

Microsoft®
SQL Server™ 2005

**데이터베이스 개발자를 위한
SQL Server 2005 개요**

요약

본 문서에서는 Microsoft SQL Server 2005에서 제공하는 새로운 데이터베이스 개발 기능을 개괄적으로 설명합니다.

본 문서는 예비 문서이며 여기에서 설명된 소프트웨어의 최종 상용 버전이 출시되기 전에 상당 부분 변경될 수 있습니다. 이 문서에 포함된 정보는 문서 발행 시에 논의된 문제들에 대한 Microsoft Corporation의 당시 관점을 나타냅니다. Microsoft는 변화하는 시장 상황에 부응해야 하므로 이를 Microsoft 측의 공약으로 해석해서는 안되며 발행일 이후 소개된 어떠한 정보에 대해서도 Microsoft는 그 정확성을 보증하지 않습니다.

이 문서는 오직 정보를 제공하기 위한 것입니다. Microsoft는 이 설명서에서 어떠한 명시적이거나 묵시적인 보증도 하지 않습니다. 해당 저작권법을 준수하는 것은 사용자의 책임입니다. 저작권에서의 권리와는 별도로, 이 설명서의 어떠한 부분도 Microsoft의 명시적인 서면 승인 없이는 어떠한 형식이나 수단(전기적, 기계적, 복사기에 의한 복사, 디스크 복사 또는 다른 방법) 또는 목적으로도 복제되거나, 검색 시스템에 저장 또는 도입되거나, 전송될 수 없습니다.

Microsoft가 이 설명서 본안에 관련된 특허권, 상표권, 저작권 또는 기타 지적 재산권 등을 보유할 수도 있습니다. 서면 사용권 계약에 따라 Microsoft로부터 귀하에게 명시적으로 제공된 권리 이외에, 이 설명서의 제공은 귀하에게 이러한 특허권, 상표권, 저작권 또는 기타 지적 재산권 등에 대한 어떠한 사용권도 허여하지 않습니다.

특별한 언급이 없는 한, 용례에 사용된 회사, 기관, 제품, 도메인 이름, 전자 메일 주소, 로고, 사람, 장소, 이벤트 등은 실제 데이터가 아닙니다. 어떠한 실제 회사, 기관, 제품, 도메인 이름, 전자 메일 주소, 로고, 사람, 장소 또는 이벤트와도 연관시킬 의도가 없으며 그렇게 유추해서도 안됩니다.

© 2005 Microsoft Corporation. 전권 보유.

Microsoft와 ActiveX는 미국, 대한민국 및/또는 기타 국가에서 Microsoft의 등록 상표 또는 상표입니다. 여기에 인용된 실제 회사와 제품 이름은 해당 소유자의 상표일 수 있습니다.

Contents

데이터베이스 개발의 새로운 패러다임	2
.NET Framework 통합	2
Transact-SQL과 관리되는 코드 중 선택하기	3
웹 서비스	4
ADO.NET	4
ADO.NET 알림 지원	4
MARS(Multiple Active Result Set)	4
기본 데이터 형식 지원	4
스냅샷 격리 지원	5
SMO(SQL Management Object)	5
SMO 및 SQL-DMO	5
XML 기술	6
XML 데이터 형식	6
XML 검색 및 작성	7
XQuery 지원	7
DML 확장	8
새로운 응용 프로그램 프레임워크	8
SQL Service Broker	8
Reporting Services	8
알림 서비스	9
SQL Server Mobile Edition	9
언어 기능 향상	10
Transact-SQL 기능 향상	10
재귀적 쿼리 및 CTE(Common Table Expression)	10
PIVOT 및 UNPIVOT 연산자	11
APPLY 연산자	11
트랜잭션의 예외 처리	11
DDL 이벤트 알림	12
전체 텍스트 검색 기능 향상	12
향상된 보안 기능	12
결론	12

데이터베이스 개발의 새로운 패러다임

Microsoft SQL Server 2005가 출시되면서 데이터베이스 개발의 판도가 변화하고 있습니다. 이제 데이터베이스 개발자들은 해당 기능과 관련된 코드를 적절하게 검색하고 XML과 같은 기본 형식의 데이터에 액세스하며 데이터베이스 서버 성능에 의해 좌우되는 복잡한 시스템을 구축할 수 있는 옵션을 갖게 되었습니다. 데이터베이스 개발은 이전보다 훨씬 통합적인 방식으로 수행되며 필요한 모든 도구를 즉시 사용할 수 있습니다.

본 문서에서는 새로운 수준의 데이터베이스 응용 프로그램을 개발할 수 있도록 지원하는 SQL Server 2005의 대표적인 새 기능을 설명하고 있습니다.

.NET Framework 통합

Microsoft SQL Server 2005가 출시됨에 따라 데이터베이스 프로그래머는 Microsoft .NET Framework 클래스 라이브러리와 최신 프로그래밍 언어를 최대한 활용해 서버 내에 기능을 구현할 수 있게 되었습니다. CLR(common language runtime) 통합을 이용해 저장 프로시저, 기능 및 트리거를 선택한 .NET Framework 언어로 코딩할 수 있습니다. Microsoft Visual Basic, .NET과 C# 프로그래밍 언어는 모두 객체 지향 구문, 구조적 예외 처리(Structured Exception Handling), 어레이, 네임스페이스(namespace), 클래스를 제공합니다. 또한 .NET Framework는 서버 측에서 사용할 수 있는 포괄적인 기본 제공 기능을 포함하여 수천 개의 클래스와 메서드를 제공합니다. Transact-SQL로 수행하기 힘들거나 어려운 여러 작업들은 관리되는 코드를 사용하여 보다 나은 결과를 거둘 수 있습니다. 또한, 새로운 2가지 형식의 데이터베이스 객체-집계(aggregate) 및 사용자 정의 형식-를 사용할 수 있습니다. 개발자는 이미 익힌 기술을 보다 효율적으로 활용해 진행 중인 코드를 작성할 수 있습니다. 요약하면 SQL Server 2005를 사용하면 데이터베이스 서버를 확장해 적절한 계산 및 작업을 백 엔드에서 더욱 쉽게 수행할 수 있습니다.

SQL Server와 CLR 간의 통합을 통해 여러 중요한 이점이 제공됩니다.

