

데이터베이스관리

6주차

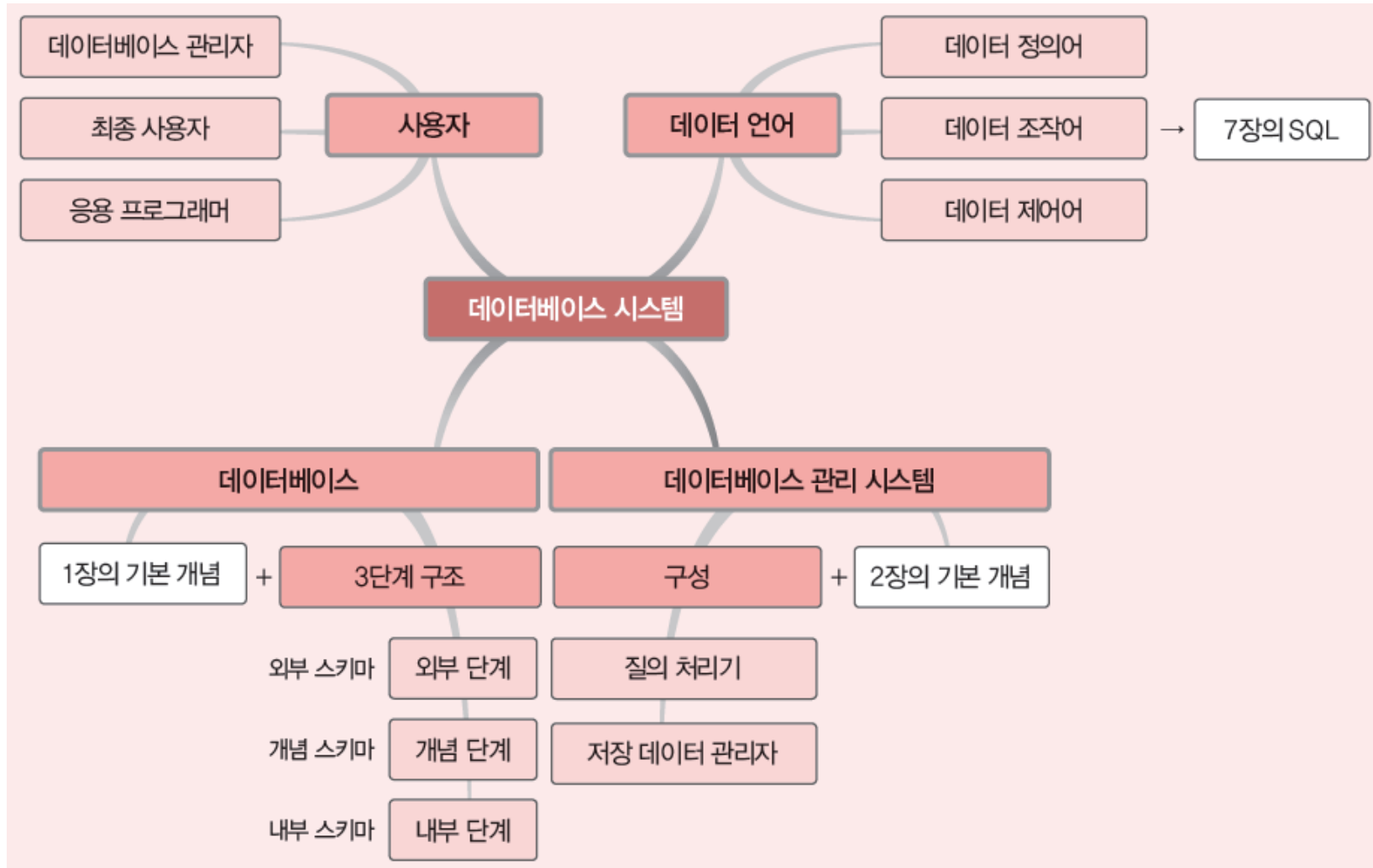
담당교수: 김희숙
(jasmin11@hanmail.net)

