

Microsoft

SQL Server 2005

Microsoft

SQL Server[™] 2005

업그레이드 가이드



Microsoft

저자 약력

김민정 (주)필라넷 주임 컨설턴트

전 그린컴퓨터 전임강사

컨설팅: 휴넷, Yes24, 호남 석유화학, 부산은행, 동양증권 외

감수자 약력

정원혁 (주)필라넷 상무 이사, DB사업부장, MCDBA, MCT

전문가로 가는 지름길, MS SQL Server 2000외 7권 저술, SQL Server 매거진 전문 번역
전, 마이크로소프트, 이랜드

컨설팅: 두산, 롯데 카드, 삼성 반도체, KT, CJ CGV, 대한 생명, 동양 증권, 동양 생명,
신한 은행, 제일 은행 외

서문

업그레이드 열풍에 사는 대한민국에서 지내는 IT 업계 종사자는, 새로운 버전이 나오면 당연히 이를 사용하고 싶습니다. 하지만 몇 차례 early adaptor로 선두에 나섰다가, 당해 본” 경험이 있으면 움츠리게 됩니다. 이런 경험은 업그레이드에 큰 걸림돌이 되며, 업그레이드 이전에 스트레스로 이어지게 됩니다.

이 소책자는 그런 걸림돌을 줄일 수 있도록 돕기 위해 작성 되었습니다. SQL Server 2005에는 이런 불신을 덜어줄 수 있는 좋은 도구가 포함되어 있는데 이것은 바로 Upgrade Advisor(업그레이드 점검도구)가 있으며 이를 사용해서 업그레이드 전에 생길 수 있는 문제점 혹은 해결해야 할 과제를 알려주고 있습니다. 이 책자는 이 기능에 대해 소개합니다. 진작 이런 도구가 있었으면 6.5, 7.0 시절의 업그레이드로 인한 놀란 가슴 사건을 줄일 수 있었을텐데 하는 아쉬움이 있지만, 그 만큼 이 도구는 탁월한 도구입니다. 하지만, 그래도 아직은 시간이 부족하여 발견하지 못한 업그레이드에 관련된 이슈가 있다면 그에 대해서는 이 책은 다루지 못합니다. 이런 것들은 발견되는 대로 Microsoft 웹사이트, 세미나, 뉴스레터등을 통해 공지 할 것입니다.

SQL Server 2005는 그 놀라운 향상된 기능과 속도, 편리함으로 인해서, 특별히 선전하지 않아도 **사용하고 싶어지는 데이터베이스**입니다. 이 책은 기존 SQL Server 2000 사용자를 위해 바로 이런 기능들을 어떻게 활용할 수 있는지, SQL Server 2000에서 어떻게 업그레이드를 할 수 있는지 하나의 지침서가 될 수 있도록 작성되었습니다. (보다 자세한 SQL Server 2005의 새로운 기능에 대해서는 개발자/관리자를 위한 두 권의 별도의 소책자가 준비 됩니다.) 그 동안, 필라넷 DB 사업부는 SQL Server 2005의 업그레이드 프로그램을 지원하는 협력 업체로 몇 고객사의 SQL Server 2005 업그레이드를 지원 했습니다. 이런 경험을 바탕으로 소책자를 만들었습니다.

경기가 풀리나 봅니다. SQL Server에 유능한 인력을 구한다는 소식이 여기 저기서 들립니다. SQL Server 2005의 출시와 함께 우리 경기에도 새 바람이 불기를 기대합니다. SQL Server에 유능한 인력들이 더 많아지길 기대해 봅니다.

목차

업그레이드 계획세우기	6
업그레이드와 마이그레이션 개념 정의	6
업그레이드 및 마이그레이션 대상	6
마이그레이션 장점과 단점	7
업그레이드 장점과 단점	8
업그레이드 계획 수립	8
업그레이드 사전 점검사항.....	9
서버 환경 점검	9
SQL Server 2005 에디션	9
SQL Server 2005 하드웨어 요구사항 점검(32비트 시스템 기준).....	10
확장성 및 성능	11
SQL Server 2005의 운영체제 요구사항(32비트기준)	11
이전 버전에서의 응용프로그램 호환성 점검	12
업그레이드 점검도구 설치와 수행.....	13
업그레이드 어드바이저(Upgrade Advisor) 도구	13
업그레이드 어드바이저 설치방법	13
업그레이드 어드바이저 실행전 준비사항.....	14
업그레이드 어드바이저 설치	16
업그레이드 점검 도구 결과 분석	26
90호환성 모드 지원 분석 결과	26
문서화되지 않은 시스템 테이블 참조	27
기타 데이터베이스 엔진 문제.....	29

업그레이드 점검 도구 결과에서 나타난 문제의 해결	30
-----------------------------------	----

SQL Server 2005 설치	31
--------------------------	----

SQL Server 2005 설치 개요	31
시스템 구성 검사(System Consistency Checker (SCC))	32
SQL Server 2005 구성요소 설치 마법사	35
설치후 파일 위치(file locations)	52
셋업 로그파일 검사하기	52
레지스트리에서 인스턴스 이름 찾아보기	53
SQL Server 서비스 확인	53
SQL Server 2005 마이그레이션시 인스턴스명	54
SQL Server 2000에서 SQL Server 2005로 업그레이드시 인스턴스	56
마이그레이션시 SQL Server 2000로부터 사용자 데이터베이스 가져오기	57
무인 설치 수행하기	59
원격 서버 설치 수행하기	60
클러스터에 SQL 서버 설치하기	61
SQL Server 2005 구성요소 추가 또는 삭제	65
SQL Server 2005 삭제	65

SQL Server 2005의 새로운 기능	66
-------------------------------	----

SQL Server 2005 개요	66
SQL Server 2005 구성요소(components)	66
데이터베이스 관리자를 위한 새로운 기능	68
데이터베이스 개발자를 위한 새로운 기능	70
새로운 관리도구 - SQL Server Management Studio	71

새로운 기능을 사용할 수 있는 쿼리 찾아 변경하기	74
TOP(@n)구문.....	74
CTE(Common Table Expression) 테이블	75
RANK()함수	79
ROW_NUMBER()함수.....	81
PIVOT연산자.....	84
TRY..CATCH 구문	86
SQL Server management에서의 성능 자료 수집	94
프로필러를 이용한 성능 자료 수집	98

업그레이드 계획세우기

업그레이드를 할지 마이그레이션을 할지 결정하여야 하며 순조로운 업그레이드를 위해서 사전작업,수행작업,업그레이드 이후의 점검사항은 무엇인지 계획을 세워야합니다

업그레이드와 마이그레이션 개념 정의

■ 업그레이드(Upgrade)

- 기존 설치를 업데이트하는 것입니다
- 인스턴스명이 그대로 유지됩니다
- 프로세스가 자동화됩니다

■ 마이그레이션(Migration)

- 새로운 설치부터 시작되는 것입니다
- 이전 인스턴스와 새로운 인스턴스가 각각 유지됩니다
- 대부분의 프로세스가 수작업입니다

업그레이드 및 마이그레이션 대상

■ 버전(Version)

- SQL Server 7.0(이후서비스팩)
- SQL Server 2000(이후 서비스팩)

■ 구성요소(Component)

- 데이터베이스 엔진(Database Engine)
- 분석 서비스(Analysis Services)
- 리포팅 서비스(Reporting Services)
- 알림 서비스(Notification Services)

- 데이터 변환 서비스(Migration of Data Transformation Services to Integration Services)

■ 에디션(Edition)

- Desktop, Workgroup, Personal, Standard, Developer, Enterprise

■ 플랫폼(Platform)

- 32-bit & 64-bit

■ 언어(Language)

- 모든 SQL Server 7.0 and SQL Server 2000 릴리즈 언어

마이그레이션 장점과 단점

■ 장점

- 업그레이드 프로세스에 관한 세부적인 관리를 제공합니다
- 이전 인스턴스와 새로운 인스턴스에 대한 테스트 및 확인을 합니다
- 마이그레이션중 레거시 인스턴스를 유지합니다
- 마이그레이션 오류를 수정합니다

■ 단점

- 새로운 혹은 추가적인 하드웨어 자원을 요구합니다
- 응용프로그램의 새로운 인스턴스에 대한 수정 요구가 발생합니다

업그레이드 장점과 단점

■ 장점

- 소규모의 시스템에서 더 쉽고 빠릅니다
- 추가적인 하드웨어가 불필요합니다
- 인스턴스의 변화가 없어 응용프로그램의 수정이 불필요합니다

■ 단점

- 업그레이드 진행중 오프라인 상태가 발생합니다
- 모든 구성요소에 대해 가장 좋은 방법은 아닙니다
- 분석서비스의 큐브들은 마이그레이션을 권장합니다

업그레이드 계획 수립

■ 업그레이드 계획 단계

■ 업그레이드 사전작업

■ 업그레이드 작업

■ 업그레이드 이후 점검

■ 예행 연습

■ 계획 실행

업그레이드 사전 점검사항

서버 환경 점검

- SQL Server 2005 하드웨어 및 소프트웨어 최소요구사항을 확인합니다
- 응용 프로그램 및 레거시 시스템을 점검합니다
 - 예) 릴리즈, 구성요소, 플랫폼..
- 2005에서 지원되지 않는 것에 어떤 것들이 있는지 확인합니다
 - 예) 문서화되지 않은 시스템 프로시저, 가상큐브, 가상차원
- 업그레이드 어드바이저를 실행하여 이전 버전과의 호환성 이슈를 점검합니다
- 업그레이드 실행 이전의 벤치마크 내용을 기록합니다

SQL Server 2005 에디션

SQL Server 2005는 여러 에디션이 있습니다. 여러분의 환경에 어떤 것이 적절한지 각 에디션 간의 차이를 이해하는 것이 중요합니다

SQL Server 2005는 익스프레스 에디션을 제외하고 모든 에디션이 32-bit 와 64-bit에 설치 가능합니다

아래와 같은 정식 에디션 이외 120일 평가판 에디션이 있습니다

에디션	설명
엔터프라이즈 에디션 (Enterprise Edition) (32bit와 64bit)	대기업이 주고객이 되는 버전이다. 가장 크고 복잡한 기업 환경의 온라인 트랜잭션 처리(OLTP), 고도의 데이터 분석 서비스, 데이터 웨어 하우징, 고가용성기능을 제공합니다.
표준 에디션 (Standard Edition) (32bit와 64bit)	소규모와 중간크기 조직에 맞는 데이터 관리 및 분석 플랫폼이다 2000버전에서는 엔터프라이즈 버전에서만 지원되었던 클러스터링기능을 지원하며 모든 영역의 BI기능을 지원합니다.

워크그룹 에디션 (Workgroup Edition) (32bit와 64bit)	새로 나온 버전이다. 중소기업의 기업과 단위 부서별 설치에 맞게 설계되었다. 이 버전은 BI와 표준 및 엔터프라이즈에서 가지고 있는 고가용성 기능을 제외한 관계형 데이터베이스를 지원합니다.
익스프레스 에디션 (Express Edition) (32bit와 64bit)	MSDE(Microsoft SQL Server Desktop Engine)를 대체하기 위해 만든 버전이다 CLR통합기능을 지원합니다.
개발자 에디션 (Developer Edition) (32bit와 64bit)	개발과 테스트를 위한 데이터베이스 에디션입니다. 엔터프라이즈 에디션의 모든 기능을 사용하여 볼 수 있으나 프러덕션 서버에서는 사용할 수 없습니다.

SQL Server 2005 하드웨어 요구사항 점검(32비트 시스템 기준)

하드웨어 최소 요구사항을 확인해야만 합니다. 만족되지 않는 경우 전체 또는 일부의 구성요소(components)가 설치 되지 않을 수 있습니다

하드웨어	요구사항
프로세서	500-megahertz (MHz) 또는 그 이상 (1 gigahertz 또는 그 이상 수준을 권장)
메모리	엔터프라이즈 에디션 : 512MB (1GB 또는 그 이상 권장) 표준 에디션 : 512MB ,1GB 또는 그 이상 권장) 워크그룹 에디션 : 512MB(1GB 또는 그이상 권장;최대 3GB) 평가 에디션 : 512MB ,1GB 또는 그 이상 권장) 개발자 에디션 : 512MB ,1GB 또는 그 이상 권장) 익스프레스 에디션: 512MB(1GB 또는 그이상 권장;최대 1GB)
디스크	전체 설치를 위해 350MB의 사용가능한 공간 필요 예제 데이터베이스를 위해 390MB 공간 필요

확장성 및 성능

기능	Express	Workgroup	Standard	Enterprise
CPU수	1	2	4	제한없음
RAM	1GB	3GB	제한없음	제한없음
64비트 지원	WOW(Windows On Windows)	WOW	✓	✓
데이터베이스크기	4GB	제한없음	제한없음	제한없음

SQL Server 2005의 운영체제 요구사항(32비트기준)

	엔터 프라이즈	표준	워크그룹	평가	개발자	익스 프레스
Windows 2003 Enterprise SP1 (all pc editions)	✓	✓	✓	✓	✓	✓
Windows XP Professional SP2			✓	✓	✓	✓
Windows XP Home Edition SP2					✓	✓
Windows 2000 professional SP4			✓	✓	✓	✓
Windows 2000 Server SP4(all editions)	✓	✓	✓	✓	✓	✓

■ 추가적인 소프트웨어 요구사항입니다

- Microsoft Internet Explorer 6.0 서비스팩 1 또는 그이상(Microsoft Management Console에 필요합니다)
- Internet Information Services (IIS) 5.0 또는 그이상 (Reporting Services에 필요합니다)
- TCP/IP networking
- Microsoft .NET Framework 2.0 (SQL Server설치에 필요하다면)
- Microsoft Windows Installer 2.0 (SQL Server 설치에 필요하다면)

이전 버전에서의 응용프로그램 호환성 점검

■ 이전버전과의 호환성 (Backward compatibility)

- SQL Server 2005 는 이전 버전 데이터베이스 호환성 설정을 위해 sp_dbcmtlevel 프로시저를 제공합니다 하나의 인수를 갖습니다

@new_cmptlevel 설정값	SQL Server 버전
60	SQL Server 6.0
65	SQL Server 6.5
70	SQL Server 7.0
80	SQL Server 2000
90	SQL Server 2005

[참고]

SQL Server Management Studio 와 SQL Management Objects(SMO)는 60레벨을 지원하지 않습니다

■ 정렬(Collation Options)

설치하는 동안 일반적으로 Windows 기본값 정렬옵션을 선택합니다. 그러나 SQL Server 7.0 또는 그 이상버전 과의 호환이 필요하다면 설치 시에 또는 추후 변경 가능합니다

업그레이드 점검도구 설치와 수행

업그레이드 과 마이그레이션 중 어떤 것이든 업그레이드 어드바이저를 실행하여야 합니다. 업그레이드 어드바이저는 CD 또는 Microsoft.com에서 무료로 다운로드 받을 수 있습니다.