- **향상된 프로그래밍 모델** : NET Framework와 호환되는 프로그래밍 언어는 여러 측면에서 Transact-SQL보다 풍부하기 때문에, SQL 개발자들은 이전까지 지원되지 않았던 구문 및 기능을 활용할 수 있습니다.
- **안전성 및 보안 강화** : 관리되는 코드는 데이터베이스 엔진이 호스팅하는 CLR 환경에서 실행됩니다. .NET Framework 데이터베이스 객체는 이전 버전의 SQL Server에서 사용할 수 있었던 확장된 저장 프로시저 보다 안전하며 철저하게 보호됩니다.
- **사용자 정의 형식 및 집계** : CLR을 호스팅하여 SQL Server의 스토리지 및 쿼리 기능을 확장하는 2개의 새로운 데이터베이스 객체를 지원할 수 있습니다.
- **공동 개발 환경** : 데이터베이스 개발은 Microsoft Visual Studio 2005 개발 환경으로 통합됩니다. 개발자는 중간 계층 또는 클라이언트 계층 .NET Framework 구성 요소 및 서비스를 작성할 때 사용하는 것과 동일한 도구로 데이터베이스 객체 및 스크립트를 개발 및 디버깅할 수 있습니다.

- **성능 및 확장성** : 관리되는 코드는 복잡한 절차 및 과학적 프로세싱에 최적화되어 있으며 고유 코드에 따라 실행하기 때문에 관리되는 코드를 사용하면 일부 시나리오에서 상당한 성능 향상을 이룰 수 있습니다.

Visual Basic .NET 및 C# 같은 언어를 사용함으로써 논리가 복잡하고 계산 작업에 더욱 적합한 코드를 작성하는 데 CLR 통합을 활용할 수 있습니다. 또한, Visual Basic .NET과 C#는 캡슐화(encapsulation), 상속(inheritance) 및 다형성(polymorphism)과 같은 객체 지향 기능을 제공합니다. 관련 코드를 클래스와 네임스페이스로 쉽게 구성할 수 있기 때문에 이제 대량의 코드 작업 시 코드 투자를 더욱 쉽게 구성하고 유지할 수 있습니다. 코드를 논리적/물리적으로 어셈블리 및 네임스페이스로 구성하는 기능은 매우 큰 이점을 제공하며, 대규모 데이터베이스 구현에서 서로 다른 코드 조각을 찾아 연관시키는 기능을 향상시킬 수 있습니다.

관리되는 코드는 숫자 처리 및 복잡한 실행 논리 관리 면에서 Transact-SQL보다 훨씬 효율적이며 문자열 처리, 정규식 등을 광범위하게 지원합니다. 또한, .NET Framework 클래스 라이브러리에서 지원되는 기능을 이용하면 모든 저장 프로시저, 트리거 또는 사용자 정의 함수에서 쉽게 액세스할 수 있는 수천 개의 사전 구축된 클래스와 루틴을 완벽하게 활용할 수 있습니다. 향상된 문자열 처리 함수, 수학 함수, 날짜 연산(date operation), 시스템 자원 액세스, 첨단 암호화 알고리즘, 파일 액세스, 이미지 처리, XML 데이터 조작 등 모든 기능은 관리되는 저장 프로시저, 함수, 트리거 및 집계(aggregate)에서 쉽게 액세스할 수 있습니다.

관리되는 코드의 주요 이점 중 하나는 안정성 타입(type safety)입니다. 관리되는 코드가 실행되기 전에 CLR은 코드가 실행하기에 안전한지 확인하기 위해 검증(verification)이라는 프로세스를 통해 몇 가지 검사를 시행합니다. 예를 들어 코드를 검사하여 쓰여지지 않은 메모리를 읽지 않도록 보장합니다.

Transact-SQL과 관리되는 코드 중 선택하기

저장 프로시저, 트리거 및 사용자 정의 함수 등을 작성할 때 기존 Transact-SQL을 사용할지, Visual Basic .NET 또는 C#과 같이 .NET Framework와 호환되는 프로그래밍 언어를 사용할지 결정해야 합니다. 이 질문에 대한 대답은 주어진 상황에 따라 다릅니다. 즉, 어떤 상황에서는 Transact-SQL이 유용하고 또 다른 상황에서는 관리되는 코드를 사용하는 것이 효과적일 수 있습니다.

Transact-SQL은 절차상의 로직이 거의 또는 전혀 없으며 코드가 주로 데이터 액세스를 수행하는 경우에 적합합니다. .NET Framework와 호환되는 프로그래밍 언어는 복잡한 로직이 특징인 계산 집약적인 기능 및 절차나, .NET Framework 클래스 라이브러리의 이점을 활용하려는 경우에 가장 적합합니다.

코드 배치도 중요합니다. Transact-SQL과 관리되는 코드 모두 서버에서 실행되기 때문에 기능과 데이터가 더욱 가까운 곳에 위치하게 되어 서버의 처리 성능을 최대한 활용할 수 있습니다. 또한 데이터와 중간 계층 간의 트래픽이 감소하여 상당한 이점을 얻을 수 있습니다. 데이터 바로 옆에서 계산이 이루어지기 때문에, I/O-바운드 시스템에서 이 방법은 매우 효과적입니다. CLR 기능이 SQL Server 쿼리 프로세서를 활용할 수 있기 때문에 실행을 동기화하고 최적화할 수 있습니다. 한편, 데이터베이스 서버에서 프로세서 집약적인 작업을 처리하고 싶지 않을 수도 있습니다. 오늘날 대다수 컴퓨터들은 매우 강력하기 때문에 클라이언트에 가능한 많은 코드를 배치함으로써 이 처리 성능을 활용할 수 있습니다. 그러나 모든 요구 사항을 단 한가지 솔루션으로 해결할 수는 없습니다.