데이터베이스 시스템

6주차 6-01

담당교수: 김희숙
(jasmin11@hanmail.net)

[요약] 데이터베이스 시스템



[요약] 데이터베이스 시스템



* 데이터베이스 시스템 구성 요소:

데이터(데이터베이스), 하드웨어, DBMS, 사용자

□ 데이터베이스

- ✓ 사용자 데이터베이스
- ✓ 시스템 데이터베이스
 - ✓ 데이터 사전(Data Dictionary), 시스템 카탈로그, 메타 데이터(meta data)

□ 데이터베이스 관리 시스템

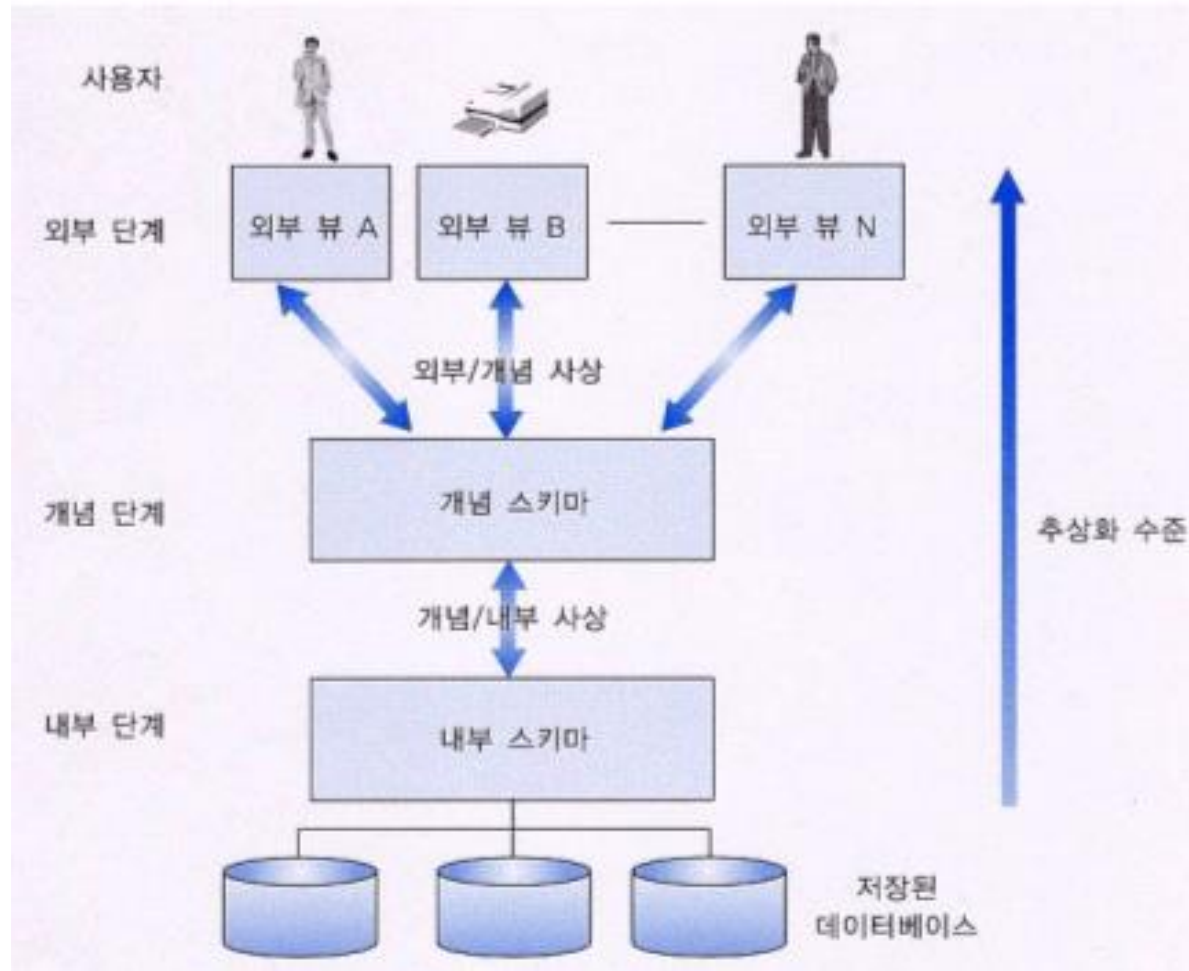
- ✓ 데이터베이스를 관리하는 시스템 소프트웨어

□ 데이터베이스 시스템

- ✓ 데이터를 데이터베이스로 저장하고 관리하여 정보를 생성하는 컴퓨터 중심의 시스템

[요약] 3단계 구조

ANSI/SPARC 아키텍처 (스키마 3단계 구조)

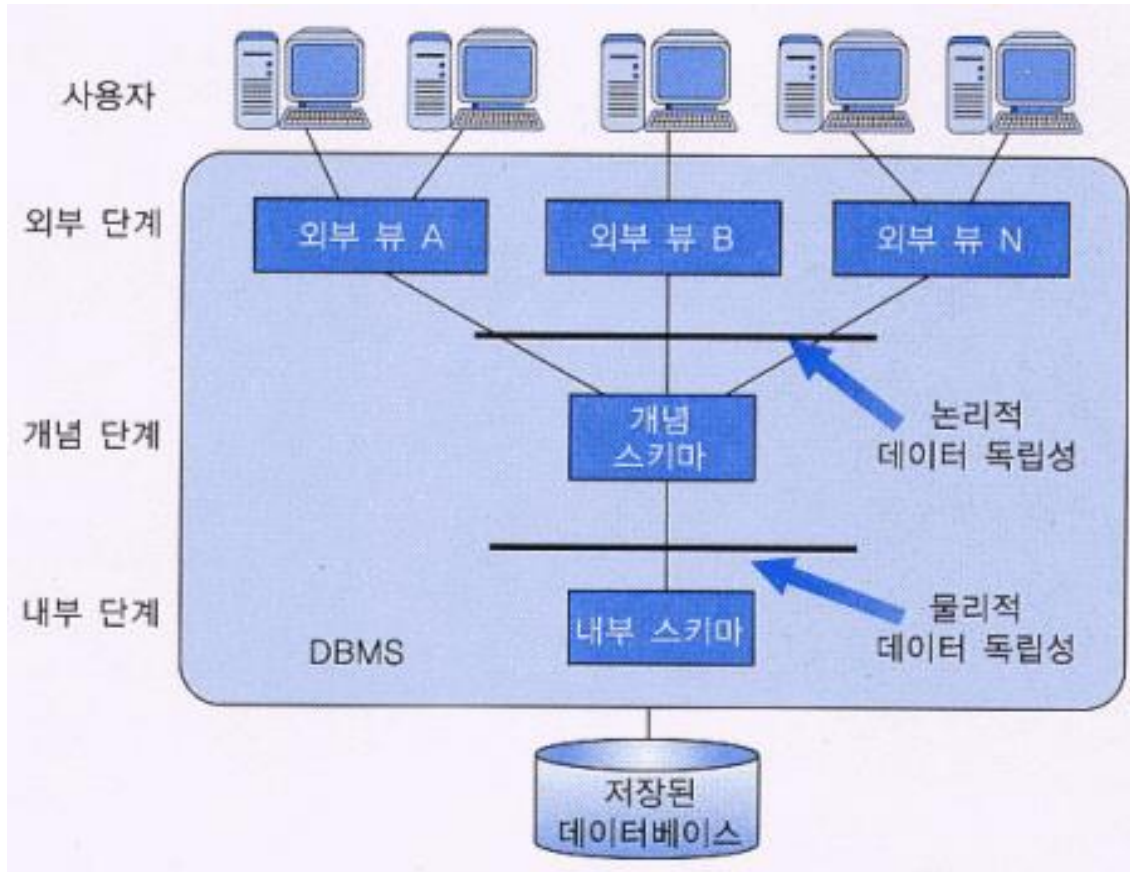


(그림 출처: "데이터베이스배움터", 홍의경 저, 생능, 2012)



