

Chapter 03

데이터베이스 시스템

01 데이터베이스 시스템의 정의

02 데이터베이스의 구조

03 데이터베이스 사용자

04 데이터 언어

05 데이터베이스 관리 시스템의 구성

요약

연습문제

Preview



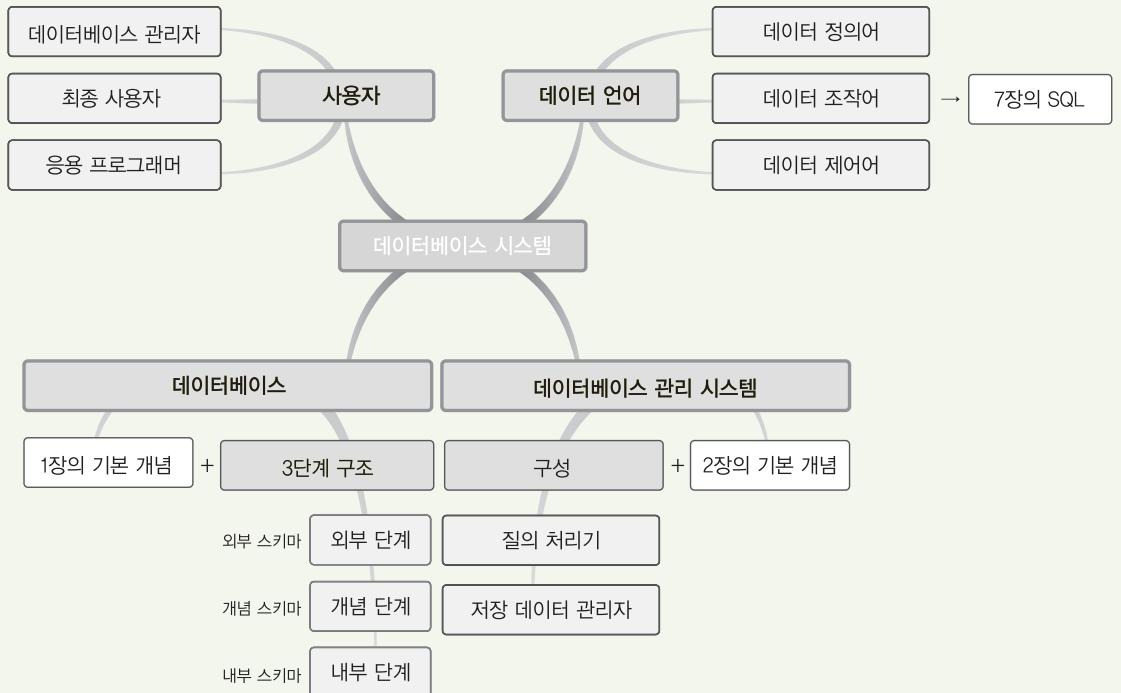
모든 조각을 제자리에 맞춰야 그림을 완성
할 수 있는 퍼즐을 해 본 적이 있는가? 앞
서 공부한 데이터베이스와 데이터베이스 관리

시스템은 퍼즐 조각과 같다. 이들을 제자리에 맞춰야 하나의 데이터베이스 시스템을 완성할
수 있다.

그리고 더 큰 데이터베이스 전문가 퍼즐을 완성하려면 퍼즐 조각이 몇 개 더 필요하다. 데이터베이스 관련 기술을 공부하는 이유는 보통 데이터 관리가 필요한 조직의 요구 사항에 딱 들어맞는 데이터베이스 시스템을 개발하거나, 이미 개발된 데이터베이스 시스템을 잘 활용하기 위해서이다. 그러므로 데이터베이스 시스템의 개념과 구성이 데이터베이스 전문가 퍼즐을 위한 조각이 될 수 있다.

이 장에서는 데이터베이스 전문가 퍼즐을 완성하기 위한 추가 퍼즐 조각을 몇 개 더 소개한다.
각 퍼즐에 숨겨진 놀라운 기능에 맞게, 퍼즐 조각을 제자리에 잘 맞춰 데이터베이스 전문가가
되기 위한 또 한걸음을 내딛어 보자.

| 학 | 습 | 목 | 표 |



- ▶ 데이터베이스, 데이터베이스 관리 시스템, 데이터베이스 시스템의 차이를 이해한다.
- ▶ 데이터베이스 시스템의 구성 요소를 살펴본다.
- ▶ 데이터베이스 3단계 구조에서 데이터 독립성의 개념을 실현하는 방법을 이해한다.
- ▶ 데이터 언어별 특징을 알아본다.
- ▶ 데이터베이스 사용자별 특징을 알아본다.
- ▶ 데이터베이스 관리 시스템의 구성을 알아본다.

01

데이터베이스 시스템의 정의

데이터베이스 시스템 DataBase System은 데이터베이스에 데이터를 저장하고, 저장된 데이터를 관리하여 조직에 필요한 정보를 생성해주는 시스템이다. 따라서 다양한 목적의 정보 처리 시스템을 구축하는 데 필요한 핵심 요소다.

흔히 데이터베이스, 데이터베이스 관리 시스템, 데이터베이스 시스템을 같은 의미로 사용하지만 각 용어의 개념을 구분하고 관계를 명확히 이해할 필요가 있다. 데이터를 저장해두는 곳, 다시 말해 저장된 데이터의 집합이 1장에서 배운 데이터베이스다. 그리고 2장에서 배운 데이터베이스 관리 시스템은 데이터베이스에 저장되어 있는 데이터가 일관되고 무결한 상태로 유지되도록 관리하는 역할을 한다. 그리고 이 장에서 알아보는 데이터베이스 시스템은 데이터베이스와 데이터베이스 관리 시스템을 이용해 조직에 필요한 정보를 제공해주는 전체 시스템이다. 데이터베이스와 데이터베이스 관리 시스템은 데이터베이스 시스템의 핵심 구성 요소다. 데이터베이스 시스템 내에서 데이터베이스와 데이터베이스 관리 시스템이 중요한 역할을 하여 일반적으로 이 두 용어를 더 자주 사용하는 것이다.

데이터베이스 시스템은 [그림 3-1]과 같이 데이터베이스와 데이터베이스 관리 시스템 외에도 사용자, 사용자가 데이터베이스에 접근하는 데 사용하는 데이터 언어, 데이터베이스와 데이터베이스 관리 시스템을 설치하고 데이터 처리 연산을 담당하는 컴퓨터로 구성된다.

각 구성요소에 대해서는 3절부터 하나씩 살펴보기로 하고, 우선 다음 절에서는 데이터베이스를 이해하기 위한 가장 기본이 되는 데이터베이스의 구조부터 살펴보자.

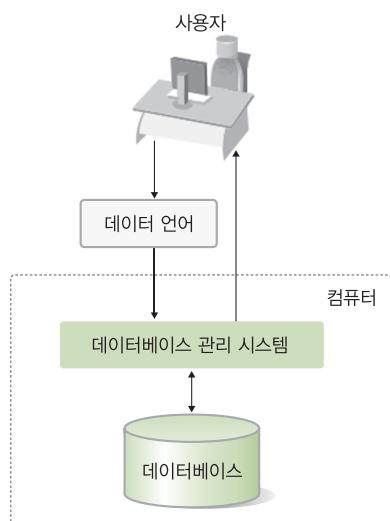


그림 3-1 데이터베이스 시스템의 구성

02

데이터베이스의 구조

이 절에서는 데이터베이스의 구조를 이해하기 위해 필요한 개념인 스키마를 알아본 후, 데이터베이스를 3단계로 나누어 구조를 살펴본다. 그리고 3단계 데이터베이스 구조에서 데이터 독립성의 개념을 실현하는 방법을 알아본다.

2.1 스키마

스키마 schema는 데이터베이스에 저장되는 데이터 구조와 제약조건을 정의한 것이다. [그림 3-2]는 스키마를 그림으로 간략히 표현한 것이다. 고객과 관련된 데이터인 고객번호, 이름, 나이, 주소를 저장한다고 가정했다. 고객번호는 정수로, 이름은 최대 10자의 문자열로, 나이는 정수로, 주소는 최대 20자의 문자열만 허용하기로 했다면 정해진 이 모든 내용이 스키마다. 그리고 정의된 스키마에 따라 데이터베이스에 실제로 저장된 값이 인스턴스 instance다. 보통 스키마는 한번 정의되면 자주 변경되지 않지만, 인스턴스는 계속 변하는 특성이 있다. 이는 한번 지어진 집의 구조는 잘 변하지 않지만 이사 등을 통해 사는 사람들이 계속 바뀌는 것과 같다.

고객	고객번호 INT	이름 CHAR(10)	나이 INT	주소 CHAR(20)
----	-------------	----------------	-----------	----------------

그림 3-2 스키마의 예

2.2 3단계 데이터베이스 구조

데이터를 데이터베이스에 저장하고 관리하기는 쉽지 않은 일이지만, 데이터베이스의 여러 장점 때문에 많은 사용자가 데이터베이스를 이용한다. 그런데 데이터베이스의 복잡한 내부 구조와 동작 방식을 모두 이해하고 있어야 한다면 일반 사용자에게 데이터베이스는 그림의 떡일 수

밖에 없다. 미국의 표준화 기관인 ANSI/SPARC에서는 데이터베이스의 복잡한 내부 구조를 감추고 일반 사용자가 데이터베이스를 쉽게 이해하고 이용할 수 있도록 3단계 데이터베이스 구조^{3-level database architecture}를 제안하였다.

2.2.1 3단계 데이터베이스 구조의 개념

3단계 데이터베이스 구조는 하나의 데이터베이스를 세 단계로 나누어 이해한다. 즉, 개별 사용자 관점에서 바라보는 외부 단계^{external level}, 조직 전체의 관점에서 바라보는 개념 단계^{conceptual level}, 물리적인 저장 장치의 관점에서 바라보는 내부 단계^{internal level}로 나눈다. 데이터베이스 하나를 세 단계로 나누고, 각 단계별로 다른 추상화^{abstraction}를 제공하면 데이터베이스를 효과적으로 관리할 수 있다. 일반적으로 내부 단계에서 외부 단계로 갈수록 추상화 레벨이 높아진다. 3단계 데이터베이스 구조를 통해 모든 데이터가 어떻게 저장되고 유지되는지와 관련된 복잡한 내용을 숨기고 필요한 데이터만 단순화한 외부 단계의 관점을 일반 사용자들에게 제공할 수 있다.

3단계 데이터베이스 구조의 개념을 아파트의 예를 통해 좀 더 이해해보자.

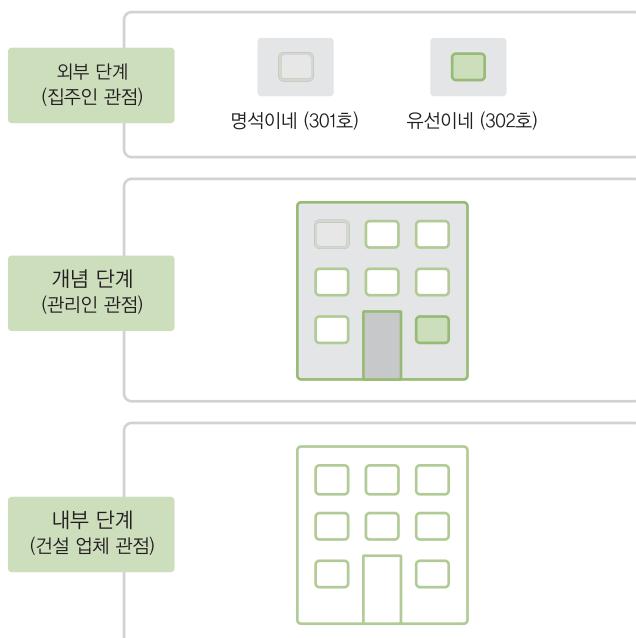


그림 3-3 3단계 데이터베이스 구조의 개념

[그림 3-3]의 아파트에는 다수의 세대가 살고 있다. 각 세대의 주민들은 자신의 집에만 관심을 가지면 되지 굳이 다른 집까지 알 필요가 없다. 유선이가 302호의 구조나 크기, 인테리어 등에 관심을 가질 뿐 명석이네인 301호나 다른 집이 어떤지 관심이 없는 것과 같다. 이렇게 유선이나 명석이 관점에서 아파트를 바라보는 것이 외부 단계다. 이와 달리 아파트 관리인은 어느 한 집에만 관심을 가지면 안 된다. 아파트 전체를 문제없이 관리하려면 아파트 전체를 잘 알고 있어야 하는데, 이처럼 관리인 관점에서 전체 아파트를 바라보는 것이 개념 단계다. 하지만 아파트 관리인도 아파트 뼈대, 즉 철근 콘크리트 구조가 어떻게 생겼는지, 시멘트를 얼마나 사용했는지 등을 잘 모른다. 이것은 아파트 건설 업체의 관심사다. 아파트를 건설한 업체 관점에서 전체 아파트를 바라보는 것이 바로 내부 단계가 된다.

데이터베이스를 서로 다른 추상화 레벨에서 바라보는 3단계 데이터베이스 구조를 단계별로 좀 더 자세히 알아보자.

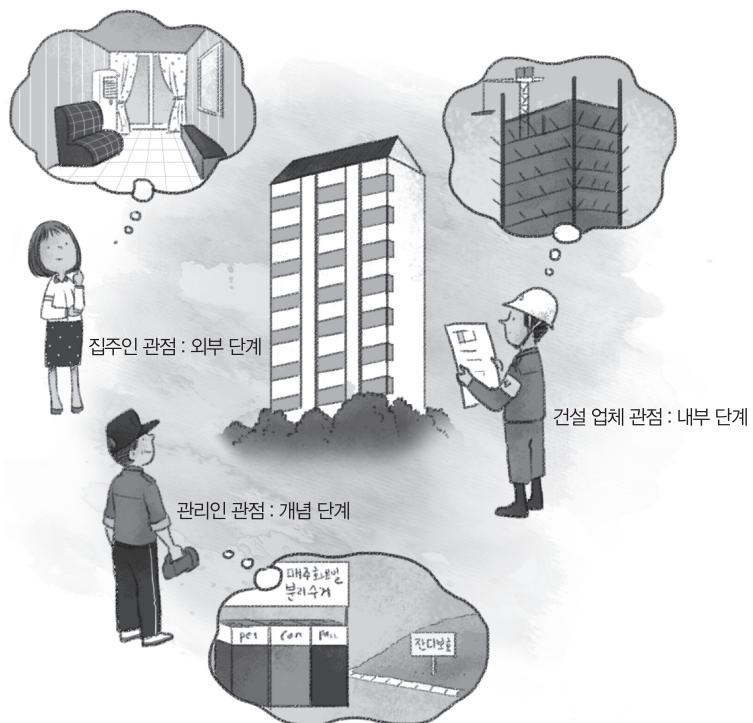


그림 3-4 3단계 데이터베이스 구조의 이해

2.2.2 외부 단계

외부 단계^{external level}에서는 개별 사용자 관점에서 데이터베이스를 이해하고 표현한다. 하나의 데이터베이스를 조직 내의 사용자들이 함께 사용하지만 각 사용자가 데이터베이스 전체에 관심이 있는 것은 아니다. 사용자마다 업무 내용과 사용 목적이 달라 필요한 데이터 내용이 다를 수 있다. 예를 들어 쇼핑몰을 운영하기 위해 필요한 데이터베이스에는 고객, 상품, 직원, 주문 등과 관련한 데이터들이 존재할 것이다. 그러면 사용자 중에서 고객 관리를 담당하는 직원은 데이터베이스에서 고객과 관련된 데이터에만, 상품 관리를 담당하는 직원은 상품과 관련된 데이터에만 관심을 가질 것이다.