웹 서비스

SQL Server 2005을 활용하면 데이터베이스 계층에서 XML 웹 서비스를 개발하여 SQL Server를 HTTP 리스너(listener)로 만들 수 있습니다. 이에 따라 웹 서비스 중심으로 중앙 집중화되는 응용 프로그램에 새로운 형식의 데이터 액세스 기능이 제공됩니다. SQL Server 2005가 Microsoft Windows Server 2003 또는 Microsoft Windows XP SP2에서 사용될 경우, 현재 운영 체제에 포함되어 있는 경량 웹 서버인 HTTPSYS를 활용함으로써 Microsoft IIS(Internet Information Services)와 같은 중간 계층을 사용하지 않고 HTTP에서 직접 SQL Server로 액세스할 수 있습니다. SQL Server는 웹 서비스 인터페이스를 제시하여 SQL 문을 실행하고 함수 및 프로시저를 호출합니다. 쿼리 결과는 XML 형식으로 반환되며 Visual Studio의 웹 서비스 인프라를 활용할 수 있습니다.

ADO.NET

차기 버전의 ADO.NET에서는 여러 새로운 기능을 선 보이게 됩니다. 새로운 쿼리 변경 알림 지원에서 MARS(Multiple Active Result Set), 그리고 Visual Studio 2005의 고유 형식 지원에 이르기까지, ADO.NET은 데이터 집합 액세스 및 처리 기능을 발전시켜 더욱 우수한 확장성과 융통성을 발휘합니다.

ADO.NET 알림 지원

SQL Server 2005는 SQL Server 쿼리에 대한 알림 기능을 새롭게 지원합니다. 이 지원을 사용하여 SQL Server에 명령을 전달하고 동일한 명령을 다시 실행할 때 처음 얻은 결과와 다른 결과가 나오는 경우 알림 표시가 생성하도록 요청할 수 있습니다. 원본 데이터가 변경되는 시점을 감지하는 종속성 객체를 사용해 이 기능을 수행할 수 있습니다. ADO.NET, OLE DB, ODBC(Open Database Connectivity), Microsoft ADO(ActiveX Data Objects) 또는 SOAP 같은 모든 클라이언트 API를 통해 서버로 보내진 명령에는 알림이 필요한 태그가 포함되어 있을 수 있습니다. 요청의 일부분으로 실행된 각 문에 대해 서버는 요청에 포함된 각 문마다 한 번만 실행되는 “알림 구독”을 만듭니다. 알림 메시지는 비동기 SQL Service Broker 대기열을 통해 전달됩니다. 쿼리 알림은 데이터베이스 중심 웹 사이트 같은 응용 프로그램에서 결과를 캐싱하는 데 유용합니다. ASP.NET 2.0이 사용되고 있다면 SQL Server Query Notifications에 대한 지원이 해당 제품에 직접 내장되며 알림 옵션을 선택하는 것과 같은 간단한 방법으로 실행할 수 있습니다.

MARS(Multiple Active Result Set)

MARS(Multiple Active Result Set)는 단일한 연결상에 다수의 보류 요청이 전달될 수 있도록 하는 기능으로서, 특히 연결당 하나 이상의 기본 결과 집합이 열려 있을 수 있습니다. 기본 결과 집합은 전달 전용/읽기 전용 결과 집합입니다. 기본 결과 집합의 경우 클라이언트 드라이버는 대형 청크(테이블 형식 데이터 스트림 버퍼 크기 청크)에서 데이터를 있는 그대로 검색하여 서버 커서의 경우처럼 서버와 정보를 주고 받지 않고도 응용 프로그램 요청이 충족되도록 합니다. 응용 프로그램은 성능 저하 없이 간단한 한 번에 한 행(row-at-a-time)씩 프로그래밍하는 모델을 사용할 수 있습니다.

MARS(Multiple Active Result Sets)는 전체 결과 집합이 사용될 때까지 드라이버가 서버에 요청을 전송하지 않도록 차단하는 열린 기본 결과 집합 내에서 현재의 제한을 제거합니다.

기본 데이터 형식 지원

ADO.NET 2.0을 통해 많은 새로운 데이터 액세스 옵션을 사용할 수 있을 뿐 아니라 새 SQL Server 데이터 형식 모듈에 대한 지원이 관리되는 코드 언어에 구현됩니다. Visual Studio 개발자들은 이제 Visual Studio 환경에서 직접 SQLXML, 사용자 정의 형식 및 varchar(max)와 같은 새로운 SQL Server 2005 데이터 형식을 모두 활용할 수 있습니다.

스냅샷 격리 지원

SQL Server 2005에서는 새로운 스냅샷 격리 수준을 선보입니다. 스냅샷 격리는 데이터 버전들이 데이터 판독기를 위해 행 버저닝 메커니즘으로 저장되는 것을 말합니다. 새롭게 추가된 이 격리 수준은 다음과 같은 이점을 제공합니다.

- 읽기 전용 응용 프로그램에 대한 데이터 가용성 향상 OLTP 환경에서 허용된 읽기 작업 비차단(nonblocking)
- 쓰기 트랜잭션에 대한 자동 명령 충돌 탐지
- Oracle에서 SQL Server로의 응용 프로그램 마이그레이션 간소화

예를 들어 잠금을 설정하면 동일한 데이터를 동시에 읽고 쓰는 응용 프로그램 사이를 차단하게 되며 트랜잭션 중 행이 변경될 경우 다른 트랜잭션에서는 쓰기가 커밋될 때까지 그 행을 읽을 수 없습니다. 스냅샷 격리를 실행하면 판독기는 이전에 커밋된 행 값에 액세스할 수 있습니다.

스냅샷 격리 수준이 지원되며 ADO, OLE DB, SQLOLEDB, Shape Provider, SQLODBC, OLE DB Managed Provider 및 SQL Managed Provider를 통해 제시됩니다.

SMO (SQL Management Object)

SMO(SQL Management Objects) 모델은 SQL Server 2005의 관리 객체 모델로서 SQL Server 관리 객체 모델이 디자인 및 아키텍처 면에서 크게 향상된 것입니다. 이는 사용이 간편하면서도 .NET Framework 관리되는 코드를 기반으로 하는 풍부한 객체 모델로서 .NET Framework를 사용하는 데이터베이스 관리 응용 프로그램 개발이나 SQL Server 2005 관리 작업 자동화를 위한 핵심 도구입니다. SMO는 SQL Server Management Studio의 모든 대화 상자에서 사용되며 SQL Server Management Studio로 수행할 수 있는 모든 관리 작업을 SMO를 사용해서도 수행할 수 있습니다.