업그레이드 어드바이저(Upgrade Advisor) 도구

- Microsoft SQL Server 2005 업그레이드 어드바이저(Upgrade Advisor)는 SQL Server 2005 업그레이드 프로세스를 진행하는 도구입니다. 이는 설치된 SQL Server 구성요소를 분석하는 성공적인 업그레이드를 위해 확인해야 하는 확인된 문제들에 관한 보고서를 만들어 냅니다
- SQL Server 2005 업그레이드 실행 경험을 증진합니다
- 업그레이드 진행 또는 이후의 예상외의 사항을 체크합니다
- SQL Server 2000 과 SQL 7.0 인스턴스를 분석합니다
- 사용중인 서버,스크립트,추적파일을 분석합니다
- 발견된 이슈들에 대한 보고서를 제공합니다
- 수정되어야 할 발견된 이슈들에 관한 지침을 제공합니다
- 이슈들에 관한 수정 및 작업형태를 기술합니다
- 추가 사항에 관한 정보를 제공합니다

업그레이드 어드바이저 설치방법

두 가지 방법으로 설치할 수 있습니다

- SQL Server 2005 CD1로부터 `₩Servers₩redist`폴더를 열어 봅니다
 - SQLUasetup.msi 파일을 구동시킨 후 화면 지시에 따라합니다

■ 마이크로 소프트 웹사이트에서 다운로드합니다

- 웹사이트에서 열거나 저장할 수 있습니다. 만약 열기를 선택하면 압축파일을 풀어낼 경로를 지정하여야 할 것 입니다. 만약 저장한다면 대화상자를 열어 저장하고 exe 확장자 파일을 저장할수 있습니다. 그러한 이후에 내용을 풀어냅니다

업그레이드 어드바이저 실행전 준비사항

■ Windows 2000 SP4 또는 이상 , Windows XP SP2 또는 그 이상, Windows Server 2003 SP1 이상을 필요로 합니다

■ 마이크로소프트 인스톨러 3.1또는 그이상을 요구합니다

■ 닷넷 프레임워크 버전 2.0이 윈도우 인스톨러 3.1을 필요로 합니다. 마이크로 웹사이트로부터 인스톨러를 윈도우에 설치할 수 있습니다

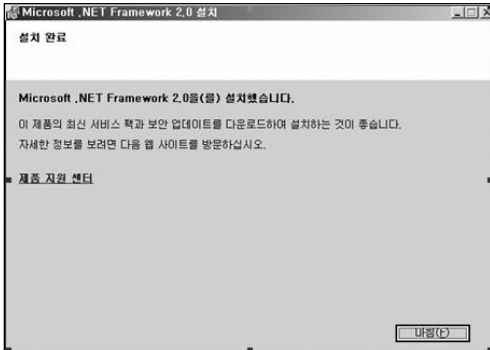
■ 32-bit 버전의 Microsoft .NET Framework 2.0을 필요로 32-bit 버전의 Microsoft .NET Framework 2.0은 SQL Server 2005 제품 미디어에 사용됩니다.

■ 닷넷 프레임워크2.0을 SQL Server 2005 미디어로부터 설치하기 위하여 CD/DVD 드라이브의 redistributable - 2.0 폴더를 열어 dotnetfx.exe (for 32 bit) 또는dotnetfx64.exe (for 64 bit)를 실행시킵니다.

■ 업그레이드 어드바이저가 닷넷 프레임을 필요로 하는 경고가 발생합니다



■ the 32-bit version of Microsoft[williamc1].NET Framework 2.0을 설치합니다



- SQL Server 2000 DSO(decision support objects)는 분석서비스에서 업그레이드 문제를 검색할 수 있습니다. DSO를 설치하기 위하여 SQL Server 2000 미디어를 CD/DVD드라이브에 삽입합니다. SQL Server 2000 설치 프로그램을 시작합니다 SQL Server 2000 구성요소를 선택합니다. 분석 서비스 설치 프로그램을 시작하여 분석 서비스를 선택합니다. 구성요소에서 DSO를 확인합니다.
- SQL Server 2000 클라이언트 구성요소는 DTS 패키지를 검색합니다. SQL Server 2000 CD를 CD/DVD드라이브에 설치한후 클라이언트 구성요소를 설치합니다.

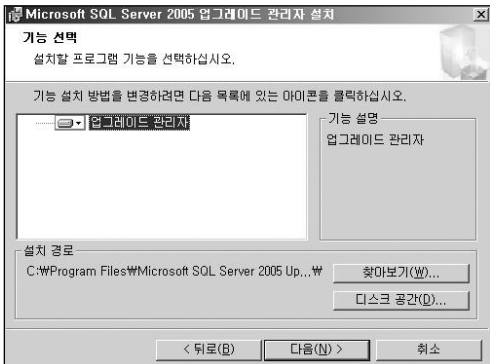
업그레이드 어드바이저 설치

- 업그레이드 어드바이저 분석 마법사는 총 다섯 단계를 가집니다
 - 분석하고자하는 서버와 구성요소
 - 추가적인 매개변수들을 수집합니다
 - 인증정보를 수집합니다
 - 선택한 구성요소를 분석합니다
 - 업그레이드 문제 보고서를 생성합니다
- 업그레이드 어드바이저(Upgrade Advisor)를 다운로드하거나 CD에서 실행합니다
 - <http://www.microsoft.com/sql/2005/default.mspix>

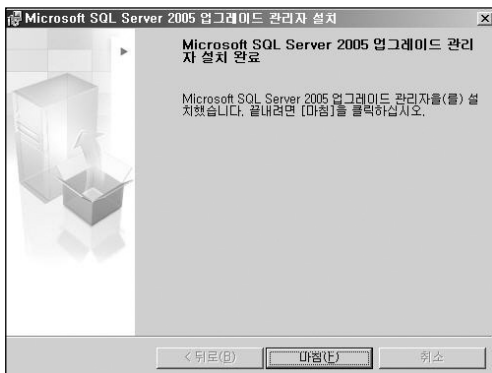
■ Upgrade Advisor 설치를 시작합니다



■ 기능 선택을 합니다

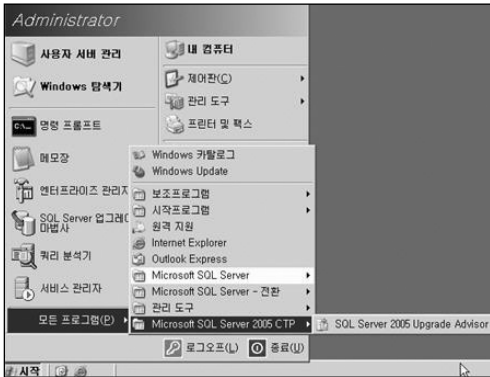


■ 마침을 누릅니다

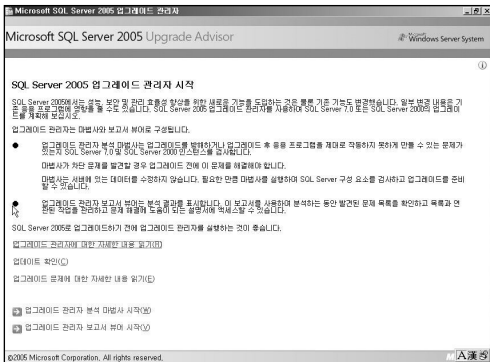


■ 업그레이드 어드바이저(Upgrade Advisor) 실행하기

시작 - 모든프로그램-Microsoft SQL Server 2005 - SQL Server 2005 Upgrade Advisor를 선택합니다



■ 업그레이드 관리자 분석 마법사를 선택합니다



■ 시작화면입니다



■ SQL Server 구성요소를 선택합니다

■ 서버를 확인하고 업그레이드 분석에 사용될 구성요소를 확인합니다

■ 업그레이드 마법사는 로컬 컴퓨터 이름의 서버 이름상자를 가지고 있습니다 로컬 컴퓨터에 연결하기 위해 "."과 "local host"를 사용할 수도 있습니다.

■ 만약 다른 컴퓨터를 분석하고자 한다면 서버 이름상자(name box)에 그 서버 이름을 입력 하십시오. 그런 다음 분석하고자 하는 구성요소를 선택합니다

■ 현재 서버에 설치된 모든 구성요소들을 선택하기 위하여 Detect버튼을 클릭할 수 있습니다

■ 이후 현재 서버에 설치된 구성요소들 옆에 체크마크가 나타납니다. 물론 수동으로 구성 요소옆의 체크박스를 선택하거나 해제하여 분석하고자 하는 구성소를 선택할 수 있습니다

- SQL Server 구성요소 목록
- SQL Server
- Analysis Services
- Notification Service

- Reporting Service
- 데이터 변환 서비스

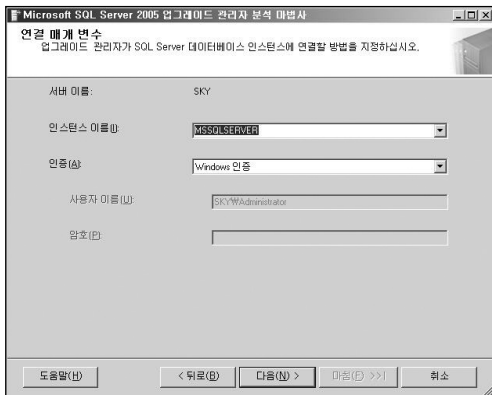


■ 연결 매개변수

서버타입의 업그레이드 분석을 위하여, 예를 들어 데이터베이스 엔진처럼 분석에 사용될 특정한 서버 인스턴스를 선택해야 합니다. 이번 페이지에서는 업그레이드 어드바이저가 분석하고자 하는 SQL Server 데이터베이스에 연결될 수 있도록 정보를 입력해야 합니다.

매개변수	값
서버이름	SQL Server 구성요소 창에 입력한 서버이름입니다
인스턴스 이름	현재 서버의 사용가능한 인스턴스 목록으로부터 선택합니다. 만약 인스턴스목록이 없다면 기본값 인스턴스를 검색하여 MSSQLSERVER의 기본값으로 사용하십시오. 이는 특히 원격 서버에 관련됩니다. 디폴트 인스턴스를 검색하여 "default" 라는 단어를 사용할 수 있습니다

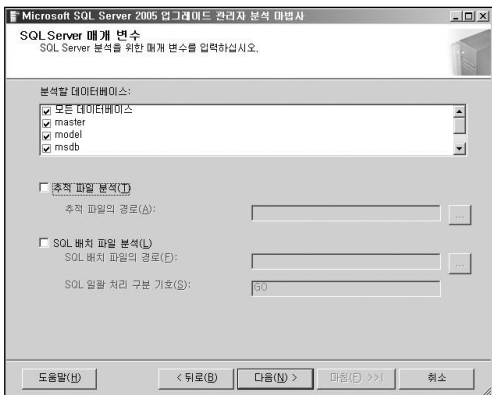
인증(Authentication)	현재 서버의 사용가능한 인증 방법의 목록을 선택하십시오. 기본적으로 인증모드는 마이크로 윈도우 인증입니다
사용자이름	상자에 사용자이름을 입력하십시오. 그 이름은 현재 서버에 관리자(administrator)권한을 가지고 있어야 합니다
암호	입력한 사용자이름에 대한 암호를 입력합니다.