외부 단계에서는 개별 사용자가 데이터베이스를 어떻게 보는지를 표현하므로 사용자마다 생각하는 데이터베이스의 구조가 다르다. 이처럼 외부 단계에서 사용자에게 필요한 데이터베이스를 정의한 것을 외부 스키마^{external schema}라 한다. 외부 스키마는 각 사용자가 생각하는 데이터베이스의 모습인 논리적인 구조로, 사용자마다 다르다.

데이터베이스 하나에는 외부 스키마가 여러 개 존재할 수 있고, 외부 스키마 하나를 사용 목적이나 같은 사용자들이 공유할 수 있다. 외부 스키마는 전체 데이터베이스 중 사용자가 관심을 가지는 일부분으로 볼 수 있어 서브 스키마^{sub schema}라고도 한다.

2.2.3 개념 단계

개념 단계^{conceptual level}에서는 데이터베이스를 이용하는 사용자들의 관점을 통합하여, 데이터베이스를 조직 전체의 관점에서 이해하고 표현한다. 데이터베이스 관리 시스템이나 데이터베이스 관리자는 데이터베이스의 일부분이 아닌 전체 데이터베이스에 관심을 가지는데, 개념 단계에서는 데이터베이스 관리 시스템이나 관리자의 관점에서 모든 사용자에게 필요한 데이터를 통합한 전체 데이터베이스의 논리적 구조를 정의한다. 그리고 이를 개념 스키마^{conceptual schema}라 한다.

개념 스키마는 조직 전체의 관점에서 생각하는 데이터베이스의 모습이며, 모든 사용자가 생각하는 데이터베이스의 모습을 하나로 합친 모습이다. 개념 스키마는 전체 데이터베이스에 어떤 데이터가 저장되는지, 데이터들 간에는 어떤 관계가 존재하고 어떤 제약조건이 존재하는지에 대한 정의뿐만 아니라, 데이터에 대한 보안 정책이나 접근 권한에 대한 정의도 포함한다. 하지만 데이터를 물리적으로 저장하는 방법이나 데이터를 저장하는 저장 장치와는 독립적이다.

데이터베이스 하나에는 개념 스키마가 하나만 존재하고, 각 사용자는 개념 스키마의 일부분을 사용한다. 즉, 외부 스키마는 개념 스키마를 기초로 하여 사용자의 이용 목적에 맞게 만들어진다. 일반적으로 스키마라고 하면 개념 스키마를 의미한다.

2.2.4 내부 단계

내부 단계(internal level)에서는 데이터베이스를 디스크나 테이프 같은 저장 장치의 관점에서 이해하고 표현한다. 내부 단계는 전체 데이터베이스가 저장 장치에 실제로 저장되는 방법을 정의하며 이를 내부 스키마(internal schema)라고 한다.

데이터베이스는 저장 장치에 파일 형태로 저장되는데 내부 스키마는 파일에 데이터를 저장하는 레코드의 구조, 레코드를 구성하는 필드 크기, 인덱스를 이용한 레코드 접근 경로 등을 정의한다.

내부 스키마는 데이터베이스의 개념 스키마에 대한 물리적인 저장 구조를 표현하므로 하나의 데이터베이스에 하나만 존재한다.

[그림 3-5]는 쇼핑몰의 데이터베이스를 3단계 데이터베이스 구조로 표현한 예다. 쇼핑몰 운영에는 상품, 주문, 직원 등의 데이터도 필요하지만 고객 데이터만 저장한다고 단순화하였다.

외부 단계에는 고객 분석팀과 상품 배송팀 사용자가 존재한다. 두 사용자는 자신의 팀에 필요한 데이터로 구성된 외부 스키마를 각각 가지고 있다. 고객 분석팀은 상품을 구매한 고객의 성향을 분석하는 것이 주 업무이므로 고객의 성별·나이·직업 데이터를 필요로 한다. 따라서 데이터베이스가 성별·나이·직업으로 구성되어 있다고 생각하고 다른 데이터에는 관심이 없다. 상품 배송팀도 고객에게 상품을 배송하는 업무를 담당하므로 데이터베이스가 고객번호·고객이름·주소·연락처로만 구성되어 있다고 생각한다. 만약 고객 분석팀에 팀원이 여러 명이라면 팀원들은 성별·나이·직업으로 구성된 외부 스키마를 함께 사용할 것이다. 외부 단계에서는 사용자별로 외부 스키마를 정의하여 불필요한 데이터 접근을 사전에 막아 보안 측면에서도 효과적이다.

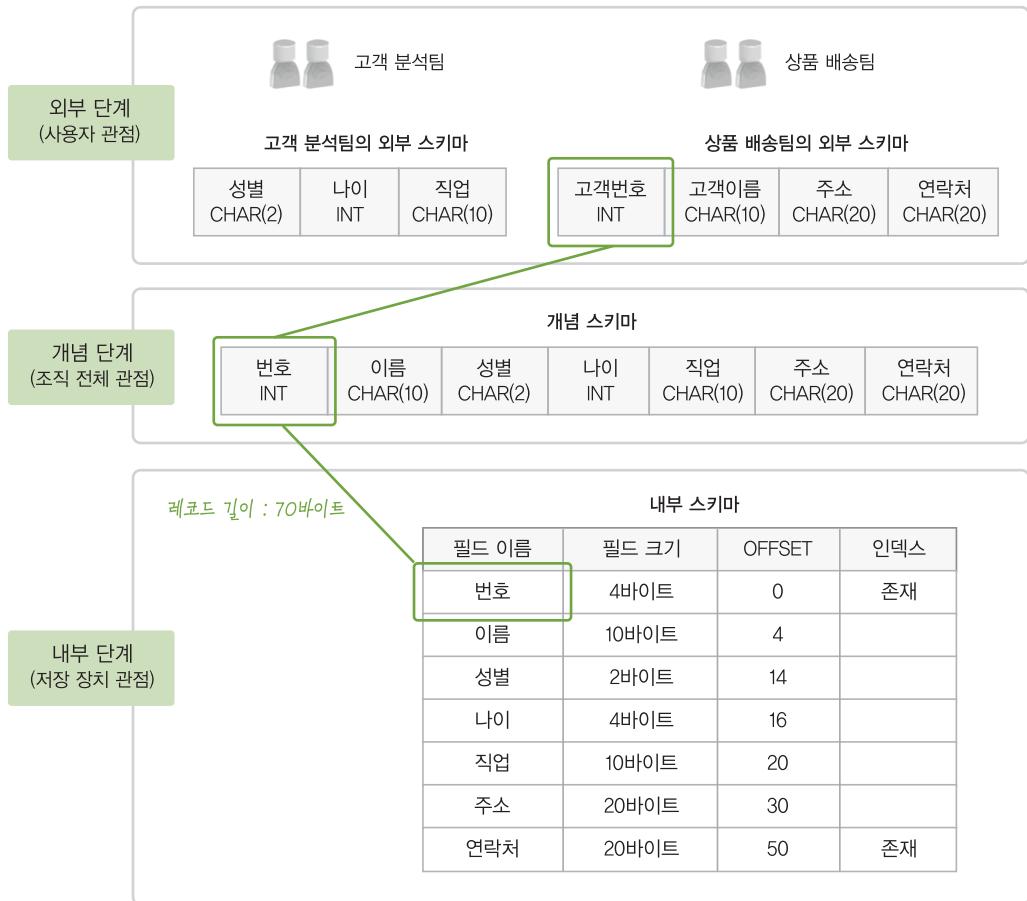


그림 3-5 3단계 데이터베이스 구조의 예

개념 단계에는 고객 데이터베이스 전체에 대한 논리적 구조를 정의하는 개념 스키마가 하나 존재한다. 개념 스키마는 고객 데이터베이스를 이용하는 모든 사용자에게 필요한 데이터를 종합하여 번호·이름·성별·나이·직업·주소·연락처로 데이터베이스를 구성하고, 각 데이터의 타입도 함께 정의한다.

내부 단계에는 고객 데이터베이스를 저장 장치에 저장하는 파일을 구성하는 레코드의 구조를 정의한 내부 스키마가 하나 존재한다. 내부 스키마에 정의된 고객 레코드는 필드 7개로 구성되어 있고, 레코드 총 길이는 70바이트다. 이 내부 스키마는 번호와 연락처 필드에 인덱스를 정의하고 있어, 번호나 연락처 필드의 값을 이용해 해당 고객 레코드에 빠르게 접근할 수 있다.

2.3 데이터 독립성

하나의 데이터베이스에는 세 가지 유형의 스키마가 존재하지만, 이는 데이터베이스를 바라보는 관점이 다를 뿐 모두 같은 데이터베이스를 표현한다. 실제 데이터는 물리적 저장 장치에 저장된 데이터베이스에만 존재하므로 사용자가 자신의 외부 스키마를 통해 원하는 데이터를 얻으려면 내부 스키마에 따라 저장된 데이터베이스에 접근해야 한다. 그러므로 세 가지 스키마 사이에는 유기적인 대응 관계가 성립해야 한다.

예를 들어 [그림 3-5]에서 상품 배송팀의 외부 스키마에 있는 고객번호 데이터는 개념 스키마에 있는 번호 데이터에 대응하고, 개념 스키마에 있는 번호 데이터는 내부 스키마에 있는 번호 필드에 대응한다는 연결 관계가 미리 정의되어 있어야 한다. 그래야 사용자가 물리적 저장 장치에 저장된 고객번호 데이터에 접근할 수 있다.

스키마 사이의 대응 관계를 사상 또는 매핑 mapping이라 한다. 외부 스키마와 개념 스키마는 외부/개념 사상에 의해 대응되고, 개념 스키마와 내부 스키마는 개념/내부 사상에 의해 대응된다. 데이터베이스 관리 시스템은 미리 정의된 외부/개념 사상과 개념/내부 사상 정보를 이용해 사용자가 원하는 데이터에 접근할 수 있다.

데이터베이스를 3단계 구조로 나누고, 단계별로 스키마를 유지하며 스키마 사이의 대응 관계를 정의하는 궁극적인 목적은 데이터 독립성 data independency을 실현하기 위해서이다. 데이터 독립성은 데이터베이스 관리 시스템의 중요한 장점이자 데이터베이스 관리 시스템이 필요한 이유이기도 하다. 데이터 독립성은 하위 스키마를 변경하더라도 상위 스키마가 영향을 받지 않는 특성이다. 3단계 데이터베이스 구조에는 논리적 데이터 독립성과 물리적 데이터 독립성이 존재한다.

[그림 3-6]은 3단계 데이터베이스 구조에 존재하는 외부/개념 사상과 개념/내부 사상이 어떤 스키마 간의 대응 관계를 의미하는지 그림으로 보여준다. 논리적 데이터 독립성과 물리적 데이터 독립성이 어떤 사상 정보와 관련이 있는지도 확인할 수 있다.

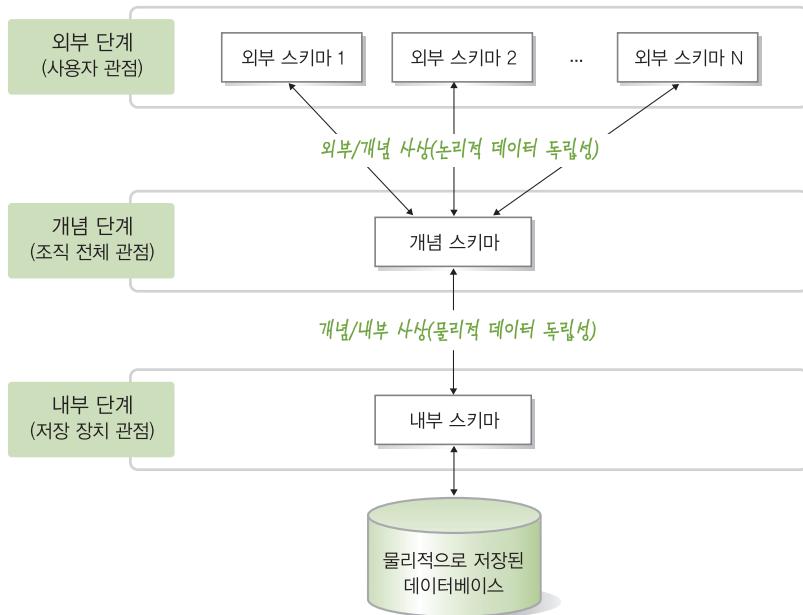


그림 3-6 3단계 데이터베이스 구조에서 스키마 간의 사상

2.3.1 논리적 데이터 독립성

논리적 데이터 독립성은 개념 스키마가 변경되더라도 외부 스키마가 영향을 받지 않는 것이다. 그래서 전체 데이터베이스의 논리적인 구조가 변경되어도 관련된 외부/개념 사상 정보만 적절히 수정해주면 직접 관련이 없는 사용자를 위한 외부 스키마는 변경할 필요가 없다. 외부/개념 사상은 외부 스키마와 개념 스키마의 대응 관계를 정의한 것으로, 응용 인터페이스 application interface라고도 한다. 개념 스키마가 변경되어도 외부 스키마가 영향을 받지 않는다는 것은 결국 외부 스키마의 사용자가 전체 데이터베이스의 논리적 구조가 변경되었다는 사실을 알 필요가 없음을 의미한다.