새로운 SMO 객체 모델과 Microsoft WMI(Windows Management Instrumentation) API는 SQL-DMO를 대체합니다. 가능한 경우, SMO는 사용 용이성을 위해 유사한 객체를 SQL-DMO로 통합합니다. SQL Server 2005는 여전히 SQL-DMO와 함께 사용할 수 있지만 SQL-DMO는 SQL Server 2005 고유 기능을 관리하도록 업데이트되지 않습니다.

SMO 및 SQL-DMO

SMO 객체 모델은 SQL-DMO에서 수행된 작업의 논리적인 연속체입니다. SMO SQL-DMO와 기능적으로 호환되며 많은 동일한 객체를 포함하고 있습니다. 가능한 경우라면 원래의 SQL-DMO 설계를 따르지만 SMO에는 SQL-DMO보다 더 많은 추가 기능이 포함되어 있습니다. 최대 DDL(Data Definition Language)과 SQL Server 2005에 대한 관리 범위를 확보하기 위해 SMO는 150개 이상의 새로운 클래스를 추가합니다.

SMO의 주요 이점은 그 탁월한 성능과 확장성입니다. SMO는 SQL Server에 대한 변경을 적용하기 전에 객체의 여러 속성을 변경할 수 있도록 하는 캐싱된 객체 모델을 보유하고 있습니다. 따라서 SMO는 서버 간의 왕복을 줄이고 객체를 더욱 유연하게 만듭니다. 또한 최적화된 인스턴스화 기능을 통해 객체를 부분적으로 또는 전체적으로 인스턴스화할 수 있습니다. 따라서 객체들의 모든 속성을 인스턴스화하지 않아도 되기 때문에 신속하게 많은 객체를 로드할 수 있습니다.

단일 응용 프로그램 루트 디렉터리에서 생성된 모든 서버 객체에 대한 참조를 유지하는 SQL-DMO와 달리, SMO에서는 새로운 연결을 설정하지 않아도 서버에 대해 여러 루트를 설정할 수 있습니다. SMO는 SQL-DMO 스타일 스크립팅을 지원할 뿐 아니라 향상된 단계 스크립팅을 구현합니다. 또한 실제로 서버에 변경사항을 적용하지 않아도 객체를 캡처 모드로 전환하여 해당 객체에 대해 누락됐을 수 있는 모든 DDL을 포착할 수 있습니다.

SQL-DMO는 또한 SMO 객체 인터페이스를 통해 WMI 모니터링과 서버 구성을 지원할 수 있도록 WMI로의 인터페이스를 간소화하는 관리되는 컴퓨터 객체를 보유하고 있습니다.

XML 기술

XML은 데이터를 저장 및 전송하는 일반적인 형식이며, 마크 업되거나 구조적 또는 반구조적인 정보에 가장 널리 사용되는 방법입니다. 이러한 데이터 종류에는 텍스트(문서 구조를 식별하기 위해 마크업되고 반전 표시됨), 중첩된 객체(구조적), 그리고 인스턴스 간에 구조가 변경될 수 있는 이기종 데이터(반구조적)가 있습니다. XML은 또 로컬 네트워크 및 인터넷에서 이기종 응용 프로그램 간에 데이터를 전달할 수 있는 중요하고 널리 받아 들여진 표준이기도 합니다.

Microsoft SQL Server 2000은 Microsoft SQLXML을 통해 XML을 지원하기 때문에 관계형 데이터를 XML 데이터로 전환하여 관계형 테이블에 XML 데이터를 저장할 수 있습니다. Microsoft SQL Server 2005는 최상급 데이터 형식으로 XML을 지원함으로써, 그리고 XML 문서에 새로운 쿼리 언어를 제공하여 사용자가 적절히 수정할 수 있도록 함으로써 이 기능에 바탕을 두고 구축된 것입니다.

XML 데이터를 사용하는 사용자의 요구를 보다 효과적으로 지원하기 위해 새로운 XML 데이터 형식이 개발되었습니다. 이 데이터 형식은 현재 최종 심의가 진행 중인 중요한 XQuery(XML Query) 사양 세트를 구현하는 query(), exist(), value(), nodes() 및 modify() 등의 메서드를 가지고 있습니다. 사실상, 이 사양은 XML 데이터 수정 구문이 추가되면서 SQL Server 2005에서 크게 확장되었으며 XML 형식을 지원하기 위해 XML 스키마를 등록하고 관리하기 위해 키워드가 추가되었습니다. 또한 관계형 데이터에서 XML을 생성하고 XML에서도 관계형 데이터를 생성할 수 있도록 지원하기 위해 SQL Server 2000에서 처음 소개되었던 FOR XML과 OPENXML에 대해서도 XML 데이터 형식을 지원하는 기능이 추가되어 한층 향상되었습니다.

XML 데이터 형식

XML은 SQL Server가 지원하는 스칼라 형식을 비롯한 복잡한 데이터를 모델링할 수 있습니다. 따라서 char 또는 varchar 같은 문자열 기반의 기본 제공 데이터 형식은 XML의 강력한 기능과 여러 이점을 완벽하게 효율적으로 활용하기에는 부족합니다. 예를 들어 XML이 문자열로 저장된 경우 전체 문서를 삽입 또는 선택하거나 전체 문서에서 연속된 바이트를 검색할 수는 있지만 문서 자체의 내용을 쿼리할 수는 없습니다. XML 데이터 형식을 제공하여 SQL Server 2005는 사용자가 XML 문서의 일부분을 쿼리하고 문서가 XML 스키마를 준수하는지 확인하며 XML 문서의 내용을 현재 위치에서 수정할 수도 있도록 합니다. 또한 SQL Server 2000에서는 지원되지 않았던 방법으로 전통적인 관계형 데이터와 비구조적 또는 반구조적 XML 문서의 데이터를 통합합니다. SQL Server 2005에서는 효율적으로 구문을 재분석하고 쿼리하며 어느 정도 압축이 가능한 내부 표현을 사용하여 XML 데이터를 BLOB(Binary Large Objects)로서 저장합니다.

SQL Server 2005에서 XML 데이터를 이용해 작업하는 경우 성능을 높이기 위해 XML 데이터 형식에 대한 색인을 노드 수준까지 적용할 수도 있습니다. SQL Server 2005는 XML 데이터에서 주 색인과 보조 색인을 지원하며 XML 데이터를 쿼릴 때 적절한 색인을 활용할 수 있도록 기존 옵티마이저 엔진의 성능을 이용합니다.



