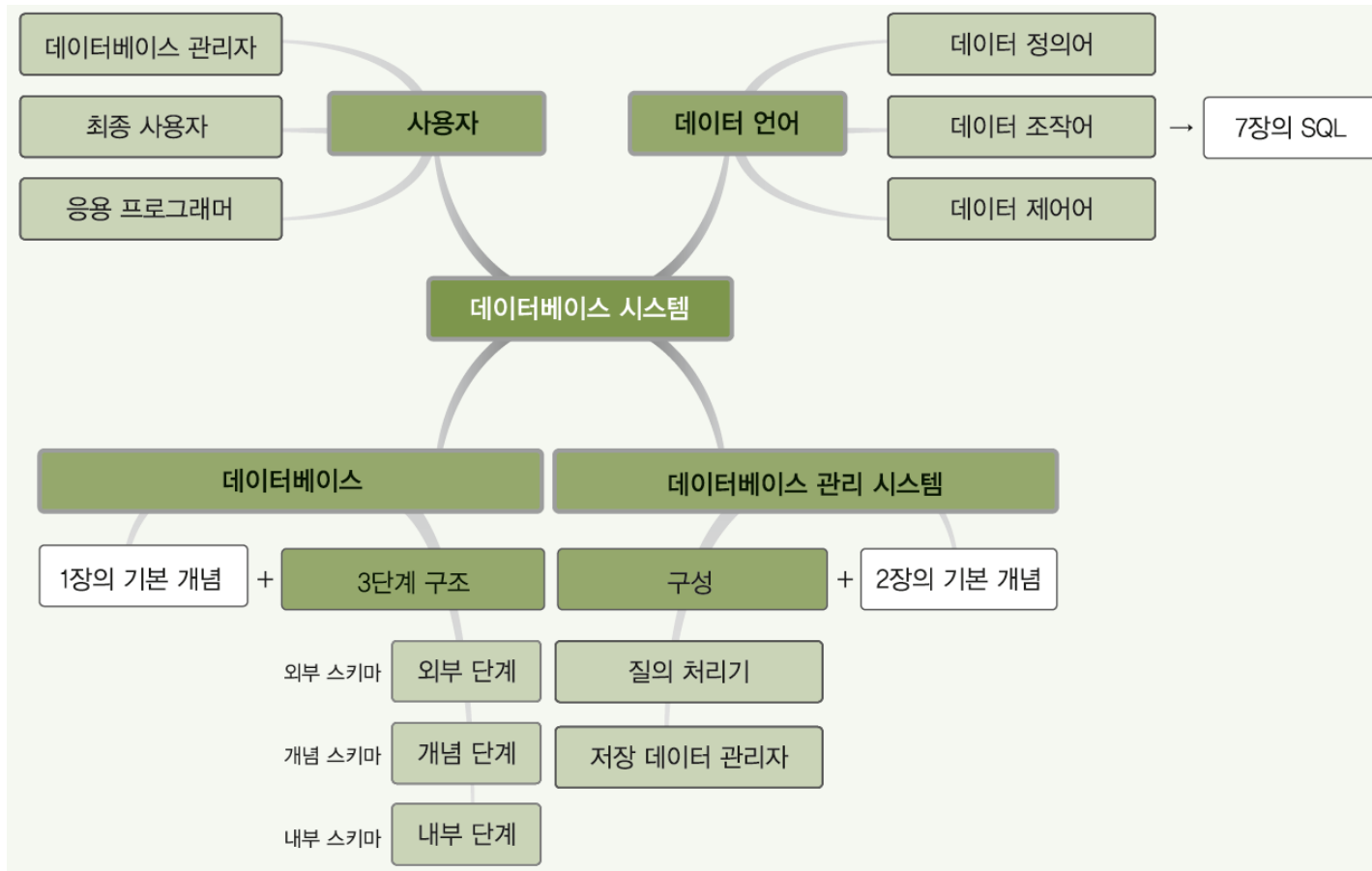




3장. 데이터베이스 시스템

- 데이터베이스 시스템의 정의
- 데이터베이스의 구조
- 데이터베이스 사용자
- 데이터 언어
- 데이터베이스 관리 시스템의 구성

학습목표



- ▶ 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(DBS; DataBase System)