사용자는 응용 프로그램을 통해 데이터베이스 시스템을 이용하는 경우가 많으므로 여기서는 사용자와 응용 프로그램을 동일하게 생각한다.

예를 들어 [그림 3-5]의 개념 스키마에서 연락처 데이터의 이름이 전화번호로 바뀌는 경우, 상품 배송팀의 외부 스키마에 있는 연락처가 이후부터는 개념 스키마의 전화번호와 대응 관계에 있다고 외부/개념 사상만 정확히 수정해주면 된다. 그러면 상품 배송팀의 외부 스키마에 있는 연락처 데이터의 이름을 변경할 필요가 없고, 변경된 내용과 직접 관련이 없는 고객 분석팀의

외부 스키마도 변경할 필요가 없다. 개념 스키마에 새로운 내용이 추가되거나 기존 내용이 삭제되는 경우에도 외부 스키마는 영향을 받지 않는다.

2.3.2 물리적 데이터 독립성

물리적 데이터 독립성은 내부 스키마가 변경되더라도 개념 스키마가 영향을 받지 않는 것이다. 그래서 결과적으로는 외부 스키마도 영향을 받지 않는다.

물리적 데이터 독립성이 실현되면 데이터베이스의 저장 구조가 변경되어도 관련된 개념/내부 사상 정보만 적절히 수정해주면 직접적으로 관련이 없는 데이터베이스의 논리적 구조는 영향을 받지 않는다. 개념/내부 사상은 개념 스키마와 내부 스키마의 대응 관계를 정의한 것으로, 저장 인터페이스 storage interface라고도 한다.

예를 들어 [그림 3-5]의 내부 스키마에서 주소 다음에 연락처 필드가 저장되는 순서가 바뀌어도 두 필드와 관련된 개념/내부 사상 정보만 수정해주면 된다. 그러면 개념 스키마에는 변경할 내용이 없고 외부 스키마도 변경할 필요가 없다. 내부 스키마에 새로운 인덱스가 추가되거나 기존 인덱스가 삭제되는 경우에도 개념 스키마는 영향을 받지 않는다.

2.4 데이터 사전

데이터베이스는 조직 운영을 위해 필요한 실제 데이터를 저장하는데, 저장된 데이터를 올바르게 관리하고 이용하려면 필요한 부가 정보도 저장해야 한다. 대표적인 부가 정보가 스키마와 사상 정보다.

데이터 독립성을 실현하면서 데이터베이스를 다양한 관점에서 이해하기 위해 정의되는 세 가지 스키마와, 스키마 간의 사상 정보도 어떤가에 저장되어 있어야 필요할 때 사용할 수 있다. 데이터베이스에 저장되는 데이터에 관한 정보를 저장하는 곳을 데이터 사전 data dictionary 또는 시스템 카탈로그 system catalog라고 한다. 데이터 사전은 일반 사전처럼 데이터베이스에 저장되어 있는 데이터를 정확하고 효율적으로 이용하기 위해 참고해야 되는 스키마, 사상 정보, 다양한 제약조건 등을 저장하고 있다. 데이터베이스에 저장되는 데이터에 관한 정보(데이터 사전 정보)이므로 데이터에 대한 데이터 data about data를 의미해 메타 데이터 meta data라고도 한다.

데이터 사전도 데이터를 저장하는 데이터베이스의 일종이기 때문에 시스템 데이터베이스^{system database}라고 한다. 그리고 이와 구별하기 위해 사용자가 실제로 이용하는 데이터가 저장되는 일반 데이터베이스를 사용자 데이터베이스^{user database}라고 하기도 한다. 데이터 사전은 데이터베이스 관리 시스템이 스스로 생성하고 유지하는 것으로, 데이터베이스 관리 시스템이 주로 접근하지만 일반 사용자도 접근할 수 있다. 단, 데이터베이스 관리 시스템은 데이터 사전에 내용을 새로 추가하거나 수정할 수 있는 반면, 사용자는 저장된 내용을 검색만 할 수 있다. 데이터 사전에 있는 데이터에 실제로 접근하는 데 필요한 위치 정보는 데이터 디렉토리^{data directory}라는 곳에서 관리한다. 데이터 사전과 데이터 디렉토리는 둘 다 시스템을 위한 데이터베이스라는 공통점이 있지만, 데이터 사전은 사용자가 접근할 수 있고 데이터 디렉토리는 시스템만 접근할 수 있다는 차이가 있다.

03

데이터베이스 사용자

데이터베이스 시스템을 구성하는 또 하나의 중요 요소가 사용자 user다. 사용자는 데이터베이스를 이용하기 위해 접근하는 모든 사람을 의미한다. 데이터베이스를 이용하는 사용자는 매우 다양한데, 이용 목적에 따라 크게 데이터베이스 관리자, 최종 사용자, 응용 프로그래머로 나눌 수 있다.

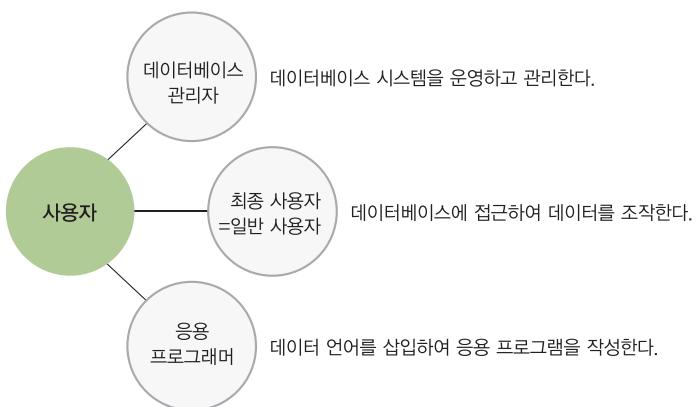


그림 3-7 데이터베이스 사용자

3.1 데이터베이스 관리자

데이터베이스 관리자 DBA: DataBase Administrator는 데이터베이스 시스템을 운영·관리한다. 데이터베이스를 직접 활용하기보다는 조직 내의 사용자를 위해 데이터베이스를 설계 및 구축하고, 제대로 서비스할 수 있도록 데이터베이스를 제어한다. 데이터베이스 관리자는 다음 절에서 살펴볼 데이터 언어 중 주로 데이터 정의어와 데이터 제어어를 이용해 데이터베이스에 접근한다. 데이터베이스 관리자는 데이터베이스 운영·관리를 책임지므로 컴퓨터 시스템이나 데이터베이

스와 관련해 지식과 경험을 많이 갖추고 있어야 한다. 큰 조직에서는 데이터베이스 설계 업무만 담당하는 데이터베이스 설계자 database designer를 따로 두기도 한다.

데이터베이스 관리자의 주요 업무는 다음과 같다.

- **데이터베이스 구성 요소 선정**

사용자의 요구 사항을 분석하여 데이터베이스를 구성할 데이터를 결정한다.

- **데이터베이스 스키마 정의**

선정된 데이터베이스의 구성 요소를 토대로 데이터베이스 스키마를 설계하고, 데이터 정의어를 이용해 설계한 스키마를 데이터베이스 관리 시스템에 설명한다.

- **물리적 저장 구조와 접근 방법 결정**

데이터베이스를 물리적으로 저장하기 위한 레코드 구조를 설계한다. 레코드들 간의 저장 순서와 레코드에 빠르게 접근하기 위해 인덱스를 만들 기준 필드 등도 결정한다.

- **무결성 유지를 위한 제약조건 정의**

현실 세계의 실제 데이터와 일치하는, 즉 결합이 없는 데이터만 데이터베이스에 저장할 수 있도록 필요한 규칙을 정의한다. 그리고 데이터베이스를 이 규칙에 따라 재어하여 데이터의 정확성과 유효성을 유지한다.

- **보안 및 접근 권한 정책 결정**

허가되지 않는 사용자가 데이터베이스에 불법적으로 접근하는 것을 방지하고, 허가된 사용자에게 적절한 권한을 부여하는 보안 관련 정책을 결정한다.

- **백업 및 회복 기법 정의**

시스템 장애에 대비하여 데이터베이스를 백업하거나, 손상된 데이터베이스를 일관된 상태로 복구하는 방법을 정의한다.

- **시스템 데이터베이스 관리**

데이터 사전 같은 시스템 데이터베이스를 관리한다.

- **시스템 성능 감시 및 성능 분석**

시스템 성능을 저해하는 병목 현상 bottleneck 등이 발생하지 않는지 확인하고, 시스템 자원의 활용도 분석 등을 통해 시스템의 성능을 감시한다.

- 데이터베이스 재구성

데이터베이스는 사용자의 요구 사항이나 사용 형태가 달라지면, 변화된 내용에 맞게 데이터베이스를 재구성하여 사용자의 만족도를 높여야 한다. 그리고 시스템의 전체 성능을 향상시켜야 하거나 시스템 장비가 교체된 경우에도 데이터베이스를 재구성해야 한다. 데이터베이스 관리자는 항상 사용자의 요구 사항에 귀기울이고 시스템 성능을 분석하여, 새로운 변화에 부응하면서 성능을 향상시키는 방향으로 데이터베이스를 재구성해야 한다.

3.2 최종 사용자

데이터를 조작(삽입·삭제·수정·검색)하기 위해 데이터베이스에 접근하는 사람들을 일반 사용자 또는 최종 사용자 *end user*라 한다. 최종 사용자는 컴퓨터 시스템이나 데이터베이스 전문 지식이 꼭 필요하지는 않다. 그리고 데이터 정의어도 사용할 수 있지만 주로 데이터 조작어를 사용한다.

최종 사용자는 캐주얼 사용자 *casual end user*와 초보 사용자 *naive end user*로 구분할 수 있다. 캐주얼 사용자는 데이터베이스에 대한 이론적 지식이 있으며, 주로 데이터 조작어를 이용해 원하는 데이터와 데이터에 대한 처리를 데이터베이스 관리 시스템에 직접 설명한다. 초보 사용자는 데이터베이스를 초보 수준으로 이용할 수 있어 데이터 조작어로 자신의 요구를 직접 표현하기보다는 메뉴나 GUI Graphic User Interface 형태의 응용 프로그램을 통해 데이터베이스를 사용한다.

3.3 응용 프로그래머

응용 프로그래머 *application programmer*는 C 언어, 자바 *Java* 등과 같은 프로그래밍 언어로 응용 프로그램을 작성할 때 데이터베이스에 접근하는 데이터 조작어를 삽입하는 사용자다. 데이터 정의어를 삽입할 수도 있지만 주로 데이터 조작어를 삽입한다.

최종 사용자는 응용 프로그래머가 작성한 응용 프로그램을 이용해 데이터베이스에 접근할 수 있다. 도서의 위치를 검색하거나 고객의 구매 요청 처리를 위해 서점 직원에게 제공되는 응용 프로그램이 좋은 예다.



그림 3-8 최종 사용자와 응용 프로그래머의 예

04

데이터 언어

미국 사람에게 길을 물어봐야 한다면 어떤 언어로 부탁할까? 대부분 영어를 사용할 것이다. 데이터베이스에서도 사용자를 대신해 데이터베이스를 구축하고 활용 및 관리하는 데이터베이스 관리 시스템에 부탁할 때 사용하는 언어가 있다. 이것이 바로 데이터 언어 *data language*다. 데이터 언어는 사용자가 데이터베이스를 구축하고 접근하기 위해 데이터베이스 관리 시스템과 통신하기 위한 수단이다.

인간의 언어도 때와 장소에 맞는 용법이 있는 것처럼 데이터 언어에도 상황에 따른 용법이 있다. 데이터 언어는 데이터베이스 관리 시스템의 정의·조작·제어 기능을 이용하기 위한 것이기 때문에 사용 목적에 따라 데이터 정의어, 데이터 조작어, 데이터 제어어로 나눈다. 이는 하나의 데이터 언어를 기능에 따라 내부적으로 구분 짓는 것일 뿐 독립적으로 존재하는 언어들은 아니다.

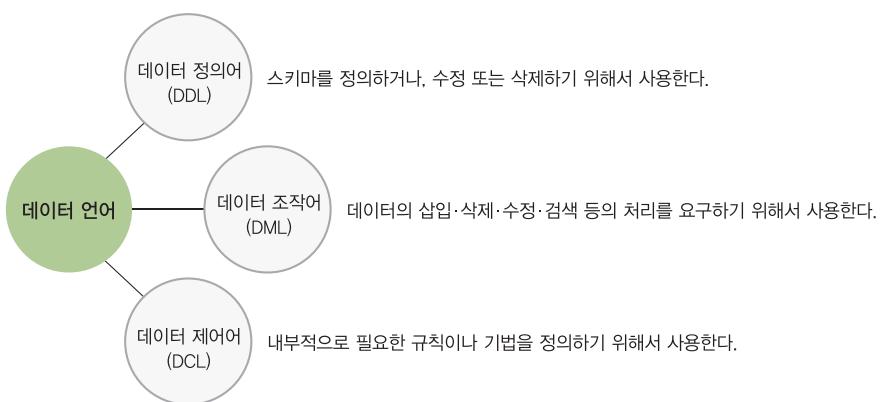


그림 3-9 데이터 언어의 종류와 용도

앞에서도 몇 번 등장한 데이터 정의어, 데이터 조작어, 데이터 제어어를 어떤 용도로 사용하는지 하나씩 살펴보자.

4.1 데이터 정의어

데이터 정의어 DDL:Data Definition Language는 새로운 데이터베이스를 구축하기 위해 스키마를 정의하거나, 기존 스키마의 정의를 삭제 또는 수정하기 위해 사용하는 데이터 언어다. 즉, 새로 만들려는 데이터베이스의 스키마를 설명하거나, 이미 정의된 스키마의 구조나 제약조건 등을 변경 또는 삭제하고 싶어 이를 데이터베이스 관리 시스템에 알릴 때 사용한다. 데이터 정의어로 정의된 스키마는 데이터 사전에 저장되고, 삭제나 수정이 발생하면 이 내용도 데이터 사전에 반영된다. 데이터 사전에 저장된 스키마 정보는 사용자나 데이터베이스 관리 시스템이 필요할 때 참고할 수 있다.