XML 스키마의 집합을 XML 형식의 열에 연결할 수 있습니다. 이에 따라 제약 조건, 삽입 및 업데이트에 대한 유효성 검사와 저장된 XML 내부의 값 입력은 물론 스토리지 및 쿼리 처리에 대한 최적화 작업도 수행할 수 있습니다. SQL Server 2005에서는 서버에서 스키마를 관리하기 위한 여러 DDL 문도 제공합니다.

XML 검색 및 작성

SQL Server 2005에는 SQL Server 2000에서 처음 선보인 FOR XML 및 OPENXML의 향상된 여러 기능이 포함되어 있습니다.

FOR XML

SQL Server 2000의 FOR XML 구문은 서버에서 XML 결과를 사용할 수 있는 방법을 제공하지 않습니다. 테이블에 XML 결과를 저장할 수 없거나(물론 클라이언트로 먼저 반환하지 않고) 변수에 지정할 수 없습니다. SQL Server 2005는 XML 데이터 형식에 대한 지원을 추가하고 서버에서 XML 사용을 허용함으로써 FOR XML을 향상시켰습니다. 이는 FOR XML로 작성된 TYPE 지시어를 추가하는 방식으로 이루어졌습니다. 예를 들어, SELECT...FOR XML TYPE 문의 결과로 로컬 XML 변수에 지정되거나 XML 데이터 형식 열을 생성하기 위해 후속 INSERT 문에서 사용할 수 있는 XML 데이터 형식 인스턴스가 생성됩니다. PATH 모드는 열 값이 표시되는 XML 트리의 경로를 지정합니다. FOR XML에 포함된 TYPE 옵션(들)은 복잡한 XML 생성을 간소화하며 FOR XML EXPLICIT 쿼리보다 사용이 더 간편합니다. FOR XML은 SQL Server 2005의 XML 데이터 형식 열에 작용합니다.

OPENXML

SQL Server 2000는 기본적으로 FOR XML 문과 OPENXML rowset 함수를 역관계로 처리합니다. 즉 FOR XML의 경우 관계형 데이터를 XML로 검색할 수 있는 반면, OPENXML에서는 XML을 관계형 데이터로 반환하며, 이를 SQL 조인 설정 또는 쿼리 실행의 기준으로 삼을 수 있습니다. SQL Server 2005에서는 OPENXML의 기능이 한층 강화되었습니다. XML 데이터 형식에 대한 지원은 물론, 사용자 정의 형식과 같은 여러 가지 새로운 데이터 형식에 대한 지원이 제공됩니다. OPENXML WITH 문에서 이러한 형식을 사용할 수 있으며, XML 데이터 형식 인스턴스를 sp_preparedocument로 전달할 수도 있습니다.

XQuery 지원

XML 쿼리 언어, 즉 XQuery는 모든 형식의 XML 데이터를 쿼리할 수 있도록 최적화된 지능적이며 강력한 언어입니다. XQuery를 사용하면 XML 데이터 형식과 연결된 메서드를 사용하여 XML 데이터 형식의 변수와 열에 대한 쿼리를 실행할 수 있습니다. 여러 XML 표준과 마찬가지로, W3C(World Wide Web Consortium)는 XQuery의 개발을 감독합니다. XQuery는 XPath(XML Path Language) 버전 1.0, XQL 및 SQL 등과 같은 여러 다양한 쿼리 언어를 기반으로 하는 Quilt라는 쿼리 언어에서 발전한 것입니다. XPath 2.0을 하위 집합으로 포함하기도 합니다. 따라서 XPath 1.0을 사용한 경험이 있다면 그 기술을 사용할 수 있으며 새 쿼리 언어를 완전히 다시 배울 필요가 없습니다. 그러나 입력, 특수 함수, 그리고 효과적인 반복, 결과 정렬 및 구성 지원 등을 비롯해 XPath 1.0보다 훨씬 향상된 기능들을 제공합니다.

SQL Server 2005에는 데이터 계층에서 XML 객체 조작을 가능하게 하는 고급 XQuery 기능이 함께 제공됩니다. 이는 2003년 11월 15일 XQuery 1.0 Working Draft의 statically typed 하위 집합을 지원합니다.

DML 확장

XQuery 사양은 현재 쿼리에 필요한 구문과 의미를 포함하고 있지만, XML 문서를 수정하는 데 사용되지는 않습니다. XML DML(Data Modification Language)은 데이터 수정을 위한 XQuery의 향상된 기능입니다. SQL Server 2005는 삽입(insert), 업데이트(update), 삭제(delete)라는 키워드를 추가했습니다. 이들 각 키워드는 XML 데이터 형식의 modify() 메서드 내에서 사용됩니다.

새로운 응용 프로그램 프레임워크

SQL Server는 개발자들이 데이터베이스 서비스의 적절한 활용을 통해 자체 응용 프로그램 성능을 대폭 확장시킬 수 있도록 하는 다수의 새롭거나 향상된 응용 프로그램 프레임워크를 제공합니다.

SQL Service Broker

지난 10년 간 전자 상거래 응용 프로그램이 널리 확산됨에 따라 데이터베이스 응용 프로그램 간의 워크플로우 관리를 한층 강화해야 한다는 요구가 높아졌습니다. 온라인 고객이 책을 주문한 경우 이 주문은 재고, 출하 및 신용 카드 시스템에 트랜잭션을 커밋하고 다른 웹 응용 프로그램을 통해 주문 확인도 보내야 합니다. 이러한 각 프로세스가 순차적으로 진행되도록 기다리는 것은 비생산적입니다. SQL Server 2005는 비동기 메시지 라우팅을 구현할 수 있도록 확장 가능한 새로운 아키텍처를 제공합니다.

SQL Server 2005는 새로운 SQL Server 응용 프로그램 프레임워크인 Service Broker를 제공합니다. Service Broker는 데이터베이스 대 데이터베이스 수준에서 신뢰할 수 있는 비동기 메시지를 제공하는 분산 응용 프로그램 프레임워크입니다. Service Broker 기술을 통해 내부 또는 외부 프로세스에서 일반 Transact-SQL 데이터 조작 언어의 확장을 사용하여 신뢰할 수 있는 비동기 메시지 스트림을 주고 받을 수 있습니다. 메시지는 보낸 사람과 동일한 데이터베이스에 있는 대기열, 동일한 SQL Server 인스턴스에 있는 다른 데이터베이스 또는 동일한 서버나 원격 서버에 있는 다른 SQL Server 인스턴스로 보내집니다.