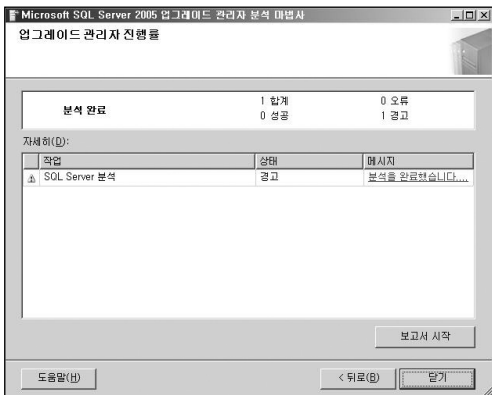
■ 데이터베이스와 추적파일 또는 배치파일을 선택하여 분석할 수 있습니다

하나 또는 여러 개의 데이터베이스를 선택할 수 있으며 MSSQL Server에 의해 생성된 프로필러 추적파일, SQL 배치 파일을 분석할 수 있습니다

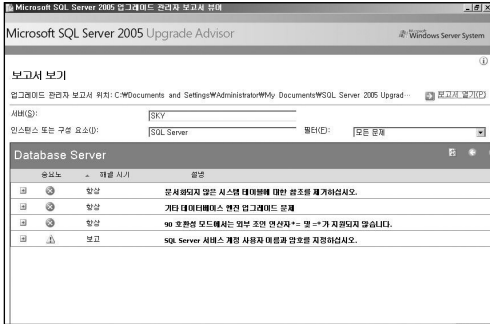
매개변수	값
데이터베이스 분석	모든 데이터베이스를 선택하기 위해서는 All Databases 를 선택합니다. 하나의 데이터베이스만을 선택하기 위해서는 해당 데이터베이스이름 앞의 체크박스를 클릭합니다
추적파일 분석	추적 파일을 분석하기 위해서는 이 옵션을 선택하여야 합니다. 기본적으로 선택되어 있지 않습니다
추적파일 경로	하나 또는 그 이상의 파일을 분석할 수 있습니다. 하나 이상의 파일이 선택되면 세미콜론이 분리자로 됩니다. 글상자 안에 직접 적을 수도 있고 UNC 또는 로컬 경로로 추적파일을 브라우징할 수 있습니다. 경로를 선택하지 않으면 NEXT 버튼은 활성화되지 않습니다
SQL 배치 파일 경로	하나 또는 그 이상의 *.sql 파일이나 T-SQL코드가 들어있는 텍스트 파일을 선택할 수 있습니다



■ 분석을 시작합니다



- 분석이 모두 끝난후에 Launch Report버튼을 누르면 보고서를 볼 수 있습니다



- 첫 화면인 SQL Server 2005 업그레이드 관리자에서 업그레이드 관리자 보고서 뷰어 시작 버튼을 선택하여 다시 결과화면을 볼 수 있습니다

업그레이드 점검 도구 결과 분석

내용	예	비고
90호환성 모드 지원	T-SQL Outer Join (=)	그림1
문서화 되지 않은 시스템 테이블 참조	spt_values,sysxlogins	그림3
기타 데이터베이스 엔진 업그레이드 문제		그림7

90호환성 모드 지원 분석 결과

■ 이 문제에 대한 상세 정보 및 해결 방법 보기 플러스버튼을 눌러 자세한 설명을 볼 수 있습니다

2005 업그레이드시 T-SQL 외부조인방식(=)을 사용하는 경우 90호환성 모드로 바꾼다면 더 이상 이 연산자를 지원하지 않으며 OUTER JOIN키워드를 사용해야 한다는 설명이 있습니다

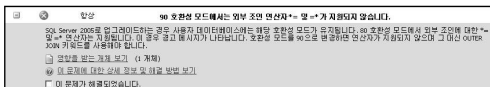


그림1

- 영향받는 개체 보기 를 클릭하여 봅니다 어떤 데이터베이스의 어떤 객체에 그러한 지적을 하였는지 알 수 있습니다



그림2

문서화되지 않은 시스템 테이블 참조

- 문서화 되지 않은 이전의 많은 시스템 테이블이 바뀌거나 더 이상 존재하지 않음을 알고 있습니다. 예고 없이 테이블이 바뀌어 질 수 있으므로 문서화되지 않은 객체들은 사용하지마십시오.

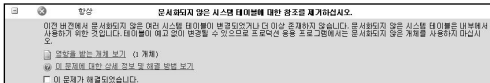


그림3

- 영향 받는 개체 보기 버튼을 클릭합니다



그림4

- 이 문제에 대한 상세정보 및 해결 방법 보기 버튼을 클릭합니다
- 업그레이드 어드바이저에서 다음의 문서화되지 않은 시스템 테이블이 SQL Server 2005 에서 제거 되었음을 알려줍니다 또한 그 대신 어떤 방식으로 수정할 수 있는지 알려줍니다

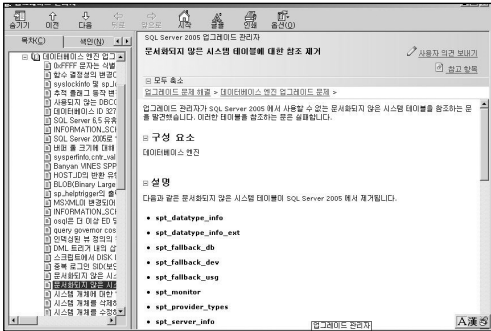


그림5

이전 사용	현재 사용
sysfulltextnotify	OBJECTPROPERTYEX 함수의 TableFulltextPendingChanges 속성
syslocks	sys.dm_tran_locks 동적 관리 뷰, sp_lock 또는 sys.syslockinfo 호환성 뷰
sysproperties	sys.extended_properties 카탈로그 뷰 또는 fn_listextendedproperty 함수
syslogins	sys.server_principals 카탈로그 뷰 또는 syslogins 호환성 뷰
모든 spt_ 테이블	대신 사용할 수 있는 항목 없음

그림6

기타 데이터베이스 엔진 문제

이 문제에 대한 상세 정보 및 해결 방법 보기를 선택하여 봅니다. 이 문제는 발견된다 하더라도 오해에서 비롯된 양성 반응(false positive) 결과 일 가능성이 크므로 업그레이드 문제라기 보다는 검토사항으로서 적절한 업데이트를 권고하고 있습니다.

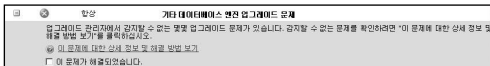


그림7

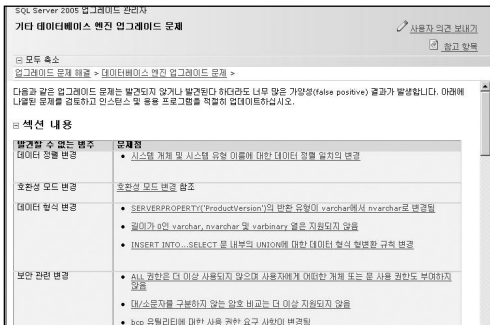


그림8

업그레이드 점검 도구 결과에서 나타난 문제의 해결

■ T-SQL형식의 외부조인 수정하기

[따라하기] T-SQL형식의 외부조인을 ANSI형식의 외부조인으로 수정합니다

```
use pubs
go
sp_helptext proc1
go
alter proc proc1
as
select au_lname, au_fname, title_id
from authors a left outer join titleauthor ta
on a.au_id = ta.au_id
```

■ 수정한 후 다시 Upgrade Advisor를 실행하여 봅니다. 외부조인 경고가 나타나지 않습니다

■ 문서화 되지 않은 시스템 테이블의 참조

- sysxlogins를 syslogins로 수정합니다

SQL Server 2005 설치

SQL 서버 데이터베이스 관리자는 SQL 서버 설치 요구사항, 이전 버전과의 공존 문제, SQL 서버 구성요소들의 추가 또는 삭제 절차에 대하여 잘 알고 있어야 합니다.

SQL Server 2005 설치, SQL Server 설치 유지에 대하여 알아봅니다.

SQL Server 2005 설치 개요

SQL Server 2005 설치는 크게 두 단계로 이루어집니다. 구성요소 업데이트와 SQL 셋업 MSI 패키지입니다.

■ 구성요소 업데이트

SQL Server 셋업 프로그램은 다음의 구성요소를 체크하고 필요하다면 인스톨합니다.

- 윈도우 인스톨러 2.0
- 닷넷 프레임워크 2.0
- SQL Server 셋업 지원 파일들

[참고]

구성요소 업데이트 단계가 끝난 후 셋업이 취소된다면 이 과정에서 설치된 구성요소는 제거되지 않습니다.

■ SQL 셋업 MSI 패키지

구성요소 업데이트 단계 이후에, 윈도우 인스톨러는 SQL 셋업 MSI 패키지를 설치합니다. 셋업은 다음절차로서 이루어집니다

1. 로컬 또는 원격 설치중에서 결정합니다.
2. 인스톨 형식(기본값 인스턴스 또는 명명된 인스턴스)을 결정합니다.
3. 시스템 구성 검사(System Consistency Checker)를 이용하여 컴퓨터를 분석합니다.

4. 인스톨 되어야될 기능을 결정하고 적절한 인스톨을 수행합니다.

시스템 구성 검사(System Consistency Checker (SCC))

이는 SQL 서버 설치 프로세스중 하나입니다. 많은 시스템 검사를하고 성공적인 SQL Server 2005 설치에 요구되는 항목에 필요한 것들을 비교합니다. 만약 호환되지 않는 것이 발견되면 시스템 구성 검사는 문제를 치료하기 위한 방법을 제안합니다.

SCC 검사는 크게 다섯 개의 범주로 나누어집니다. 시스템 구성, 시스템 가용성, 보안 구성, 버전 구성, 원격과 클러스터 구성입니다.

SCC 검사가 셋업하는 동안 무엇을 하는가를 이해하는 것은 인스톨을 준비하는데 도움을 줄 것입니다.

■ 시스템 구성 검사

검사 항목	설명	실패시 치료
운영체제 최소수준 요구사항	운영체제가 지원되는지 확인합니다	운영체제 요구사항에 관하여 사용자에게 문의합니다
운영체제 서비스팩 수준 요구사항	필요한 서비스팩이 운영체제에 적용되어 있는지 확인합니다	서비스팩에 필요한 윈도우 업데이트 구성요소에 대해 사용자에게 문의합니다
운영체제와 SQL 호환성	설치되고 있는 SQL Server 에디션을 운영체제가 지원하는지 확인합니다	운영체제/SQL 에디션 매트릭스를 참조합니다
32비트 또는 64비트 구성 요구사항	사용자가 32비트플랫폼에 32비트 릴리즈를 설치하고자 하는지 64비트 플랫폼에 64비트 릴리즈를 하고자 하는지 확인합니다	사용자에게 비호환성에 대하여 알려주고 셋업을 끝냅니다

Windows Management Interface(WMI) 서비스 요구사항	WMI가 인스톨되어 있는지 검사합니다	사용자에게 WMI가 반드시 필요함을 알리고 셋업을 중지합니다
최소 하드웨어 요구사항	최소 하드웨어 요구사항을 충족하는지 확인합니다.	설치되고 있는 에디션에 대한 하드웨어 요구사항을 사용자에게 문의합니다

■ 시스템 가용성 검사

검사 항목	설명	실패시 치료
배타적 뮤텍스 객체	다른 실행중인 셋업 인스턴스가 있는지 확인합니다	기존의 셋업 프로세스가 끝나고 재시작하는 동안 기다려야 함을 사용자에게 알려줍니다
재 부팅 요구사항에서 계속 기다리는 경우	필요한 셋업 파일들이 다른 서비스나 프로세스에 의해 잠겨있는지 확인합니다	서비스, 프로세스, 응용 프로그램 목록을 만들고 멈추어야만 한다. 셋업을 시작하기 전에 재 부팅을 요구할 수도 있습니다
세부기능 가용성	어떤 기능들이 현재 시스템 구성에 인스톨될 수 있는지 결정합니다	SCC 보고에 따라 사용할 수 없는 기능을 알려줍니다
기본값 설치 경로 허가 요구사항	드라이브가 포맷되었는지 읽기전용인지 또는 압축되었는지 검사합니다	사용자에게 문제를 알려주고 셋업을 중지합니다

■ 보안 구성 검사

검사 항목	설명	실패시 치료
시스템 관리자 권한	셋업 실행이 관리자(administrator)의 권한으로 수행된 것인지 확인합니다	시스템 관리자로서 수행해야 함을 알려줍니다

■ 버전 구성 검사

검사 항목	설명	실패시 치료
인터넷 탐색기 요구사항	인터넷 탐색기 6.0 서비스팩1 이상이 설치되어 있는지 확인합니다	브라우저 요구사항을 인스톨하기 위해 윈도우 구성요소 업데이트가 필요함을 알려줍니다
SQL Server 버전 검사	설치될 SQL Server 버전의 적합성을 검사합니다	서비스팩 인스톨 파일에 대해 문의합니다
MSXML요구사항	MSXML이 인스톨되어 있는지 확인합니다	MSXML인스톨을 위한 윈도우 구성요소 업데이트를 문의합니다

■ 원격과 클러스터 구성 검사

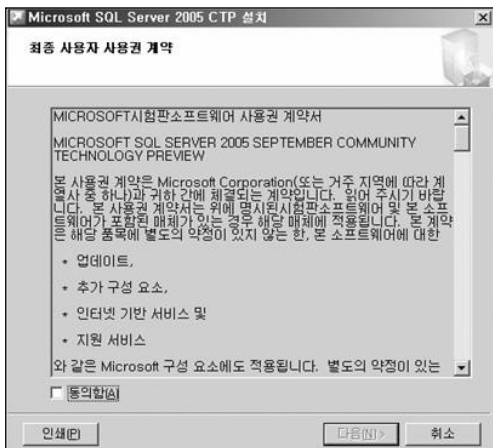
검사 항목	설명	실패시 치료
가능한 기본 관리상의 공유 디렉토리	원격 컴퓨터 또는 클러스터 노드에 대하여 관리상의 공유(admin\$)를 가능하게 합니다	공유를 하도록 촉구합니다