4.2 데이터 조작어

데이터 조작어 DML:Data Manipulation Language는 사용자가 데이터의 삽입·삭제·수정·검색 등의 처리를 데이터베이스 관리 시스템에 요구하기 위해 사용하는 데이터 언어다. 데이터 정의어를 이용해 스키마를 정의하면 스키마에 따라 조직에 필요한 실제 데이터 값(인스턴스)이 저장되는 데, 사용자가 실제 데이터 값을 활용하기 위해 사용하는 것이 데이터 조작어다. 데이터 조작어는 설명 방식에 따라 절차적 데이터 조작어와 비절차적 데이터 조작어로 나눈다.

■ 절차적 데이터 조작어

절차적 데이터 조작어 procedural DML는 사용자가 어떤 what 데이터를 원하고 해당 데이터를 얻기 위해서는 어떻게 how 처리해야 하는지를 설명한다. [그림 3-10]에서 위의 상황과 같이 어머니가 두부를 사오라고 심부름을 시키실 때 1번 버스를 타고 세 정거장을 지나 한빛 마트에 가야 두부를 사올 수 있다고 설명하는 것과 같다.

■ 비절차적 데이터 조작어

비절차적 데이터 조작어 nonprocedural DML는 사용자가 어떤 what 데이터를 원하는지만 설명한다. 즉, 해당 데이터를 얻으려면 어떻게 how 처리해야 하는지는 데이터베이스 관리 시스템에 맡긴다. 비절차적 데이터 조작어는 사용자가 어떤 데이터를 원하는지만 데이터베이스 관리 시스템에 선언하는 방식이기 때문에 선언적 언어 declarative language라고도 한다. [그림 3-10]에서 아래의 상황과 같이 두부를 사오라고만 하고 어떻게 사와야 하는지는 따로 설명하지 않는 것과 같다. 사오는 방법은 심부름하는 사람의 몫이다.



그림 3-10 절차적 데이터 조작어와 비절차적 데이터 조작어의 이해

4.3 데이터 제어어

데이터 제어어 DCL:Data Control Language는 데이터베이스에 저장된 데이터를 여러 사용자가 무결성과 일관성을 유지하며 문제없이 공유할 수 있도록, 내부적으로 필요한 규칙이나 기법을 정의하는 데 사용하는 데이터 언어다. 사용자는 데이터베이스를 올바르게 관리하기 위해 필요한 규칙과 기법을 데이터 제어어를 이용해 데이터베이스 관리 시스템에 설명한다. 그러면 데이터 베이스 관리 시스템이 이 규칙과 기법에 따라 데이터베이스를 제어하고 보호한다. 데이터 제어어를 이용해 규칙이나 기법을 정의하는 이유는, 데이터베이스 관리 시스템의 장점이기도 한 다음을 보장하기 위해서다.

- **무결성**^{integrity} : 데이터베이스에 정확하고 유효한 데이터만 유지한다.
- **보안**^{security} : 허가받지 않는 사용자가 데이터에 접근하는 것을 차단하거나, 허가된 사용자가 접근 권한이 있는 데이터에만 접근할 수 있게 한다.
- **회복**^{recovery} : 장애가 발생해도 데이터의 일관성을 유지한다.
- **동시성**^{concurrency} : 여러 사용자가 같은 데이터에 동시에 접근하여 처리할 수 있게 한다.

■ 데이터 제어어는 원래 데이터 정의어로 분류되었지만, 데이터베이스 제어 기능이 중요해지고 다양한 제어 기능이 소개되면서 독립되었다.

05

데이터베이스 관리 시스템의 구성

데이터베이스를 관리하고 사용자의 데이터 처리 요구를 수행하는 데이터베이스 관리 시스템은 데이터베이스 시스템의 주요 구성 요소다. 사용자와 데이터베이스 사이에 위치하며, 기능에 따라 크게 질의 처리기와 저장 데이터 관리자로 구분할 수 있다.

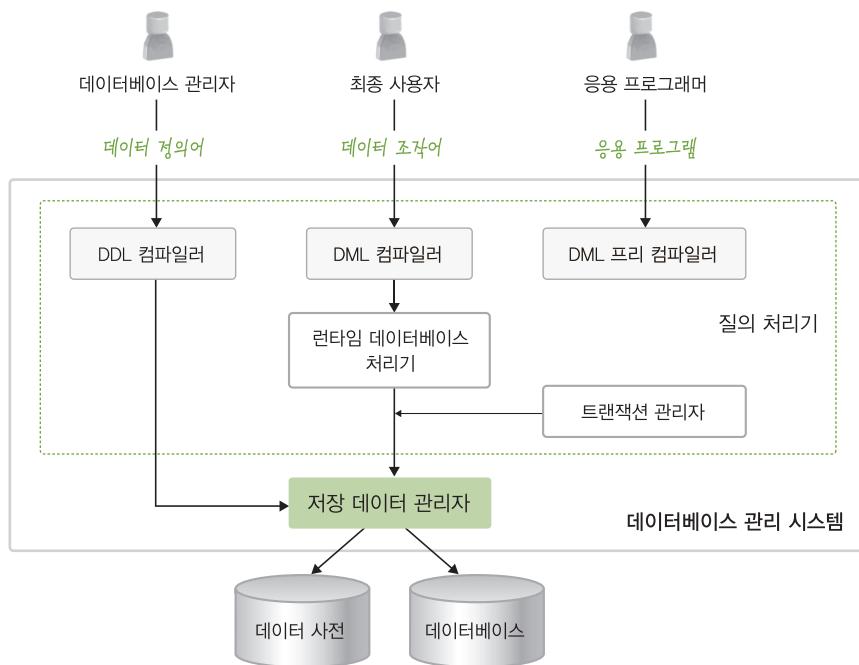


그림 3-11 데이터베이스 관리 시스템의 구성

5.1 질의 처리기

질의 처리기 query processor는 사용자의 데이터 처리 요구를 해석하여 처리하는 역할을 담당하고, 다음의 주요 구성 요소들을 포함한다.

- **DDL 컴파일러** DDL compiler : 데이터 정의어로 작성된 스키마의 정의를 해석한다. 그리고 저장 데이터 관리자의 도움을 받아 새로운 데이터베이스를 구축하고, 스키마의 정의를 데이터 사전에 저장한다. 데이터 정의어로 작성된 기존 스키마의 삭제나 수정 요청도 처리하여, 변경된 내용을 데이터 사전에 적용한다.
- **DML 프리 컴파일러** DML precompiler : 응용 프로그램에 삽입된 데이터 조작어를 추출하여 DML 컴파일러에 전달한다. 단, 데이터 조작어와 관련이 없는 나머지 코드들은 해당 언어의 컴파일러에 보내진다.
- **DML 컴파일러** DML compiler : 데이터 조작어로 작성된 데이터의 처리(삽입·삭제·수정·검색) 요구를 분석하여 런타임 데이터베이스 처리기가 이해할 수 있도록 해석한다.
- **런타임 데이터베이스 처리기** run-time database processor : 저장 데이터 관리자를 통해 데이터베이스에 접근하여, DML 컴파일러로부터 전달받은 데이터 처리 요구를 데이터베이스에서 실제로 실행한다.
- **트랜잭션 관리자** transaction manager : 데이터베이스에 접근하는 과정에서 사용자의 접근 권한이 유효한지를 검사하고, 데이터베이스 무결성을 유지하기 위한 제약조건 위반 여부를 확인한다. 회복이나 병행 수행과 관련된 작업도 담당한다.

5.2 저장 데이터 관리자

저장 데이터 관리자 stored data manager는 디스크에 저장되어 있는 사용자 데이터베이스와 데이터 사전을 관리하고, 여기에 실제로 접근하는 역할을 담당한다. 그런데 디스크에 저장된 데이터에 접근하는 것은 운영체제의 기본 기능이므로 저장 데이터 관리자는 운영체제의 도움을 받아 데이터베이스에 대한 접근을 수행한다.

요약

1 데이터베이스 시스템

- 데이터베이스에 데이터를 저장하고, 이를 관리하여 조직에 필요한 정보를 생성해주는 시스템이다.
- 사용자, 데이터 언어, 데이터베이스 관리 시스템, 데이터베이스, 컴퓨터로 구성된다.

2 스키마와 인스턴스

- 스키마 : 데이터베이스에 저장되는 데이터 구조와 제약조건을 정의한 것이다.
- 인스턴스 : 스키마에 따라 데이터베이스에 실제로 저장된 값이다.

3 3단계 데이터베이스 구조

데이터베이스를 쉽게 이해하고 이용할 수 있도록 하나의 데이터베이스를 관점에 따라 세 단계(외부 단계, 개념 단계, 내부 단계)로 나눈 것이다.

- 외부 단계 : 데이터베이스를 개별 사용자 관점에서 이해하고 표현한다. 사용자에게 필요한 데이터베이스를 정의한 외부 스키마가 여러 개 존재할 수 있다.
- 개념 단계 : 데이터베이스를 조직 전체의 관점에서 이해하고 표현한다. 데이터베이스 전체의 논리적 구조를 정의하는 개념 스키마가 하나만 존재한다.
- 내부 단계 : 데이터베이스를 저장 장치의 관점에서 이해하고 표현한다. 데이터베이스가 저장 장치에 저장되는 방법을 정의한 내부 스키마가 하나만 존재한다.

4 데이터 독립성

3단계 데이터베이스 구조의 목적은 데이터 독립성을 실현하기 위해서인데, 데이터 독립성에는 논리적 데이터 독립성과 물리적 데이터 독립성이 존재한다.

- 논리적 데이터 독립성 : 개념 스키마가 변경되어도 외부 스키마는 영향을 받지 않는다(외부/개념 사상).
- 물리적 데이터 독립성 : 내부 스키마가 변경되어도 개념 스키마는 영향을 받지 않는다(개념/내부 사상).

5 데이터 사전(시스템 카탈로그)

데이터베이스에 저장되는 데이터에 관한 정보, 즉 메타 데이터를 유지하는 시스템 데이터베이스다.

6 데이터베이스 사용자

데이터베이스를 이용하기 위해 접근하는 모든 사람이다. 데이터베이스 관리자, 최종 사용자, 응용 프로그래머로 나눈다.

- 데이터베이스 관리자 : 데이터베이스 시스템을 운영·관리한다.
- 최종 사용자 : 데이터베이스에 접근하여 데이터를 조작(삽입·삭제·수정·검색)한다.
- 응용 프로그래머 : 데이터 언어를 삽입하여 응용 프로그램을 작성한다.

7 데이터 언어

사용자와 데이터베이스 관리 시스템 간의 통신 수단이다. 데이터 정의어, 데이터 조작어, 데이터 제어어로 나눈다.

- 데이터 정의어DDL : 스키마를 정의하거나, 수정 또는 삭제하기 위해서 사용한다.
- 데이터 조작어DML : 데이터의 삽입·삭제·수정·검색 등의 처리를 요구하기 위해서 사용한다.
- 데이터 제어어DCL : 동시 공유가 가능하면서도 무결성과 일관성을 유지하도록 내부적으로 필요한 규칙이나 기법들을 정의하기 위해서 사용한다.

8 데이터베이스 관리 시스템

주요 기능은 데이터베이스 관리와 데이터 처리 요구의 수행이다. 질의 처리기와 저장 데이터 관리자로 나눈다.

- 질의 처리기 : 사용자의 데이터 처리 요구를 해석하여 처리한다.
- 저장 데이터 관리자 : 디스크에 저장된 사용자 데이터베이스와 데이터 사전을 관리하고, 여기에 실제로 접근한다.

연습문제

1 3단계 데이터베이스 구조에서 다음 설명과 관련 있는 스키마는?

데이터베이스를 물리적 저장 장치의 관점에서 이해한 구조다. 레코드의 구조, 레코드를 구성하는 필드 크기, 레코드의 물리적 순서, 인덱스를 이용한 레코드의 접근 경로 등과 같이 실제로 저장되는 방법을 정의한다.

- ① 외부 스키마-external schema
- ② 개념 스키마-conceptual schema
- ③ 내부 스키마-internal schema
- ④ 슈퍼 스키마-super schema

2 3단계 데이터베이스 구조에서 다음 설명과 관련 있는 스키마는?

데이터베이스를 사용자 관점에서 이해한 구조다. 각 사용자에게 필요한 데이터베이스의 구조를 정의하여 하나의 데이터베이스에 여러 개가 존재할 수 있다.

- ① 외부 스키마-external schema
- ② 개념 스키마-conceptual schema
- ③ 내부 스키마-internal schema
- ④ 슈퍼 스키마-super schema

3 개념 스키마-conceptual schema에 대한 설명으로 옳은 것을 모두 고르시오.

- ① 사용자 관점에서 본 데이터베이스의 구조다.
- ② 조직 전체의 관점에서 본 데이터베이스의 구조다.
- ③ 저장 장치의 관점에서 본 데이터베이스의 구조다.
- ④ 여러 개가 존재할 수 있다.
- ⑤ 데이터베이스에 저장되는 데이터들 간의 관계와 제약조건을 정의한다.
- ⑥ 접근 권한, 보안 정책을 정의한다.
- ⑦ 데이터를 물리적으로 저장하는 방법을 정의한다.
- ⑧ 서브 스키마-sub schema라고도 한다.

4 데이터 사전에 대한 설명으로 옳지 않은 것은?

- ① 데이터 사전에 저장된 데이터를 메타 데이터라고도 한다.
- ② 시스템 자신이 필요로 하는 스키마 및 여러 객체에 관한 정보를 저장한다.
- ③ 사용자가 데이터 사전에 내용을 직접 추가하거나 수정할 수 없다.
- ④ 시스템 데이터베이스이므로 일반 사용자는 내용을 검색할 수 없다.

5 데이터베이스 관리자의 주요 업무와 거리가 먼 것은?

- ① 데이터베이스 스키마 정의
- ② 보안 및 접근 권한 정책 결정
- ③ 응용 프로그램의 개발
- ④ 무결성 유지를 위한 제약조건 정의

6 사용자가 데이터의 삽입·삭제·수정·검색 등의 처리를 데이터베이스 관리 시스템에 요구하기 위해 사용하는 데이터 언어는?

- ① 데이터 정의어 DDL
- ② 데이터 조작어 DML
- ③ 데이터 제어어 DCL
- ④ 데이터 요청어 DRL

7 데이터 제어어 DCL의 기능으로 거리가 먼 것은?

- ① 무결성 유지
- ② 회복 및 동시 공유 제어
- ③ 접근 제어 및 권한 부여
- ④ 스키마 정의

8 다음 설명에서 Ⓐ와 Ⓛ가 각각 무엇인지 답하시오.