Reporting Services

SQL Server 2005의 출시로 Microsoft는 자사의 통합 비즈니스 인텔리전스 플랫폼의 새로운 주요 구성 요소를 확장할 수 있게 되었습니다. SQL Server Reporting Services는 모든 비즈니스 환경에서 적격의 사람들이 올바른 정보를 쉽게 이용할 수 있도록 함으로써 Microsoft의 비즈니스 인텔리전스 비전을 확장하고 있습니다.

Reporting Services는 기존 보고서 및 대화형 보고서를 생성, 관리 및 제공하는 완벽한 서버 기반 플랫폼입니다. 여기에는 보고서를 즉시 생성, 배포 및 관리하는 데 필요한 모든 것이 들어 있습니다. 동시에 Reporting Services의 모듈형 설계와 확장 가능한 API(Application Programming Interfaces)를 이용해 소프트웨어 개발자, 데이터 제공업체 및 기업은 레거시 시스템 및 타사 응용 프로그램에 보고 기능을 통합할 수 있습니다.



Reporting Services는 SQL Server 2005에서 제공되며 주요 구성 요소는 다음과 같습니다.

- 보고서를 작성, 관리 및 검토할 수 있도록 완벽한 도구를 제공합니다.
- 보고서 호스팅 및 프로세싱을 위한 엔진을 제공합니다.
- 다양한 IT 환경에서 보고서를 포함하거나 보고 솔루션을 통합하기 위한 확장 가능한 아키텍처 및 개방형 인터페이스를 제공합니다.
- 클라이언트에 대한 제어력을 강화하여 Smart Client 및 웹 응용 프로그램 모두에서 보고 기능을 실행할 수 있습니다.
- 사용자가 데이터 하위 집합을 기본으로 보고서를 생성하여 엔터프라이즈 서버로 쉽게 배포할 수 있도록 하는 도구인 Report Builder를 제공합니다.

알림 서비스

Microsoft SQL Server Notification Services는 알림을 생성하여 사용자에게 보내는 응용 프로그램을 개발 및 배포하기 위한 플랫폼입니다.

알림이란 다양한 장치로 전송할 수 있는 개인화된 시기 적절한 메시지를 의미합니다.

알림은 구독자의 기본 설정을 반영합니다. 구독자는 구독을 입력하여 특정 정보에 대해 관심이 있다는 것을 나타냅니다. 예를 들면 “Adventure Works 주식 시세가 \$70.00에 도달하면 알림” 또는 “우리 팀에서 작성 중인 전략 문서가 업데이트되는 경우 알림” 등과 같이 지정할 수 있습니다.

알림은 트리거 이벤트가 발생하는 즉시 생성하여 사용자에게 보낼 수도 있고, 사용자가 지정한 미리 결정된 일정에 따라 생성하여 보낼 수도 있습니다. 사용자의 구독은 알림을 생성하여 보낼 시점을 지정합니다.

알림은 다양한 장치, 즉 사용자의 휴대 전화, PDA(personal digital assistant), Microsoft Windows Messenger 또는 e-메일 계정 등으로 보낼 수 있습니다. 이들 장치는 대개의 경우 사용자들이 휴대하고 있기 때문에, 알림은 우선 순위가 높은 정보를 보내는 데 적합합니다.

SQL Server Mobile Edition

SQL Server 2000 Windows CE Edition과 함께 제공되는 SQL Server 2000은 이제 SQL Server Mobile Edition이 되었습니다. SQL Server Mobile Edition은 개발자들을 위해 많은 새로운 주요 기능을 제공합니다.

- SQL Server Management Studio에서 직접 데스크톱 또는 장치에 SQL Server Mobile Edition 데이터베이스를 작성할 수 있습니다. 데이터베이스가 모바일 장치에 있든, 데스크톱에 있든 관계없이 Management Studio에서 바로 SQL Server Mobile Edition 데이터베이스의 스키마를 조작할 수도 있습니다. SQL Server Management Studio를 사용하여 장치 또는 데스크톱에서 SQL Server Mobile Edition 데이터베이스를 대상으로 하는 쿼리를 실행할 수 있습니다. 기본 SQL Server처럼 GUI 형식으로 렌더링되는 XML 실행 계획을 포함하는 새로운 SQL Server Mobile Edition 기능과 SQL Server Mobile Edition의 쿼리 최적화 프로그램을 재정의하는 쿼리 힌트를 사용할 수도 있습니다. 처음으로 개발자가 장치에서 최적화 계획을 제어할 수 있습니다.
- 이제 DTS(Data Transformation Services) 객체를 코드화하여 데이터를 교환할 수 있습니다.

- 새 SqlCeResult 집합은 SQL Server 2005에 있는 SqlCeResult 집합에서 파생된 것으로서 SQL Server Mobile Edition에서 커서를 완벽하게 스크롤 및 업데이트할 수 있도록 지원합니다. 또한 장치에 있는 데이터 객체에 대한 바인딩을 가능하게 합니다.
- 주 응용 프로그램을 열어 놓은 상태에서 응용 프로그램을 코드화하여 데이터를 동기화하고, 별도의 두 응용 프로그램이 동시에 장치에 있는 동일한 데이터베이스에 액세스하도록 할 수 있습니다.
- 동기화 상태를 표시하는 상태 표시줄로 코드화할 수 있는 알림 서비스를 받을 수 있습니다. 이전에는 동기화 상태가 얼마나 진행되었는지 알지 못했기 때문에 장치의 응답이 멈춘 것이 아니라는 것을 사용자에게 알릴 수 없었습니다.
- 보다 철저한 페이지 교정 정책(page reclamation policy)을 통해 데이터베이스의 크기를 작게 유지할 수 있습니다.
- SQL Server 구문을 사용하여 매개변수화된 쿼리 코드를 공유할 수 있습니다.

