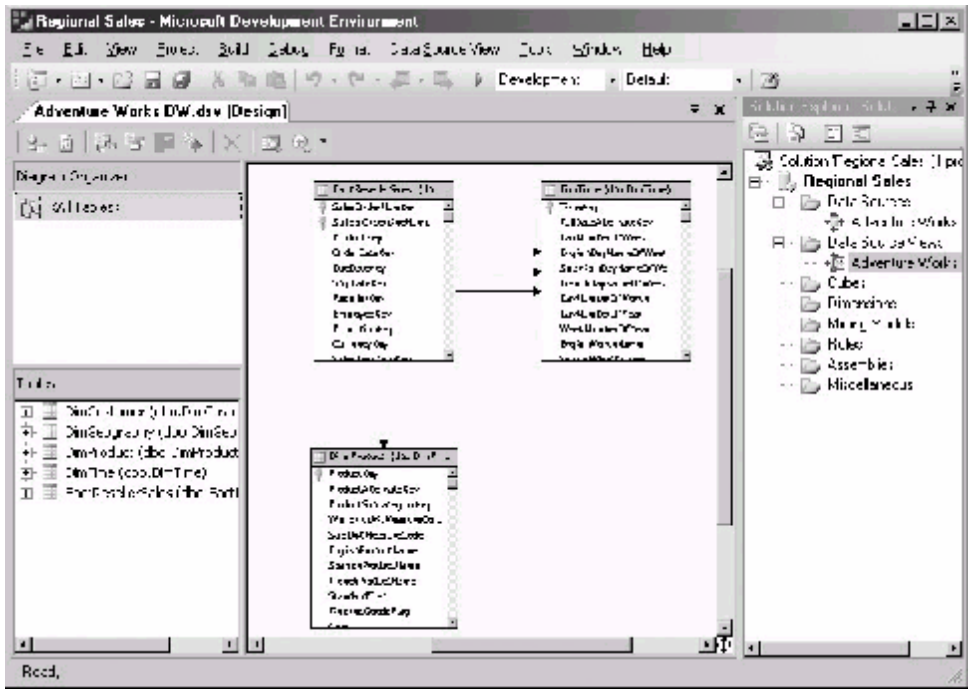
شکل ۴-۱۰ ایجاد یک دیدگاه منبع داده (b)

کادر محاوره‌ای Select Tables And Views مربوط به Data Source View Wizard به شما امکان می‌دهد تا به‌طور اختیاری لیستی را برای نشان دادن زیرمجموعه‌ای از پایگاه داده موجود فیلتر

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

کنید که می‌تواند هنگام سروکار داشتن با تعداد جداول زیاد مفید باشد. برای انتخاب جداول و دیدگاه‌ها، روی آن‌ها در ستون سمت چپ کلیک کنید. این عمل موجب پر شدن لیست Included Objects می‌شود که می‌توانید در سمت راست شکل ببینید. هنگامی که Data Source View Wizard را به پایان رساندید، Data Source View Designer که در شکل ۵-۱۰ نشان داده شده است، به‌طور خودکار شروع می‌شود.



(c)

شکل ۵-۱۰ Data Source View Designer (d)

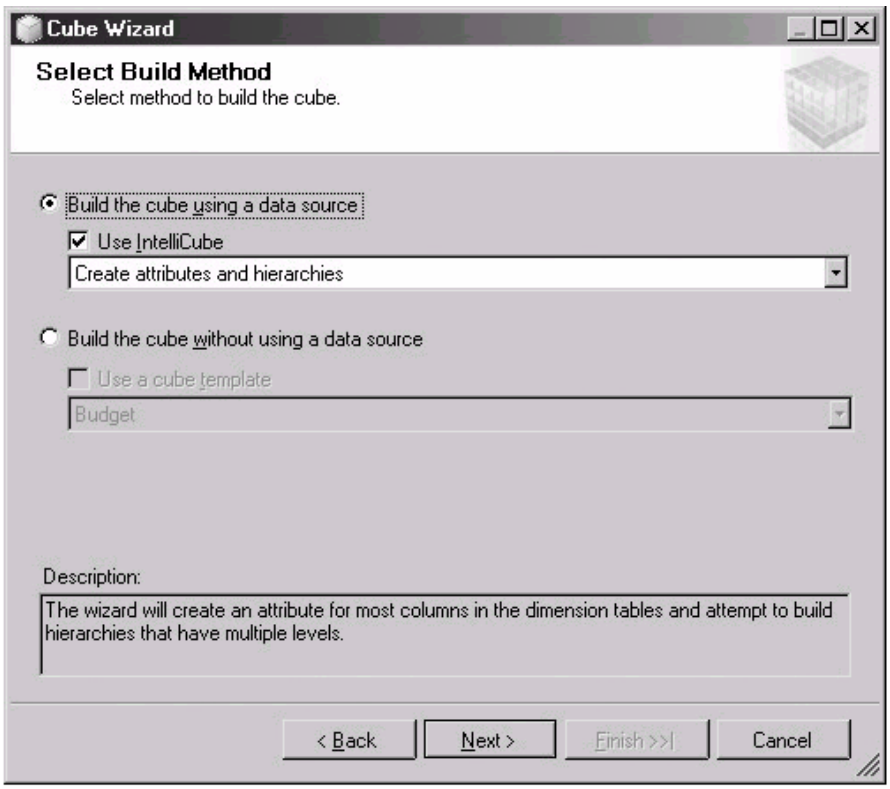
دیدگاه منبع داده لزوماً انتزاعی از یک یا چند منبع داده مرتبط است و Data Source View Designer به شما امکان می‌دهد اطلاعاتی را اختصاصی کنید که به دیدگاه منبع داده می‌روند. با استفاده از Data Source View Designer می‌توانید جنبه‌های مختلف را اختصاصی کرده و نحوه ارایه داده را کنترل کنید. برای نمونه، می‌توانید روابط پایگاه داده را تعریف کرده و تغییر دهید، نام جداول را تغییر دهید و ستون‌های محاسباتی را ایجاد کنید. تغییراتی که در دیدگاه منبع داده صورت گرفته‌اند، به منابع داده مرتبط برنمی‌گردند. آن‌ها تنها در دیدگاه منبع داده قرار دارند.

Cube Wizard

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

برای کمک به طراحی مکعب‌ها، Analysis Services دارای یک Cube Wizard کاملاً جدید است. بعد از ایجاد منابع داده و دیدگاه‌های منبع داده، Cube Wizard را با کلیک راست روی گره Cube در پنجره Solution Explorer و سپس انتخاب گزینه New Cube از منوی بازشو، اجرا کنید. این امر موجب شروع Analysis Services 2005 Cube Wizard می‌شود که در شکل ۶-۱۰ نشان داده شده است.



(e)

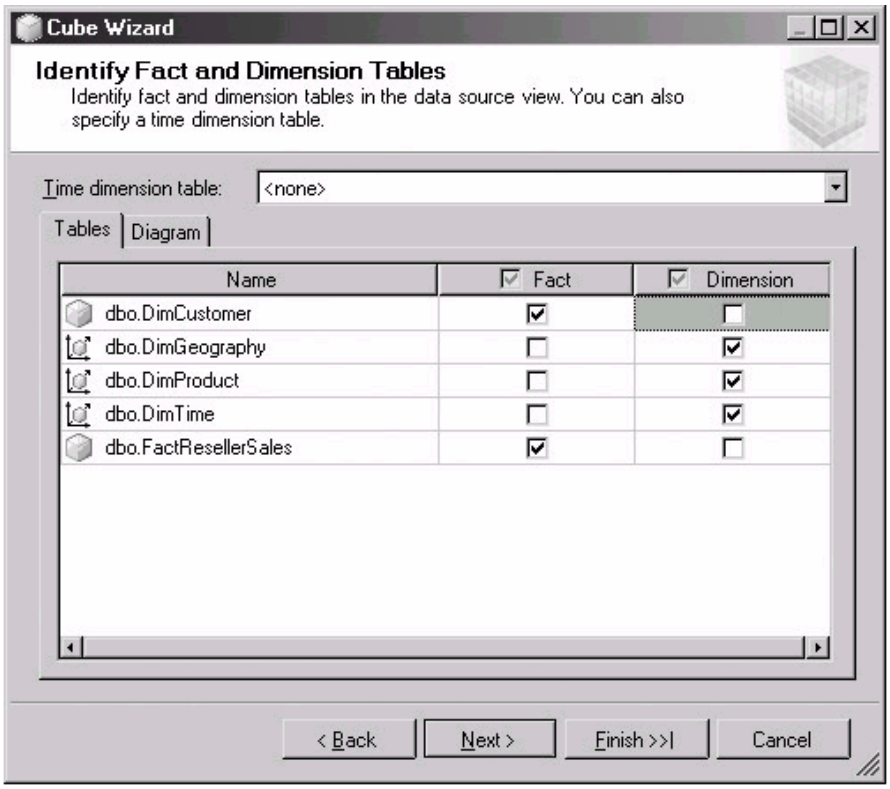
شکل ۶-۱۰ Cube Wizard (f)

Cube Wizard در SQL Server 2005 بسیار قدرتمندتر از Cube Wizard در نگارش Analysis Servicesای که بخشی از SQL Server 2000 بوده است. SQL Server 2005 Cube Wizard به شما امکان می‌دهد تا مکعبی را با الگوی پایین به بالا با انتخاب منبع داده و دیدگاه منبع داده‌ای که تعریف کرده‌اید یا به الگوی بالا به پایین با طراحی مکعب و فوق داده آن بسازید. در روش پایین به بالا، ویژگی IntelliCube جدید جداولی را تحلیل خواهد کرد که انتخاب شده‌اند و به‌طور خودکار جداول واقعیت و

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

بعد را برای تطبیق با صفات طرح‌واره آن‌ها پیشنهاد می‌کند. به روش دیگر، برای ساخت مکعب به الگویی بالا به پایین، می‌توانید گزینه Build The Cube Without A Data Source را انتخاب کنید. در این روش، به‌طور دستی تمام صفات مکعب را تعریف می‌کنید. می‌توانید نتایج انتخاب جدول IntelliCube را در شکل ۷-۱۰ ببینید.



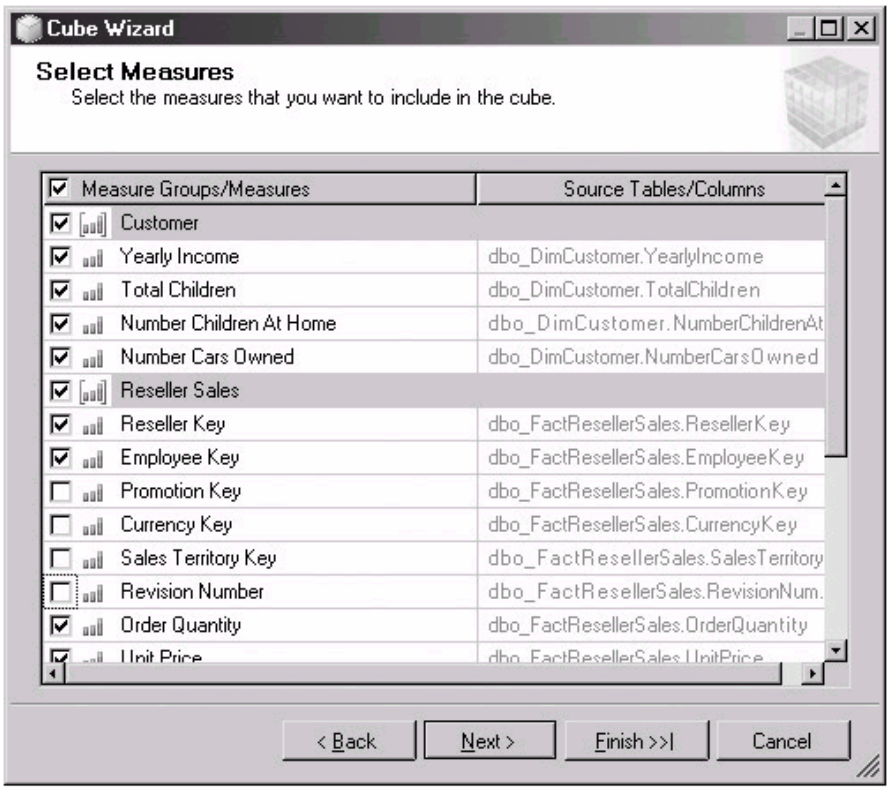
(g)