(Ⓐ)는 데이터베이스에 저장되는 데이터 구조와 제약조건을 정의한 것이다. 그리고 (Ⓑ)에 따라 데이터베이스에 실제로 저장된 값을 (Ⓒ)라고 한다.

9 다음 설명에서 Ⓛ가 무엇인지 답하시오.

3단계 데이터베이스 구조에서 (Ⓛ)는 개념 스키마와 내부 스키마의 대응 관계를 정의한 것으로, 저장 인터페이스라고도 한다.

10 다음 설명에서 Ⓛ와 Ⓜ가 각각 무엇인지 답하시오.

(Ⓛ)는 하위 스키마를 변경하더라도 상위 스키마가 영향을 받지 않는 특성을 의미한다. 3단계 데이터베이스 구조에서는 두 가지 유형으로 존재한다. 이 중 개념 스키마가 변경되더라도 외부 스키마가 영향을 받지 않는 것을 (Ⓜ)라고 한다.

11 다음 설명에서 ①와 ②가 각각 무엇인지 답하시오.

(①)는 데이터베이스에 저장되는 데이터에 관한 정보를 저장하는 곳으로, 스키마, 매펑 정보, 다양한 제약조건 등을 저장한다. 그리고 (②)에 저장되어 있는 정보에 실제로 접근하는 데 필요한 위치 정보는 (③)에서 관리한다.

12 다음 설명에서 ①와 ②가 각각 무엇인지 답하시오.

데이터베이스 관리 시스템은 내부적으로 사용자의 데이터 처리 요구를 해석하여 처리하는 역할을 담당하는 (①)과, 디스크에 저장되어 있는 사용자 데이터베이스와 데이터 사전을 관리하고 접근하는 역할을 담당하는 (②)로 구성되어 있다.

13 데이터베이스 시스템이 무엇인지 설명하시오.

14 데이터베이스 시스템의 주요 구성 요소 5가지를 설명하시오.

15 데이터 독립성의 의미를 설명하고, 3단계 데이터베이스 구조에서 데이터 독립성을 실현하는 방법을 설명하시오.

16 데이터베이스 관리자가 담당하는 주요 업무를 간단히 설명하시오

17 데이터 언어를 사용 목적에 따라 세 가지 유형으로 분류하고, 각각을 설명하시오.

Appendix A

MS SQL 서버의 설치와 데이터베이스 구축

01 MS SQL 서버의 소개

02 MS SQL 서버의 설치

03 MS SQL 서버를 이용한 데이터베이스 구축

04 MS SQL 서버를 이용한 데이터베이스 활용

01

MS SQL 서버의 소개

MS SQL 서버를 이용해 데이터베이스를 실제로 구축하고 활용하는 과정을 살펴보기에 앞서 MS SQL 서버에 대해 간단히 살펴보자. 마이크로소프트 사의 MS SQL 서버는 대표적인 관계 DBMS다. 집필 시점을 기준으로 MS SQL 2012까지 출시되었고, 다음 네 가지 제품군 형태가 주로 사용된다.



그림 A-1 MS SQL 서버 2012

■ 엔터프라이즈 에디션 enterprise edition

- AlwaysOn이라고 불리는 고가용성은 물론 보안, 복제, 데이터 웨어하우징 등과 관련하여 MS SQL 서버가 포함하고 있는 대부분의 기능들을 모두 제공한다. 확장성과 성능이 가장 뛰어난 제품군이다.
- 대규모의 데이터 센터를 구축하거나, 높은 성능의 데이터 웨어하우스 솔루션을 개발하는데 주로 활용된다.

■ 비즈니스 인텔리전스 에디션 business intelligence edition

- 고가용성, 보안, 복제, 데이터 웨어하우징 등과 관련하여 엔터프라이즈 에디션보다 기능을 적게 제공하지만, 안전하고 관리하기 쉬운 BI Business Intelligence 솔루션을 개발하기에 충분하다.

- 데이터 탐색과 시각화를 기반으로 하는 브라우저, 데이터 분해 및 통합 관리에 뛰어난 기능을 제공한다.

■ 스탠다드 에디션 standard edition

- 기본 데이터 관리는 물론 클라우드용 공동 개발 도구 등을 지원하므로 최소한의 자원으로 데이터베이스 관리 기능을 제공하고자 할 때 적합하다.
- 엔터프라이즈 에디션이나 비즈니스 인텔리전스 에디션을 사용하는 조직보다 상대적으로 작은 조직에서 주로 활용된다.

■ 익스프레스 에디션 express edition

- SQL 서버의 확장형 버전으로, SQL 서버의 기본 기능을 제공하면서도 무료로 배포된다.
- 데스크톱이나 소규모의 서버에서 데이터베이스 관리 기능을 이용한 응용 프로그램을 분석하고 개발하는 데 주로 활용된다.
- 설치가 빠르고 적은 수의 필수 구성 요소를 포함하여, 개발자나 취미로 데이터베이스 기반의 응용 프로그램을 개발하고자 하는 사용자들에게 적합하다.

이 책에서는 MS SQL 서버와 오라클이라는 두 가지 DBMS를 소개한다. MS SQL 서버와 오라클 모두 대표적인 관계 DBMS로, 데이터베이스를 기반으로 하는 응용 프로그램을 개발하기 위해 필요한 기본 데이터 관리 기능을 제공한다. 제품별로 특화된 고급 기능이 존재하지만 데이터베이스 기본을 공부하는 수준에서는 둘 중 어느 것을 사용해도 상관없다.

일반적으로 DBMS는 다수의 사용자가 접근할 수 있도록 서버에 설치한다. 하지만 개인적으로 실습하거나 간단한 데이터베이스 응용 프로그램을 개발하고자 할 경우에는 단일 사용자 환경에서 이용해도 상관 없다. 여기서는 MS SQL 서버 2012 익스프레스 에디션을 중심으로 MS SQL 서버의 기본 기능을 익힌다.

▣ 오라클은 윈도우, 유닉스/리눅스 등 대부분의 운영체제에서 동작하지만, MS SQL 서버는 윈도우 계열의 운영체제에서만 동작한다. 특히 MS SQL 서버 2012는 Windows Vista Service Pack 2, Windows Server 2008 Service Pack 2, Windows 2008 R2 Service Pack 1 및 Windows 7 Service Pack 1 운영체제에만 설치된다. 하지만 오라클보다 가격이 저렴한 편이고, 환경 설정이 비교적 쉬워 초보자가 다루기가 쉽다.

02

MS SQL 서버의 설치

2.1 MS SQL 서버의 설치 과정

MS SQL 서버 2012는 윈도우 운영체제에서만 동작하기 때문에 설치 전에 시스템 환경을 점검해야 한다. 특히, MS SQL 서버 2012를 설치할 컴퓨터의 운영체제가 Windows Vista이거나 Windows Server 2012인 경우에는 Service Pack 2가 설치되어 있어야 하며, Windows 7과 Windows Server 2012 R2인 경우에는 Service Pack 1이 설치되어 있어야 한다.

만약 컴퓨터에 비주얼 스튜디오 2010^{Visual Studio 2010}이 설치되어 있다면 Service Pack 1 이상이 적용되어 있어야 한다. Service Pack 1 이상이 설치되어 있지 않으면 설치 도중 ‘Visual Studio 2010 인스턴스입니다.’라는 메시지와 함께 설치가 잠시 중단될 것이다. 이 경우에는 안내 메시지를 참조하거나 <http://www.microsoft.com/ko-kr/download/details.aspx?id=23691>에서 Service Pack 1을 다운받아 설치하면 계속 진행할 수 있다.

2.1.1 MS SQL 서버 설치 파일의 검색과 다운로드

MS SQL 서버는 마이크로소프트 홈페이지에서 다운로드할 수 있다. 마이크로소프트 다운로드 페이지(<http://www.microsoft.com/ko-kr/download>)의 검색 창에서 ‘sql server 2012 express’로 검색한 후 [Microsoft® SQL Server® 2012 서비스 팩 1(SP1) Express]를 클릭한다.

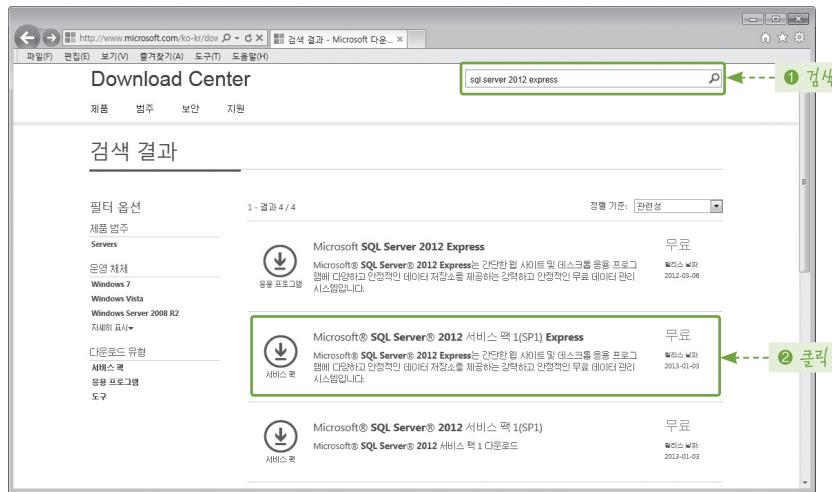


그림 A-2 MS SQL 서버 2012 설치 파일 검색과 선택

자신의 시스템 환경에 맞게 설치 파일을 다운로드한다. 컴퓨터가 32비트인 경우에는 SQLEXPRWT_x86_KOR.exe를 선택하고, 64비트인 경우에는 SQLEXPRWT_x64_KOR.exe를 선택한다.

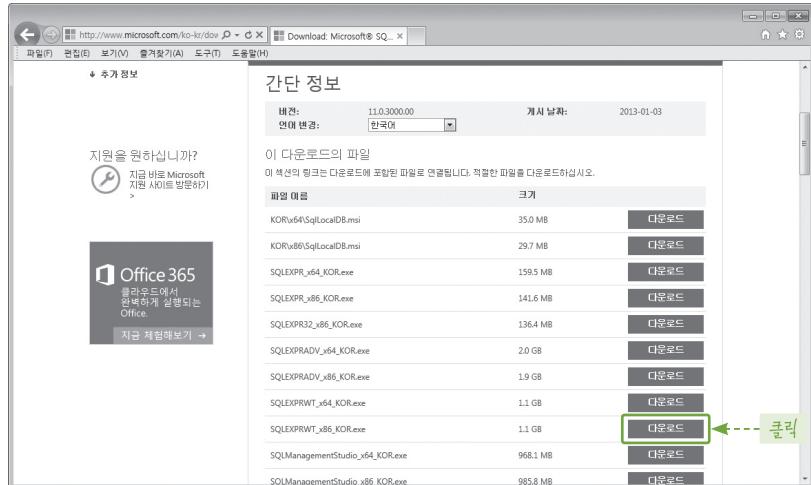


그림 A-3 MS SQL 서버 2012 설치 파일 다운로드

2.1.2 설치 파일의 압축 해제와 설치

MS SQL 서버 2012 설치를 위한 압축 파일을 더블 클릭하면 압축이 풀린다. 잠시 후 SQL Server 설치 센터가 나타나면 [새 SQL Server 독립 실행형 설치 또는 기존 설치에 기능 추가]를 클릭한다.



그림 A-4 설치 시작

[사용 조건] 대화상자에서 [동의함]을 체크하고 <다음>을 누른다.

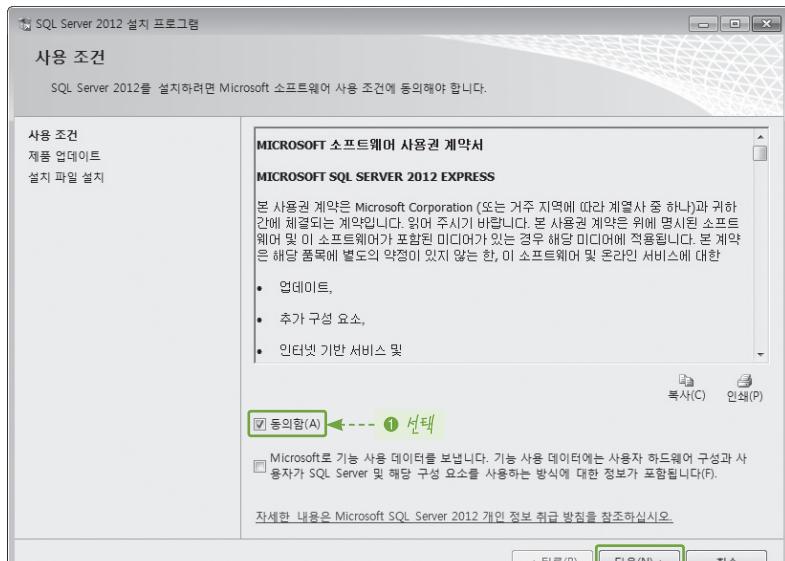


그림 A-5 사용 조건 동의

MS SQL 서버와 관련해 최신으로 업데이트할 필요가 있는 경우에는 다음과 같은 업데이트 설치 단계를 거치게 된다. 이 단계는 시스템 환경에 따라 업데이트 목록이 다르고, 업데이트가 필요 없는 경우에는 생략되기도 한다. 업데이트를 요청하는 창이 나타나면 <다음>을 눌러 업데이트를 설치한다.