- 데이터베이스에 데이터를 저장하고, 이를 관리하여 조직에 필요한 정보를 생성해주는 시스템

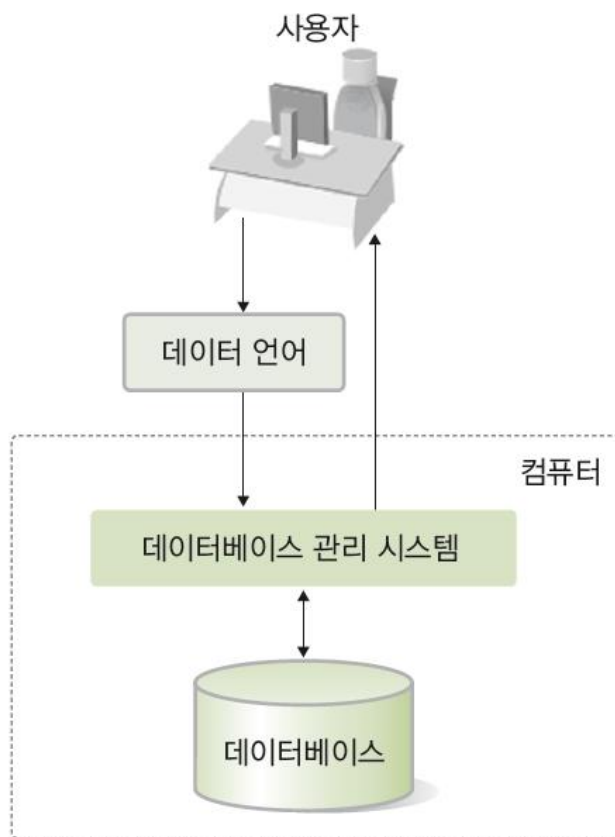


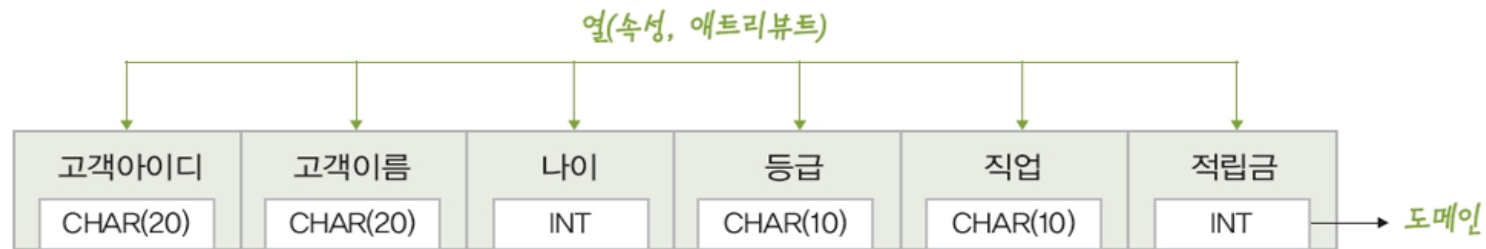
그림 3-1 데이터베이스 시스템의 구성

02 데이터베이스의 구조

❖ 스키마와 인스턴스

- 스키마(schema) (뜻 : 지식의 추상적 구조)

- 데이터베이스에 저장되는 데이터 구조와 제약조건을 정의한 것



행(튜플)

apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

- 인스턴스(instance)

- 스키마에 따라 데이터베이스에 실제로 저장된 값

02 데이터베이스의 구조

❖ 3단계 데이터베이스 구조

- 미국 표준화 기관인 ANSI/SPARC에서 제안
- 데이터베이스를 쉽게 이해하고 이용할 수 있도록 하나의 데이터베이스를 관점에 따라 세 단계로 나눈 것
 - 외부 단계(external level) : 개별 사용자 관점
 - 개념 단계(conceptual level) : 조직 전체의 관점, 관리자의 관점
 - 내부 단계(internal level) : 물리적인 저장 장치의 관점
- 각 단계별로 다른 추상화(abstraction) 제공
 - 내부 단계에서 외부 단계로 갈수록 추상화 레벨이 높아짐
- 이렇게 DB구조를 3단계로 나누어 설계하는 이유???
- 데이터의 독립성 확보