شکل ۷-۱۰ تعیین جداول Fact و Dimension (h)

IntelliCube یک کار بسیار خوب در مورد انتخاب خودکار جداول واقعیت و بُعد مناسب انجام می‌دهد، ولی کامل نیست و می‌توانید دسته‌بندی‌های جدول را تغییر دهید که IntelliCube تولید کرده است. بعد از انتخاب جداول مناسب، Cube Wizard شما را از طریق فرآیند انتخاب اندازه‌ها برای مکعب راهنمایی می‌کند. اصولاً، Cube Wizard تمام ستون‌های عددی را به عنوان اندازه‌های ممکن انتخاب می‌کند. سپس می‌توانید ستون‌هایی را انتخاب کنید که می‌خواهید استفاده کنید. می‌توانید کادر محاوره‌ای Cube Wizard Select Measures را در شکل ۸-۱۰ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(i)

شکل ۸-۱۰ انتخاب اندازه‌ها (j)

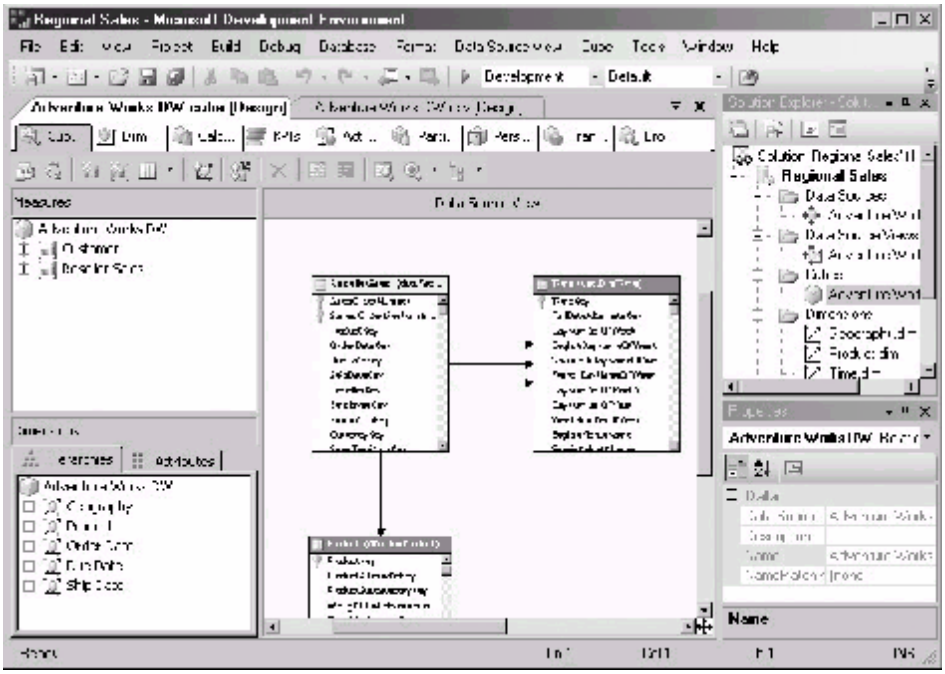
بعد از انتخاب جداول واقعیت و بُعد و تعریف اندازه‌های مناسب، Cube Wizard ظاهر داده را برای روابط ممکن نمونه‌سازی کرده و سلسله مراتب داده را ایجاد می‌کند.

Cube Editor

هنگامی که Cube Wizard تمام شد، Cube Editor به‌طور خودکار نمایش داده می‌شود. Analysis Services 2005 Cube Editor بسیار بهبود یافته است و ارتباط بزرگی از عملکرد ماورای ویژگی‌های فراهم شده در نگارش‌های قبلی Analysis Services ارایه می‌دهد. می‌توانید Cube Editor را در شکل ۹-۱۰ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(k)

شکل ۹-۱۰ Cube Editor (l)

SQL Server 2005 Analysis Services Cube Editor نه برگه مجزا را فراهم می‌کند و هر برگه به شما امکان می‌دهد تا با جنبه متفاوتی از مکعب کار کنید. نه دیدگاه مکعب که توسط Cube Editor فراهم شده است، عبارتند از:

- **Cube Builder** با اندازه‌های مکعب کار می‌کند.
- **Dimensions** با ابعاد مکعب کار می‌کند.
- **Calculations** با محاسباتی برای مکعب کار می‌کند.
- **KPIs** با Key Performance Indicators برای مکعب کار می‌کند.
- **Actions** با اعمال مکعب کار می‌کند.
- **Partitions** با پارتیشن‌های مکعب کار می‌کند.
- **Perspectives** با دیدگاه‌های مکعب کار می‌کند.
- **Translations** ترانزیشن‌هایی اختیاری برای مکعب تعریف می‌کند.
- **Browser** به شما امکان مرور مکعب توزیع شده را می‌دهد.

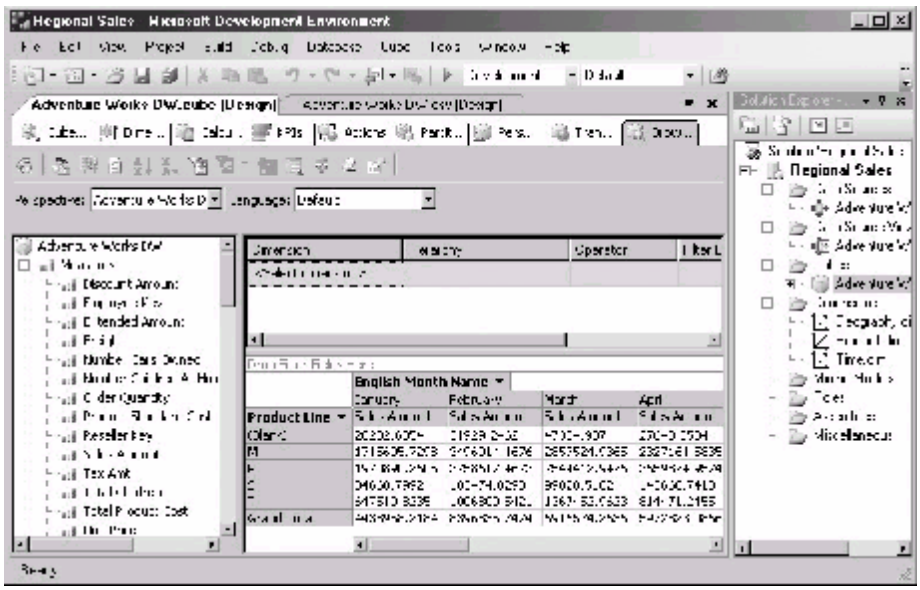
www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

بعد از تعریف پروژه، می‌توانید گزینه Build/Deploy Solution را برای ساخت مکعب در سرور Analysis Services انتخاب کنید. گزینه‌های پروژه کنترل می‌کنند آیا مکعب بعد از توزیع به سرور، پردازش خواهد شد. به‌طور پیش‌فرض، مکعب هنگام توزیع اولیه پردازش خواهد شد.

Cube Browser

هنگامی که مکعبی توزیع و پردازش شد، می‌توانید ابعاد و اندازه‌های مکعب را با استفاده از مرورگر تعبیه شده Cube Editor دیده و در بین آن‌ها حرکت کنید. می‌توانید مثالی از Cube Browser را در شکل ۱۰-۱۰ ببینید.



(m)

شکل ۱۰-۱۰ Cube Browser (n)

Cube Browser با استفاده از OWC[1] ساخته می‌شود. از Cube Browser با کشیدن ابعاد از صفات مکعب نشان داده شده در سمت چپ صفحه نمایش و انداختن در محورهای سطر و ستون OWC که در وسط صفحه نمایش نشان داده شده است، استفاده کنید. سپس اندازه‌های مناسب را انتخاب کرده و آن‌ها را در فیلد داده بیندازید. Cube Browser به‌طور خودکار داده را بازیابی کرده و آن را در پنجره مرورگر نمایش می‌دهد.

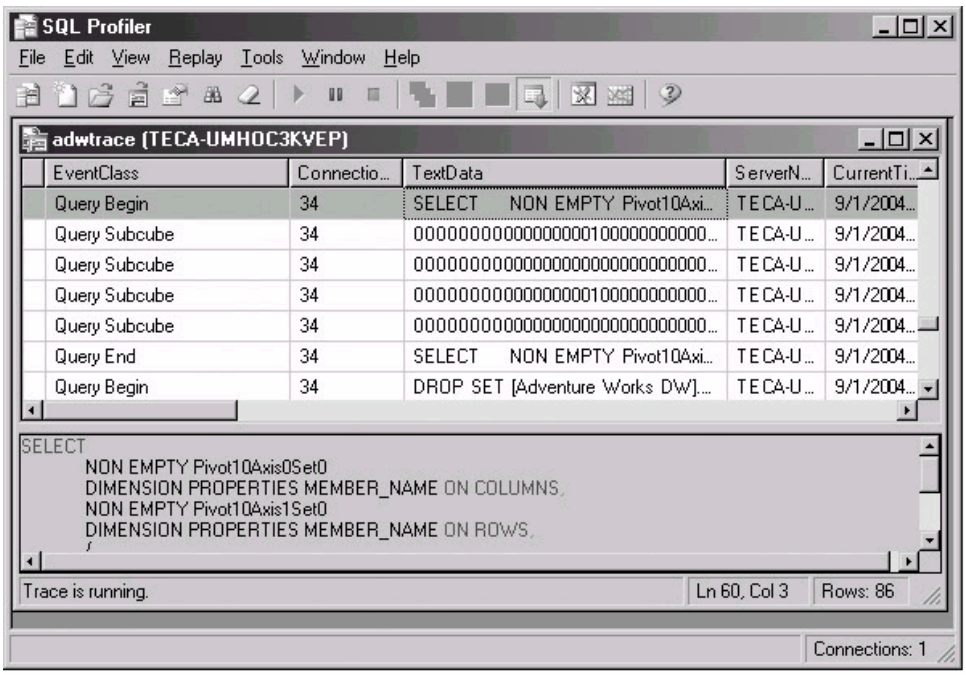
1- Office Web Component

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۶۲،۰۲ Profiler

در SQL Server 2005، Analysis Services دیگر یک سرور جعبه سیاه نیست. SQL Server 2000 Analysis Services و SQL Server 7 OLAP Services واقعاً هیچ روشی برای راهبر جهت مشاهده این مسأله که سرور چه کاری انجام می‌دهد، نداشتند. در SQL Server 2005، Profiler قادر به ردیابی تمام اعمال مختلفی است که در Analysis Services در حال اجرا هستند (این که چه کاری در پایگاه داده SQL Server رابطه‌ای انجام می‌شود). Profiler را با انتخاب گزینه Profiler در حال اجرا در Analysis Services در شکل ۱۱-۱۰ ببینید.