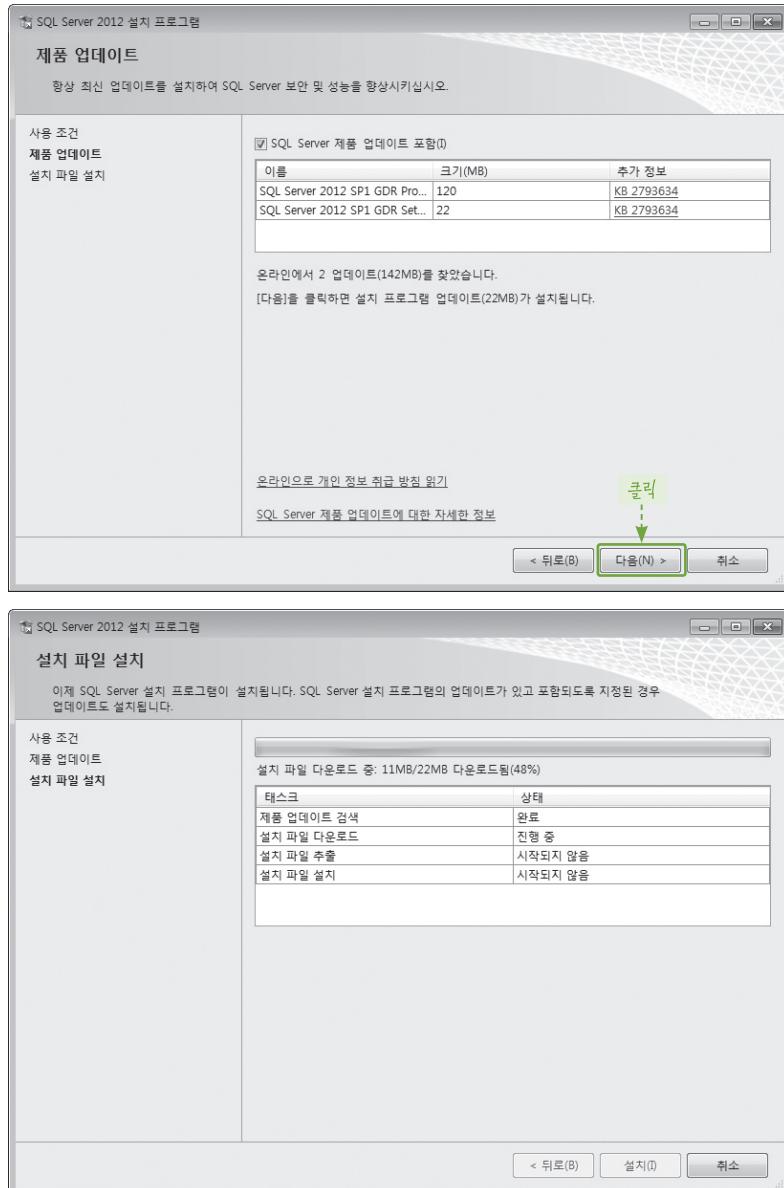


그림 A-6 업데이트 여부 확인 및 설치

[기능 선택] 대화상자에서 설치하고 싶은 기능을 지정한다. 기본으로 선택된 기능만으로도 설치하는 데 충분하므로 기본 설정으로 두고 <다음>을 누른다.

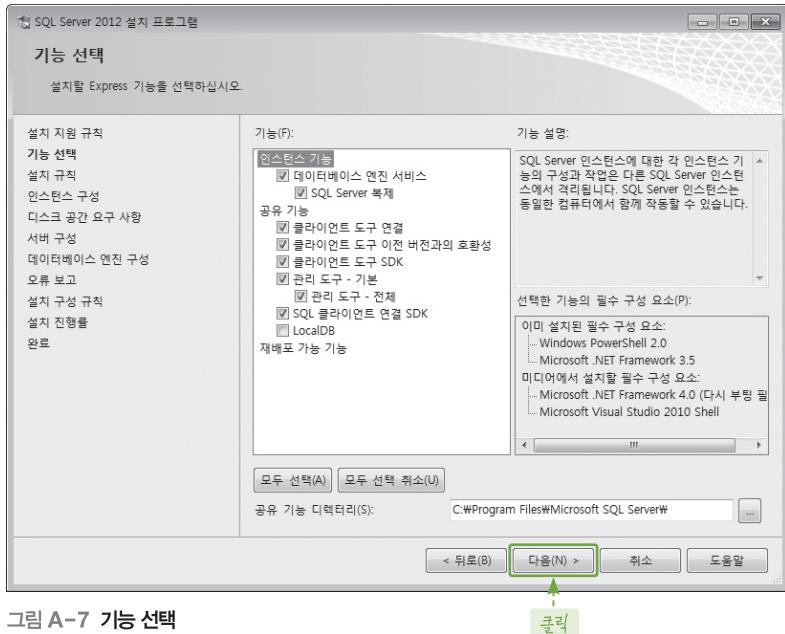


그림 A-7 기능 선택

비주얼 스튜디오 2010이 설치되어 있다면 Service Pack 1 이상이 적용되어 있어야 한다. '업데이트가 필요한 이전 Visual Studio 2010 인스턴스입니다.'라는 메시지와 함께 설치가 잠시 중단되면 안내 메시지를 참조하거나 Service Pack 1을 받아 설치한다.

[인스턴스 구성] 대화상자에서 [기본 인스턴스] 항목을 선택한다. [기본 인스턴스] 항목을 선택하면, 해당 인스턴스의 이름으로 자동 지정된다.



그림 A-8 인스턴스 구성

[서버 구성] 대화상자에서는 컴퓨터가 시작할 때 MS SQL 서버 관련 서비스를 실행할지 여부와 데이터 정렬 방식을 지정할 수 있다. 기본 설정으로 두고 <다음>을 누른다.

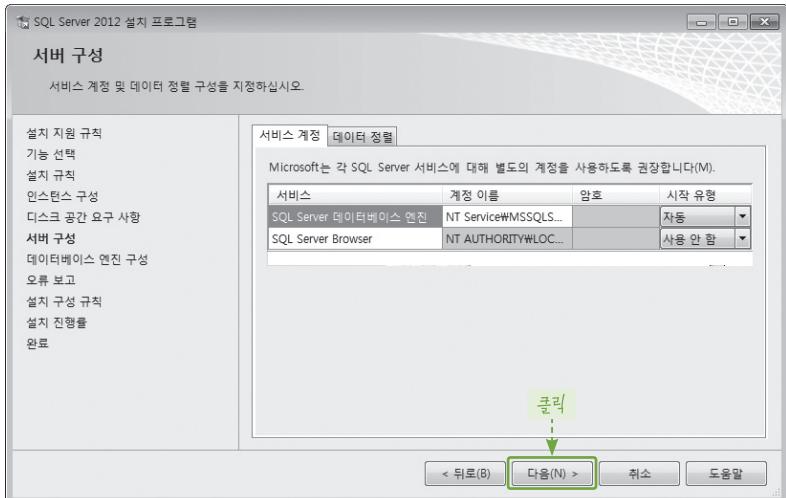


그림 A-9 서버 구성

[데이터베이스 엔진 구성]–[서버 구성] 탭에서 SQL 서버에 접속할 때 사용할 인증 모드와 SQL 서버 시스템 관리자의 암호를 지정한다. 인증 모드를 [Windows 인증 모드]로 선택한다. 원도우 사용자 계정으로 SQL 서버에 접속할 수 있도록 하고 <다음>을 누른다.

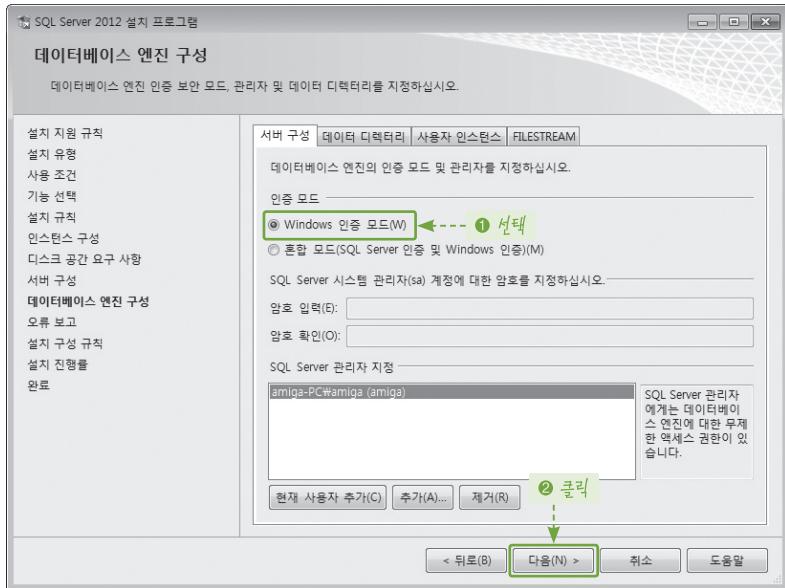


그림 A-10 데이터베이스 엔진 구성

MS SQL 서버에 다음 두 가지 인증 모드로 접속할 수 있다. 마이크로소프트 사에서는 원도우 운영체제가 제공하는 보안 기능을 그대로 사용할 수 있도록 Windows 인증 모드의 사용을 권장하지만, SQL 서버를 원격으로 사용할 때는 혼합 모드를 사용하는 것이 더 간편할 수도 있다. 개인 컴퓨터에 MS SQL 서버를 설치하여 간단한 예제 데이터베이스를 구축하고 활용하는 연습을 하는 데는 Windows 인증 방식이 간편하기 때문에 여기서는 설치 시 Windows 인증 모드를 선택하였다.

■ Windows 인증 모드를 선택했다고 해서 설치 후에 혼합 모드로 접속할 수 없는 것은 아니다. 설치 후에도 변경할 수 있다.

- **Windows 인증 모드** : MS SQL 서버가 설치된 윈도우 운영체제에 등록된 사용자 계정을 이용해 접속하는 방식이다. 별도의 사용자 이름이나 암호를 입력할 필요 없이 윈도우 운영체제의 사용자 계정으로 SQL 서버에 접속할 수 있다.
- **혼합 모드** : Windows 인증 모드도 사용하면서 SQL 서버에서만 사용되는 계정을 이용하는 SQL 서버 인증 방식도 함께 지원한다. 이 방식을 이용하면 SQL 서버에 등록된 사용자 이름과 암호를 입력하여 SQL 서버에 접속할 수 있다.

[오류 보고] 대화상자에서는 설치 시 발생한 오류 내용을 마이크로소프트 사에 보내도록 지정할 수 있다. 옵션을 선택하지 않아도 설치에 영향을 주지 않으므로 그대로 두고 <다음>을 누른다.

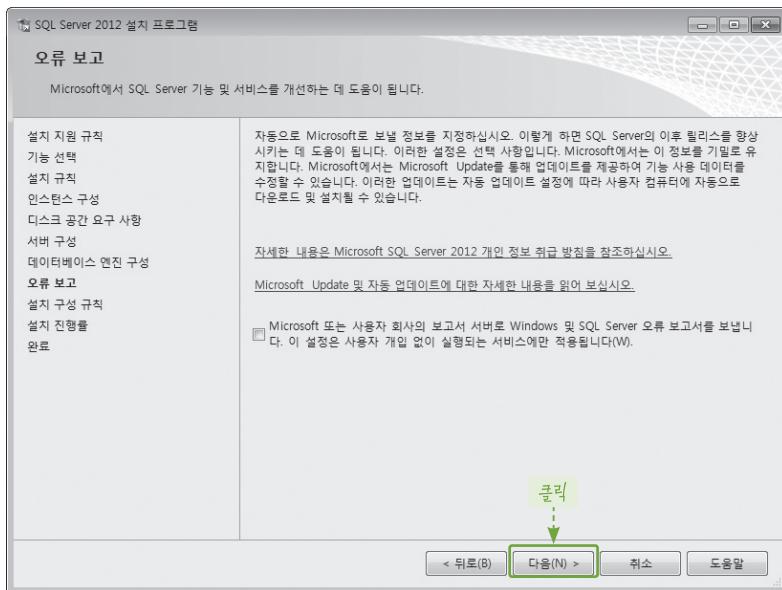


그림 A-11 오류 보고

본격적으로 설치가 진행된다. 설치가 완료되면 <닫기>를 눌러 설치를 종료한다.

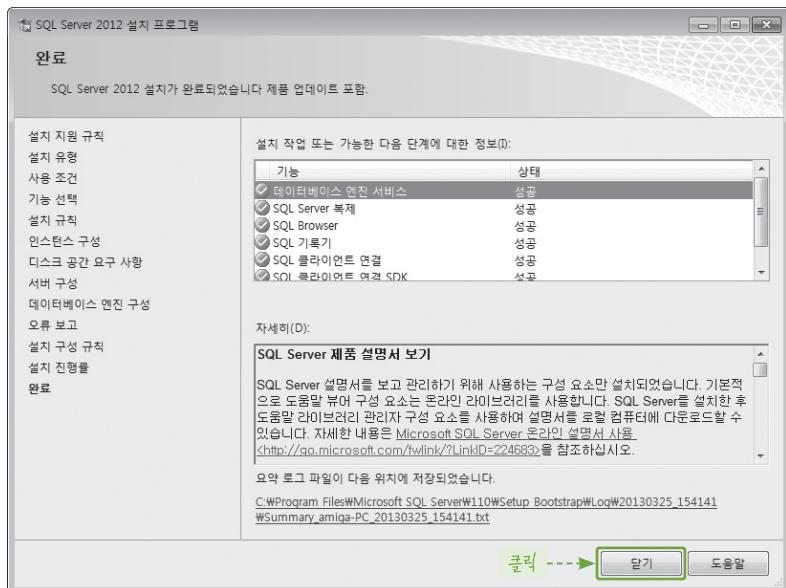
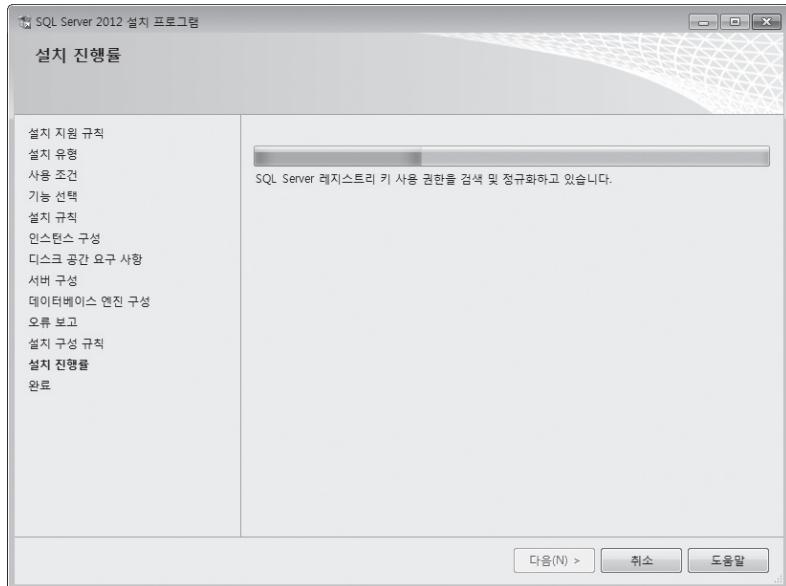


그림 A-12 설치 완료

설치 센터 화면은 수동으로 닫아준다.