02 데이터베이스의 구조

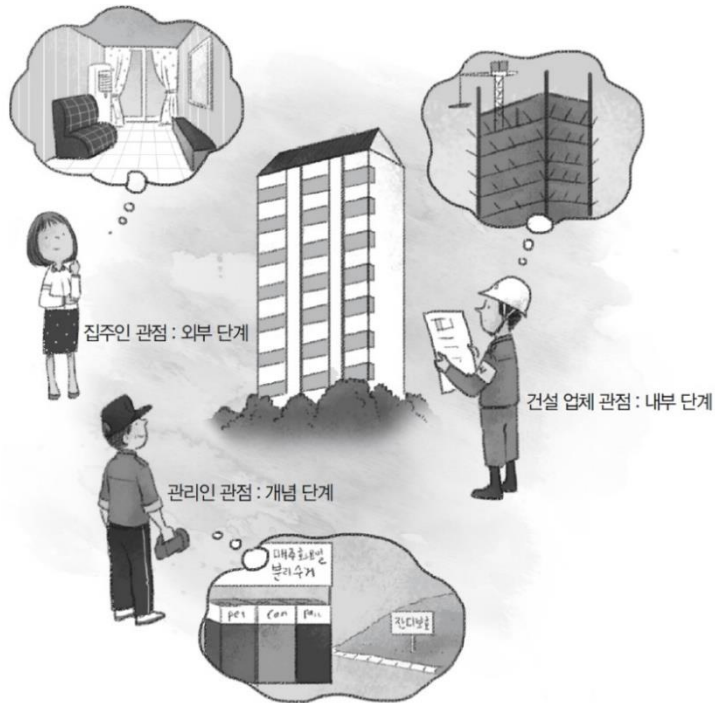
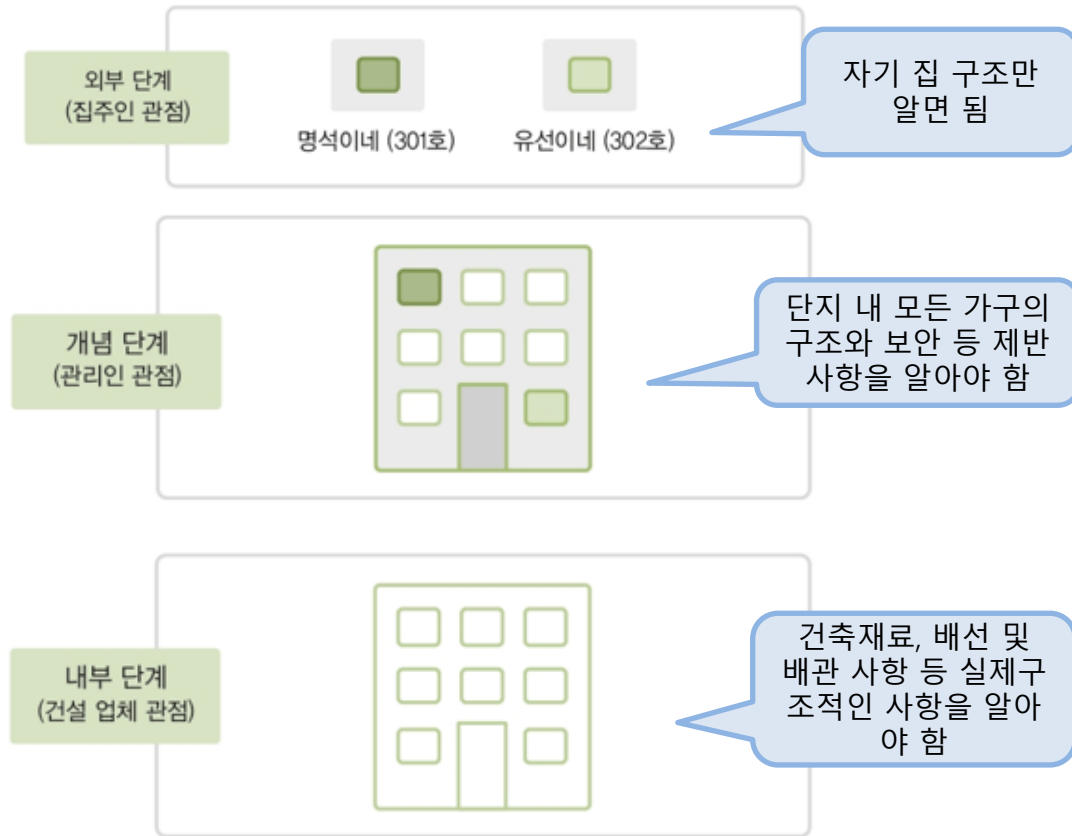


그림 3-4 3단계 데이터베이스 구조의 이해



02 데이터베이스의 구조(전철노선의 예)

5호선 종로3가 → 6호선 효창공원앞

이용정보

5호선 종로3가 17:17 소요시간 19분
운임구간 6가구간
6호선 효창공원앞 17:37 환승횟수 1회
요금정보 1,250원(카드)
1,350원(현금)

역차 시간표

역명	종로3가	시간
평일	00:13:30 (방화철)	
토요일	23:24:30 (방화철)	
공휴일	23:24:30 (방화철)	
평일	00:40:30 (한강전철)	
토요일	23:51:50 (한강전철)	
공휴일	23:51:50 (한강전철)	
평일	00:41:50 (한강전철)	
토요일	23:53:10 (한강전철)	
공휴일	23:53:10 (한강전철)	



5호선 화곡 → 2호선 홍대입구

이용정보

5호선 화곡 17:14 소요시간 27분
운임구간 9가구간
2호선 홍대입구 17:43 환승횟수 1회
요금정보 1,350원(카드)
1,450원(현금)

역차 시간표

역명	화곡	시간
평일	00:28:50 (여의가철)	
토요일	23:22:50 (홍익리철)	
공휴일	23:22:50 (홍익리철)	
평일	00:53:30 (홍대입구철)	
토요일	23:54:30 (홍대입구철)	
공휴일	23:54:30 (홍대입구철)	
평일		
토요일		
공휴일		
평일		
토요일		
공휴일		

외부단계



개념단계

02 데이터베이스의 구조(외부단계)

❖ 외부 단계 (외부 스키마)

- 데이터베이스를 개별 사용자 관점에서 이해하고 표현하는 단계
- 데이터베이스 하나에 외부 스키마가 여러 개 존재할 수 있음
 - 고객관리 직원과 상품관리 직원이 바라보는 데이터베이스는 다름
- 외부 스키마(external schema)
 - 외부 단계에서 사용자에게 필요한 데이터베이스를 정의한 것
 - 각 사용자가 생각하는 데이터베이스의 모습, 즉 논리적 구조로 사용자마다 다름
 - 서브 스키마(sub schema)라고도 함
- 외부스키마의 예 (영화예매시스템)
 - 관리자
 - » 상영관 설정, 영화등록, 영화시간표 등록, 회원관리 등
 - 회원
 - » 영화검색, 예매, 결제, 발권 등

02 데이터베이스의 구조(개념단계)

❖ 개념 단계 (개념 스키마)