(a)

(b) **شکل ۱۱-۱۰ پروفایل کردن Analysis Services**

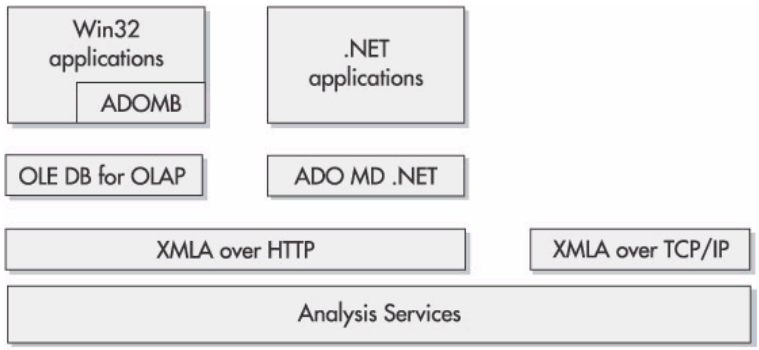
Profiler همچنین ابزار بزرگی برای یادگیری مطالبی درباره MDX است. علاوه بر این، Profiler همچنین یک ابزار عیب‌یابی قوی برای ردیابی فعالیت‌های سرور است. می‌توانید از Analysis Services Profiler برای گرفتن و انجام مجدد رویدادها در سرور استفاده کنید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۶۲،۰۳ XMLA[1]

XMLA یک پروتکل مستقل از محیط است که بر طبق سرویس‌های وب و SOAP است. Microsoft SQL Server 2005 Analysis Services از XMLA برای مدیریت تمام ارتباطات برنامه کلاینت با Analysis Services استفاده می‌کند. این امر شامل ابزارهای برنامه‌نویسی و مدیریتی مایکروسافت است: SQL Server Management Studio و Business Intelligence Development Studio. XMLA برای اینترنت بهینه شده است و برای کاهش رفت و برگشت‌ها به سرور طراحی شده است. در شکل ۱۲-۱۰ می‌توانید نحوه استفاده از XMLA توسط برنامه‌های مدیریتی و کلاینت OLAP برای اتصال به محیط Analysis Services را ببینید.



(a)

(b) **شکل ۱۲-۱۰ XMLA کلاینت‌ها را به Analysis Services متصل می‌کند.**

XMLA از دو نوع عملکرد پایه پشتیبانی می‌کند: اجرای درخواست‌ها و کشف درخواست‌ها.

اجرای درخواست‌ها

بدیهی است که اجرای درخواست‌ها یک عمل را انجام می‌دهد؛ آن‌ها حالت اشیا را در سرور تغییر می‌دهند. می‌توانید از اجرای درخواست‌ها برای ایجاد، تغییر و حذف اشیا و پردازش مکعب‌ها استفاده کنید.

کشف درخواست‌ها

کشف درخواست‌ها برای بازیابی اطلاعاتی درباره اشیا در سرور است. با استفاده از کشف درخواست‌ها، می‌توانید تعاریف امنیتی، پارتیشن و مکعب را بازیابی کنید. هم‌چنین می‌توانید حالت

1- XML for Analysis

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

سیستم را از سرور پرس‌وجو کنید تا مطالبی درباره تعداد اتصالات در حال استفاده و کاربرد منبع سرور بدانید.

Section ۶۲،۰۴ بهبودهای ODL

یکی از مهم‌ترین بهبودهای مدیریتی در Analysis Services for SQL Server 2005، ODL[1] جدید است. ODL همان نوع قابلیت‌های ایجاد شئ قابل اسکریپت‌نویسی برای Analysis Services را به همراه دارد که SQL Server رابطه‌ای همیشه داشته است. Analysis Services ODL به شما امکان نوشتن اسکریپت‌هایی را می‌دهد که می‌توانند به‌طور خودکار تمام اشیای پایگاه داده Analysis Services شما را ایجاد کنند. هم‌چنین می‌توانید از این اسکریپت‌ها برای کنترل نگارش‌گذاری پایگاه داده Analysis Services استفاده کنید.

Analysis Services ODL جدید SQL Server 2005 یک مشخصه باز است که در XMLA ساخته شده است. برطبق یک استاندارد باز، اسکریپت‌های SQL Server 2005 ODL می‌توانند با استفاده از یک ویراستار مبتنی بر XML ایجاد شوند.

Section ۶۲،۰۵ بهبودهای MDX

MDX زبان پرس‌وجوی اصلی برای پایگاه‌های داده Analysis Services باقی مانده است و در SQL Server 2005، MDX بهبودهای مهمی داشته است.

اسکریپت‌های MDX

یکی از بزرگ‌ترین تغییرات برای MDX در SQL Server 2005، توانایی گروه‌بندی چندین عبارت MDX در یک اسکریپت با یکدیگر است. اسکریپت‌نویسی، اجرای چندین عبارت MDX را به‌طور متوالی ممکن می‌سازد.

ساختار دستوری ساده شده

MDX هم‌اکنون یک ساختار دستوری کوتاه جدید برای اعضای محاسباتی است. این ساختار دستوری کوتاه جدید، ابعادی را برای تعیین اعضا در نظر می‌گیرد و آن را برای مشخص کردن صریح تکراری ابعاد در عبارات MDX غیرضروری می‌کند.

1- Object Definition Language

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

تبدیل نوع خودکار

تبدیلات نوع خودکار، تبدیل خودکار از یک عضو به یک زوج برای یک مجموعه و برعکس را ممکن می‌سازند. این امر نیاز به نوشتن کروشه‌ها و پرانتزها را هنگام مشخص کردن مجموعه‌ها برطرف کرده و MDX را ساده‌تر و خواناتر می‌کند.

مدیریت اعضای از بین رفته

بهبود دیگر در SQL Server 2005 MDX، توانایی مدیریت اعضای از بین رفته است. در SQL Server 2000، هنگامی که یک گزارش را تعریف می‌کردید و یک یا چند عضو مورد استفاده گزارش دیگر وجود نداشتند، گزارش ناموفق بود. فاکتورهای مختلف می‌توانند موجب این نوع شرط شوند، از جمله تغییرات صریح برای ساختار مکعب مرتبط و تغییرات کند در ابعاد. در MDX جدید SQL Server 2005، می‌توانید از خصوصیت بُعد MDXMissingMemberMode برای دادن امکان ادامه عملکرد به یک گزارش استفاده کنید، حتی هنگامی که اعضای یک بعد از بین رفته باشند.

انبوهه‌سازی اعضای شمارش مجزا

قابلیت دیگری که در SQL Server 2000 Analysis Services نبود و اضافه شده است، MDX در SQL Server 2005 است که هم‌اکنون امکان انبوهه کردن محتویات اعضای شمارش مجزا را دارد.

مجموعه‌ها در بخش WHERE

استفاده از مجموعه‌ها در بخش MDX WHERE، بهبود مهم دیگری برای Analysis Services در SQL Server 2005 است. استفاده از مجموعه‌ها در بخش MDX WHERE لزوماً شبیه استفاده از کلمه کلیدی OR در بخش SQL WHERE است. مجموعه‌ها به پرس‌وجو امکان برگرداندن نتایج از تمام اعضایی را که در بخش WHERE وجود دارند، می‌دهند.

Section ۶۲،۰۶ ADOMD.NET

ADOMD.NET یک تأمین کننده داده NET. طبیعی کاملاً جدید است که برای دستیابی به منابع داده چند بُعدی طراحی شده است. ADOMD.NET به عنوان جایگزینی برای کتابخانه شیئی دستیابی داده چند بُعدی ADO MD مبتنی بر COM قدیمی طراحی شده است. برنامه‌های کلاینتی که با استفاده از هر یک از زبان‌های NET. نظیر Visual Basic، C#، Managed C++ یا J# ساخته

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

می‌شوند، می‌توانند از ADOMD.NET برای بازیابی داده و اطلاعات فوق داده از SQL Server 2005 Analysis Services استفاده کنند. در پشت صحنه، ADOMD.NET از پروتکل XMLA برای اتصال به سرور Analysis Services استفاده می‌کند. برنامه‌های ADOMD.NET می‌توانند به دو روش به Analysis Server متصل شوند: XMLA تحت HTTP یا XMLA تحت TCP/IP.

Section ۶۲،۰۷ اشیای مدیریتی Analysis Services (AMO)

AMO چارچوب کاری شیئی کاملاً جدید دیگری است که مایکروسافت در SQL Server 2005 معرفی کرده است. AMO نسل بعدی DSO[1] مبتنی بر COM قدیمی است که در SQL Server 2000 فراهم شده بود. DSO کنار نگذاشته شده است. نگارشی از DSO در SQL Server 2005 وجود دارد، ولی در اصل برای سازگاری با قبل در دسترس قرار گرفته است. AMO با استفاده از .NET Framework ساخته شده و به‌طور انحصاری برای مدیریت Analysis Services طراحی شده است. AMO در سطح بالاتری از XMLA کار می‌کند و شبیه سایر چارچوب‌های کاری شیئی Analysis Services جدید، AMO از XMLA برای ارتباط با سرور Analysis Services استفاده می‌کند. SQL Server Management Studio و Business Intelligence Development Studio هر دو از AMO استفاده می‌کنند.

MAO از اتصالات مدیریتی ایمن برای Analysis Services پشتیبانی می‌کند. AMO از تعیین هویت Windows و یک کانال ارتباطی رمزگذاری شده بین برنامه کلاینت و سرور پشتیبانی می‌کند. AMO هم‌چنین اتصال به سرور را با فشرده‌سازی XMLای که بین کلاینت و سرور Analysis Services ارسال می‌شود، بهینه می‌کند.

AMO مزایای دیگری نیز نسبت به مدل DSO قدیمی دارد. اول این که، AMO به‌طور هوشمند اشیا را شمارش می‌کند و هنگامی که تعداد آیتم‌های زیادی را لیست می‌کند، کارآیی بهتری را ارایه می‌دهد. AMO هم‌چنین توانایی پشتیبان‌گیری و بازیابی سیستم را فراهم می‌کند. علاوه بر این، AMO تحلیل فشرده‌ای را برای برنامه‌های شما در دسترس قرار می‌دهد. یک تحلیل فشرده به برنامه شما امکان می‌دهد تا تعیین کنید کدام اشیای Analysis Services تحت تأثیر عمل معینی قرار خواهند گرفت. مثلاً، اگر عبارتی، ابعاد یک مکعب را تغییر دهد، می‌تواند نیاز به پردازش مجدد مکعب داشته باشد. تحلیل فشرده در نشان دادن تأثیرات این عبرات به شما کمک می‌کنند.

1- Decision Support Objects

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Article LXIII. Data Mining

Data Mining به سازمان‌ها اجازه می‌دهد تا اطلاعات مشتق شده از یک خط تجاری را به کار برید و از آن اطلاعات برای انجام پیش‌بینی‌هایی درباره تمایلات تجاری آتی استفاده کنید. پیش‌بینی‌های Data Mining می‌توانند به یک حرفه کمک کنند تا تصمیمات بهتری درباره مسیر آتی خود و نحوه کاربرد بهتر منابع بگیرند. SQL Server 2000 دو الگوریتم Data Mining بنیادی را فراهم کرده بود: درخت‌های تصمیم‌گیری و کلاسترینگ. برای این موارد، SQL Server 2005 چندین الگوریتم Data Mining جدید اضافه کرده است. الگوریتم‌های Data Mining که در SQL Server 2005 وجود دارند، عبارتند از: Decision Trees، Time Series، Sequence Clustering، Naïve Bayes و Association Rules.

Section ۶۳.۰۱ Decision Trees

الگوریتم Decision Trees (DT) مایکروسافت در اصل برای پیش‌بینی طراحی شده است. این الگوریتم برای پیش‌بینی متغیرهای پیوسته و گسسته استفاده می‌شود.

Section ۶۳.۰۲ Time Series

الگوریتم Time Series جدید مفهوم گذشته، حال و آینده را در کار پیش‌بینی معرفی می‌کند. این الگوریتم برای پیش‌بینی مراحل بعدی دنباله عددی طراحی شده است و نه تنها بهترین پیش‌بینی‌ها را برای یک هدف معین انتخاب می‌کند، بلکه هم‌چنین بهترین دوره‌های زمانی را برمی‌گزیند که باید با توجه به تأثیر هر عامل پیش‌بینی انتظار داشته باشید.

Section ۶۳.۰۳ Clustering و Sequence Clustering

الگوریتم Clustering برای یافتن یک شماره کلاستر خوب برای مدل شما با خصوصیاتی معین از داده آموزشی طراحی شده است. Sequence Clustering به شما اجازه می‌دهد تا کلاسترهای دنباله‌هایی از داده‌ها را بیابید. به عبارت دیگر، این الگوریتم Clustering حساس به ترتیب است.