2.2 MS SQL 서버의 설치 확인

MS SQL 서버가 정상적으로 설치되었는지는 [시작] – [제어판] – [시스템 및 보안] – [관리 도구] – [서비스] 메뉴를 차례로 실행한 후, 설치 과정 중에 지정한 인스턴스 이름의 SQL Server 서비스가 목록에 존재하는지를 확인하면 된다. 앞서 설치 과정에서 기본 인스턴스의 이름을 MSSQLSERVER로 설정하였으므로 SQL Server (MSSQLSERVER)라는 이름의 서비스가 목록에 존재할 것이다. 이 화면에서 MS SQL 서버를 시작하거나 중지시킬 수 있으며, 현재 상태도 확인할 수 있다.

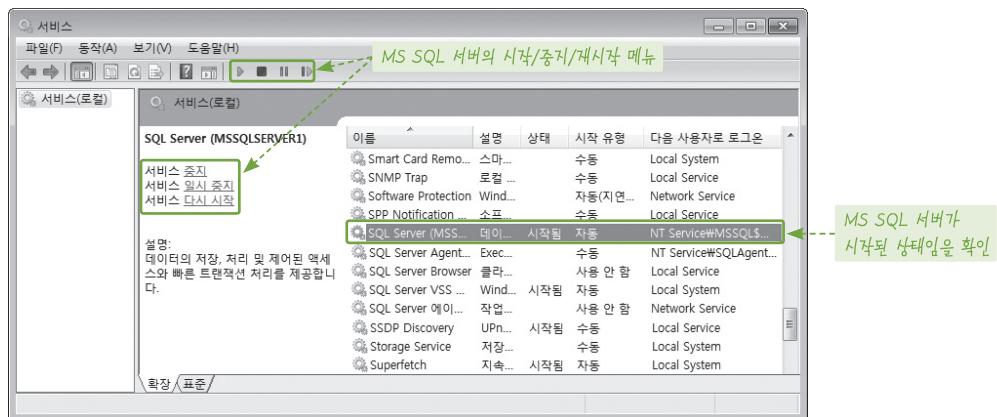


그림 A-13 MS SQL 서버의 설치 확인

MS SQL 서버를 이용한 데이터베이스 구축

3.1 MS SQL 서버 접속

설치한 MS SQL 서버 2012에 접속해서 테이블을 생성하고 데이터를 조작할 때는 SQL Server Management Studio(SSMS) 도구를 이용한다. SQL Server Management Studio는 SQL 문을 작성하고 실행할 수 있는 GUI 환경을 제공한다.

SQL Server Management Studio를 실행시키기 위해 [시작] – [모든 프로그램] – [Microsoft SQL Server 2012] – [SQL Server Management Studio] 메뉴를 차례로 누른다.

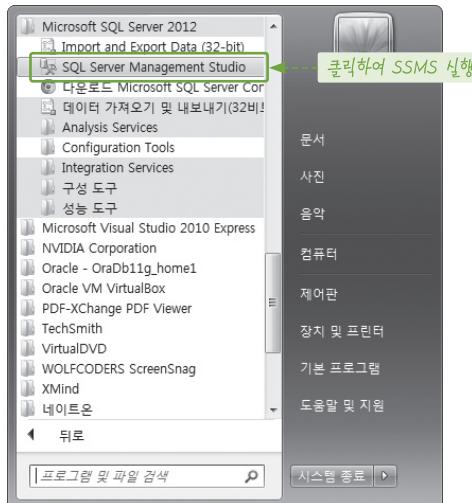


그림 A-14 SQL Server Management Studio 선택

[그림 A-15]와 같이 MS SQL 서버에 접속하기 위한 사용자 인증 창이 나타나면, 원도우 사용자 계정으로 접속할 수 있도록 인증 항목을 [Windows 인증]으로 선택하고 〈연결〉을 누른다. 운영체제와 상관없이 별도의 사용자 계정을 생성하여 접속해야 하는 오라클과 달리 MS SQL

서버는 두 가지 인증 방식 중 하나를 선택할 수 있다. MS SQL 서버가 제공하는 Windows 인증과 SQL Server 인증 방식 중 [Windows 인증] 방식을 선택하면 별도의 사용자 이름이나 암호를 입력할 필요 없이 윈도우 운영체제의 사용자로 MS SQL 서버에 접속할 수 있다. 여기서는 설치 시 선택한 [Windows 인증] 방식으로 접속한다.



그림 A-15 MS SQL 서버 2012에 로그인

MS SQL 서버에 접속이 성공하면 [그림 A-16]과 같이 SQL Server Management Studio 가 실행된다.

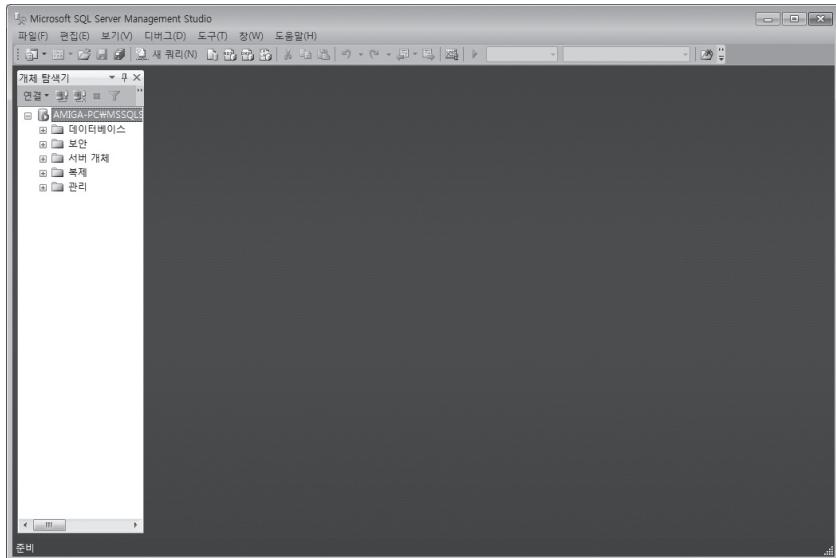


그림 A-16 SQL Server Management Studio의 시작 화면

3.2 데이터베이스의 생성

SQL Server Management Studio(이하 SSMS로 통일)를 이용해 새 데이터베이스 만들어 보자. SSMS의 화면 왼쪽 영역에 있는 [개체 탐색기]를 이용하면 실습에 사용할 예제 데이터베이스를 생성할 수 있다. [개체 탐색기]의 [데이터베이스] 폴더에서 마우스 오른쪽 버튼을 누른 다음 [새 데이터베이스] 메뉴를 선택한다.

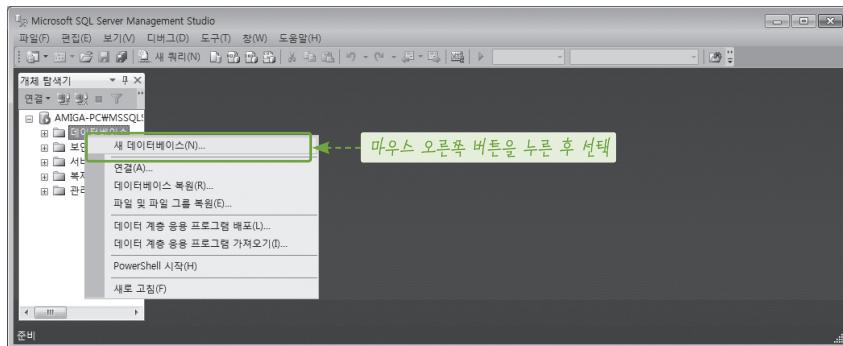


그림 A-17 새 데이터베이스 메뉴 선택

[새 데이터베이스] 대화상자의 [데이터베이스 이름] 항목에 원하는 데이터베이스의 이름을 입력하고 <다음>을 누른다. 여기서는 데이터베이스 이름에 “SALES”를 입력했다.

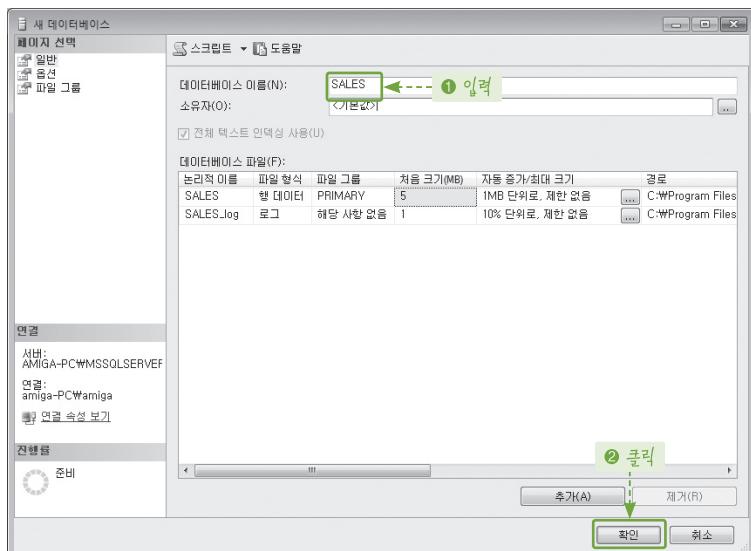


그림 A-18 데이터베이스 이름 입력

데이터베이스를 생성한 다음 [개체 탐색기]에서 [데이터베이스] 폴더를 펼쳐보면 SALES 데이터베이스가 생성된 것을 확인할 수 있다.

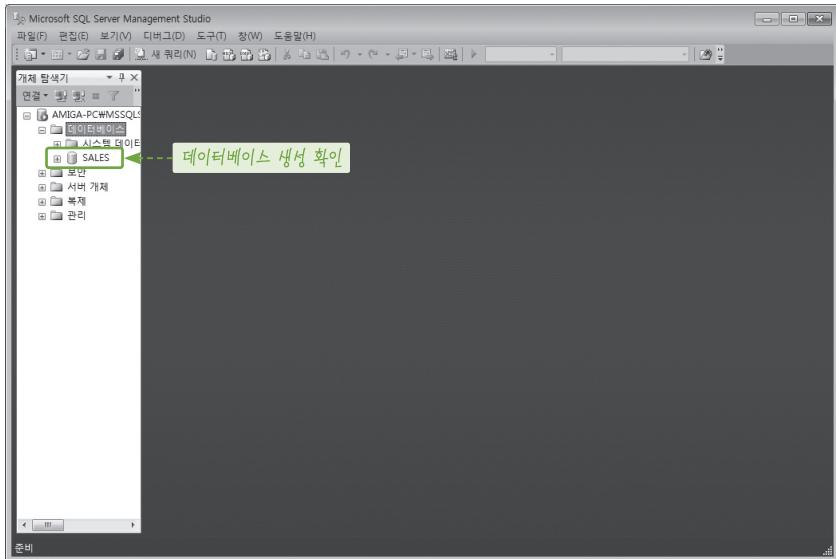


그림 A-19 데이터베이스 생성 확인

3.3 테이블의 생성

새로 만든 데이터베이스에 테이블을 생성하고 수정하는 간단한 실습을 수행해보자.

3.3.1 생성할 테이블과 SQL 문

실습에서 생성할 테이블은 7장에서 활용한 [그림 A-20] 고객, 제품, 주문 테이블이다.

고객 테이블

<u>고객아이디</u>	고객이름	나이	등급	직업	적립금
apple	정소화	20	gold	학생	1000
banana	김선우	25	vip	간호사	2500
carrot	고명석	28	gold	교사	4500
orange	김용욱	22	silver	학생	0
melon	성원용	35	gold	회사원	5000
pear	채광주	31	silver	회사원	500
peach	오형준	NULL	silver	의사	300

제품 테이블

<u>제품번호</u>	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운쫄면	2500	5500	민국푸드
p03	쿵떡파이	3600	2600	한빛제과
p04	만나초콜렛	1200	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	통통우동	1000	1550	민국푸드
p07	달콤비스켓	1650	1500	한빛제과

주문 테이블

<u>주문번호</u>	고객아이디	주문제품	수량	배송지	주문일자
o01	apple	p03	10	서울시 마포구	2013-01-01
o02	melon	p01	5	인천시 계양구	2013-01-10
o03	banana	p06	45	경기도 부천시	2013-01-11
o04	carrot	p02	8	부산시 금정구	2013-02-01
o05	melon	p06	36	경기도 용인시	2013-02-20
o06	banana	p01	19	충청북도 보은군	2013-03-02
o07	apple	p03	22	서울시 영등포구	2013-03-15
o08	pear	p02	50	강원도 춘천시	2013-04-10
o09	banana	p04	15	전라남도 목포시	2013-04-11
o10	carrot	p03	20	경기도 안양시	2013-05-22

그림 A-20 실습에서 생성할 테이블 : 고객, 제품, 주문 테이블

고객, 제품, 주문 테이블을 생성하는 SQL 문은 다음과 같다.

표 A-1 고객, 제품, 주문 테이블을 생성하는 SQL 문

테이블	테이블 생성 SQL 문
고객	<pre>CREATE TABLE 고객 (고객아이디 VARCHAR(20) NOT NULL, 고객이름 VARCHAR(10) NOT NULL, 나이 INT, 등급 VARCHAR(10) NOT NULL, 직업 VARCHAR(20), 적립금 INT DEFAULT 0, PRIMARY KEY(고객아이디));</pre>
제품	<pre>CREATE TABLE 제품 (제품번호 CHAR(3) NOT NULL, 제품명 VARCHAR(20), 재고량 INT, 단가 INT, 제조업체 VARCHAR(20), PRIMARY KEY(제품번호), CHECK (재고량 >= 0 AND 재고량 <=10000));</pre>
주문	<pre>CREATE TABLE 주문 (주문번호 CHAR(3) NOT NULL, 주문고객 VARCHAR(20), 주문제품 CHAR(3), 수량 INT, 배송지 VARCHAR(30), 주문일자 DATETIME, PRIMARY KEY(주문번호), FOREIGN KEY(주문고객) REFERENCES 고객(고객아이디), FOREIGN KEY(주문제품) REFERENCES 제품(제품번호));</pre>

▣ 기본키로 지정한 속성은 굳이 NOT NULL을 표기하지 않아도 자동으로 NOT NULL 특성을 갖지만, 기본키로 사용할 속성은 널 값을 가질 수 없다는 제약조건을 더 명확히 표현하기 위해 NOT NULL을 표기했다.