- 데이터베이스를 조직 **전체의 관점**에서 이해하고 표현하는 단계
- 데이터베이스 하나에 개념 스키마가 **하나**만 존재함
 - 개념 스키마(conceptual schema)
 - 개념 단계에서 데이터베이스 전체의 논리적 구조를 정의한 것
 - 조직 전체의 관점에서 생각하는 데이터베이스의 모습
 - 전체 데이터베이스에 **어떤 데이터**가 저장되는지, 데이터들 간에는 어떤 **관계**가 존재하고 어떤 **제약조건**이 존재하는지에 대한 정의뿐만 아니라, 데이터에 대한 **보안** 정책이나 **접근 권한**에 대한 정의도 포함

개념스키마를 정의 하는 것이 DB설계의 핵심

02 데이터베이스의 구조(내부단계)

❖ 내부 단계 (내부 스키마)

- 데이터베이스를 **저장 장치의 관점**에서 이해하고 표현하는 단계
- 데이터베이스 하나에 내부 스키마가 하나만 존재함
 - 내부 스키마(internal schema)
 - 전체 데이터베이스가 저장 장치에 실제로 저장되는 방법을 정의한 것
 - 레코드 구조, 필드 크기, 레코드 접근 경로 등 물리적 저장 구조를 정의
 - 대부분의 내부스키마는 DDL에 의해 구현됨