Section ۶۳.۰۴ Naïve Bayes

الگوریتم Naïve Bayes جدید یک الگوریتم پیش‌بینی است. این الگوریتم برای کارآیی بسیار سریع طراحی شده است و روابط بین دسته‌های آیتم‌ها را پیش‌بینی می‌کند.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Section ۶۳،۰۵ Association Rules

الگوریتم Association Rules برای تحلیل داده تراکنشی طراحی شده است و برای یافتن گروه‌هایی از آیتم‌ها استفاده می‌شود که در یک تراکنش وجود دارند.

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

پیوست الف

نصب و ارتقا

در این پیوست، مروری بر گزینه‌های نصب و ارتقای SQL Server 2005 خواهیم داشت.

Article LXIV. ویرایش‌های SQL Server 2005

همانند نسخه‌های قبلی SQL Server، مایکروسافت ویرایش‌های مختلفی از SQL Server 2005 را فراهم کرده است. مایکروسافت این نگارش‌های SQL Server را فراهم کرده است:

- **SQL Server 2005 Enterprise Edition:** Enterprise Edition مجموعه ویژگی کاملی را فراهم می‌کند که در دسترس SQL Server 2005 است.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

- **SQL Server 2005 Standard Edition:** Standard Edition مجموعه ویژگی اصلی SQL Server را فراهم می‌کند، ولی تمام ویژگی‌های جهانی را فراهم نمی‌کند که از پایگاه‌های داده خیلی بزرگ و مقیاس‌پذیری عالی پشتیبانی می‌کنند.
- **SQL Server 2005 Developer Edition:** Developer Edition همان مجموعه ویژگی Enterprise Edition را فراهم می‌کند. هرچند، برای کاربرد تولیدی معتبر نیست.
- **SQL Server 2005 Express:** SQL Server 2005 Express جایگزینی برای MSDE است. شبیه MSDE، می‌توانید با آسودگی اجرا کنید. برخلاف MSDE، دارای کنترل کننده کار بار نیست.

Article LXV. نصب SQL Server 2005

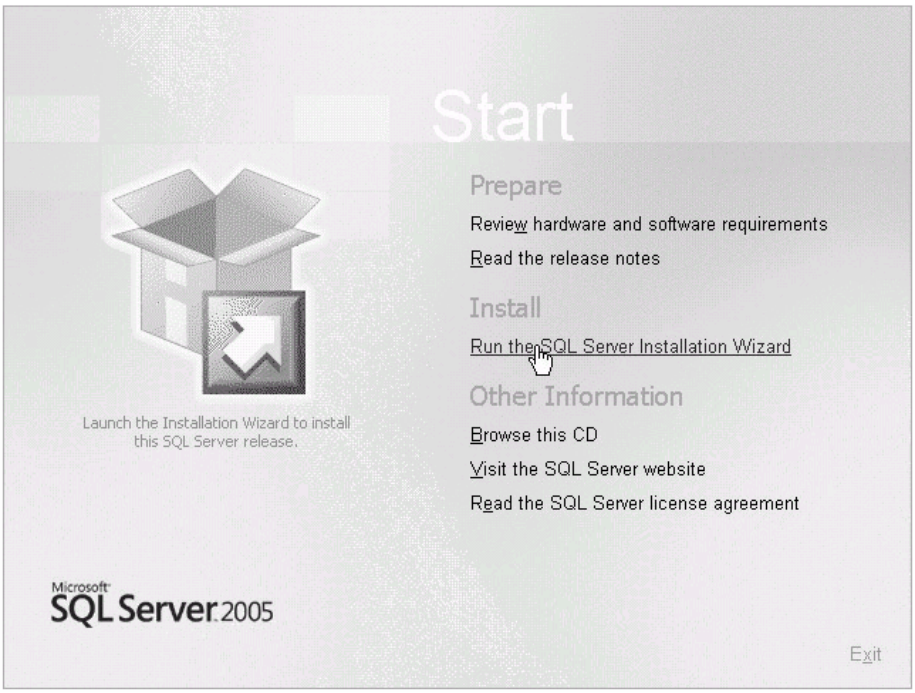
نصب SQL Server 2005 متفاوت از فرآیند نصب هر یک از نگارش‌های قبلی SQL Server بوده و تعدادی از مواردی را که در نصب نگارش‌های قبلی وجود داشت، برطرف کرده است. برخلاف برنامه‌های نصب SQL Server قبلی که مبتنی بر EXE. بود، برنامه نصب برای SQL Server 2005 روی Microsoft Installer (NSI) 3.0 استاندارد شده است. مبتنی بر MSI بودن موجب آسان‌تر شدن استفاده SQL Server برای نصب‌های غیرعمدی و برای توزیع نرم‌افزار با استفاده از SMS[1] Microsoft می‌شود. برنامه نصب جدید هم‌چنین نصب یک مرحله‌ای را برای کلاسترهای failover مایکروسافت فراهم می‌کند.

نصب SQL Server 2005 با استفاده از فایل Autorun روی CD یا با اجرای برنامه setup.exe شروع می‌شود. برنامه setup.exe با صفحه splash اولیه‌ای به کاربران خوشامد می‌گوید که می‌توانید در شکل الف-۱ ببینید.

1- System Management Server

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(a)

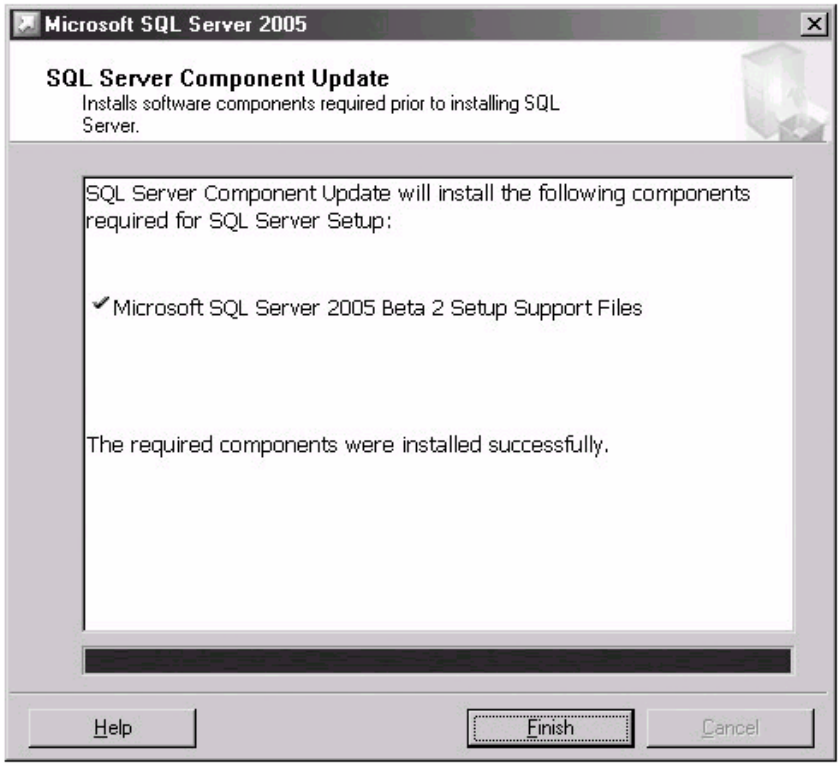
شکل الف-۱ صفحه splash نصب SQL Server 2005 (b)

کلیک کردن لینک SQL Server Installation Wizard موجب شروع نصب با ارایه EULA[1] به کاربر می‌شود. بعد از پذیرفتن توافق گواهینامه، صفحه نصب اولیه را خواهید دید که در شکل الف-۲ نشان داده شده است.

1- End User License Agreement

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(c)

شکل الف-۲ SQL Server Component Update (d)

کادر محاوره‌ای SQL Server Component Update بررسی می‌کند تا مطمئن شود که تمام اجزایی که برای نصب SQL Server 2005 مورد نیاز هستند، روی سیستم نصب می‌شوند. اگر اجزا پیدا نشوند، در سیستم کپی می‌شوند. این مرحله آخرین NET Framework. را نصب کرده و فایل‌های نصب را روی سیستم محلی کپی می‌کند. بعد از نصب تمام اجزای نصب مورد نیاز، کلیک کردن Finish موجب نمایش صفحه Welcome To The SQL Server Installation Wizard می‌شود که در شکل الف-۳ نشان داده شده است.

کادر محاوره‌ای Welcome لزوماً برای آگاه کردن شما از این موضوع می‌باشد که می‌خواهید نصب محصول SQL Server را شروع کنید. کلیک کردن کادر محاوره‌ای Welcome موجب نمایش کادر محاوره‌ای System Configuration Check می‌شود که در شکل الف-۴ نشان داده شده است.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

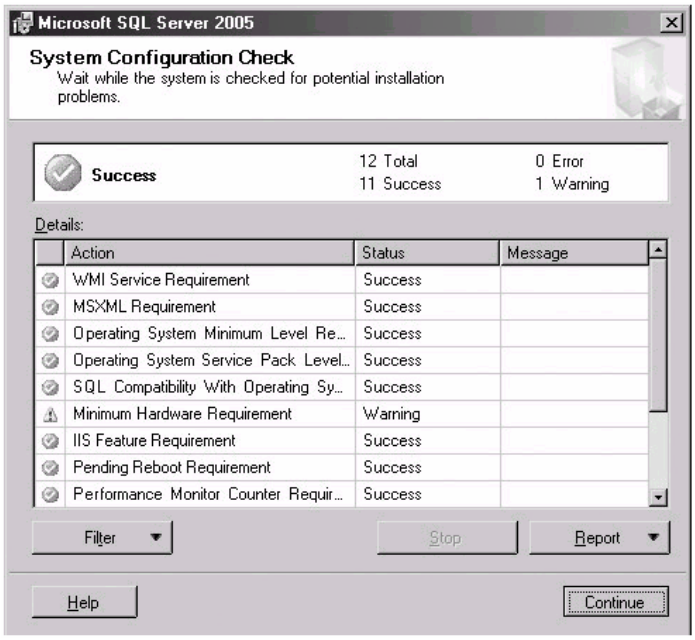
www.IrPDF.com

(e)

Welcome To The SQL Server Installation Wizard ۳-شکل الف (f)

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(g)

شکل الف-۴ System Configuration Check (h)

کادر محاوره‌ای System Configuration Check سیستم را پویش می‌کند تا مطمئن شود تمام الزمات نصب SQL Server 2005 را برآورده می‌کند. برنامه نصب مجموعه‌ای از بررسی‌های سیستمی را انجام می‌دهد تا مطمئن شود که سیستم قادر به اجرای SQL Server 2005 است. هنگامی که تمام آیتم‌ها بررسی شدند، برنامه نصب آن را با یک چک مارک سبز رنگ مشخص می‌کند تا موفقیت کار را نشان می‌دهد که یک خطای جدید وجود دارد. نصب می‌تواند با هشدارها جلو برود، ولی اگر یک X قرمز وجود داشته باشد، این آیتم باید اصلاح شود قبل از این که بتوانید نصب را ادامه دهید. انتخاب آیتم پرچم‌دار و کلیک کردن Report می‌تواند اطلاعات بیشتری درباره هر آیتم به شما بدهد. جدول الف-۱ بررسی‌های سیستمی را فهرست می‌کند که باید قبل از نصب SQL Server 2005 انجام شوند.

جدول الف-۱ بررسی‌های سیستمی (i)

شرط	شرح
WMI Service Requirement	سرویس WMI باید نصب شود.
MSXML Requirement	بهنگام‌رسانی QFE[1] .NET Framework. باید نصب شود.