[요약] 3단계 구조

데이터 독립성



- **논리적 데이터 독립성:**
개념스키마가 변화해도
외부스키마는 영향 받지 않는다
- **물리적 데이터 독립성:**
내부스키마가 변화해도
개념스키마, 외부스키마는 영향 받지 않는다

(그림 출처: "데이터베이스배움터", 홍의경 저, 생능, 2012)

01 데이터베이스 시스템의 정의

- 데이터베이스 시스템(DBS; DataBase System)
 - 데이터베이스에 데이터를 저장하고, 이를 관리하여 조직에 필요한 정보를 생성해주는 시스템

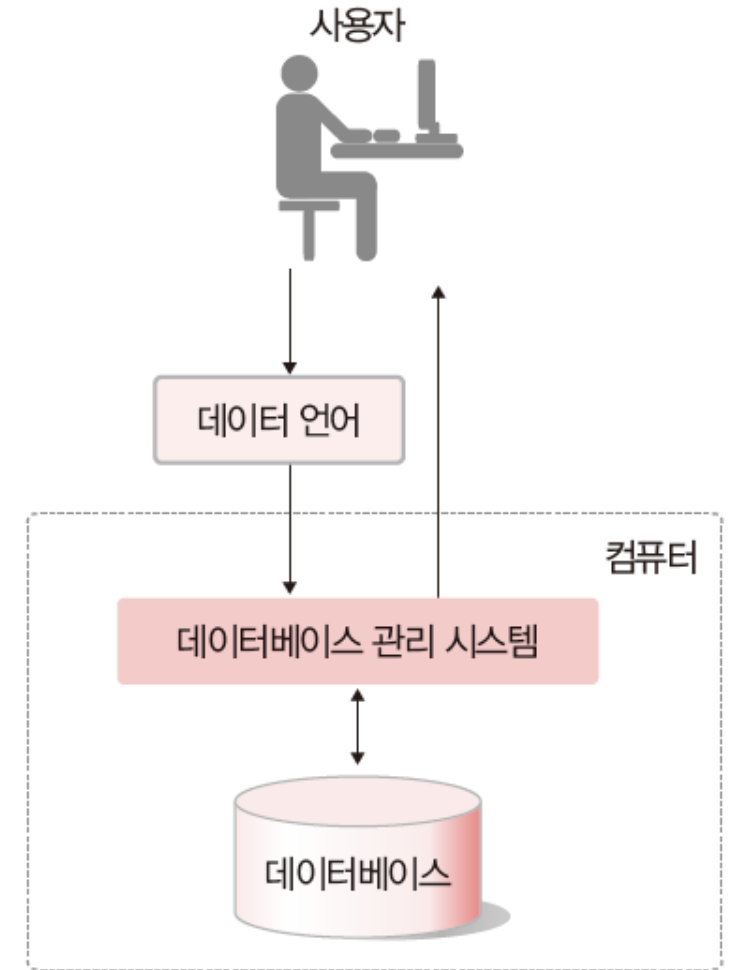


그림 3-1 데이터베이스 시스템의 구성

02 데이터베이스의 구조

- 스키마와 인스턴스

- 스키마(schema)
 - 데이터베이스에 저장되는 데이터 구조와 제약조건을 정의한 것
- 인스턴스(instance)
 - 스키마에 따라 데이터베이스에 실제로 저장된 값



그림 3-2 스키마의 예

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
 - 내부 단계(internal level) : 물리적인 저장 장치의 관점
- 각 단계별로 다른 추상화(abstraction) 제공
 - 내부 단계에서 외부 단계로 갈수록 추상화 레벨이 높아짐