02 데이터베이스의 구조

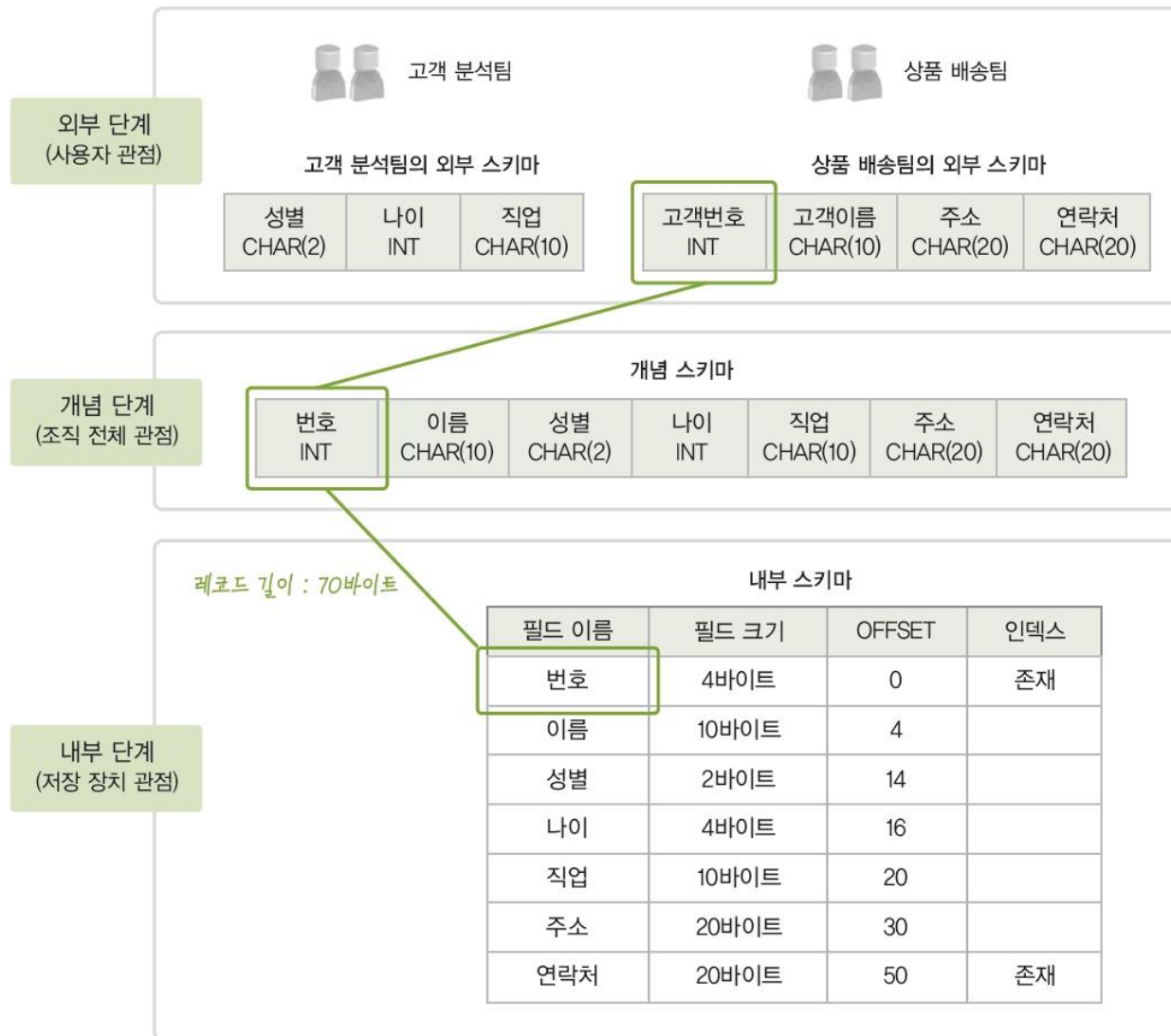


그림 3-5 3단계 데이터베이스 구조의 예

02 데이터베이스의 구조

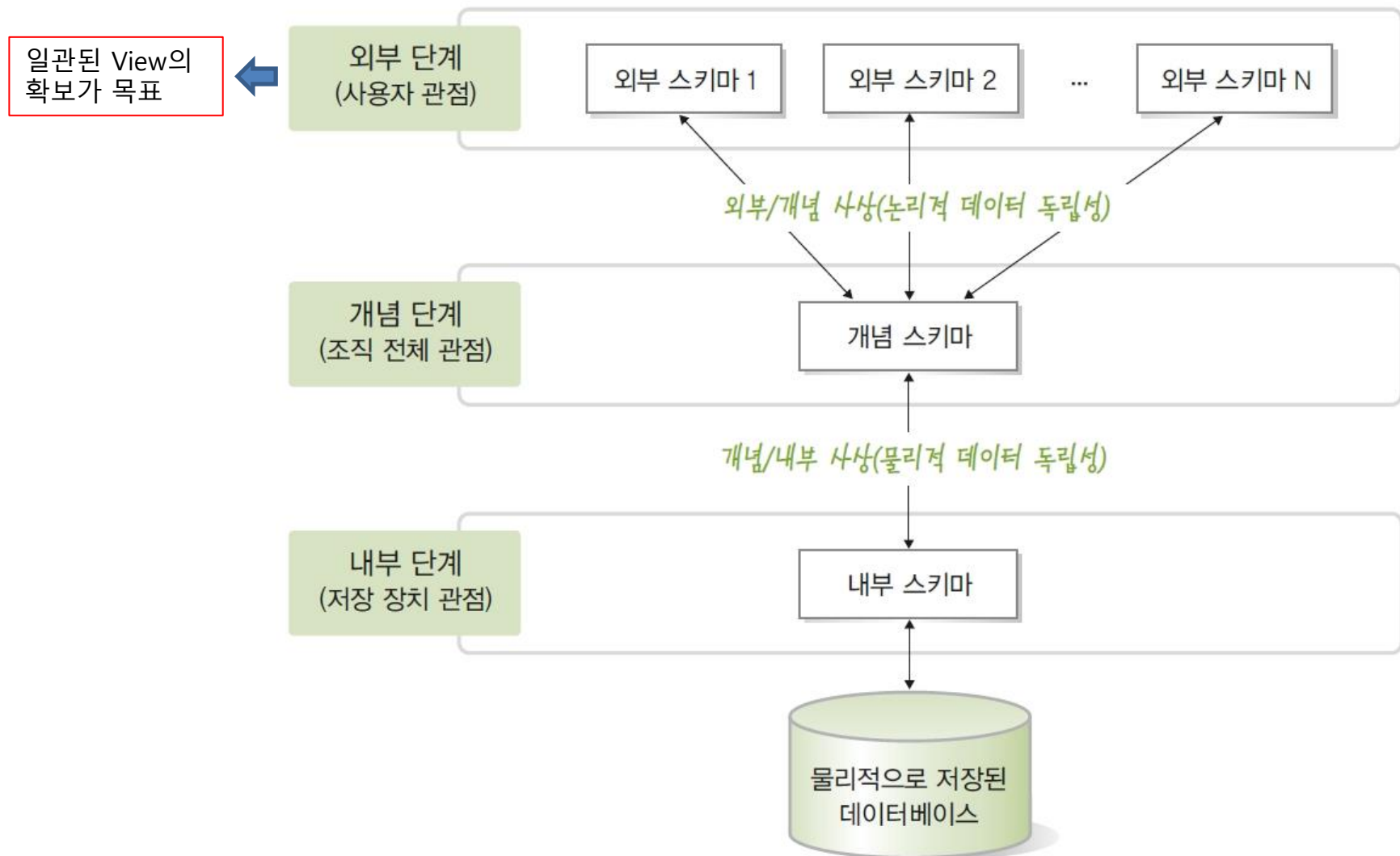


그림 3-6 3단계 데이터베이스 구조에서 스키마 간의 사상

02 데이터베이스의 구조

❖ 3단계 데이터베이스 구조의 사상 또는 매핑

- 스키마 사이의 대응 관계
 - 외부/개념 사상 : 외부 스키마와 개념 스키마의 대응 관계
 - 응용 인터페이스(application interface)라고도 함
 - 개념/내부 사상 : 개념 스키마와 내부 스키마의 대응 관계
 - 저장 인터페이스(storage interface)라고도 함
- 미리 정의된 사상 정보를 이용하여 사용자가 원하는 데이터에 접근

데이터베이스를 3단계 구조로 나누고 단계별로 스키마를 유지하며
스키마 사이의 대응 관계를 정의하는 궁극적인 목적

➔ **데이터 독립성의 실현**

02 데이터베이스의 구조

❖ 데이터 독립성(data independency)

- 하위 스키마를 변경하더라도 상위 스키마가 영향을 받지 않는 특성
 - 즉, 일관된 **뷰**의 확보가 매우 중요함
- **논리적 데이터 독립성**
 - 개념 스키마가 변경되어도 외부 스키마는 영향을 받지 않음
 - 즉, 응용프로그램에 영향을 주지 않음
 - 개념 스키마가 변경되면 관련된 외부/개념 사상만 정확하게 수정해주면 됨
- **물리적 데이터 독립성**
 - 내부 스키마가 변경되어도 개념 스키마는 영향을 받지 않음
 - 응용 프로그램에 영향을 주지 않고, DB의 물리적 구조를 변경 시킬 수 있는 성질