SQL Server 2005 구성요소 설치 마법사

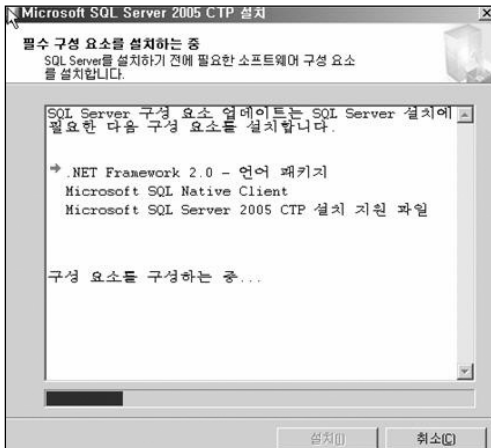
- 설치는 SQL Server 2005 9월 CTP버전을 기준으로 합니다
- 첫번째 CD를 삽입한 후 첫화면입니다. 자동실행이 되지 않으면 Setup.exe를 실행합니다
- 서버 구성 요소, 도구, 온라인 설명서 및 예제를 선택합니다



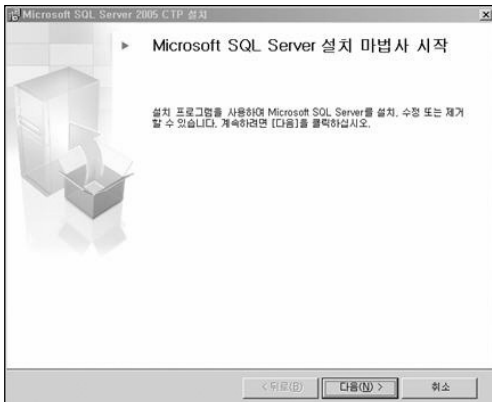
- 동의함 항목에 체크를 하고 다음버튼을 누릅니다



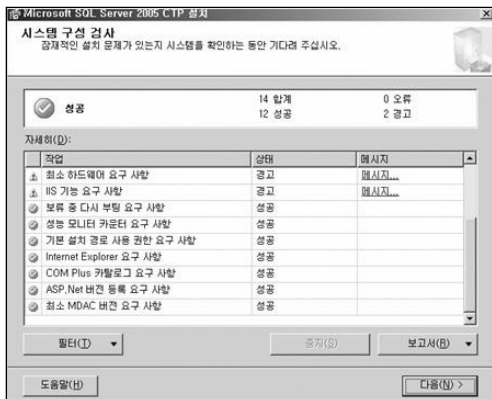
- 설치 버튼을 누르면 SQL Server 필수 구성요소 항목들이 올라옵니다



- 설치 마법사가 시스템의 구성을 검사합니다
- SQL Server 환영 페이지가 나타납니다

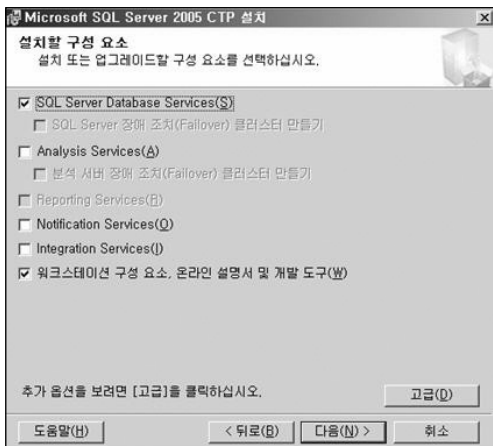


- 시스템 검사에 대한 결과가 나타납니다. IIS가 없으면 SQL 일부 기능을 설치할 수 없습니다



■ 윈도우에 설치될 구성요소들이 올라옵니다.

SQL Server, 분석 서비스, 알림서비스, 데이터변환 서비스, 워크스테이션 구성요소, 온라인 설명서와 개발 툴에 체크표시를 합니다.



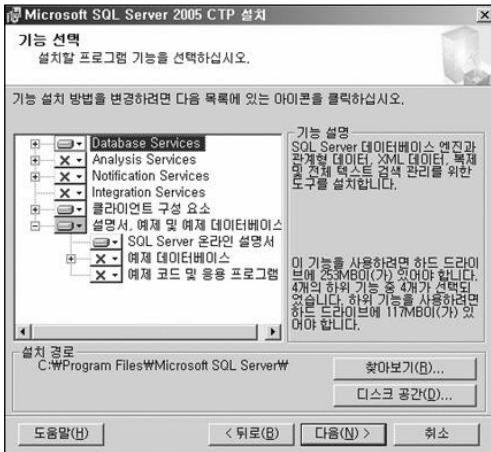
■ 고급버튼을 누릅니다



■ 설명서, 예제 및 예제 데이터베이스의 플러스 버튼을 눌러봅니다

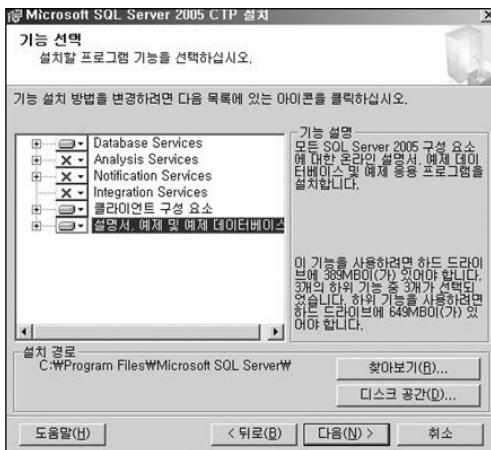
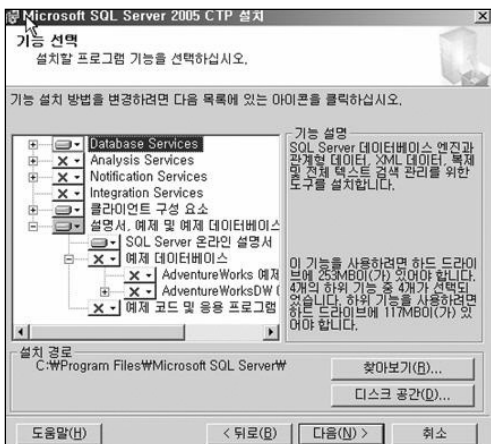
예제 데이터베이스와 예제 코드 및 응용 프로그램 선택이 되어 있지 않습니다

예제 데이터베이스에서도 AdventureWorks 와 AdventureWorksDW를 선택하여 설치할 수 있습니다



[참고]

이전 버전인 SQL 2000처럼 예제데이터베이스가 자동으로 설치되지 않습니다. 반드시 예제 데이터베이스를 선택하여야 설치됩니다



■ 업그레이드와 마이그레이션의 선택

- 업그레이드라면 기본값을 선택합니다
- 마이그레이션이라면 명명된 인스턴스를 사용하여 이름을 적습니다
- 업그레이드 하기 - 기본 인스턴스를 지정합니다

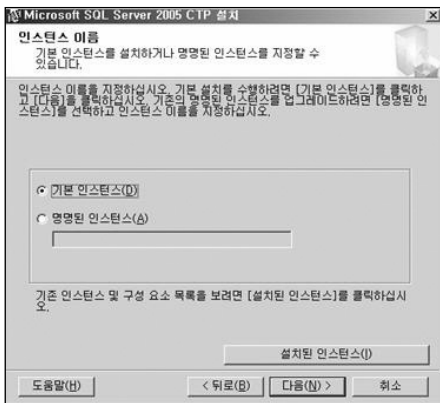


그림: 업그레이드 1

- 설치된 인스턴스 버튼을 누르면 현재 설치된 인스턴스에 대한 정보가 정리되어 있습니다
- 확인을 누른 후 다시 돌아가 다음버튼을 누릅니다
- 업그레이드할 구성요소가 나타납니다

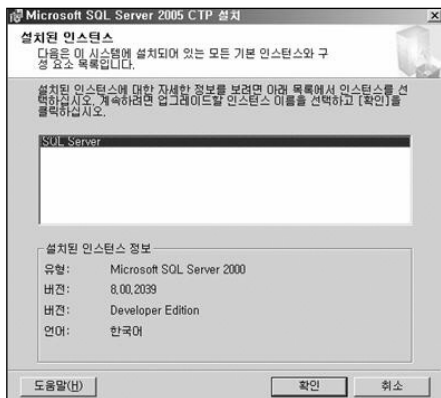


그림: 업그레이드 2

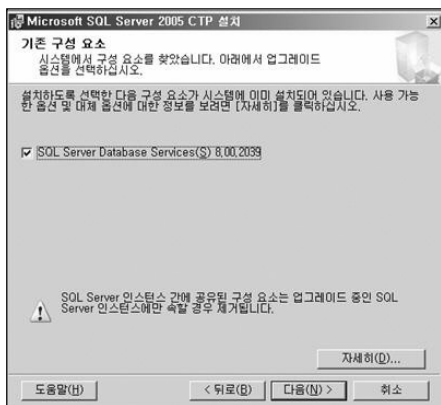


그림: 업그레이드 3

- 업그레이드 하기에서는 먼저 인증모드선택으로 들어갑니다
- 업그레이드 분석을 한 후 이전 SQL2000의 서비스 계정 설정을 기반으로 하여 업그레이드 서비스 계정 설정이 진행됩니다

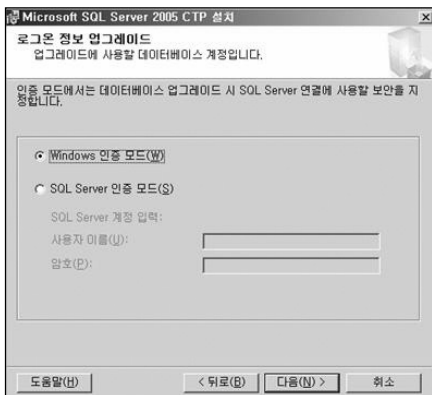


그림: 업그레이드 4

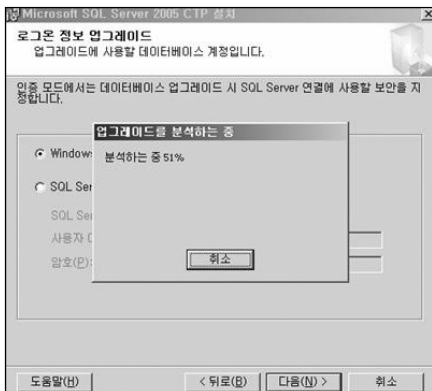


그림: 업그레이드 5

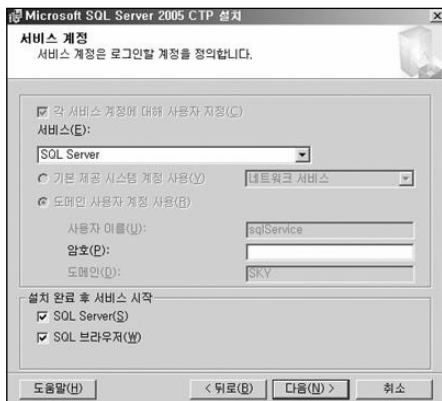


그림: 업그레이드 6



그림: 업그레이드 7

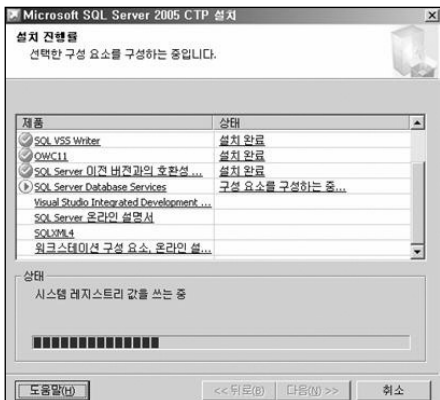
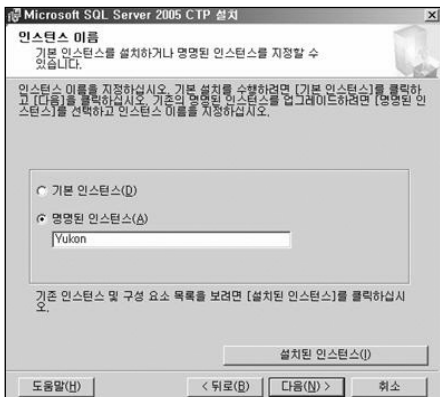