02 데이터베이스의 구조

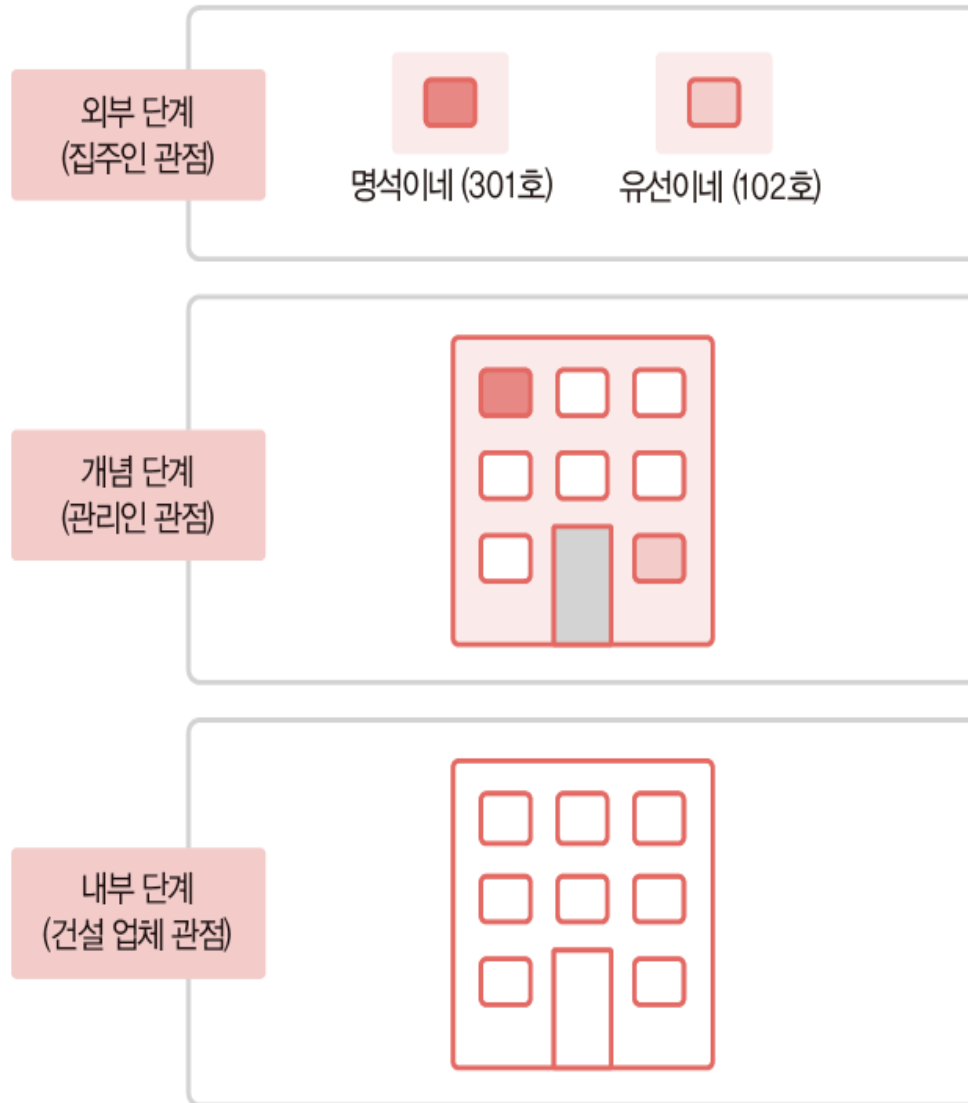


그림 3-4 3단계 데이터베이스 구조의 이해

02 데이터베이스의 구조

- 3단계 데이터베이스 구조 : 외부 단계

- 데이터베이스를 개별 사용자 관점에서 이해하고 표현하는 단계
- 데이터베이스 하나에 외부 스키마가 여러 개 존재할 수 있음
 - 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