02 데이터베이스의 구조

❖ 데이터 사전(data dictionary)

- 시스템 카탈로그(system catalog)라고도 함
- 데이터베이스에 저장되는 데이터에 관한 정보, 즉 메타 데이터를 유지하는 시스템 데이터베이스
 - 메타 데이터(meta data) : 데이터에 대한 데이터
- 스키마, 사상 정보, 다양한 제약조건 등을 저장
- 데이터베이스 관리 시스템이 스스로 생성하고 유지함
- 일반 사용자도 접근이 가능하지만 저장된 내용을 검색만 함.

02 데이터베이스의 구조

❖ 데이터 디렉토리(data directory)

- 데이터 사전에 있는 데이터에 실제로 접근하는 데 필요한 위치 정보를 저장하는 시스템 데이터베이스
 - 시스템테이블을 저장하는 곳
 - 일반 사용자의 접근은 허용되지 않음
 - 관리자 권한으로 생성 /삭제가 가능함

❖ 사용자 데이터베이스(user database)

- 사용자가 실제로 이용하는 데이터가 저장되어 있는 일반 데이터베이스

03 데이터베이스 사용자

❖ 데이터베이스 사용자

- 데이터베이스를 이용하기 위해 접근하는 모든 사람
- 이용 목적에 따라 데이터베이스 관리자, 최종 사용자, 응용 프로그래머로 구분

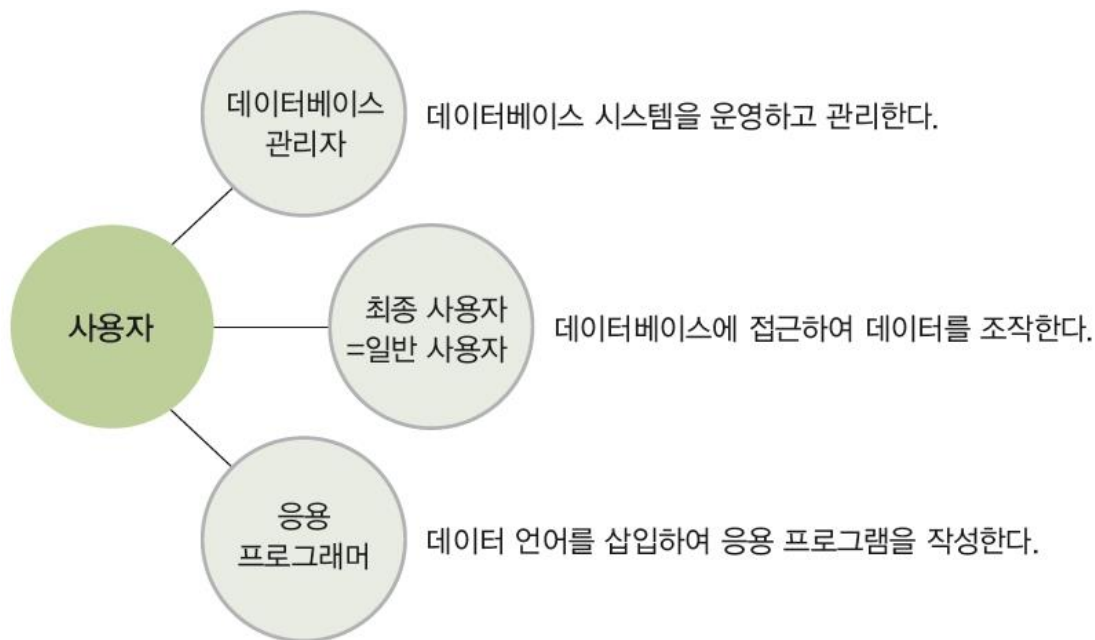


그림 3-7 데이터베이스 사용자

03 데이터베이스 사용자

❖ 데이터베이스 관리자(DBA; DataBase Administrator)

- 데이터베이스 시스템을 운영/관리하는 사람
- 주로 데이터 정의어와 데이터 제어어를 사용
- 주요 업무
 - 데이터베이스 구성 요소 선정
 - 데이터베이스 스키마 정의
 - 물리적 저장 구조와 접근 방법 결정
 - 무결성 유지를 위한 제약조건 정의
 - 보안 및 접근 권한 정책 결정
 - 백업 및 회복 기법 정의
 - 시스템 데이터베이스 관리
 - 시스템 성능 감시 및 성능 분석
 - 데이터베이스 재구성

03 데이터베이스 사용자

❖ 최종 사용자(end user)

- 데이터베이스에 접근하여 데이터를 조작(삽입·삭제·수정·검색)하는 사람
- 주로 데이터 조작어를 사용
- 캐주얼 사용자와 초보 사용자로 구분

❖ 응용 프로그래머(application programmer)