1- Quick Fix Engineering

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Operating System Minimum Level Requirement	یکی از این سیستم‌های عامل مورد نیاز است: Windows 2000، Windows XP و Windows Server 2003.
Operating System Minimum Service Pack Requirement	یکی از این سطوح بسته خدماتی سیستم عامل مورد نیاز است: Windows 2000 SP4 و Windows XP SP1.
Minimum Hardware Requirement	سطوح سخت‌افزار حداقل مورد نیاز برای نصب SQL Server 2005 عبارتند از: CPU: Intel Pentium 700 MHz RAM: 128MB برای SQL Server Standard و Enterprise Editions مورد نیاز است. 64MB برای SQL Server Developer و Explress Editions مورد نیاز است. حافظه: حداقل 95MB برای SQL Server تا حداکثر 300MB مورد نیاز است، Analysis Services نیاز به 50MB و Repoting Service نیاز به 50MB دارد.
IIS Feature Requirement	IIS باید برای اجرای برنامه‌های SQLXML و Repoting Service نصب شود. اگر IIS نصب نشده باشد، نصب می‌تواند جلو رود، ولی برنامه Repoting Service برای نصب در دسترس نخواهد بود.
Pending Reboot Requirement	قبل از مراحل نصب، ممکن است نیاز به یک reboot سیستم باشد. اگر این‌طور باشد، آن‌گاه یک reboot سیستم باید انجام شود.
Performance Monitor Counter Requirement	کلیدهای رجیستری شمارنده کارآیی باید قادر به نمو صحیح باشند.
Default Installation Path Permission Requirement	کاربر اجرا کننده نصب باید دارای امتیازات راهبری باشد.
Internet Explorer Requirement	Internet Explorer 6.0 SP1 برای اجرای Business Intelligence Development Studio مورد نیاز است.
Manifest Requirement	نصب، این فایل‌ها را برای استفاده از MDAC 9.0 پیکربندی می‌کند: mmc.exe، wmiprvs.exe، msftefd.exe، inetinfo.exe، dllhost.exe، w3wp.exe و aspnet_wp.exe.

اگر تمام الزامات سیستم برآورده شوند، دکمه Continue فعال خواهد شد. کلیک کردن Continue موجب نمایش کادر محاوره‌ای Registration Information می‌شود که نیاز به وارد کردن

www.IrPDF.com www.IrPDF.com www.IrPDF.com

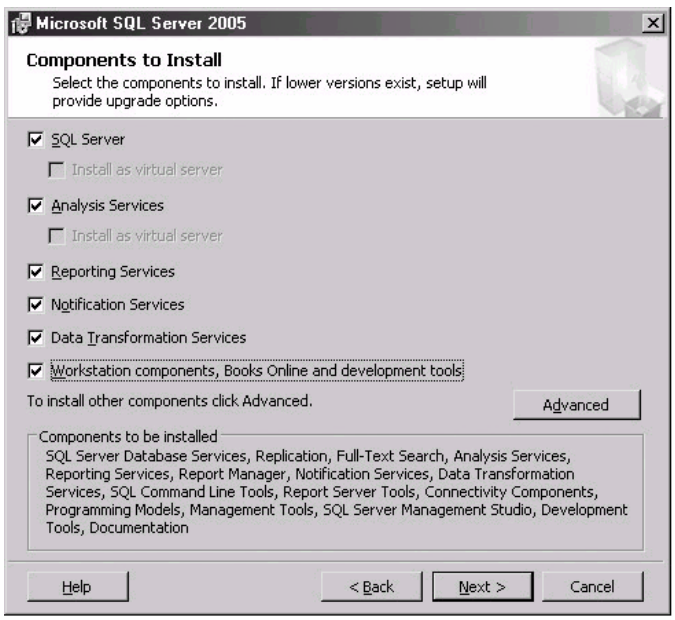
www.IrPDF.com

نام، نام شرکت و کلید محصول ۲۵ رقمی دارد. بعد از پر کردن این مقادیر و کلیک کردن Next، برنامه نصب کادر محاوره‌ای Components To Install را نمایش خواهد داد که در شکل الف-۵ نشان داده شده است.

کادر محاوره‌ای Components To Install به شما امکان می‌دهد تا اجزای SQL Server 2005 مورد نظر خود را برای نصب انتخاب کنید. اگر IIS در سیستم وجود ندارد، گزینه Repoting Service در دسترس نخواهد بود. اگر SQL Server 2005 را روی یک کلاستر نصب کرده‌اید، آن‌گاه گزینه Install As Virtual Server فعال خواهد شد.

تـــوجـه : برای نصب یک سیستم راهبری، فقط گزینه Workstation Components را انتخاب کنید.

در حالی که این سطح انتخاب جزء برای بیشتر کاربران کافی خواهد بود، هم‌چنین یک کادر محاوره‌ای انتخاب ویژگی سنجیده‌تر وجود دارد که توانایی انتخاب دقیق اجزایی را که نصب خواهند شد، به شما می‌دهد. می‌توانید کادر محاوره‌ای Feature Selection را با کلیک کردن Advanced نمایش دهید که در شکل الف-۶ نشان داده شده است.

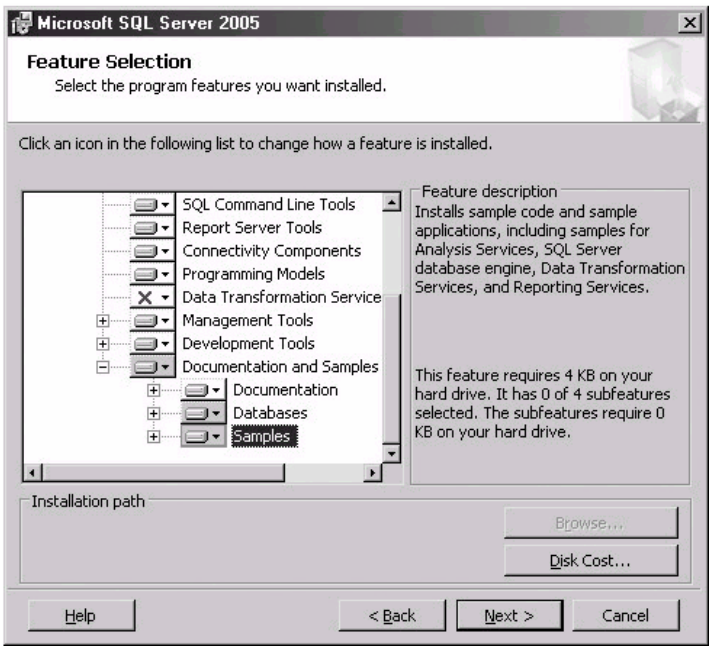


(j)

شکل الف-۵ Components To Install (k)

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(l)

شکل الف-۶ Feature Selection (m)

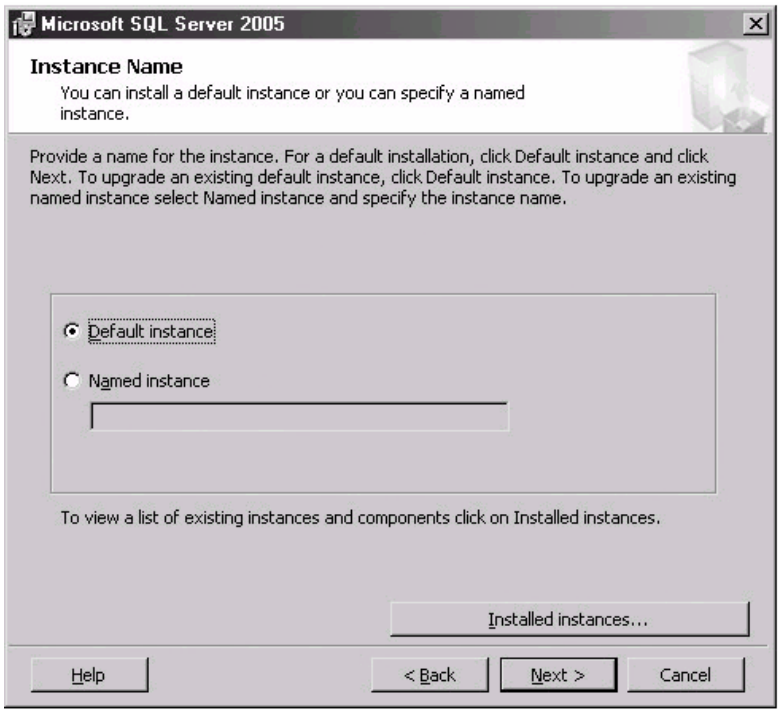
کادر محاوره‌ای Feature Selection به شما امکان می‌دهد تا در هر گزینه و جزء نصب وارد شده و ویژگی‌های گوناگون را کنار بگذارید.

توجه : به‌طور پیش‌فرض، هیچ برنامه یا پایگاه داده نمونه‌ای نصب نمی‌شود. اگر بخواهید پایگاه داده AdvantureWorks نمونه و یا نمونه‌های برنامه‌نویسی را نصب کنید، سپس باید از کادر محاوره‌ای Feature Selection برای انتخاب گزینه Databases And Samples استفاده کنید.

بعد از انتخاب اجزایی که برای نصب در نظر دارید، Next را برای نمایش کادر محاوره‌ای Instance Name کلیک کنید که می‌توانید در شکل الف-۷ ببینید.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com



(n)

شکل الف-۷ Instance Name (o)

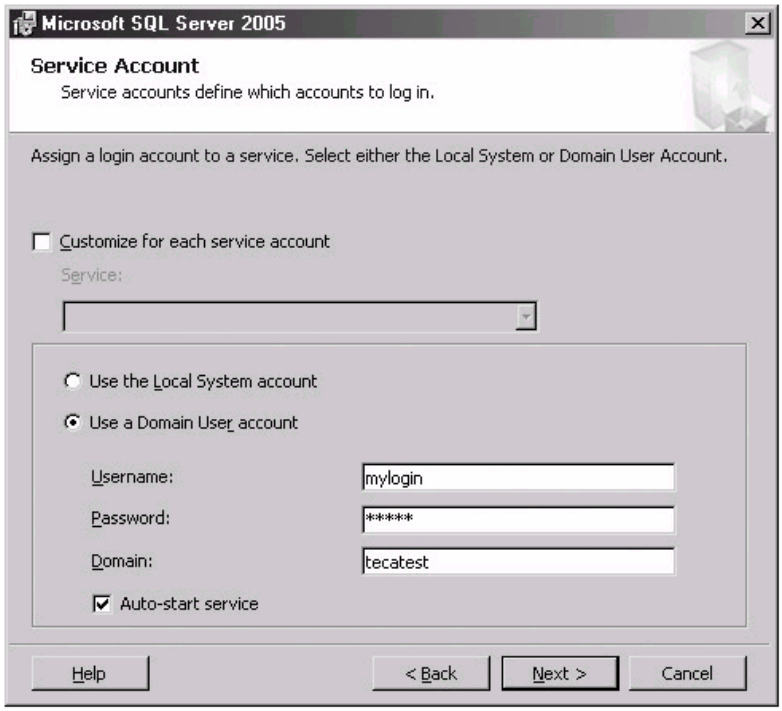
اگر این اولین باری است که برنامه نصب را اجرا می‌کنید، گزینه‌ای برای نصب نمونه پیش‌فرض یا نصب یک نمونه معین دارید. اگر نمونه SQL Server پیش‌فرض را نصب کرده‌اید، برنامه نصب آن نمونه را تشخیص خواهد داد و تنها گزینه‌ای را برای نصب یک نمونه معین ارایه می‌دهد. می‌توانید تا ۵۰ نمونه معین در یک سیستم داشته باشید. بیشتر پیاده‌سازی‌ها از یک نمونه تکی استفاده خواهند کرد، ولی نصب برای ISPها و ASPها اغلب بیشتر مورد نیاز خواهند بود. اگر یک نمونه معین را ایجاد کنید، هر نام باید ۱۶ کاراکتر یا کمتر باشد. نام نمونه‌ها حساس به حروف کوچک و بزرگ نیستند، ولی اولین کاراکتر نام باید یک حرف باشد. آن‌ها نمی‌توانند دارای فاصله باشند و نباید حاوی backslash (\)، کاما (,)، دو نقطه (:)، گیومه تکی (')، خط فاصله (-)، آمپرسند (&)، علامت (#) یا (@) باشند. نام‌های نمونه هم‌چنین نمی‌توانند حاوی کلمات رزرو شده "Default" یا "MSSQLServer" باشند.