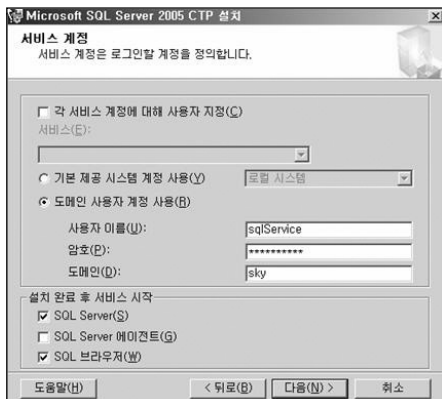
그림: 업그레이드 8

■ 마이그레이션 하기- 명명된 인스턴스를 지정합니다

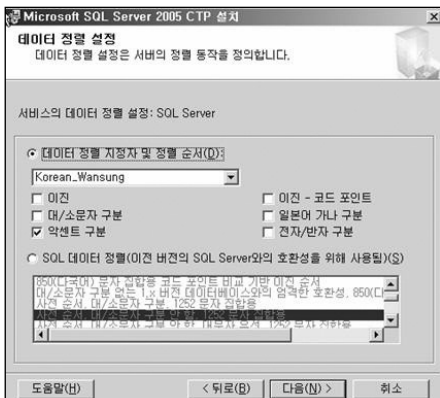


마이그레이션 1

- 마이그레이션하기에서는 서비스 계정 설정 부분이 먼저 나타납니다
- 서비스 계정 설정하기
 - SQL 서비스를 위한 계정을 따로 만들어 사용하는 것을 권장합니다
 - 업그레이드시에는 이 부분이 생략됩니다
- 각 서비스 계정에 대해 사용자 지정에 선택표시가 없음을 확인하십시오. 이는 동일한 계정이 모든 서비스에 사용됨을 의미합니다
- 서비스 계정을 windows 계정으로 미리 생성합니다



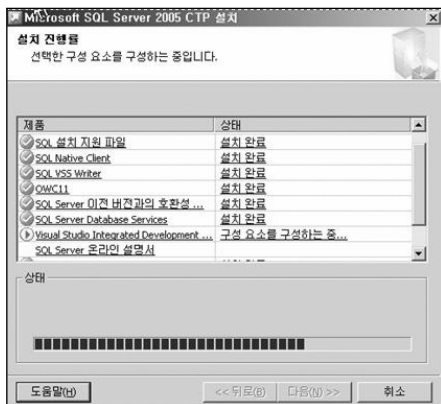
마이그레이션 2



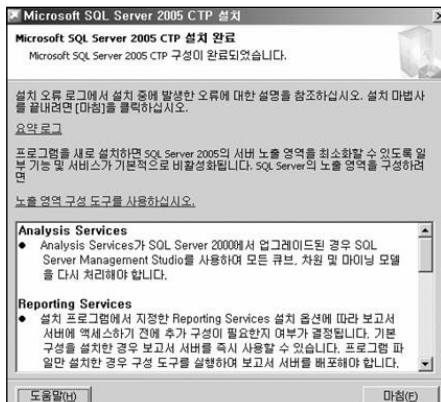
마이크로이전 3



마이크로이전 4



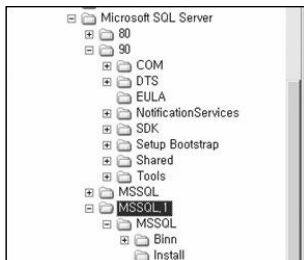
마이크로이전 5



마이크로이전 6

설치후 파일 위치(file locations)

- MSSQL,1 : 다른 서비스를 위한 다른 인스턴스 디렉토리입니다



셋업 로그파일 검사하기

- 향상된 에러 로깅과 실패 보고기능

- 요약 파일(Summary.txt)이 다음 위치에 생성됩니다
C:\Program Files\Microsoft SQL Server\90\Setup Bootstrap\LOG
- 확장 기능 경고들이 원인을 알려주고 올바른 방향을 알려줍니다



레지스트리에서 인스턴스 이름 찾아보기

■ HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\Instance Names\SQL

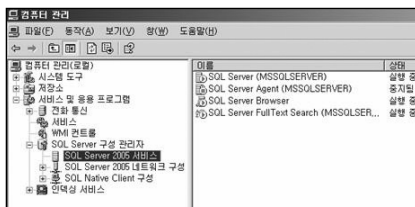


SQL Server 서비스 확인

컴퓨터 관리자 부분이 없어지고 컴퓨터 관리의 MMC부분이나 SQL Server 구성 관리자에서 확인 할 수 있습니다

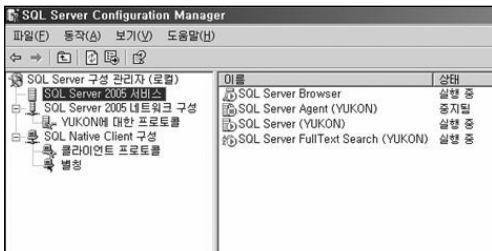
[따라하기]

1. 시작-관리도구-컴퓨터 관리를 선택합니다
2. 서비스 및 응용 프로그램을 선택합니다
3. SQL Server 구성 관리자에서 SQL Server 2005 서비스를 확인합니다
4. 업그레이드 한 경우의 화면입니다



[따라하기]

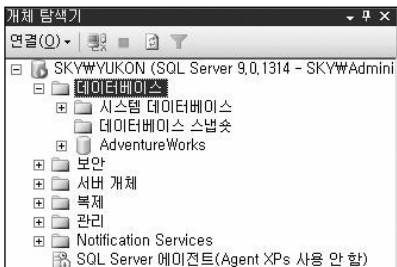
1. 시작-모든 프로그램-Microsoft SQL Server 2005 - 구성도구- SQL Server 구성 관리자를 선택합니다
2. 마이그레이션한 경우의 화면입니다



SQL Server 2005 마이그레이션시 인스턴스명

[따라하기]모든 프로그램-SQL Server 2005 - SQL Server Management Studio 선택합니다

- 1.서버명\인스턴스명으로 등록되어 있습니다



2. SQL Server 2000의 엔터프라이즈 관리자도 실행시킬 수 있습니다.
3. SQL Server Management Studio에 SQL Server 2000에서 기본값으로 등록되어있는 SQL Server 2000인스턴스를 등록할 수 도 있습니다.

[따라하기] 개체 탐색기 - 연결 - 데이터베이스 엔진 을 선택합니다 SQL Server 2000과 SQL Server 2005 인스턴스 두 개를 등록하여 보겠습니다

1. 연결을 위한 메뉴가 나타납니다



2. 서버 이름에서 기본값으로 사용되는 서버명을 선택합니다



3. 두 개의 인스턴스가 등록되었습니다



4. SQL Server 2000의 서버명 - 연결 끊기 를 클릭하면 사라집니다

SQL Server 2000에서 SQL Server 2005로 업그레이드시 인스턴스

- 마그레이션과 달리 서버명으로 등록되어 있습니다
- 사용자 데이터베이스인 northwind 와 pubs 가 업그레이드 되어 있습니다

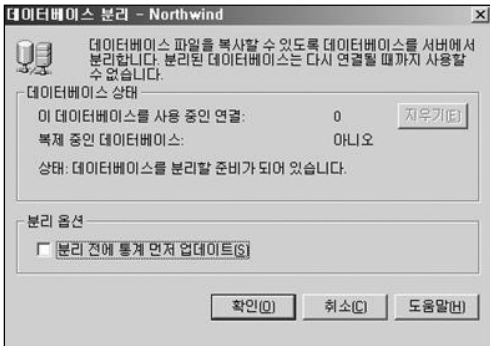


마이그레이션시 SQL Server 2000로부터 사용자 데이터베이스 가져오기

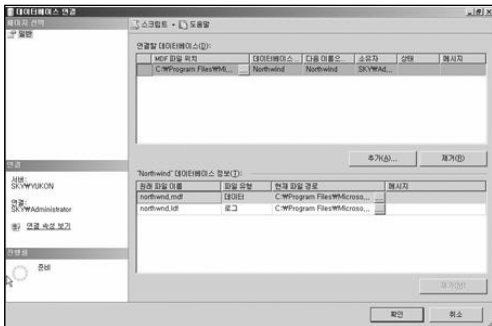
■ 여러 방법중 데이터베이스 분리과 연결 방법을 이용합니다.

[따라하기]

1. SQL Server 2000에서 **엔터프라이즈 관리자**를 실행합니다
만약의 경우를 대비하여 사용자 데이터베이스를 백업합니다
2. northwind를 분리합니다



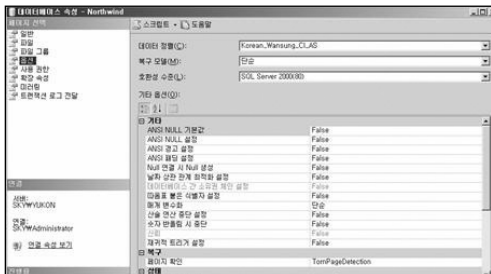
3. SQL Server 2005의 **SQL Server Management Studio - 데이터베이스 - 연결** 을 선택합니다
4. **데이터베이스 연결** 창에서 추가 버튼을 눌러 northwind의 데이터파일을 선택합니다
5. 확인을 누릅니다



6. 붙여진 northwind 데이터베이스의 속성을 알아봅니다



7. 특히 **옵션**에서 **호환성 수준**을 살펴봅니다 Northwind의 경우 **SQL Server 2000(80)**으로 되어 있습니다.



무인 설치 수행하기

필요한 셋업 정보를 가진 .ini 파일을 작성하고 명령 프롬프트에서 setup.exe를 실행하여 무인 설치를 할 수 있습니다.

기업에서 여러 개의 동일한 설치를 진행하거나 다른 엔지니어에게 설치를 위임할 경우 유용하게 사용할 수 있습니다

■ 확장자 ini 파일 만들기

- 메모장과 같은 에디터를 사용하여 만들 수 있습니다
- SQL Server CD 내의 template.ini 를 기반으로 여러분의 파일을 만듭니다
- .ini파일은 각기 다른 기능 또는 셋팅 구성을 담고 있는 여러 개의 매개변수들이 있습니다

■ 무인설치 시작

- 다음의 명령프롬프트를 이용하여 시작합니다

```
setup.exe /settings <path to .ini file>
```

- 예를 들어 C:\WSETUP 폴더인 경우

```
setup.exe Wsettings c:\wsetup\installsettings.ini
```

- /qn 스위치 : 모든 설치 대화 상자 및 오류 메시지가 표시되지 않습니다 오류 메시지를 비롯한 모든 설치 메시지가 설치 로그 파일에 기록됩니다 (silent installation)
- /qb 스위치 : 기본 설치 대화 상자를 표시할 수 있습니다. 오류 메시지도 표시됩니다. 진행상황이 표시되는 것입니다.

원격 서버 설치 수행하기

다른 위치에 있는 여러 대의 SQL서버 설치가 필요할 때 또는 설치하고자 하는 SQL서버에 대한 물리적 접근이 용이하지 않을 때 유용하게 이용될 수 있습니다

■ 원격 설치 요구사항

- 원격 설치는 .ini 파일을 이용하여 명령 프롬프트로 해야만 합니다
- ini파일은 다음의 매개변수를 포함하고 있어야 합니다

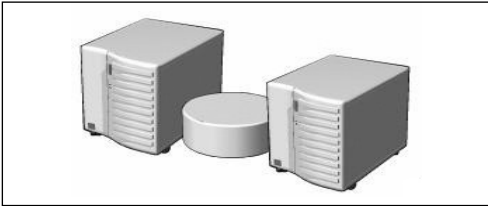
매개변수	설명
TARGETCOMPUTER	SQL 서버가 설치되어야 할 네트워크 이름
ADMINACCOUNT	원격 서버에 로그인 할 수 있는 어드민 권한의 계정
ADMINPASSWORD	원격 서버에 로그인 할 수 있는 어드민 권한의 계정의 암호

■ 원격 설치 는 도메인 환경에서만 지원됩니다

- 워크그룹 환경에 있는 원격 컴퓨터에 SQL 서버를 설치할 수 없습니다

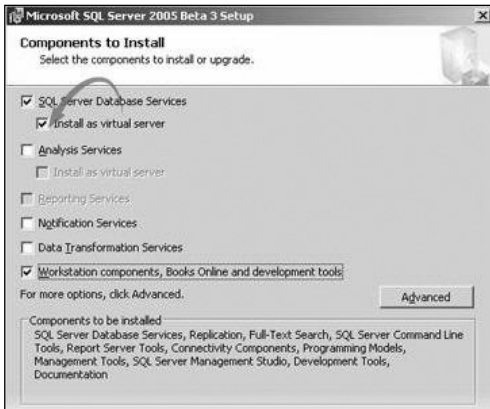
클러스터에 SQL 서버 설치하기

많은 기업이 SQL Server에 대한 고가용성 요구를 가지고 있습니다. 이를 위해 클러스터 서비스를 사용할 수 있습니다.



■ 먼저 Windows Server 2003 클러스터를 구성하고 SQL Server 설치를 위한 클러스터 그룹을 만듭니다

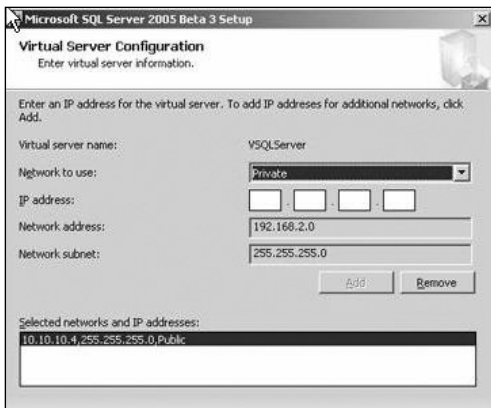
■ 구성요소 설치 페이지에서 가상서버(virtual server)옵션 설치를 선택합니다.