언어 기능 향상

Transact-SQL 기능 향상

Transact-SQL은 오랫동안 SQL Server의 모든 프로그래밍을 위한 기본 토대로 활용됐습니다. SQL Server 2005는 확장 가능한 데이터베이스 응용 프로그램을 개발하는 데 필요한 여러 새로운 언어 기능을 제공합니다. 이처럼 향상된 기능에는 오류 처리, 새로운 재귀적 쿼리 기능, 새 SQL Server Database Engine 기능에 대한 지원 등이 포함됩니다. SQL Server 2005의 Transact-SQL 기능 개선을 통해 쿼리 작성의 표현력이 향상되어 코드의 성능을 향상시키고 오류 관리 기능을 확장할 수 있습니다. Transact-SQL을 향상시키기 위한 지속적인 노력을 통해 SQL Server 내에서 Transact-SQL이 매우 중대한 역할을 담당한다는 확고한 신념을 확인할 수 있습니다.

재귀적 쿼리 및 CTE (Common Table Expression)

CTE(Common Table Expression)는 정의 구문이 참조할 수 있는 임시 명명된 결과 세트입니다. 간단한 형식 때문에 CTE를 비영구적 형식의 뷰와 매우 비슷한 파생 테이블의 향상된 버전으로 생각할 수도 있습니다. 파생 테이블과 뷰를 참조하는 것처럼 유사한 방식으로 쿼리의 FROM 구문에서 CTE를 참조합니다. 단 한번만 CTE를 정의하면 쿼리에서 여러 차례 참조할 수 있습니다. CTE의 정의에서 동일 배치에서 정의된 변수들을 참조할 수 있습니다. 뷰를 사용하는 것과 유사한 방식으로 INSERT, UPDATE, DELETE 및 CREATE VIEW 문에서 CTE를 사용할 수도 있습니다. 그러나 CTE의 진정한 위력은 바로 CTE가 이들에 대한 참조를 포함하고 있는 재귀적 기능에 있습니다. 쿼리 결과를 참조하고 싶지만, 데이터베이스에서 영구 뷰를 생성하고 싶지는 않을 때 마치 테이블처럼 파생 테이블을 사용할 수 있습니다. 그러나 파생 테이블에는 결정적으로 CTE와 구별되는 한계점이 존재합니다. 즉, 쿼리에서 파생 테이블을 한 번 정의하여 여러 차례에 걸쳐 사용할 수 없기 때문에 동일한 쿼리에서 여러 파생 테이블을 정의해야 한다는 것입니다. 또는 CTE를 한 번 정의하여 데이터베이스에서 영구화하지 않고 쿼리에서 여러 차례 사용할 수도 있습니다.

비재귀적 CTE는 사용자의 표현 능력을 향상시킵니다. 그러나 비재귀적 CTE를 사용하는 각 코드 부분의 경우에는 파생 테이블과 같은 다른 Transact-SQL 구문을 사용하여 동일한 결과를 얻을 수 있는 더 긴 코드를 작성할 수 있습니다. 이 경우는 재귀적 CTE와 다릅니다. CTE가 그 자체를 참조하는 경우 재귀적이라고 간주합니다. 재귀적 CTE는 최소 2개의 쿼리 부분(또는 구성원, 재귀적 쿼리 어법에서)에서 생성됩니다. 한 부분은 비재귀적 쿼리 부분이면서 AM(Anchor Member)으로서 참조되며 다른 부분은 재귀적 쿼리 부분이면서 RM(Recursive Member)으로서 참조됩니다. 쿼리 부분들은 UNION ALL 연산자에 의해 단일 CTE로 조인됩니다.

PIVOT 및 UNPIVOT 연산자

SQL Server 2005는 쿼리의 FROM 구문에서 사용하는 PIVOT과 UNPIVOT이라는 2개의 새로운 관계형 연산자를 제공합니다. 이들 연산자는 입력 테이블 반환식에서 일부 조작을 수행하여 결과로서 출력 테이블을 반환합니다. PIVOT 연산자는 행을 열로 회전시켜 계속해서 집계를 수행합니다. 주어진 피벗 열을 기준으로 입력 테이블 반환식의 범위를 넓혀 피벗 열의 고유한 각각의 값에 대한 열이 있는 출력 테이블을 생성합니다.

PIVOT 연산자는 오픈 스키마 시나리오를 처리할 때와 크로스 탭 보고서를 생성할 때 유용합니다. 오픈 스키마 시나리오에서는 이전에 알려지지 않았거나 각 엔터티 형식이 다른 속성 집합을 가진 엔터티를 유지합니다. 응용 프로그램 사용자들은 동적으로 속성을 정의합니다. 많은 열을 미리 정의하고 많은 NULL을 테이블에 저장하는 대신에 속성을 다른 열로 분리하여 각 엔터티 인스턴스에 관련 속성만 저장합니다. PIVOT을 사용하면 오픈 스키마는 물론, 행을 열로 회전하는 다른 시나리오에서 크로스 탭 보고서를 생성할 수 있어 계속해서 총계를 계산할 수 있으며 유용한 형식으로 해당 데이터를 표시할 수 있습니다.

UNPIVOT 연산자는 열을 행으로 회전시켜 PIVOT과 반대의 연산을 수행합니다. 이는 피벗 열을 기준으로 입력 테이블 반환 식의 범위를 좁힙니다. UNPIVOT 연산자를 사용하여 이전에 피벗된 데이터를 정상화할 수 있습니다.

APPLY 연산자

APPLY 연산자를 통해 SQL Server 2005에서는 관련 하위 쿼리에서 테이블 반환 함수를 참조할 수 있습니다. APPLY 관계형 연산자를 사용하면 외부 테이블 반환식의 각 행에 지정된 테이블 반환 함수를 한 번 호출할 수 있습니다. JOIN 관계형 연산자를 사용하는 것과 유사한 방식으로 쿼리의 FROM 구문에서 APPLY를 지정합니다. APPLY는 CROSS APPLY와 OUTER APPLY의 2가지 형식으로 나타납니다.

CROSS APPLY는 외부 테이블 반환식의 각 행에 테이블 반환 함수를 호출합니다. 테이블 반환 함수의 인수로서 외부 테이블의 열을 참조할 수 있습니다. CROSS APPLY는 테이블 반환 함수의 개별 호출로 반환된 모든 결과가 컴파일된 통합 결과 집합을 반환합니다. 테이블 반환 함수가 주어진 외부 행의 빈 집합을 반환하면 해당 외부 행은 결과로 반환되지 않습니다.