توجه : اگر نام معینی را ایجاد کنید، نام سرویس SQL Server به این صورت گذاشته می‌شود: MSSQL$InstanceName (که InstanceName با نام نمونه‌ای که ایجاد کرده‌اید جایگزین می‌شود).

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

بعد از پذیرفتن نمونه پیش‌فرض یا ایجاد یک نمونه معین و کلیک کردن Next، کادر محاوره‌ای Server Account نمایش داده می‌شود که می‌توانید آن را در شکل الف-۸ ببینید.



(p)

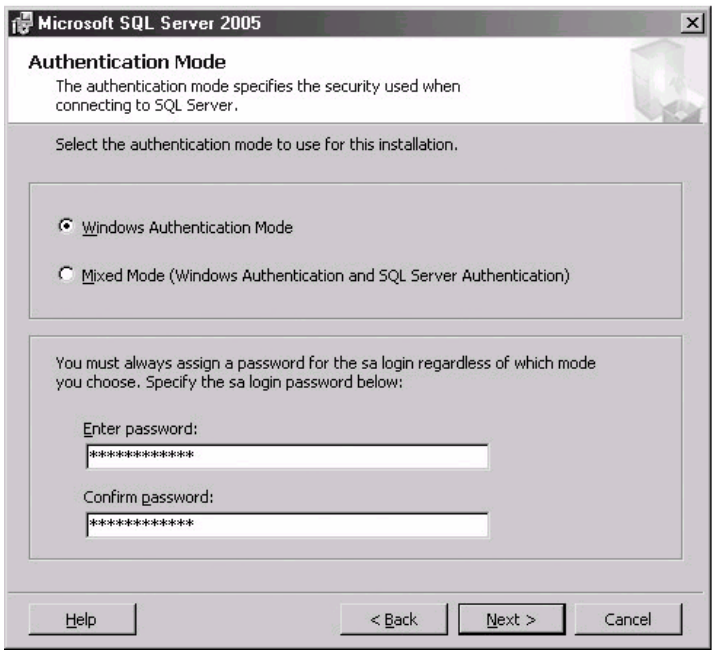
شکل الف-۸ Server Account (q)

مقادیر در کادر محاوره‌ای Server Account، حساب‌های کاربری مورد استفاده سرویس SQL Server و SQL Server Agent، Analysis Services و Repoting Server را مشخص می‌کنند. این یک انتخاب مهم است، زیرا مجوزهایی را که هر یک از این سرویس‌ها تحت آن اجرا می‌شوند، کنترل می‌کند. به‌طور پیش‌فرض، کادر محاوره‌ای Server Account شما را برای انتخاب یک حساب Domain User برای سرویس SQL Server راهنمایی می‌کند. می‌توانید از حساب راهبری استفاده کنید، ولی به دلیل سطح امتیاز بالای آن، این ایده خوبی نیست. همچنین می‌توانید حساب Local System را انتخاب کنید؛ هر چند، این حساب بسیار توانند است و مجوزهایی شبیه راهبر دارد، ولی در توانایی دستیابی به منابع شبکه محدود شده است. معمولاً، باید یک حساب کاربر حوزه را بخصوص برای SQL

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Server جهت اجرا تحت آن و انتخاب این حساب ایجاد کنید. این امر توانایی کنترل سنجیده‌تر مجوزهایی را که مالک سرویس‌های مختلف هستند، می‌دهد. بعد از تعیین حساب سرویس، برای نمایش کادر محاوره‌ای Authentication Mode که در شکل الف-۹ نشان داده شده است، روی Next کلیک کنید.



(r)

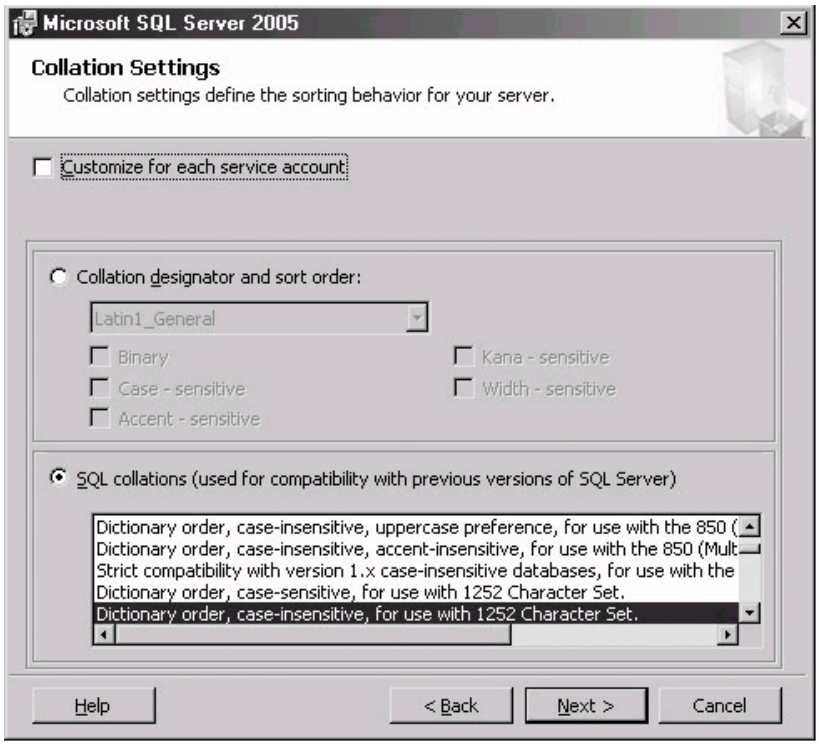
شکل الف-۹ Authentication Mode (s)

کادر محاوره‌ای Authentication Mode نوع تعیین هویت کاربری را تعریف می‌کنند که SQL Server 2005 استفاده خواهد کرد. مقدار پیش‌فرض Windows Authentication است، بدین معنی که حساب‌های کاربری Windows نیز در SQL Server به کار می‌روند. معمولاً، این چیزی است که می‌خواهید، زیرا مدیریت ساده‌تری را فراهم می‌کند که در آن تنها مجموعه‌ای از حساب‌های login باید مدیریت شوند و آن مجموعه loginها توسط سیستم عامل میزبان نگهداری می‌شوند. هم‌چنین ایمن‌تر است، زیرا باید تعیین هویت Mixed Mode را انتخاب کنید، بدین معنی که هم loginهای Windows و هم loginهای SQL Server پذیرفته می‌شوند. در مورد loginهای SQL Server، باید به‌طور دستی این loginها را به SQL Server اضافه کنید و آن‌ها به‌طور مستقل از Windows login نگهداری می‌شوند. بعد از انتخاب حالت تعیین هویت که سرور استفاده خواهد کرد، سپس باید کلمه عبوری را

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

برای sa login[1] SQL Server انتخاب کنید. به دلایل امنیتی، باید کلمه عبوری را انتخاب کنید که خالی نباشد. اکیداً باید در نظر داشته باشید که این کلمه عبور را قوی بسازید که حداقل هشت کاراکتر طول داشته باشد و حاوی کاراکترها، اعداد و کاراکترهای خاص باشد. کلیک کردن Next موجب نمایش کادر محاوره‌ای Collation Settings می‌شود که در شکل الف-۱۰ نشان داده شده است.



(t)

شکل الف-۱۰ Collation Settings (u)

کادر محاوره‌ای Collation Settings به شما امکان می‌دهد ترتیب مرتب‌سازی پیش‌فرضی را مشخص کنید که توسط SQL Server 2005 استفاده خواهند شد. در حالی که Collation Settings در طی نصب مشخص می‌شود، تطبیق پیش‌فرض SQL Server 2005 تنظیم می‌شود، ترتیب تطبیق هم‌چنین می‌تواند برای هر پایگاه داده مجزا تنظیم شود. اگر تطبیق‌های SQL را برای نصب Analysis

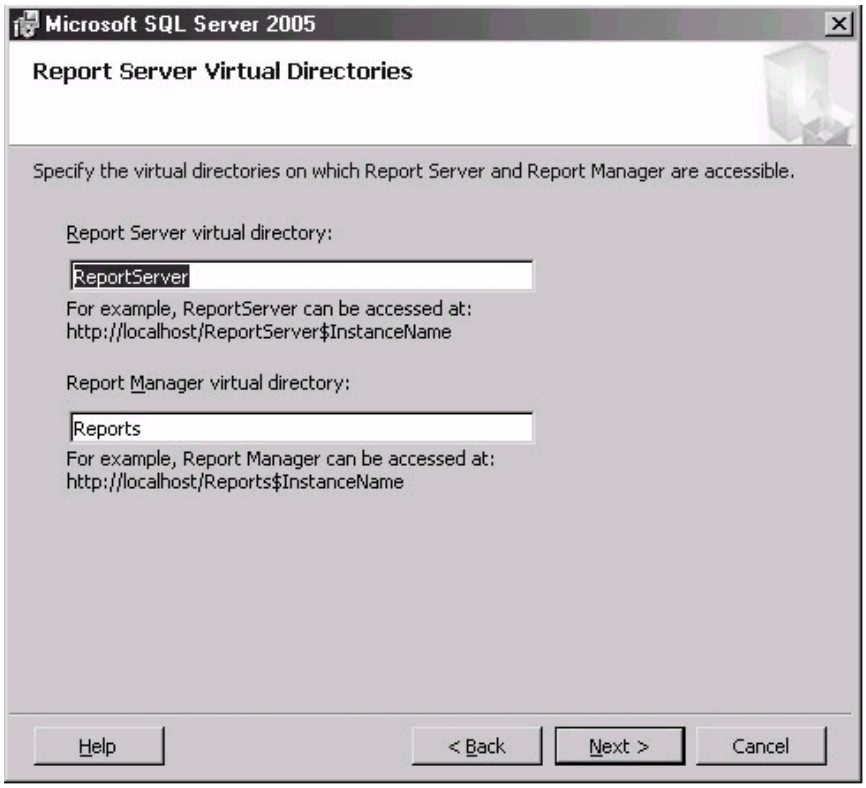
1- System Administrator

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Services انتخاب کنید، یک کادر محاوره‌ای نمایش داده خواهد شد که از شما می‌پرسد آیا می‌خواهید از تطبیق Latin1_General برای Analysis Services استفاده کنید.

اگر قصد نصب Reporting Service را دارید، کلیک کردن Next موجب نمایش کادر محاوره‌ای Report Server Virtual Directories می‌شود که در شکل الف-۱۱ نشان داده شده است.