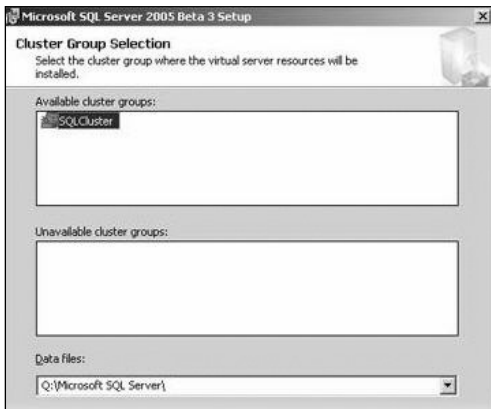
- 가상서버 이름을 기술합니다. 클라이언트가 클러스터드 SQL Server 인스턴스에 연결하는데 사용하는 넷바이오스 이름 (netbios name)입니다



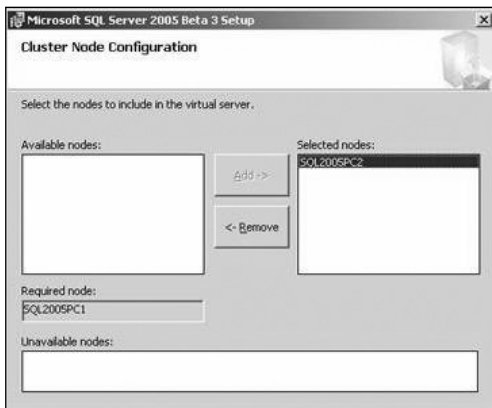
■ 가상서버 IP주소를 입력합니다



■ SQL Server 클러스터드 인스턴스에 대한 공유 자원을 클러스터 그룹에 할당해야 합니다



- 셋업 마법사에 SQL Server가 인스톨 되어야 할 클러스터 노드를 지정합니다.



- SQL Server 2005는 8개의 클러스터 노드까지 설치할 수 있습니다.
- 클러스터에서 다른 노드에 대한 관리자 권한 계정을 지정해야 합니다
 - 이 계정은 원격 서버 설치 수행에 사용되어진 것입니다.
- SQL Server 서비스에 대한 도메인 계정이 필요합니다. 이 계정은 클러스터의 모든 노드에서 유효해야만 합니다.
- 남은 페이지에서는 단독 서버 설치와 비슷합니다
- 셋업마법사를 끝마치고 실제적인 설치가 시작 되어 질 때 노드 드랍다운 목록에서 보고 원하는 노드를 선택하여 각각의 클러스터 노드의 진행상황을 모니터링 할 수 있습니다.
- 인스톨이 완성될 때 SQL Server는 윈도우에 인스톨되고 액티브 클러스터 노드에서 실패가 발생할 시 자동 장애극복(automatic failover)기능이 제공됩니다

SQL Server 2005 구성요소 추가 또는 삭제

- SQL Server 설치를 수정하고자 하는 절차들입니다
- **제어판의 프로그램 추가 변경 프로그램**을 클릭합니다
- 수정하고자 하는 SQL Server 인스턴스를 선택합니다
- **변경**을 클릭합니다
- SQL Server 설치 마법사 환경페이지에서 **다음** 을 선택합니다
- 시스템 구성 검사 페이지에서 **다음** 을 선택합니다
- 변경 또는 인스턴스 제거 페이지에서 **변경 인스톨 구성요소** 를 선택합니다

SQL Server 2005 삭제

SQL Server 2005를 컴퓨터에서 삭제할 때 제어판의 추가 삭제 프로그램을 사용하며 다음의 두 가지 방법 중 하나를 사용하면 됩니다

- 추가삭제 프로그램에서 삭제하고자 하는 SQL Server를 선택한 후 제거를 클릭합니다
- 추가삭제 프로그램에서 삭제하고자 하는 SQL Server를 선택한 후 변경을 클릭합니다. 변경제거 페이지가 나타나기 전까지 셋업프로그램에서 기본 옵션을 사용합니다.이 페이지에서 SQL Server 삭제를 클릭합니다

[참고]

SQL Server 2005 설치보다 Visual Studio 2005를 먼저 설치하는 경우 기본적으로 SQL Server Express Edition이 설치되어 다른 SQL Server 2005에디션이 설치되지 않습니다. 이런 경우 제어판에서 관련 프로그램을 제거한 후 설치하여야 합니다.

SQL Server 2005의 새로운 기능

SQL Server 2005 개요

SQL Server 2005는 데이터베이스 관리 그 이상의 시스템입니다. 기업 응용 프로그램을 위한 다양한 구성요소와 서비스들을 포괄하고 있습니다.

SQL Server 2005 구성요소(components)

구성요소	설명
관계형 데이터베이스 엔진 (Relational Database Engine) 	SQL Server 관계형 데이터베이스 엔진은 SQL Server 2005의 핵심으로 관계형 또는 XML형식의 데이터를 저장하고, 추출하고, 수정하는데 있어 탁월한 성능과 확장기능하고 안전한 환경을 제공하는 강력한 관계형 데이터베이스 엔진입니다.
분석 서비스 (SQL Server Analysis Services) 	온라인 분석 처리 어플리케이션(OLAP) 과 데이터 마이닝을 지원하는 강력한 비즈니스 인텔리전스 솔루션입니다.
통합서비스 (Integration Services, or SSIS) 	데이터를 가져오고 내보내는 솔루션으로서 데이터의 이동이 이루어질 때 변환과정을 수행합니다

알림서비스 (Notification Services) 	이벤트와 요청된 데이터에 근거하여 이메일, 텍스트 메시지 기타 다른 방식으로 알림(notifications)을 발생시킬 수 있습니다
리포팅 서비스 (Reporting Services) 	데이터 원본으로부터 데이터를 추출하여 보고서를 만들고 브라우저로 볼 수 있게 하거나 파일로 내보내거나 이메일로 보낼 수 있습니다
서비스 브로커 (Service Broker) 	소프트웨어 서비스간의 메시지기반 통신에 관한 서비스입니다
네이티브 http 서비스 (Native HTTP Service) 	Microsoft Windows Server™ 2003에 설치되어 있을시 SQL Server 2005는 HTTP (Hypertext Transfer Protocol)로 이루어진 요구에 응답할 수 있습니다. Native HTTP Service는 SQL Server 2005가 IIS (Microsoft Internet Information Services)없이도 웹서비스 인터페이스를 만들수 있도록 하여 줍니다.
SQL Server 에이전트 (SQL Server Agent) 	데이터베이스 유지 및 작업, 이벤트, 경고 관리를 자동화하도록 하는 예약 관리 업무 엔진입니다.

닷넷 CLR 기반 서비스 .NET Common Language Runtime 	CLR(Common Language Runtime)이 SQL Server 에 내재되어 있어서 데이터베이스 솔루션이 Microsoft Visual C#® .NET 또는 Microsoft Visual Basic® .NET 과 같은 언어에서 생성된 관리 코드 (managed code)를 이용 할 수 있도록 합니다.
복제 (Replication) 	한쪽 데이터베이스에서 다른 데이터베이스로 데이터 및 데이터베이스 객체들을 복사 이동시키고난 후 데이터 베이스간의 일관성이 동기화되도록 하여줍니다.
전체 텍스트 검색 (Full Text Search) 	SQL Server 데이터베이스에 있는 텍스트로 저장된 키워드 기반 쿼리에 대한 빠르고 유연한 인덱스 검사를 가능하게 하여 줍니다

데이터베이스 관리자를 위한 새로운 기능

■ 향상된 제품설치

SQL Server 2005 구성요소들은 이전에 발표된 SQL Server 보다 배치와 관리를 더 쉽게하도록 MSI(Microsoft Windows Installer)에 기반한 단일 설치 프로그램을 이용하여 설치됩니다. 설치과정은 시스템 호환성에 관한 중대한 정보와 설치에 영향을 미칠 수 있는 알려진 문제들에 관한 정보를 제공하는 엄격한 일관성 검사를 합니다. 이러한 향상은 데이터베이스 응용 프로그램을 안정적이고 보호된 환경에 설치될 수 있도록 합니다

■ 중앙 집중화된 관리 도구

SQL Server 2005는 SQL Server Management Studio 관리 도구를 가지고 있습니다. 이는 데이터베이스 서버, 분석 서비스, SQL Server CE 솔루션을 관리하는 집중화된 사용자 인터페이스입니다. 또한 버전 관리 지원을 합니다. 집중화된 이러한 관리는 솔루션에서 개발자와 DBA(database administrator)가 보다 더 협력할 수 있도록 합니다

■ 새로운 보안 아키텍처

SQL Server 2005의 보안분야는 크게 개선된 부분입니다. 새로운 계층적인 보안장치들은 시스템에 대한 security principals에 할당된 권한을 관리하는 것을 용이하게 하며 모든 권한들이 수여될 수 있습니다. 로그인에 암호 정책을 적용할 수 있습니다. 강화된 보안 기능은 데이터베이스 응용 프로그램의 보안을 한 층 더 강화할 수 있도록 합니다.

■ 고가용성 지원

클러스터링 설치에 대한 지원은 다중 인스턴스 지원기능을 향상시키며 현재 8노드 클러스터를 지원합니다. 또한 데이터베이스 미러링이라는 새로운 기술이 소개되었는데 이는 현재 주서버가 실패하였을 때 현재 운영중인 대기서버가 자동적으로 장애극복이 되도록 하는 것입니다. 이러한 기술들이 데이터베이스 응용 프로그램의 가용성 요구를 만족시키는데 사용될 수 있습니다

■ 시스템 향상

SQL Server 2005는 메모리 추가 후 서버를 재시작하지 않아도 되며 AWE(Address Windowing Extensions)지원, 32비트 한계를 넘는 물리적 메모리사용, SQL Server 서비스를 재 시작하지 않는 동적 CPU affinity 조정 능력을 가지고 있습니다. 이는 SQL Server 2005가 기업수준의 솔루션을 개발할 수 있는 고도의 확장 가능한 플랫폼임을 뜻합니다.

데이터베이스 개발자를 위한 새로운 기능

■ T-SQL 향상

새로운 데이터 형식, 구조화된 예외처리 방식 CTE 테이블, 관계형 연산자들, 순위함수들을 포함하여 많은 추가사항들이 생겼습니다.

■ Native XML 지원

SQL Server 2005는 XML을 데이터베이스 자체적으로 지원합니다. XML 데이터 형식을 지정하여 데이터를 저장하고 확인할 수 있으며 XML 형식으로 데이터를 쿼리할 수 있습니다.

■ 메시지 기반의 서비스

서비스 브로커 구성요소는 메시지 기반의 서비스 응용 프로그램 개발에 관한 플랫폼을 제공합니다. 큐에 옮겨진 메시지를 이용하여 비동기적으로 통신하는 서비스를 개발할 수 있습니다.

■ Native HTTP 서비스

SQL Server 2005에서는 HTTP 끝점을 생성할 수 있습니다. IIS에 대한 요구 없이 웹 서비스로서 데이터베이스 객체가 이용될때 이러한 끝점에 대한 HTTP요구를 감지할 수 있습니다. HTTP에 대한 지원은 SQL Server 기반 서비스를 이용하는 클라이언트 응용 프로그램의 다양한 요구를 소화해낼 수 있습니다.

■ 알림 서비스

알림 서비스는 특정한 이벤트를 지정한 구독자에게 알림을 보낼 수 있는 응용 프로그램을 만들 수 있습니다. 이는 알림 기반의 응용 프로그램 개발에 대한 강력한 플랫폼을 제공합니다.

■ CLR(Common Language Runtime) 통합

닷넷 CLR이 SQL Server 2005 데이터베이스 엔진에 통합되었습니다. 이는 Visual C# .NET 또는 Visual Basic .NET 같은 닷넷 언어로 쓰여진 관리코드(managed code)를 사용한 데이터 형식, 저장 프로시저, 함수, 트리거 같은 데이터베이스 객체들이 수행되는 것을 가능하게 합니다. CLR을 이용하면 T-SQL을 이용하여 수행하는 것이 어렵거나 불가능한 것을 가능하게 합니다.

■ 리포팅 서비스

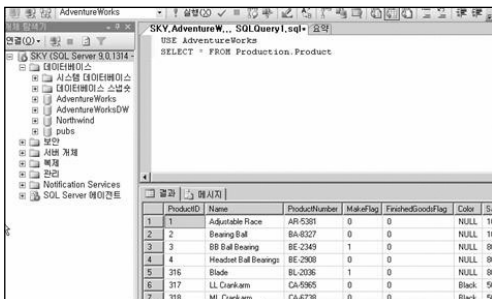
SQL Server 데이터베이스 또는 분석 서비스 큐브에 있는 데이터로부터 보고서를 개발하도록 합니다. 개발자들은 Microsoft Visual Studio® .NET 기반의 보고서 디자이너를 이용하여 보고서를 디자인할 수 있고 IIS 웹서비스에 출판할 수 있는 강력한 비즈니스 인텔리전스 응용 프로그램 플랫폼입니다.

새로운 관리도구 - SQL Server Management Studio

SQL Server 2000의 엔터프라이즈 관리자와 쿼리분석기를 통합한 형태로서 Visual Studio .Net 과 유사한 인터페이스를 가지고 있습니다.

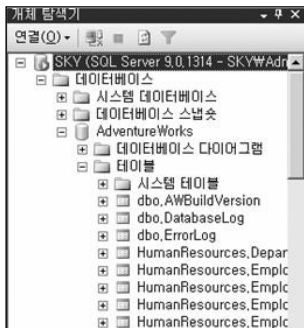
■ 어떻게 쿼리를 수행하는가

- 표준 도구모음에서 새쿼리를 클릭합니다
- SQLQuery1.sql에서 T-SQL 코드, USE AdventureWorks SELECT * FROM Production.Product를 적습니다
- 도구모음에서 표형태로서 결과 표시 아이콘을 클릭합니다
- 도구모음에서 실행을 클릭합니다.