OUTER APPLY는 CROSS APPLY와 매우 유사하지만, 테이블 반환 함수가 빈 집합을 반환하는 외부 테이블의 행도 반환한다는 차이점이 있습니다. NULL은 테이블 반환 함수의 열에 해당하는 열 값으로 반환됩니다.

트랜잭션의 예외 처리

SQL Server의 초기 버전에서는 Microsoft Visual Basic 6.0과 마찬가지로 오류의 소지가 있는 모든 구문 다음에는 오류 처리 코드를 포함해야 했으며 오류 검사 코드를 중앙화하려면 레이블과 GOTO 문을 사용해야 했습니다. 더욱이 데이터 형식 전환 오류와 같은 오류가 발생하면 배치가 종료되어 버리기 때문에 Transact-SQL에서 이를 발견할 수 없었습니다. SQL Server 2005는 Visual Basic .NET 및 C#에서와 마찬가지로 TRY/CATCH Transact-SQL 구문 형식의 단순하면서도 강력한 예외 처리 메커니즘을 도입하여 이러한 많은 문제를 해결합니다. 이제 과거 구문, 수준, 배치 또는 트랜잭션의 종료를 야기했던 오류들을 발견하고 처리할 수 있게 됐기 때문에 이들 오류들은 더 이상 연결의 단절을 야기할 만큼 심각한 문제가 아니라고 간주할 수 있습니다.

오류 처리를 실행하려면 BEGIN TRY/END TRY 블록 내에서 실행하려는 코드를 작성하여 BEGIN CATCH /END CATCH 블록에서 오류 처리 코드를 따르기만 하면 됩니다. TRY 블록은 해당 CATCH 블록이 있어야 합니다. 그렇지 않으면 구문 오류가 발생할 것입니다.

DDL 이벤트 알림

SQL Server 2005에서는 DDL과 시스템 이벤트를 포착하여 Service Broker 서비스로 이벤트 알림을 보낼 수 있습니다. 동시에 처리되는 트리거와는 반대로 이벤트 알림은 비동기 실행을 지원하는 이벤트 메커니즘입니다. 이벤트 알림은 지정된 Service Broker 서비스로 XML 데이터를 전송하고 이벤트 사용자는 비동기적으로 이 데이터를 사용합니다. 이벤트 사용자는 Service Broker Receive 문에서 WAITFOR 구문에 대한 확장을 사용하여 도착할 새 데이터를 기다릴 수 있습니다.

전체 텍스트 검색 기능 향상

SQL Server 2005에는 풍부한 전체 텍스트 응용 프로그램에 대한 지원이 포함되어 있습니다. 카탈로그 성능도 강화되어 카탈로그할 내용에 있어서 더 큰 유연성을 발휘할 수 있습니다. 쿼리 성능과 확장성도 획기적으로 향상되었으며 새로운 관리 도구가 전체 텍스트 구현에 대한 보다 큰 통찰력을 제공합니다.

향상된 보안 기능

SQL Server 2005는 보안, 개인 정보, 안정성 및 비즈니스 무결성 부문에서 고객 경험을 향상시킨다는 Microsoft의 Trustworthy Computing 프로젝트를 통해 상당한 성과를 거두었습니다. 2002년 1월 전사적으로 발표된 이 프로젝트에 따라 Microsoft는 이제 자사 제품 및 제품 배치가 설계 단계에서, 기본 설정에서, 그리고 배치 단계에서도 안전하다는 것을 보장할 수 있도록 뒷받침하는 개발 프로세스를 따르고 있습니다. Microsoft SQL Server 개발 팀은 SQL Server 2005의 개발에 이러한 프로세스를 통합했습니다. 배포 후에도 보안 문제에 대한 고객 및 파트너의 지속적인 커뮤니케이션을 지원합니다. 이에 따라 SQL Server 2005는 지금까지 출시된 모든 SQL Server 릴리스 중 가장 광범위한 보안 기능을 포함하게 될 것입니다.

일반적으로 이러한 기능과 향상점들은 다음 3가지 부문으로 구분됩니다.

- **SQL 서버에 대한 사용자 액세스 제한** : SQL Server 액세스를 보다 강력하게 제어할 수 있는 것은 물론, 관리자가 정책을 통해 SQL Server에 대한 액세스를 제어할 수 있는 메커니즘에 대한 투자를 최대한 활용할 수 있습니다.
- **서비스 사용 금지 및 서비스 구성 제한** : 관리자가 지정한 범위에서 세분화된 수준으로 SQL Server 내 자원에 대한 액세스를 제한할 수 있으며 최소 권한 원칙을 위반하지 않고도 손쉽게 시스템을 관리할 수 있도록 보장합니다. 새 서버 설치 시 기본적으로 특정 서비스를 사용할 수 없기 때문에 고객은 사용하려는 특정 추가 서비스를 결정하는 데 있어 더욱 능동적으로 개입하게 됩니다.
- **새 기능에 대한 공격 범위 축소** : SQL Server의 설치 및 설정에서 출발하기 때문에 공격 범위가 최소화됩니다. 새로운 특징들은 제품 개발 주기 전반에 걸쳐 공격 범위를 줄일 수 있도록 보안 측면에 대한 검토 및 검증 작업을 수행하게 됩니다.

이러한 서버 보안과 관련된 주요 향상 기능 외에도 SQL Server 2005는 고유 데이터 암호화 기능을 추가했습니다. SQL Server 기반으로 구축된 인증서와 주요 관리 시스템을 사용하고 일부 새로운 시스템 기능을 활용함으로써 이제 서버에서 데이터를 암호화하고 해제할 수 있습니다.

결론

Microsoft SQL Server 2005는 개발자들에게 새로운 데이터베이스 응용 프로그램 클래스를 구축하는 데 필요한 도구를 제공합니다. 코드 실행 및 스토리지 위치에 대한 장애 요인을 제거하고 XML과 같은 표준을 통합함으로써, SQL Server 2005는 데이터베이스 개발자에게 새로운 가능성을 열어주고 있습니다. 본 백서에서는 SQL Server 2005의 기능에 대해 간략히 살펴 보았습니다.

Microsoft®

한국마이크로소프트(유)

서울특별시 강남구 대치동 892번지 포스코센터 | 전화 : 080-985-2000 | 인터넷 : www.microsoft.com/korea