02 데이터베이스의 구조

- 3단계 데이터베이스 구조 : 개념 단계

- 데이터베이스를 조직 전체의 관점에서 이해하고 표현하는 단계
- 데이터베이스 하나에 개념 스키마가 하나만 존재함
 - 개념 스키마(conceptual schema)
 - 개념 단계에서 데이터베이스 전체의 논리적 구조를 정의한 것
 - 조직 전체의 관점에서 생각하는 데이터베이스의 모습
 - 전체 데이터베이스에 어떤 데이터가 저장되는지, 데이터들 간에는 어떤 관계가 존재하고 어떤 제약조건이 있는지에 대한 정의뿐만 아니라, 데이터에 대한 보안 정책이나 접근 권한에 대한 정의도 포함

02 데이터베이스의 구조

- 3단계 데이터베이스 구조 : 내부 단계
 - 데이터베이스를 저장 장치의 관점에서 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

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

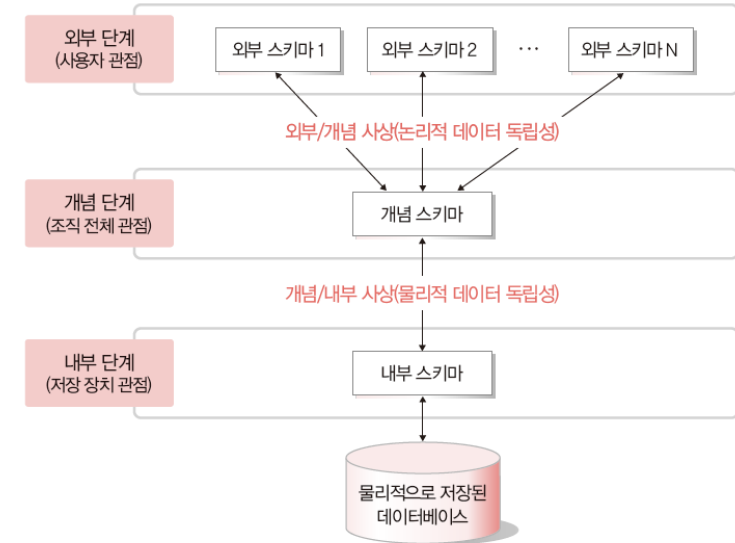


그림 3-6 3단계 데이터베이스 구조에서 스키마 간의 사상

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
- 논리적 데이터 독립성
 - 개념 스키마가 변경되어도 외부 스키마는 영향을 받지 않음
 - 개념 스키마가 변경되면 관련된 외부/개념 사상만 정확하게 수정해주면 됨
- 물리적 데이터 독립성
 - 내부 스키마가 변경되어도 개념 스키마는 영향을 받지 않음
 - 내부 스키마가 변경되면 관련된 개념/내부 사상만 정확하게 수정해주면 됨