- 데이터 언어를 삽입하여 응용 프로그램을 작성하는 사람
- 주로 데이터 조작어를 사용

응용 프로그래머



최종 사용자

그림 3-8 최종 사용자와 응용 프로그래머의 예

응용프로그램의 예

```
//conn.Open(); //데이터베이스 연결
OleDbCommand cmd = new OleDbCommand();
cmd.CommandText = "select table_name from user_tables"; //테이블목록가져 오기
cmd.CommandType = CommandType.Text; //검색명령을 쿼리 형태로
cmd.Connection = conn;

OleDbDataReader read = cmd.ExecuteReader(); //select 결과

//행 단위로 반복
while (read.Read())
{
    cmbTableList.Items.Add(read.GetValue(0)); //데이터그리드뷰에 오브젝트 배열 추가
}

read.Close();
cmbTableList.Text = "테이블선택";
```

```
//conn.Open(); //데이터베이스 연결
OleDbCommand cmd = new OleDbCommand();
cmd.CommandText = "select * from " + cmbTableList.Text;
cmd.CommandType = CommandType.Text;
cmd.Connection = conn;

OleDbDataReader read = cmd.ExecuteReader(); //select 결과

dataGridView1.ColumnCount = read.FieldCount; //read.FieldCount는 테이블의 컬럼 수를 말함
```

04 데이터 언어

❖ 데이터 언어

- 사용자와 데이터베이스 관리 시스템 간의 통신 수단
- 사용 목적에 따라 데이터 정의어, 데이터 조작어, 데이터 제어어로 구분

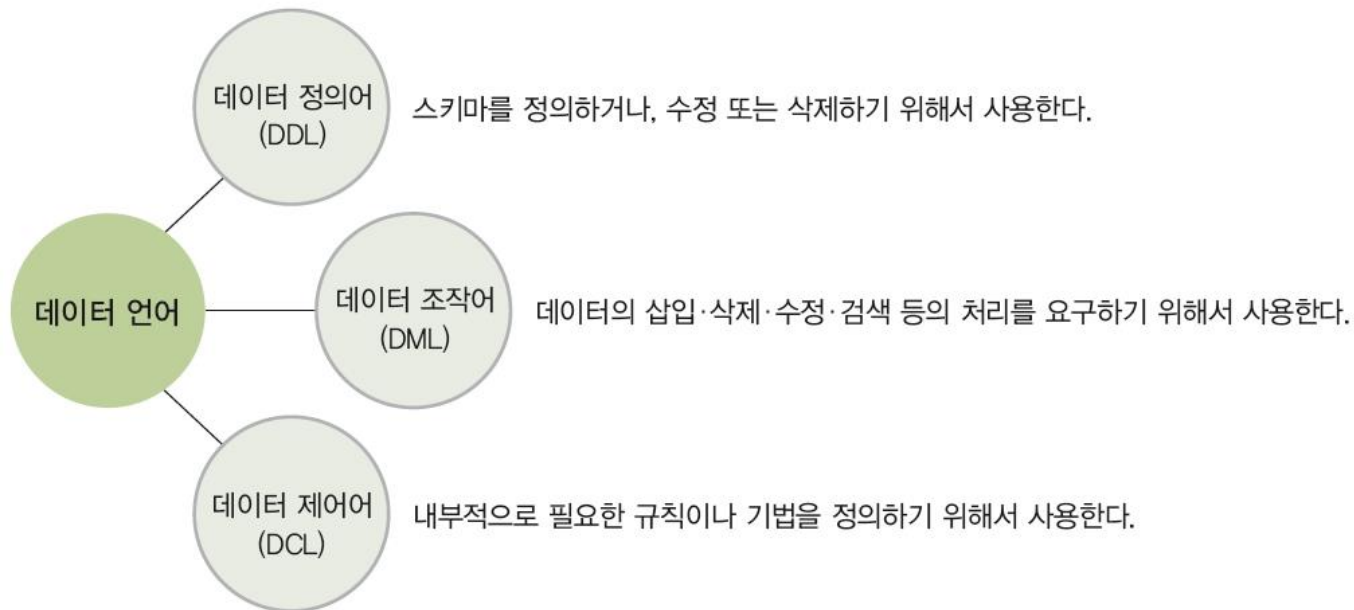


그림 3-9 데이터 언어의 종류와 용도

04 데이터 언어

❖ 데이터 정의어(DDL; Data Definition Language)

- 스키마를 정의하거나, 수정 또는 삭제하기 위해 사용

❖ 데이터 조작어(DML; Data Manipulation Language)

- 데이터의 삽입·삭제·수정·검색 등의 처리를 요구하기 위해 사용
- 절차적 데이터 조작어와 비절차적 데이터 조작어로 구분
 - 절차적 데이터 조작어(procedural DML)
 - 사용자가 어떤(what) 데이터를 원하고 그 데이터를 얻기 위해 어떻게(how) 처리해야 하는지도 설명
 - 변수, 함수 등이 사용됨
 - 일반적인 프로그래밍언어
 - 비절차적 데이터 조작어(nonprocedural DML)
 - 사용자가 어떤(what) 데이터를 원하는지만 설명
 - 선언적 언어(declarative language)라고도 함
 - 변수, 함수 등이 사용 안됨
 - 데이터베이스 언어(SQL 등), 인공지능언어(LISP 등)