3.3.2 SSMS를 이용한 테이블 생성

새로운 테이블을 생성하려면 먼저 데이터베이스를 선택해야 한다. SSMS에서 [개체 탐색기] – [데이터베이스] – [SALES] 데이터베이스를 선택한다. 테이블 생성을 위한 SQL 문을 입력하기 위해 [새 쿼리] 메뉴를 선택하면 화면 오른쪽 영역에 쿼리 입력 창이 나타난다. 쿼리 입력 창에 고객 테이블을 생성하는 SQL 문을 입력한 다음 [실행] 메뉴를 눌러 실행시킨다.

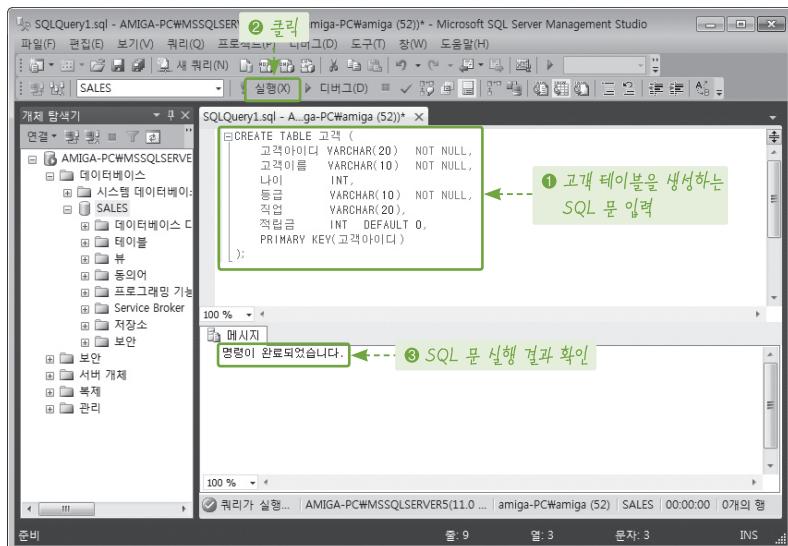


그림 A-21 고객 테이블 생성

입력한 SQL 문에 오류가 없음에도 빨간 밑줄이 생기는 경우가 있는데 이는 SSMS에 도입된 IntelliSense 기능 때문이다. IntelliSense 는 SSMS에서 SQL 문을 작성할 때 테이블이나 속성의 목록이 자동으로 나타나게 하여 선택할 수 있도록 하는 자동 완성 기능으로, 테이블이나 속성의 이름을 쉽게 입력할 수 있도록 해준다. 그리고 작성된 SQL 문에 오류가 있으면 빨간 밑줄을 그어 알려준다. 그런데 캐시 메모리의 업데이트 속도가 느려 입력한 SQL 문에 오류가 없는데도 빨간 밑줄이 나타나기도 한다. 이 문제를 해결하려면 SSMS 의 [편집] – [IntelliSense] 메뉴를 선택한 다음 [로컬 캐시 새로 고침] 메뉴를 실행하여 캐시 메모리를 업데이트시켜 주면 된다.

같은 방법으로 제품, 주문 테이블을 생성한 후 [개체 탐색기] – [SALES] – [테이블] 폴더를 클릭하면 생성한 고객, 제품, 주문 테이블을 확인할 수 있다. 만약 생성된 테이블이 보이지 않으면 [테이블] 폴더에서 마우스 오른쪽 버튼으로 누른 다음 [새로 고침] 메뉴를 선택하여 생성된 테이블을 확인할 수 있다. [테이블] 폴더에 있는 테이블들은 이름 앞에 테이블을 생성한 소유자가 표시되는데, 따로 지정하지 않으면 일반적으로 dbo가 표시된다.

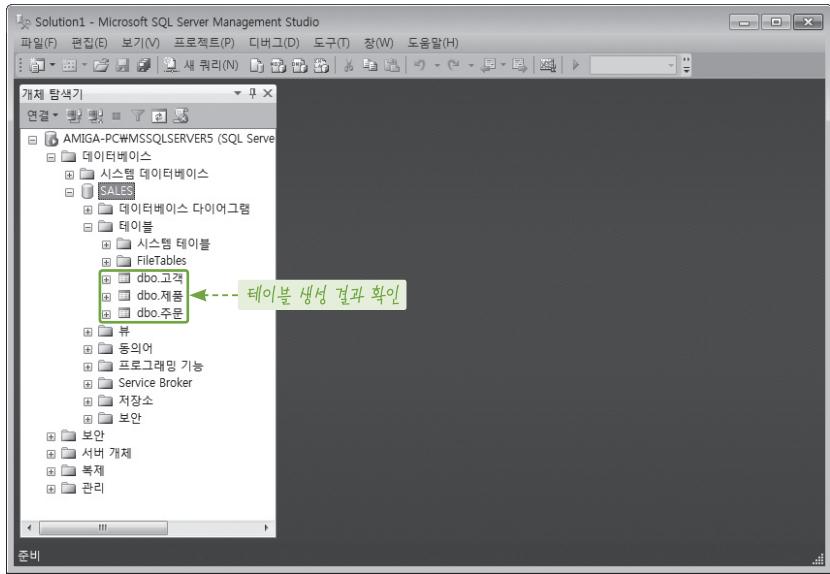


그림 A-22 고객, 제품, 주문 테이블 생성 결과 확인

▣ 고객, 제품, 주문 테이블을 생성하기 위한 세 개의 SQL 문을 모두 입력하고 한꺼번에 실행하여, 테이블 세 개를 한번에 생성할 수도 있다.

Tip



MS SQL 서버와 오라클의 데이터 타입

테이블을 생성할 때 사용하는 데이터 타입의 표현이 MS SQL 서버와 오라클이 약간 다른 경우도 있으니 주의해야 한다. 부록에서 제시하는 예제 데이터베이스를 구축하고 활용할 때는 자주 사용되는 다음 두 가지 데이터 타입의 차이를 기억해두는 것이 좋다.

- MS SQL 서버에서는 가변 길이 문자열을 VARCHAR로 표현하지만 오라클에서는 VARCHAR2로 주로 표현한다. 오라클에서도 VARCHAR를 모두 사용할 수 있지만 성능이 좀 더 개선된 VARCHAR2를 더 많이 사용한다.
- MS SQL 서버에서는 날짜와 시간을 DATETIME으로 표현하지만 오라클에서는 DATE로 표현한다.

여기서 다루는 예제는 굳이 성능이 좋은 데이터 타입을 선택해야 할 수준이 아니므로 일반적인 가변 길이 문자열은 VARCHAR로 통일하고, 주문 테이블을 생성할 때 주문일자만 MS SQL 서버에서는 DATETIME으로, 오라클에서는 DATE로 사용할 것이다.

04

MS SQL 서버를 이용한 데이터베이스 활용

앞에서 생성한 세 개의 테이블에 새로운 데이터를 입력하고, 필요한 데이터를 검색하는 간단한 실습을 SSMS를 이용해 수행해보자.

4.1 데이터의 입력

고객, 제품, 주문 테이블에 새로운 투플을 입력하는 SQL 문을 작성하고, 이를 실행한 결과를 살펴보자. 각 테이블별로 투플을 입력하기 위한 INSERT 문을 SQL 입력창에 여러 개 작성한 후 한꺼번에 실행시켜 결과를 확인하였다.

① 고객 테이블에 투플 7개 삽입하기

The screenshot shows the Microsoft SQL Server Management Studio interface. A query window titled 'SQLQuery3.sql - A - ga-PC\amiga (54)*' is open, displaying the following SQL code:

```
INSERT INTO 고객 VALUES ('apple', '정소화', 20, 'gold', '학생', 1000);
INSERT INTO 고객 VALUES ('banana', '김선우', 25, 'vip', '간호사', 2500);
INSERT INTO 고객 VALUES ('carrot', '고명석', 28, 'gold', '교사', 4500);
INSERT INTO 고객 VALUES ('orange', '김용욱', 22, 'silver', '학생', 0);
INSERT INTO 고객 VALUES ('melon', '성원용', 35, 'gold', '회사원', 5000);
INSERT INTO 고객 VALUES ('pear', '채광주', 31, 'silver', '회사원', 500);
INSERT INTO 고객 VALUES ('peach', '오현준', NULL, 'silver', '의사', 300);
```

A tooltip is displayed over the first line of the code, showing the Korean translation of the columns:

(1개 행이 영향을 받음)

그림 A-23 고객 테이블에 투플 7개 삽입

② 제품 테이블에 투플 7개 삽입하기

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the database structure under the 'AMIGA-PC\AMIGA' server, specifically the 'SALES' database. The right pane shows the results of a T-SQL script named 'SQLQuery3.sql'. The script contains seven INSERT statements for the '제품' (Product) table:

```

INSERT INTO 제품 VALUES ('p01', '그냥만두', 5000, 4500, '대한식품');
INSERT INTO 제품 VALUES ('p02', '매운쫄면', 2500, 5500, '민국푸드');
INSERT INTO 제품 VALUES ('p03', '쫑떡파이', 3600, 2600, '한빛제과');
INSERT INTO 제품 VALUES ('p04', '맛난초콜렛', 1250, 2500, '한빛제과');
INSERT INTO 제품 VALUES ('p05', '얼큰라면', 2200, 1200, '대한식품');
INSERT INTO 제품 VALUES ('p06', '통통우동', 1000, 1550, '민국푸드');
INSERT INTO 제품 VALUES ('p07', '달콤비스켓', 1650, 1500, '한빛제과');

```

The message window at the bottom indicates that all 7 rows were inserted successfully.

그림 A-24 제품 테이블에 투플 7개 삽입

③ 주문 테이블에 투플 10개 삽입하기

The screenshot shows the Microsoft SQL Server Management Studio interface. The left pane displays the database structure under the 'AMIGA-PC\AMIGA' server, specifically the 'SALES' database. The right pane shows the results of a T-SQL script named 'SQLQuery3.sql'. The script contains ten INSERT statements for the '주문' (Order) table:

```

INSERT INTO 주문 VALUES ('o01', 'apple', 'p03', 10, '서울시 마포구', '13/01/01');
INSERT INTO 주문 VALUES ('o02', 'melon', 'p01', 5, '인천시 계양구', '13/01/10');
INSERT INTO 주문 VALUES ('o03', 'banana', 'p06', 45, '경기도 부천시', '13/01/11');
INSERT INTO 주문 VALUES ('o04', 'carrot', 'p02', 8, '부산시 금정구', '13/02/01');
INSERT INTO 주문 VALUES ('o05', 'melon', 'p06', 36, '경기도 용인시', '13/02/20');
INSERT INTO 주문 VALUES ('o06', 'banana', 'p01', 19, '충청북도 보은군', '13/03/02');
INSERT INTO 주문 VALUES ('o07', 'apple', 'p03', 22, '서울시 영등포구', '13/03/15');
INSERT INTO 주문 VALUES ('o08', 'pear', 'p02', 50, '강원도 춘천시', '13/04/10');
INSERT INTO 주문 VALUES ('o09', 'banana', 'p04', 15, '전라남도 목포시', '13/04/11');
INSERT INTO 주문 VALUES ('o10', 'carrot', 'p03', 20, '경기도 안양시', '13/05/22');

```

The message window at the bottom indicates that all 10 rows were inserted successfully.

그림 A-25 주문 테이블에 투플 10개 삽입

4.2 데이터의 검색

제품 테이블에서 모든 투플을 검색하는 SQL 문과, 주문 테이블에서 고객별 주문수량의 평균을 검색하는 SQL 문을 작성하여 실행한 결과를 살펴보자.

① 제품 테이블의 모든 투플 검색하기

The screenshot shows the SQL Server Management Studio interface. A query window titled 'SQLQuery3.sql' contains the SQL statement: 'SELECT * FROM 제품;'. The results pane displays a table with 7 rows of product data:

제품번호	제품명	제고량	단가	제조업체
p01	그날만두	5000	4500	대한식품
p02	매운풀면	2500	5500	한국푸드
p03	쫄띠파이	3600	2600	한빛제과
p04	맛난초콜렛	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품
p06	事业发展	1000	1550	한국푸드
p07	달콤비스켓	1650	1500	한빛제과

그림 A-26 제품 테이블의 모든 투플 검색

② 주문 테이블에서 고객별 주문수량의 평균 구하기

The screenshot shows the SQL Server Management Studio interface. A query window titled 'SQLQuery3.sql' contains the SQL statement: 'SELECT 주문고객, AVG(수량) AS 평균주문수량 FROM 주문 GROUP BY 주문고객;'. The results pane displays a table with 5 rows of customer data:

주문고객	평균주문수량
apple	16
banana	26
carrot	14
melon	20
pear	50

그림 A-27 주문 테이블에서 고객별 주문수량 평균 검색

4.3 데이터의 수정 및 삭제

제품 테이블에서 투플의 내용을 수정하는 SQL 문과, 주문 테이블에서 투플을 삭제하는 SQL 문을 작성하여 실행한 결과를 살펴보자.

① 제품 테이블에서 한빛제과의 모든 제품 가격을 10% 인상하고 확인하기

The screenshot shows the SQL Server Management Studio interface. In the center pane, there is a results grid displaying product data from the SALES database. The columns are: 제품번호 (Product ID), 제품명 (Product Name), 재고량 (Stock Quantity), 단가 (Unit Price), and 제조업체 (Manufacturer). The data includes products like p01 (그남만두), p02 (매운쫄면), p03 (쫑떡파이), etc. An inset window shows the SQL query being run:

```
UPDATE 제품 SET 단가 = 단가 * 1.1 WHERE 제조업체 = '한빛제과';
SELECT * FROM 제품;
```

The results grid shows the updated unit price for the products belonging to '한빛제과'.

그림 A-28 한빛제과의 제품 가격을 10% 인상

② 주문 테이블에서 아이디가 apple인 고객이 주문한 내역 삭제하고 확인하기

The screenshot shows the SQL Server Management Studio interface. In the center pane, there is a results grid displaying order data from the SALES database. The columns are: 주문번호 (Order ID), 주문고객 (Customer ID), 주문제품 (Product ID), 수령 (Delivery Address), 배송지 (Delivery Location), and 주문일자 (Order Date). An inset window shows the SQL query being run:

```
DELETE FROM 주문 WHERE 주문고객 = 'apple';
SELECT * FROM 주문;
```

The results grid shows the deleted order entries for customer 'apple'.

그림 A-29 아이디가 apple 고객의 주문 내역 삭제