(v)

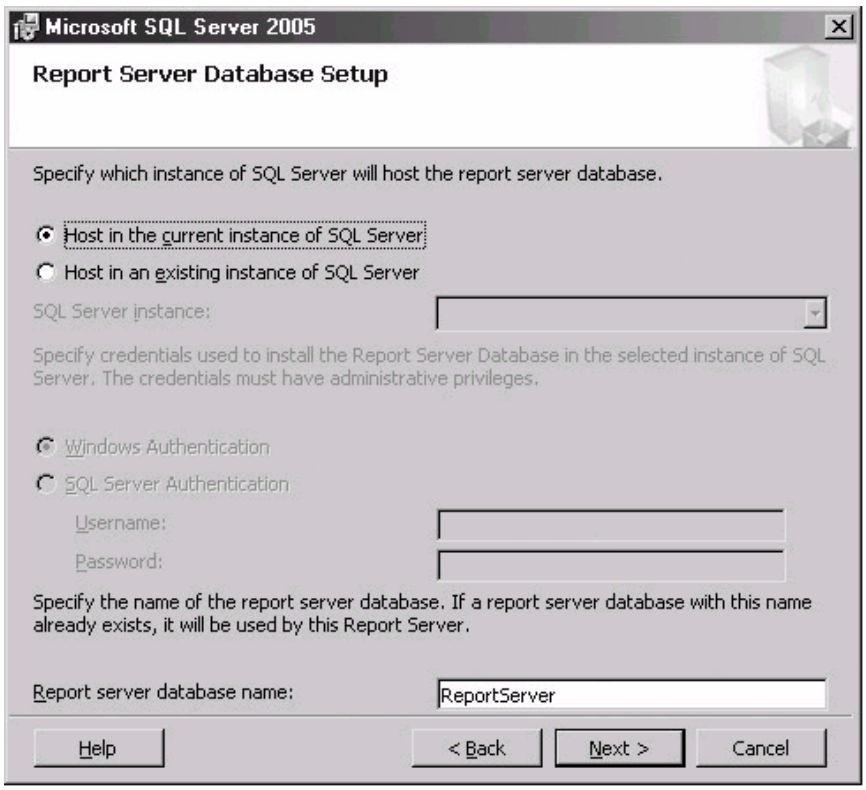
شکل الف-۱۱ Report Server Virtual Directories (w)

کادر محاوره‌ای Report Server Virtual Directories تنها در صورتی نمایش داده می‌شود که جزء Repoting Service را در کادر محاوره‌ای Components To Install نصب کنید که قبلاً در شکل الف-۵ نشان داده شد. کادر محاوره‌ای Reporting Services Virtual Directories به شما امکان می‌دهد تا دایرکتوری مجازی IIS را انتخاب کنید که برای انتشار گزارشات Reporting Service استفاده خواهد شد و Virtual Directory که می‌تواند برای مدیریت Reporting Service استفاده شود. اگر با نمونه SQL Server پیش‌فرض کار می‌کنید، نام‌های ReportingServer و Reporting استفاده

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

خواهند شد. هنگامی که نمونه معینی را ایجاد می‌کنید، برای ایجاد نام‌های دایرکتوری‌های مجازی انتشار و مدیریت نام نمونه به ثابت‌های ReportServer$ یا Reporting$ ضمیمه می‌شود. هنگامی که نام دایرکتوری‌های مجازی Reporting Service را مشخص کرده باشید، کلیک کردن Next موجب نمایش کادر محاوره‌ای Report Server Database Setup می‌شود که در شکل الف-۱۲ نشان داده شده است.



(x)

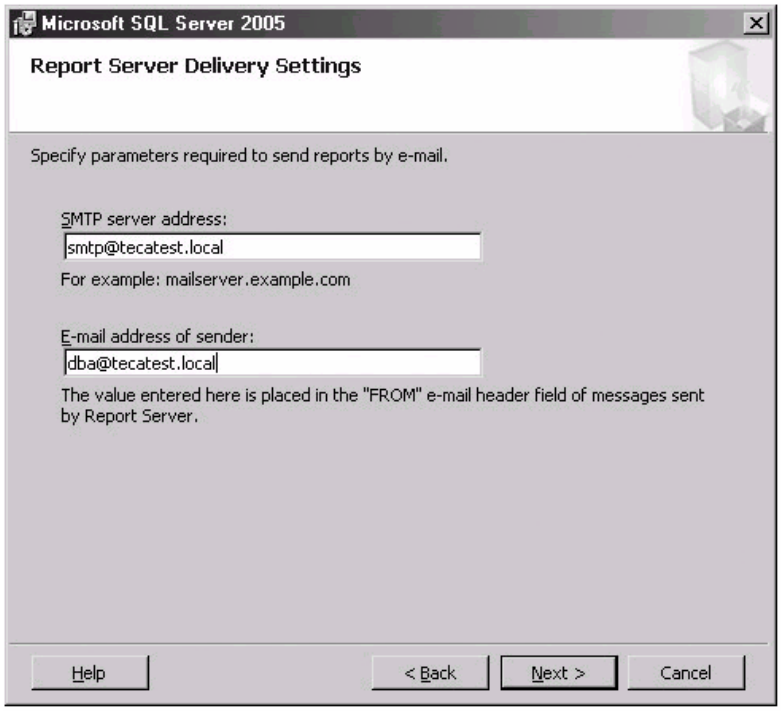
شکل الف-۱۲ Report Server Database Setup (y)

کادر محاوره‌ای Report Server Database Setup به شما اجازه می‌دهد محل پایگاه داده SQL Server مورد استفاده Reporting Service را که نصب خواهد شد، انتخاب کنید. این پایگاه داده‌ای است که Reporting Service برای ذخیره تعاریف گزارش خود استفاده خواهد کرد. خود گزارشات می‌توانند از هر تعداد منبع داده دیگری استفاده کنند. به‌طور پیش‌فرض، برنامه نصب SQL Server 2005، پایگاه داده Reporting Service را در نمونه SQL Server جاری ایجاد خواهد شد. هرچند،

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

هم‌چنین می‌توانید نصب پایگاه داده Reporting Service را در سیستم SQL Server دیگری را انتخاب کنید. برای بیشتر سازمان‌های بزرگ و متوسط، جداسازی Reporting Service در سرور خاص خود و نگهداری آن به‌طور مجزا از پایگاه داده محصول، به دلایل کارآیی ایده خوبی است. کادر محاوره‌ای Report Server Setup نیز به شما امکان می‌دهد تا نامی را مشخص کنید که برای پایگاه داده Reporting Service استفاده خواهد شد. برای نمونه پیش‌فرض، نام پایگاه داده ReportServer استفاده خواهد شد. اگر Reporting Service را در نمونه معینی نصب می‌کنید، بنابراین نام پایگاه داده ReportingServer$InstanceName خواهد بود که InstanceName توسط نام نمونه SQL Server جایگزین خواهد شد. همانند نام دایرکتوری‌های مجازی، می‌توانید این مقادیر را تغییر دهید. بعد از تنظیم گزینه‌های پایگاه داده Reporting Service، کلیک کردن Next موجب نمایش صفحه نصب نشان داده شده در شکل الف-۱۳ می‌شود.



(z)

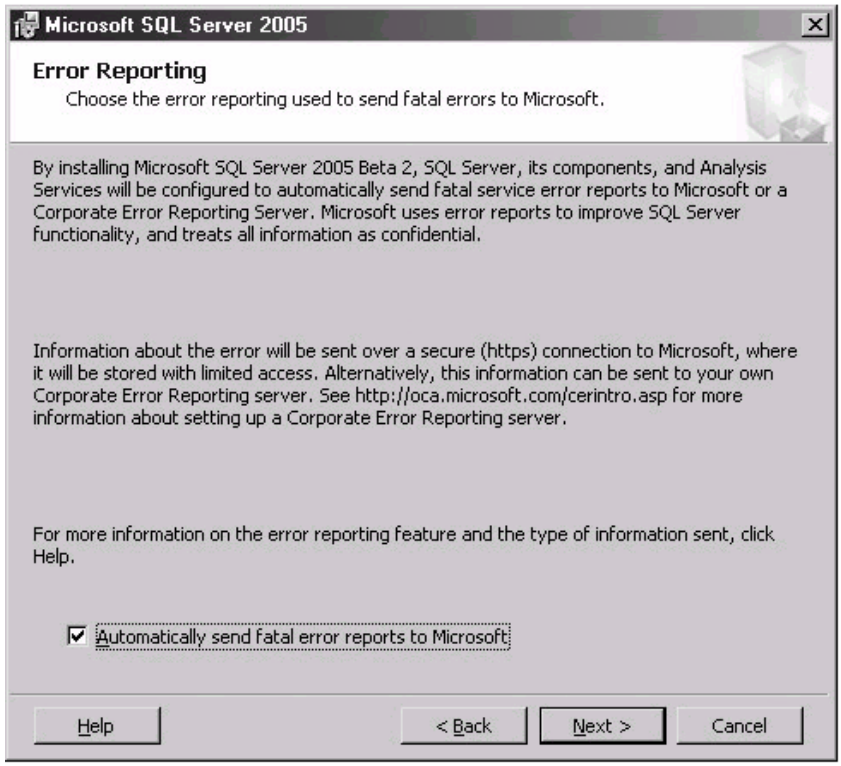
شکل الف-۱۳ Report Server Delivery Settings (aa)

کادر محاوره‌ای Report Server Delivery Settings به شما اجازه می‌دهد آدرس سرور SMTP را مشخص کنید که Reporting Service برای تحویل گزارشات و هشدارهای e-mail استفاده خواهد

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

شد. پر کردن این مقادیر اختیاری است (می‌توانید برگشته و آن‌ها را تنظیم کنید یا مقادیر را بعد از تکمیل فرآیند نصب تغییر دهید). کلیک کردن Next موجب می‌شود کادر محاوره‌ای SQL Server 2005 Error Reporting نمایش داده شود. می‌توانید کادر محاوره‌ای Error Reporting را در شکل الف-۱۴ ببینید.



(bb)

شکل الف-۱۴ Error Reporting (cc)

بیشتر شبیه پشتیبانی Watson که مایکروسافت به Windows XP اضافه کرده است، صفحه SQL Server 2005 Error Reporting به شما امکان می‌دهد تا به‌طور اختیاری خطاهای SQL Server مهلک را به مایکروسافت گزارش کنید. مایکروسافت هیچ اطلاعات پرسنلی را از این گزارشات جمع‌آوری نمی‌کند. گزارشات خطای SQL Server 2005، این اطلاعات را به مایکروسافت ارسال می‌کنند:

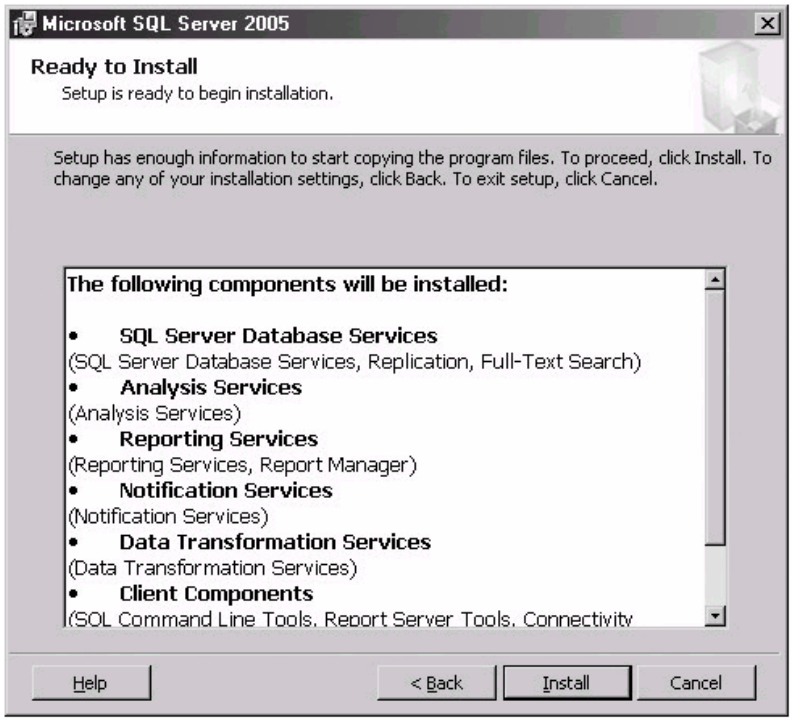
- وضعیت SQL Server هنگام رخ دادن خطا

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

- نگارش سیستم عامل
- پیکربندی سخت‌افزار پایه
- ID محصول دیجیتال SQL Server (که برای تعیین مدرک شما استفاده می‌شود)
- آدرس IP سرور
- اطلاعاتی از حافظه درباره فرآیندی که موجب خطا شده است

این گزارشگیری خطا کاملاً اختیاری است. در حالی که گزارشگیری خطای خودکار به‌طور پیش‌فرض فعال است، به آسانی می‌توانید آن را با از حالت تأیید خارج کردن کادر انتخاب Automatically Send Fatal Error Reports To Microsft، غیرفعال کنید. بعد از سامان بخشیدن به کادر محاوره‌ای Error Reporting، Next را برای نمایش کادر محاوره‌ای Ready To Install کلیک کنید که در شکل الف-۱۵ نشان داده شده است.