■ 개체 탐색기는 어떻게 사용하는가

- 객체 탐색기 안에 있는 서버명을 확장합니다
- 시스템 데이터베이스와 사용자 정의 데이터베이스가 각각 있습니다.
- 데이터베이스 - AdventureWorks - 테이블 노드를 확장합니다



■ SQL Server 솔루션은 어떻게 생성하는가

- 파일- 새로 만들기 - 프로젝트를 클릭합니다
- 템플릿 리스트에서 **SQL Server 스크립트**를 선택하여 확인 버튼을 클릭합니다
- 솔루션 탐색기의 **연결 - 새로운 연결** 을 만들어 냅니다
- 쿼리 - 새쿼리 를 만들어 냅니다



새로운 기능을 사용할 수 있는 쿼리 찾아 변경하기

이런 새로운 기능들은 기능 구현 면에서는 편리하지만, 간혹 성능 면에서는 더 나빠질 수도 있으므로 사용하는 쿼리 패턴에 따라 비교해 보고 필요하다면 튜닝하고 사용하실 것을 권장합니다.

TOP(@n)구문

SQL Server 2000에서는 TOP문장에 변수를 쓸 수 없었습니다. 꼭 사용해야 한다면 동적 SQL문을 이용하는 것이었습니다. SQL Server 2005에서는 TOP구문에 변수를 사용함으로써 반드시 문자열을 만들어야 하는 불편함을 해소하였습니다

[문법]

```
TOP (expression) [PERCENT]
[WITH TIES]
```

[따라하기] SQL Server 2000 과 SQL Server 2005에서의 top 비교

```
1. SQL Server 2000에서의 top구문과 변수
DECLARE @cnt INT, @sql NVARCHAR(800)
SET @cnt=10
SET @sql='N'
SET @sql=@sql + 'N' select top ' + CONVERT(VARCHAR,@cnt) + '
ProductID,Quantity'
SET @sql=@sql + 'N' from Production.ProductInventory'
SET @sql=@sql + 'N' order by Quantity desc'
```

```
EXEC SP_EXECUTESQL @SQL
```

2. SQL Server 2005에서의 top구문과 변수

```
DECLARE @cnt INT
```

```
SET @cnt=10
```

```
SELECT TOP(@cnt) productID,Quantity
```

```
FROM Production.ProductInventory
```

```
ORDER BY quantity DESC
```

CTE(Common Table Expression) 테이블

CTE란 SELECT 쿼리에 기반한 명명된 임시적인 결과 셋입니다. 결과 셋은 INSERT, UPDATE, DELETE, SELECT에서 사용되어 질 수 있습니다.

이는 크게 단순 CTE 와 재귀CTE로 나누어 집니다. 단순CTE는 마치 파생테이블을 사용하는 개념과 유사하다 하겠습니다.

SQL Server 2000에서 재귀관계를 표현하기 위해 루프를 이용하던 것을 SQL Server 2005에서의 재귀CTE로 바꾸어 보도록 하겠습니다. 부하직원이 누구인지 알아보는 재귀관계입니다.

[문법]

```
[WITH <common_table_expression> [...n] ]
<common_table_expression> ::=
expression_name
[(column_name [...n])]
AS
(<CTE_query_expression>)
```

■ SQL 단순 CTE 생성 절차

- CTE 이름과 컬럼 리스트를 선택합니다
- CTE SELECT 쿼리를 만듭니다
- 쿼리에서 CTE를 사용합니다

[따라하기] SIMPLE CTE 사용하기

```
1. CTE 이름을 TopSales로 정합니다.  
2. 다음 쿼리에서 CTE를 테이블처럼 사용합니다  
WITH TopSales (SalesPersonID, NumSales) AS  
( SELECT SalesPersonID, Count(*)  
FROM Sales.SalesOrderHeader GROUP BY SalesPersonID )  
SELECT * FROM TopSales  
WHERE SalesPersonID IS NOT NULL  
ORDER BY NumSales DESC
```

■ 재귀CTE 생성절차

- CTE SELECT 쿼리안의 추가적인 고려사항을 제외하면 단순CTE와 비슷합니다
- 앵커 멤버 쿼리(ANCHOR MEMBER QUERY)를 생성합니다(트리에서 꼭대기 부분이 됩니다)
- union all 연산자를 사용합니다: 재귀CTE에서 두개의 결과 셋을 연결하기 위해 꼭 필요하며 유일하게 허용된 집합연산자입니다
- CTE를 스스로 참조하는 재귀 멤버 쿼리(Recursive Member Query)를 만듭니다

■ SQL Server 2000에서의 재귀관계

```

CREATE PROC emp_two
AS
DECLARE @empTemp TABLE(
    empid INT,
    mgrid INT,
    M INT
)

DECLARE @M INT
SET @M=0

INSERT @empTemp
SELECT employeeid,managerid,@M
FROM HumanResources.Employee
WHERE ManagerID IS NULL

WHILE @@ROWCOUNT >0
BEGIN
    SET @M=@M+1
    INSERT @empTemp
    SELECT e.employeeid,e.managerid,@M
    FROM HumanResources.Employee e join @empTemp t
    ON e.managerID = t.empid and t.M=@M-1
END

SELECT mgrid,empid FROM @empTemp
ORDER BY mgrid,empid
GO

```

SQL Server 2000결과셋 - 290개의 행

결과		메시지
	mgrid	empid
1	NULL	109
2	3	4
3	3	9
4	3	11
5	3	158
6	3	263
7	3	267
8	3	270

■ 재귀CTE를 이용한 SQL Server 2005에서의 재귀관계

```
CREATE PROC ctetest
AS
WITH managers AS
(
    SELECT ManagerID,employeeid
    FROM HumanResources.Employee
    WHERE ManagerID IS NULL
    UNION ALL
    SELECT e.ManagerID,e.EmployeeID
    FROM HumanResources.Employee e
        INNER JOIN Managers mgr
    ON e.Managerid=mgr.employeeid)
SELECT * FROM managers ORDER BY managerid,employeeid
GO
```

SQL Server 2005 결과셋 - 290개의 행



	ManagerID	employeeid
1	NULL	109
2	3	4
3	3	9
4	3	11
5	3	158

RANK() 함수

순위를 구하기 위해 SQL Server 2000에서 상관 서브 쿼리 방식 또는 함수를 이용하는 경우가 많았습니다. SQL Server 2005에서는 순위를 구하는 rank() 함수가 따로 만들어져 있습니다. rank() 함수의 경우 같은 값인 경우 동순위로 순위가 매겨집니다.

[문법]

```
RANK () OVER ( [ < partition_by_clause > ] < order_by_clause > )
```

[따라하기] SQL Server 2000 과 SQL Server 2005에서의 순위구하기 비교

```
1. SQL Server 2000에서 상관서브쿼리를 이용하는 경우
USE AdventureWorks;
GO
SELECT productid, quantity,
       (SELECT COUNT(quantity) + 1
        FROM Production.ProductInventory
```



```

        WHERE p.quantity < quantity) rank
FROM Production,ProductInventory p
ORDER BY rank

```

2. SQL Server 2000에서 함수를 이용하는 경우

```

CREATE FUNCTION dbo.uf_quan_rank(@qu INT)
RETURNS INT
AS
BEGIN
    RETURN(SELECT COUNT(quantity) +1
    FROM Production,ProductInventory p
    WHERE @qu < (p.quantity)
END

```

```

SELECT productid,quantity,dbo.uf_quan_rank(quantity) AS rank
FROM Production,ProductInventory
ORDER BY rank

```

3. SQL Server 2005에서 rank() 함수 사용하기

```

SELECT productid,quantity,RANK( ) OVER (ORDER BY quantity DESC) AS rank
FROM Production,ProductInventory
ORDER BY rank

```

SQL Server 2005에서 rank() 함수

```
SELECT productid, quantity,
       rank() OVER (ORDER BY quantity DESC) AS rank
FROM Production.ProductInventory
ORDER BY rank
```

	productid	quantity	rank
1	528	924	1
2	379	897	2
3	527	888	3
4	393	780	4
5	397	763	5
6	528	729	6
7	513	724	7
8	532	715	8
9	387	710	9

ROW_NUMBER() 함수

순위를 매기는 것이 아니라 단순히 데이터를 나열하여 원하는 데이터 개수만큼 가져와야 하는 경우가 많습니다. 이를 위해 임시 테이블에 identity속성을 이용하는 방법을 흔히 사용합니다. 이를 SQL Server 2005의 새로운 함수 row_number() 함수로 바꾸어 보고자 합니다. 이는 또한 페이징을 위한 좋은 방안중 하나가 될 수도 있습니다

[문법]

```
ROW_NUMBER() OVER ([ <partition_by_clause> ] <order_by_clause> )
```

[따라하기] SQL Server 2000 과 SQL Server 2005에서의 번호 나열하기 비교

1. SQL Server 2000에서 번호를 구하려는 경우

```
CREATE TABLE #temp
(id INT IDENTITY ,
SalesOrderID INT,
OrderDate DATETIME
)

INSERT INTO #temp
SELECT SalesOrderID, OrderDate
FROM Sales.SalesOrderHeader
ORDER BY OrderDate ASC

SELECT *
FROM #temp
WHERE id BETWEEN 50 AND 60
```

SQL Server 2000에서 번호 매기기

	id	SalesOrderID	OrderDate
1	50	43708	2001-07-03 00:00:00.000
2	51	43709	2001-07-03 00:00:00.000
3	52	43710	2001-07-03 00:00:00.000
4	53	43711	2001-07-04 00:00:00.000
5	54	43712	2001-07-04 00:00:00.000
6	55	43713	2001-07-05 00:00:00.000
7	56	43714	2001-07-05 00:00:00.000
8	57	43715	2001-07-05 00:00:00.000

2. SQL Server 2005에서 row_number() 함수를 이용하여 번호를 구하는 경우

```

SELECT *
FROM(

    SELECT SalesOrderID, OrderDate,
    ROW_NUMBER() OVER (ORDER BY OrderDate)AS RowNumber
    FROM Sales.SalesOrderHeader
)a
WHERE a.RowNumber between 50 and 60

```

3. SQL Server 2005에서 row_number() 함수 와 CTE를 이용하여 번호를 구하는 경우

```

WITH OrderedOrders AS
(SELECT SalesOrderID, OrderDate,
ROW_NUMBER() OVER (order by OrderDate)as RowNumber
FROM Sales.SalesOrderHeader )
SELECT *
FROM OrderedOrders
WHERE RowNumber between 50 and 60;

```

SQL Server 2005에서 rownumber() 함수

결과		메시지	
	SalesOrderID	OrderDate	RowNumber
1	43708	2001-07-03 00:00:00.000	50
2	43709	2001-07-03 00:00:00.000	51
3	43710	2001-07-03 00:00:00.000	52
4	43711	2001-07-04 00:00:00.000	53
5	43712	2001-07-04 00:00:00.000	54
6	43713	2001-07-05 00:00:00.000	55
7	43714	2001-07-05 00:00:00.000	56
8	43715	2001-07-05 00:00:00.000	57

PIVOT연산자

SQL Server 2000에서는 행 별로 표현된 값을 컬럼 이름으로 가져가기 위해서 CASE문을 써야만 하였습니다. 그러던 것을 SQL Server 2005에서는 PIVOT연산자를 이용하여 간단하게 구현할 수 있습니다

[문법]

```
SELECT * FROM table_source
PIVOT ( aggregate_function ( value_column )
FOR pivot_column
IN ( <column_list> )
) table_alias
```

■ SQL Server 2000에서의 데이터 통계 구하기

1. 쿼리

```
SELECT DATEPART(yy,ModifiedDate)as 년도,
       SUM(CASE DATEPART(qq,ModifiedDate) WHEN 1 THEN 1 ELSE 0
END)AS [1사분기],
       SUM(CASE DATEPART(qq,ModifiedDate) WHEN 2 THEN 1 ELSE 0
END)AS [2사분기],
       SUM(CASE DATEPART(qq,ModifiedDate) WHEN 3 THEN 1 ELSE 0
END)AS [3사분기],
       SUM(CASE DATEPART(qq,ModifiedDate) WHEN 4 THEN 1 ELSE 0
END)AS [4사분기]
FROM Sales.SalesOrderDetail
GROUP BY DATEPART(yy,ModifiedDate)
ORDER BY DATEPART(yy,ModifiedDate)
```

2. 결과값

	결과	메시지				
	년도	1사분...	2사분...	3사분...	4사분기	
1	2001	0	0	2178	2973	
2	2002	2454	3060	7604	6235	
3	2003	4650	6403	18838	21346	
4	2004	19563	23804	2209	0	

■ SQL Server 2005에서 PIVOT를 이용한 데이터 통계

1. 쿼리

```

WITH quater AS
(SELECT DATEPART(yy,ModifiedDate) AS YEAR ,
    DATEPART(qq,ModifiedDate) AS qua
FROM Sales.SalesOrderDetail)
SELECT *
FROM quater
PIVOT(COUNT(qua) FOR qua IN([1],[2],[3],[4]))pvt

```

2. 결과값

	결과	메시지				
	YEAR	1	2	3	4	
1	2001	0	0	2178	2973	
2	2002	2454	3060	7604	6235	
3	2003	4650	6403	18838	21346	
4	2004	19563	23804	2209	0	

TRY..CATCH 구문

SQL Server 2005는 구조화된 예외처리 방법을 가지게 되었습니다. 이는 많은 T-SQL문에서 중요한 요구사항이었습니다. 특히 트랜잭션구문에서 그러했습니다. 많은 다른 언어에서 사용하는 예외처리와 유사한 방식입니다

SQL Server 2000에서 오류 확인은 @@error 시스템함수를 많이 이용하였습니다. 이를 SQL Server 2005에서의 TRY..CATCH구문으로 바꾸어 처리하여 봅시다

■ SQL Server 2000구문에서의 에러 처리 방식

- ta테이블을 만듭니다
- 자식 프로시저 proc2를 생성합니다 수정문 과 삽입문으로 이루어져 있습니다. 삽입문은 기본키 제약조건에 위배되므로 에러가 발생합니다
- 부모 프로시저 proc1을 생성합니다. 하나의 트랜잭션 내에서 테이블ta로 삽입하는 문장과 프로시저 proc2를 호출합니다. proc2에서 기본키 위반 오류가 발생을 하여도 이를 호출한 부모 프로시저로 오류 발생 사실이 전달 되지 않아 트랜잭션 전체가 롤백되지 못하고 오류 처리 문장이 실행되지 않습니다
- pubs데이터베이스의 titles테이블 내용

결과		메시지	
	title_id	title	
1	BU1032	The Busy Executive's Database Guide	
2	BU1111	Cooking with Computers: Surreptitious Balance Sh...	
3	BU2075	You Can Combat Computer Stress!	