그림 3-10 절차적 데이터 조작어와 비절차적 데이터 조작어의 이해

❖ 데이터 제어어(DCL; Data Control Language)

- 내부적으로 필요한 규칙이나 기법을 정의하기 위해 사용
- 사용 목적
 - 무결성 : 정확하고 유효한 데이터만 유지
 - 보안 : 허가받지 않은 사용자의 데이터 접근 차단, 허가된 사용자에게 권한 부여
 - 회복 : 장애가 발생해도 데이터 일관성 유지
 - 동시성 제어 : 동시 공유 지원

05 데이터베이스 관리 시스템의 구성

❖ 데이터베이스 관리 시스템

- 데이터베이스 관리와 사용자의 데이터 처리 요구 수행
- 주요 구성 요소
 - 질의 처리기(query processor)
 - 사용자의 데이터 처리 요구를 해석하여 처리
 - DDL 컴파일러, DML 프리 컴파일러, DML 컴파일러, 런타임 데이터베이스 처리기, 트랜잭션 관리자 등을 포함
 - 저장 데이터 관리자(stored data manager)
 - 디스크에 저장된 사용자 데이터베이스와 데이터 사전을 관리하고 실제로 접근하는 역할
 - 운영체제의 파일관리시스템의 도움을 받음

05 데이터베이스 시스템의 구성

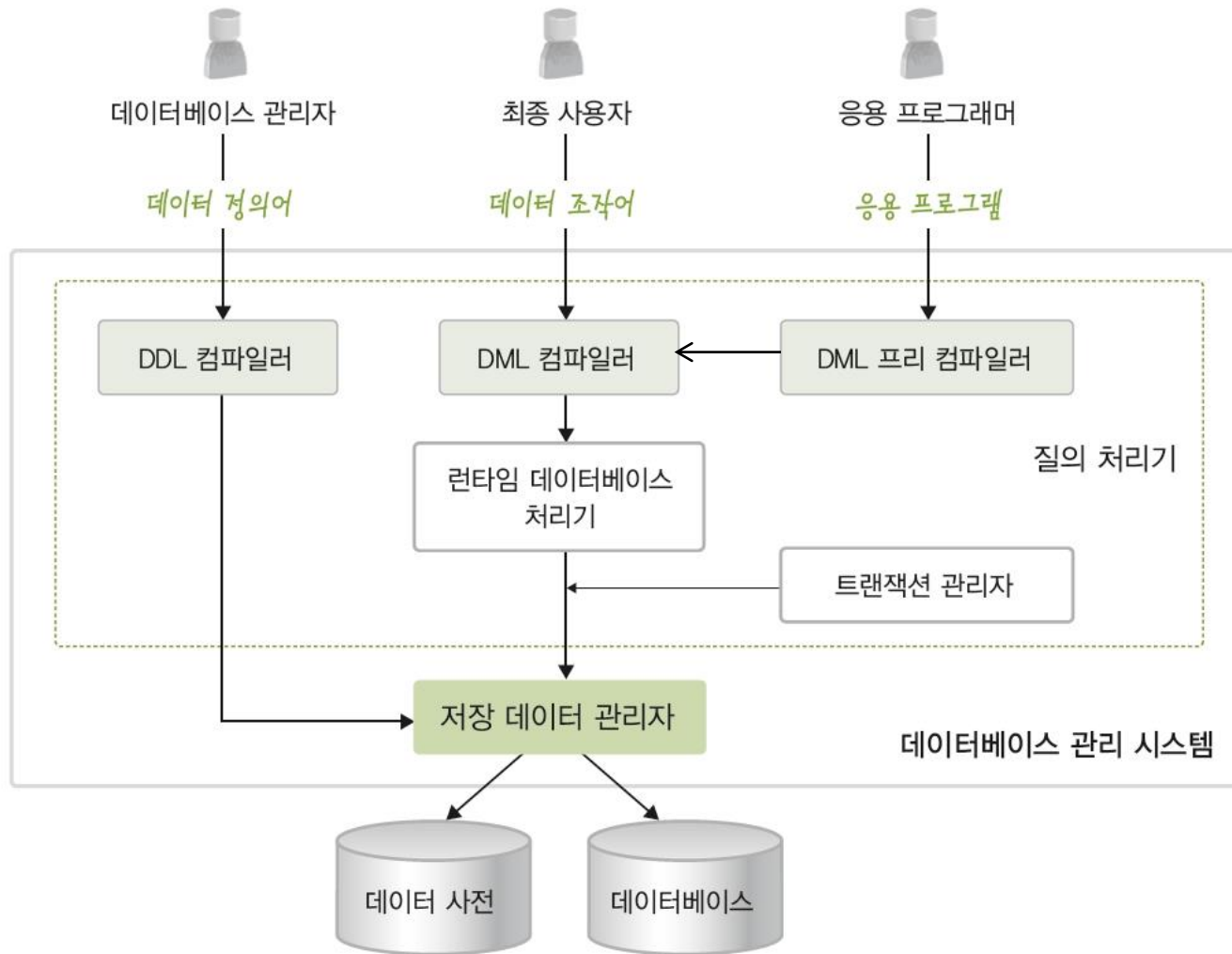


그림 3-11 데이터베이스 관리 시스템의 구성



Thank You