(dd)

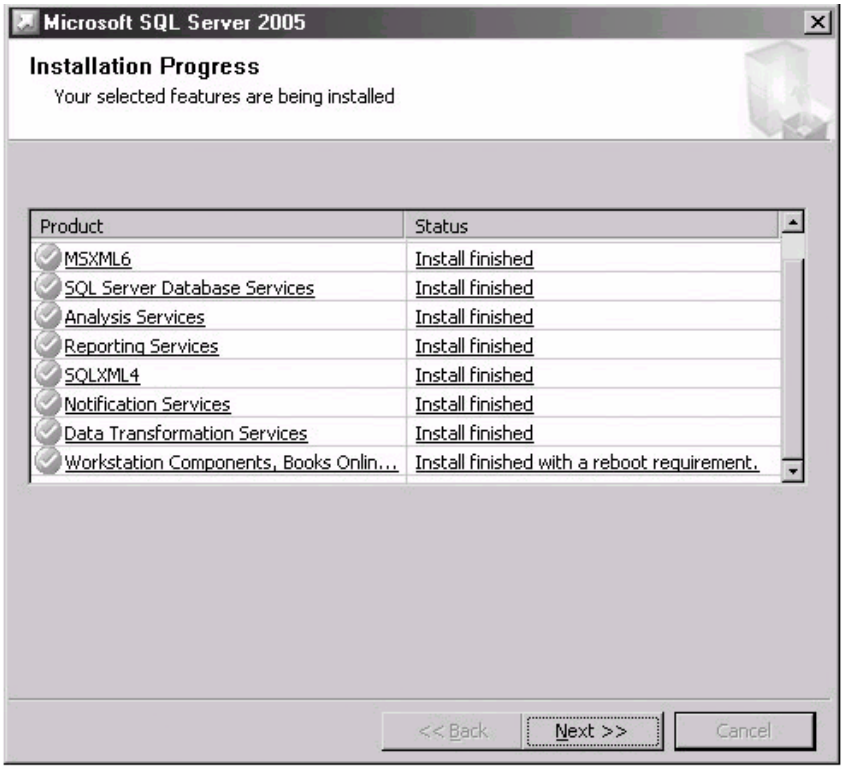
شکل الف-۱۵ Ready To Install (ee)

کادر محاوره‌ای Ready To Install به شما امکان تأیید انتخاب‌هایتان را می‌دهد. اگر نیاز به تغییر دارید، می‌توانید از دکمه Back برای برگشت به صفحات نصب قبل استفاده کنید. کلیک کردن

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Install در کادر محاوره‌ای Ready To Install موجب شروع شدن فرآیند نصب SQL Server 2005 می‌شود. هنگامی که نصب پیشرفت می‌کند، صفحه نشان داده شده در شکل الف-۱۶ نمایش داده می‌شود.



(ff)

شکل الف-۱۶ Installation Progress (gg)

صفحه Installation Progress وضعیت نصب را نمایش می‌دهد. یک علامت تأیید سبز رنگ نشان می‌دهد که یک جزء به‌طور موفق نصب شده است. یک X قرمز نشان دهنده یک خطاست. بعد از تکمیل نصب، SQL Server آماده استفاده خواهد بود.

تصحیح نصب

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

می‌توانید نصب SQL Server را با بررسی این که آیا سرویس‌های ضروری در حال اجرا هستند، تصحیح کنید. برای راهنمایی در زمینه عیب‌یابی مشکلات نصب، هم‌چنین می‌توانید فایل log نصب را ببینید.

سرویس‌ها

جدول الف-۲ سرویس‌های مورد استفاده SQL Server را فهرست کرده است. می‌توانید این سرویس‌ها را با استفاده از اپلت Start | Administrative Tools | Services ببینید.

(hh) جدول الف-۲ سرویس‌های SQL Server 2005

سرویس	شرح
SQL Server (MSSQLSERVER) and/or MSSQL$*InstanceName*	موتور پایگاه داده SQL Server و یا یک نمونه معین از موتور پایگاه داده SQL Server
SQL Server Agent (MSSQLSERVER) and/or SQLAgent$*InstanceName*	نماینده زمانبندی کار SQL Server و یا نمونه معینی از نماینده زمان‌بندی کار SQL Server
MSSQLServerOLAPService and/or MSOLAP$*InstanceName*	SQL Server Analysis Services و یا یک نمونه معین از SQL Server Analysis Services
ReportServer and/or ReportServer$*InstanceName*	SQL Server Reporting Service و یا یک نمونه معین از SQL Server Reporting Service

نصب فایل‌های Log

اگر برای نصب ناموفق باشد، می‌توانید فایل‌های Log نصب SQL Server را بررسی کنید. جدول الف-۳ فایل‌های log مورد استفاده SQL Server 2005 را فهرست کرده است.

(ii) جدول الف-۳ فایل‌های SQL Server 2005 Log

فایل log	دایرکتوری	شرح
sqlstpX.log	%temp%	این فایل حاوی اطلاعات نصب از Windows installer است. هر اجرا، یک واحد به مقدار x اضافه می‌کند.
Sqlsetup<xxxx>_machine>	%temp%	توسط نصب ایجاد می‌شود.

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

		_support.log
فایل‌های bootstrap نصب SQL Server	%sqlserver%\90 \Setup\Bootstrap\LOG	SQLSetup*.txt
حاوی اطلاعات رویه نصب است. هر اجرا، مقدار X را یک واحد افزایش می‌دهد.	%temp%	sqlrunXlog.log
Log نصب Microsoft Data Access Components.	%windows%	dasetup.log
Log خطای SQL Server.	%sqlserver%\mssql\log\errorlog	errorlog.log

Article LXVI. ارتقا به SQL Server 2005

علاوه بر ایجاد نصب‌های جدید، نصب SQL Server 2005 هم‌چنین می‌تواند برای ارتقای نصب‌های SQL Server 7 و SQL Server 2000 استفاده شود. هرچند، نمی‌تواند برای ارتقای نصب‌های SQL Server 6.5 موجود استفاده شود.

Section ۶۶.۰۱ ارتقا از SQL Server 7 و 2000

برنامه نصب از ارتقاهای مستقیم از SQL Server 7 و SQL Server 2000 به SQL Server 2005 پشتیبانی می‌کند. ساختارهای روی دیسک ضروری SQL Server مشابه هستند و برنامه نصب می‌تواند به‌طور موفقی یک انتقال یکجا را برای نصب‌های SQL Server 7 و SQL Server 2000 به SQL Server 2005 انجام دهد. جدول الف-۴ روش‌های ارتقای پشتیبانی شده برای SQL Server 7 و SQL Server 2000 را فهرست کرده است.

(a) جدول الف-۴ گزینه‌های ارتقای SQL Server 2005

ویرایش SQL Server موجود	ارتقا به ویرایش SQL Server 2005 Standard	ارتقا به ویرایش SQL Server 2005 Enterprise
MSDE 2000	Yes	No
SQL Server 7 Standard Edition	Yes	Yes
SQL Server 7 Enterprise Edition	No	Yes

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Yes	Yes	SQL Server 2000 Standard Edition
Yes	No	SQL Server 2000 Enterprise Edition

Section ۶۶،۰۲ ارتقا از SQL Server 6.5 (یا ما قبل)

از ارتقای مستقیم از SQL Server 6.5 یا هر ویرایش ما قبل SQL Server به SQL Server 2005 پشتیبانی نمی‌شود. ساختارهای روی دیسک و حافظه پایگاه داده مورد استفاده SQL Server 6.5 و نگارش‌های ماقبل با ساختارهای روی دیسک مورد استفاده SQL Server 2005 متفاوت هستند. بنابراین، نمی‌توانید مستقیماً یک ارتقای یک‌جا را انجام دهید. تنها روش برای انجام یک ارتقای یک‌جا، غیرمستقیم است که با ارتقای SQL Server 6.5 به SQL Server 2000 و سپس ارتقای SQL Server 2000 به SQL Server 2005 انجام می‌شود. هرچند، انجام دو ارتقا برای انجام دادن یکی، احتمالاً کارآمدترین روش نیست. با فرض سطوح سیستم عامل حداقل و سخت‌افزار سازگار، برنامه نصب می‌تواند برای نصب نمونه جدیدی از SQL Server 2005 در یک سیستم اجرا کننده SQL Server 6.5 استفاده شود. هرچند، بیشتر نصب‌های SQL Server 6.5 احتمالاً روی سخت‌افزاری قدیمی اجرا می‌شوند و در بیشتر موارد، باید سخت‌افزار جدیدی داشته باشید.

طرح اصلی این است که بهترین روش برای انتقال از SQL Server 6.5 به SQL Server 2005، انجام نصبی جدید از SQL Server 2005 روی سخت‌افزاری نو و سپس انجام یک ارتقای دستی با انتقال اشیای پایگاه داده و داده با DTS و کامپایل مجدد رویه‌های ذخیره شده و سایر اشیای پایگاه داده کاربر است.

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

پیوست ب

واقعیت‌های سریع

این پیوست مرجعی برای چندین محدودیت حداکثر مهم پایگاه داده و سیستم SQL Server 2005 ارایه می‌دهد.

(a) جدول ب-۱: محدودیت‌های حداکثر سیستم

دسته	ظرفیت
حداکثر حافظه قابل آدرس‌دهی	32TB (64-bit) 64GB (32-bit using PAE)
حداکثر تعداد پردازنده‌ها	64 (64-bit) 32 (32-bit)
حداکثر گره‌ها برای کلاستر	8
نمونه‌های SQL Server در هر سرور	50
قفل‌ها در هر نمونه	تنها توسط حافظه محدود شده (64-bit) 2,147,483,647 (32-bit)

(b) جدول ب-۲: محدودیت‌های حداکثر پایگاه داده

دسته	ظرفیت
پایگاه داده در هر سرور	32,767
اندازه پایگاه داده	1,048,516TB

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

دسته	ظرفیت
فایل‌ها در هر پایگاه داده	32,767
گروه‌های فایل در هر پایگاه داده	265
اندازه فایل (داده)	32TB
اندازه فایل (log)	32TB
اشیا در یک پایگاه داده	2,147,483,647
طول شناسه	128

(c) جدول ب-۳: جداول

دسته	ظرفیت
جداول در هر پایگاه داده	با تعداد اشیا در یک پایگاه داده محدود شده است
ردیف‌ها در هر جدول	با حافظه موجود محدود شده است
الزامات کلید اصلی در هر جدول	1
الزامات کلید خارجی در هر جدول	253
مراجع در هر جدول	253
تریگرها در هر جدول	با تعداد اشیا در یک پایگاه داده محدود شده است
ایندکس‌های کلاستر شده در هر جدول	1
ایندکس‌های کلاستر نشده در هر جدول	249
الزامات Unique در هر جدول	۲۴۹ کلاستر نشده، ۱ کلاستر شده

(d) جدول ب-۴: ستون‌ها

دسته	ظرفیت
ستون‌ها در هر ایندکس	16

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

دسته	ظرفیت
ستون‌ها در هر کلید اصلی	16
ستون‌ها در هر کلید خارجی	16
ستون‌ها در هر جدول	1024
اندازه کلید ایندکس	900 bytes
بایت‌ها در هر کاراکتر یا ستون باینری	8000
بایت‌ها در هر ستون text، ntext، یا image	2GB
بایت‌ها در هر ردیف	8060
بایت‌ها در هر ایندکس	900
بایت‌ها در هر کلید اصلی	900
بایت‌ها در هر کلید خارجی	900

جدول ب–۵: محدودیت‌های حداکثر T-SQL (e)

دسته	ظرفیت
اندازه دسته	65,536 ضربدر اندازه بسته شبکه
جداول در هر عبارت SELECT	۲۵۶
بایت‌ها در متن منبع یک رویه ذخیره شده	کمتر از اندازه دسته یا 250MB
پارامترها در هر رویه ذخیره شده	۱۰۲۴
زیرپرس‌وجوهای تودرتو	۳۲
سطوح تریگر تودرتو	۳۲
ستون‌ها در هر عبارت SELECT	۴۰۹۶
ستون‌ها در هر عبارت INSERT	۱۰۲۴

www.IrPDF.com www.IrPDF.com www.IrPDF.com

www.IrPDF.com

Convert To PDF By Ali643
Kylix_online@Yahoo.com
www.Barnamenevis.Org

www.IrPDF.com www.IrPDF.com www.IrPDF.com