1. 프로시저를 생성합니다

```

USE pubs
GO
CREATE TABLE dbo.ta(id INT)
GO

CREATE PROC dbo.proc2

AS
UPDATE titles
SET title=' 이걸로 수정됨'
WHERE title_id=' bu1111'

INSERT INTO titles(title_id,title) VALUES( 'bu1032' , ' happyy' )
IF @@error <> 0
BEGIN
    RETURN
END
GO

CREATE PROC dbo.proc1
AS
DECLARE @err INT
BEGIN TRAN

    INSERT INTO dbo.ta VALUES(1)
    EXEC proc2

    SELECT @err = @@error
    PRINT 'proc1 error code= ' + cast(@err AS VARCHAR(10))

```



```

IF(@err <> 0)
BEGIN
    PRINT 'proc1 오류처리 롤백'
    ROLLBACK TRAN
    RETURN
END
COMMIT TRAN

GO

EXEC proc1 --프로시저 생성

```

[참고 오류메시지]

(1개 행 적용됨)

(1개 행 적용됨)

메시지 2627, 수준 14, 상태 1, 프로시저 proc2, 줄 7

PRIMARY KEY 제약 조건 'UPKCL_titleidind' 을(를) 위반했습니다. 개체 'dbo,titles' 에
중복 키를 삽입할 수 없습니다.

문이 종료되었습니다.

proc1 error code= 0

2.데이터를 확인하여 봅니다

```

select * from dbo.ta

select * from titles
where title_id='bu1111'

```

결과	
id	1

	title_id	title	type	pub_id	price	advan...	royalty	ytd_sales
1	BU1111	이걸로 수정될	business	1389	11.95	5000.00	10	3876

3.proc2프로시저에서 raiserror()문을 사용하기

이렇게 하면 부모프로시저로 오류가 전달되어 부모 프로시저에서 오류 확인 처리를 하게 됩니다. SQL Server 2000에서는 이와 같은 에러 처리 확인 방식을 사용했습니다.

```

ALTER PROC dbo.proc2
AS
UPDATE titles
SET title=' 이걸로 수정안됨 '
WHERE title_id=' bu1111'

IF @@ERROR(>)0
BEGIN
    RAISERROR( 'proc2 update에서 에러 발생' ,16,1)
    RETURN
END

INSERT INTO titles(title_id,title) VALUES( 'bu1032' , ' happyy' )
IF @@ERROR (>)0
BEGIN

```

```
RAISERROR( 'proc2 insert 에서 에러 발생' ,16,1)
RETURN
END
GO
```

[참고 오류메시지]

1개 행 적용됨)

(1개 행 적용됨)

메시지 2627, 수준 14, 상태 1, 프로시저 proc2, 줄 13

PRIMARY KEY 제약 조건 'UPKCL_titleidind' 을(를) 위반했습니다. 개체 'dbo.titles' 에
중복 키를 삽입할 수 없습니다.

문이 종료되었습니다.

메시지 50000, 수준 16, 상태 1, 프로시저 proc2, 줄 16

proc2 insert 에서 에러 발생

proc1 error code= 50000

proc1 오류처리 롤백

■ SQL Server 2005의 TRY..CATCH구문

[문법]

```
BEGIN TRY
    {sql_statement | statement_block}
END TRY
BEGIN CATCH
    {sql_statement | statement_block}
END CATCH
```

1. SQL Server 2005의 TRY..CATCH구문을 이용한 오류코드 처리 예제

```
USE pubs
GO

CREATE PROC dbo.procb
AS
UPDATE titles
SET title=' happy'
WHERE title_id=' bu1111'

INSERT INTO titles(title_id,title) VALUES( 'bu1032' , ' happy' )
IF @@ERROR (>)0
BEGIN
    RETURN
END
GO

CREATE TABLE dbo.tb(id INT)
GO
```

```

CREATE PROC dbo.proca

AS

BEGIN TRY

    BEGIN TRAN

        INSERT INTO dbo.tb VALUES(1)
        EXEC procb

    COMMIT TRAN

END TRY

BEGIN CATCH

    DECLARE @error INT
    SELECT @error = @@error
    ROLLBACK TRAN
    PRINT 'proca error code= ' + CAST(@error AS VARCHAR(10))

END CATCH

GO

EXEC proca

```

[참고 오류메시지]

(1개 행 적용됨)

(1개 행 적용됨)

proca error code= 2627

2.데이터를 확인하여 봅니다.

원하는 대로 오류확인을 하였고 트랜잭션 전체가 롤백 되었습니다



The screenshot shows a SQL query window with the following text:

```
select * from dbo.tb
select * from titles
where title_id='bu1111'
```

Below the query window, there are tabs for '결과' (Results) and '메시지' (Messages). The '결과' tab is active, displaying a table with the following data:

	title_id	title	type	pub_id	price	advan...	royalty
1	BU1111	미결로 수정됨	business	1383	11.95	5000.00	10

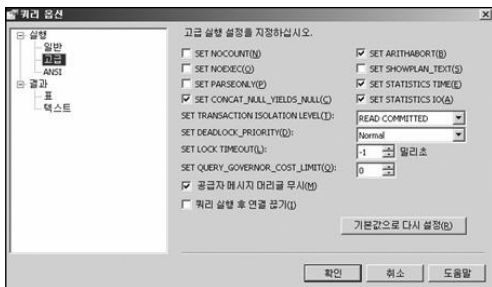
업그레이드 후 성능 자료 수집

업그레이드 이후 windows의 성능 카운터를 통해 전체적인 성능향상 자료를 수집할 수 있을 것입니다. 또한 SQL Server 2005를 이용하여 개개의 쿼리를 수정한 이후 실행계획, 수행시간, IO(Input Output)시간 등의 자료를 수집할 수 있을 것입니다. 쿼리에 대한 분석은 sql server management studio 또는 프로필러를 이용할 수 있습니다

SQL Server management에서의 성능 자료 수집

SQL Server 2000에서의 northwind 데이터베이스를 SQL Server2005로 가져옵니다. 이후 ORDERS 테이블에서 801에서 805번째까지의 데이터를 가져오는 페이징 처리 방식에 대한 SQL Server 2000 과 SQL Server 2005 성능 데이터를 수집하여 봅니다.

[따라하기] sql management studio의 쿼리 - 쿼리옵션 - 고급 을 차례대로 누릅니다 SET STATISTICS TIME, SET STATISTICS IO 항목에 체크를 한 후 쿼리를 실행하여 성능 데이터를 수집합니다. 테스트가 끝난 후에는 체크를 해제하여야 합니다. order 테이블의 논리적 읽기 수에서 2배정도의 차이를 보이고 있습니다.



[따라하기] SQL Server 2000에서의 NOT IN 방식을 이용한 페이징 처리

1. 쿼리

```
SELECT TOP 5 customerid,orderid,employeeid  
FROM orders  
WHERE orderid not in
```

```
(SELECT TOP 800 orderid  
FROM orders
```

```

ORDER BY customerid
DESC,orderid DESC
)
ORDER BY customerid DESC,orderid DESC

```

2.결과값

결과		메시지	
	customerid	orderid	
1	AROUT	11016	
2	AROUT	10953	
3	AROUT	10920	
4	AROUT	10864	
5	AROUT	10793	

3.성능 데이터 자료 수집

[참고]

(5개 행 적용됨)

테이블 'Worktable'. 검색 수 0, 논리적 읽기 수 0, 물리적 읽기 수 0, 미리 읽기 수 0, LOB 논리적 읽기 수 0, LOB 물리적 읽기 수 0, LOB 미리 읽기 수 0.

테이블 'Orders'. 검색 수 2, 논리적 읽기 수 10, 물리적 읽기 수 0, 미리 읽기 수 0, LOB 논리적 읽기 수 0, LOB 물리적 읽기 수 0, LOB 미리 읽기 수 0.

SQL Server 실행 시간:

CPU 시간 = 0ms, 경과 시간 = 7ms.

SQL Server 구문 분석 및 컴파일 시간:

CPU 시간 = 0ms, 경과 시간 = 1ms.

[따라하기]SQL Server 2005에서 ROW_NUMBER()를 이용한 페이징 처리

1.쿼리

```
SELECT * FROM  
(  
    SELECT TOP 805 customerid,orderid,  
  
    ROW_NUMBER( ) OVER (ORDER BY customerid DESC ,orderid DESC)AS  
    RowNumber  
    FROM orders  
    ) a  
WHERE a,rownumber BETWEEN 801 AND 805
```

2.결과값

결과		메시지	
	customerid	orderid	RowNumber
1	AROUT	11016	801
2	AROUT	10953	802
3	AROUT	10920	803
4	AROUT	10864	804
5	AROUT	10793	805

3. 성능데이터 자료 수집

[참고]

(5개 행 적용됨)

테이블 'Orders'. 검색 수 1, 논리적 읽기 수 5, 물리적 읽기 수 0, 미리 읽기 수 0, LOB 논리적 읽기 수 0, LOB 물리적 읽기 수 0, LOB 미리 읽기 수 0.

SQL Server 실행 시간:

CPU 시간 = 0ms, 경과 시간 = 2ms.

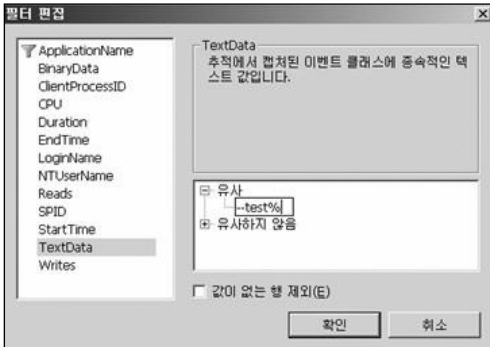
SQL Server 구문 분석 및 컴파일 시간:

CPU 시간 = 0ms, 경과 시간 = 1ms.

3. 열구성 버튼 - 원하는 컬럼 값을 보기 필요한 순서대로 합니다



4. 열필터링 - 특정 단어로 시작하도록 유사항목에 조건검색을 적용합니다.



5. 실행하여 결과를 비교합니다 - cpu, reads, duration 비교하여 봅시다

SQL Server 프로파일러 - [제목 없음 - 1 (SKYWYUKON)]						
파일(F) 편집(E) 보기(V) 재생(R) 도구(T) 창(W) 도움말(H)						
EventClass	TextData	CPU	Reads	Duration	Writes	
Trace Start						
SQL:BatchCompleted	--test 2000 SELECT TOP 5 customeri...	10	10	5	0	
SQL:BatchCompleted	--test 2000 SELECT TOP 5 customeri...	10	10	8	0	
SQL:BatchCompleted	--test 2000 SELECT TOP 5 customeri...	10	10	5	0	
SQL:BatchCompleted	--test yukon SELECT * FROM (S...	10	13	7	0	
SQL:BatchCompleted	--test yukon SELECT * FROM (S...	0	5	1	0	
SQL:BatchCompleted	--test yukon SELECT * FROM (S...	0	5	2	0	

비매품

SQL Server 2005 업그레이드 가이드

- 저 자: 김 민 정 (쥬필라넷)
- 감 수: 정 원 혁
- Contents 관련문의: mjkim@feelanet.com

- 발 행: 한국마이크로소프트(유)
- 제 작: (쥬필라넷)
- 초판 발행일: 2006년 4월 1일

본 책에 실린 글과 그림, 사진 및 프로그래밍 코드등의 저작권 및 배포권은 (쥬필라넷과 한국마이크로소프트(유)에 있으며, 저작권자의 동의 없이는 사용할 수 없습니다.

Microsoft
SQL Server 2005
업그레이드 가이드

Microsoft®

한국마이크로소프트(유)

서울특별시 강남구 대치동 892번지 포스코센터 서관 5층
고객지원센터 : 1577-9700
인터넷 : <http://www.microsoft.com/korea/sql>