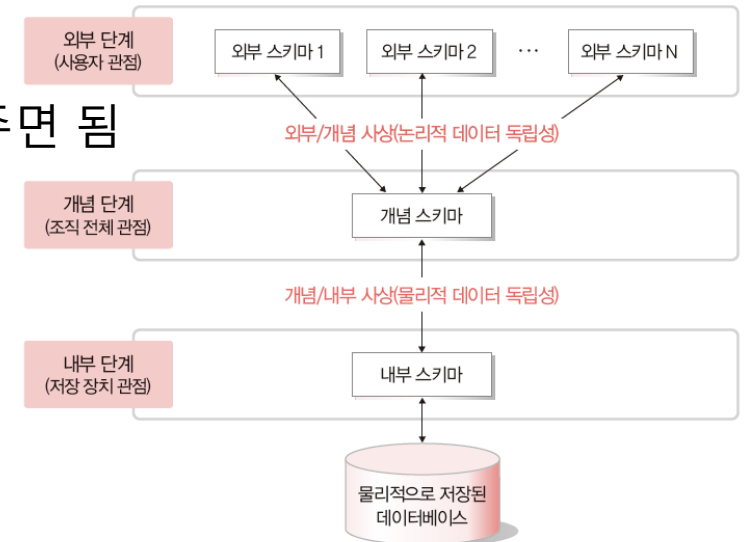


그림 3-6 3단계 데이터베이스 구조에서 스키마 간의 사상

02 데이터베이스의 구조

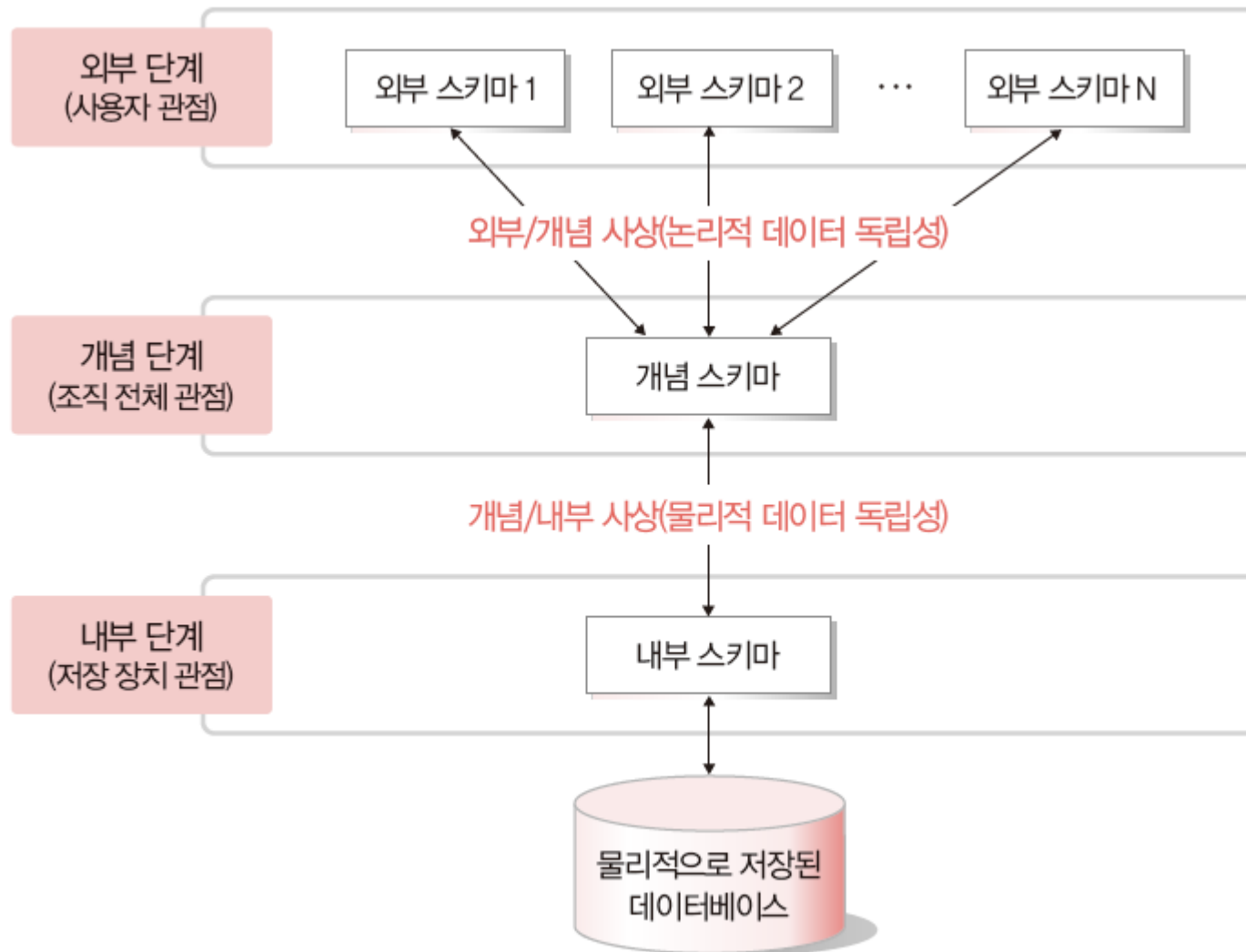


그림 3-6 3단계 데이터베이스 구조에서 스키마 간의 사상

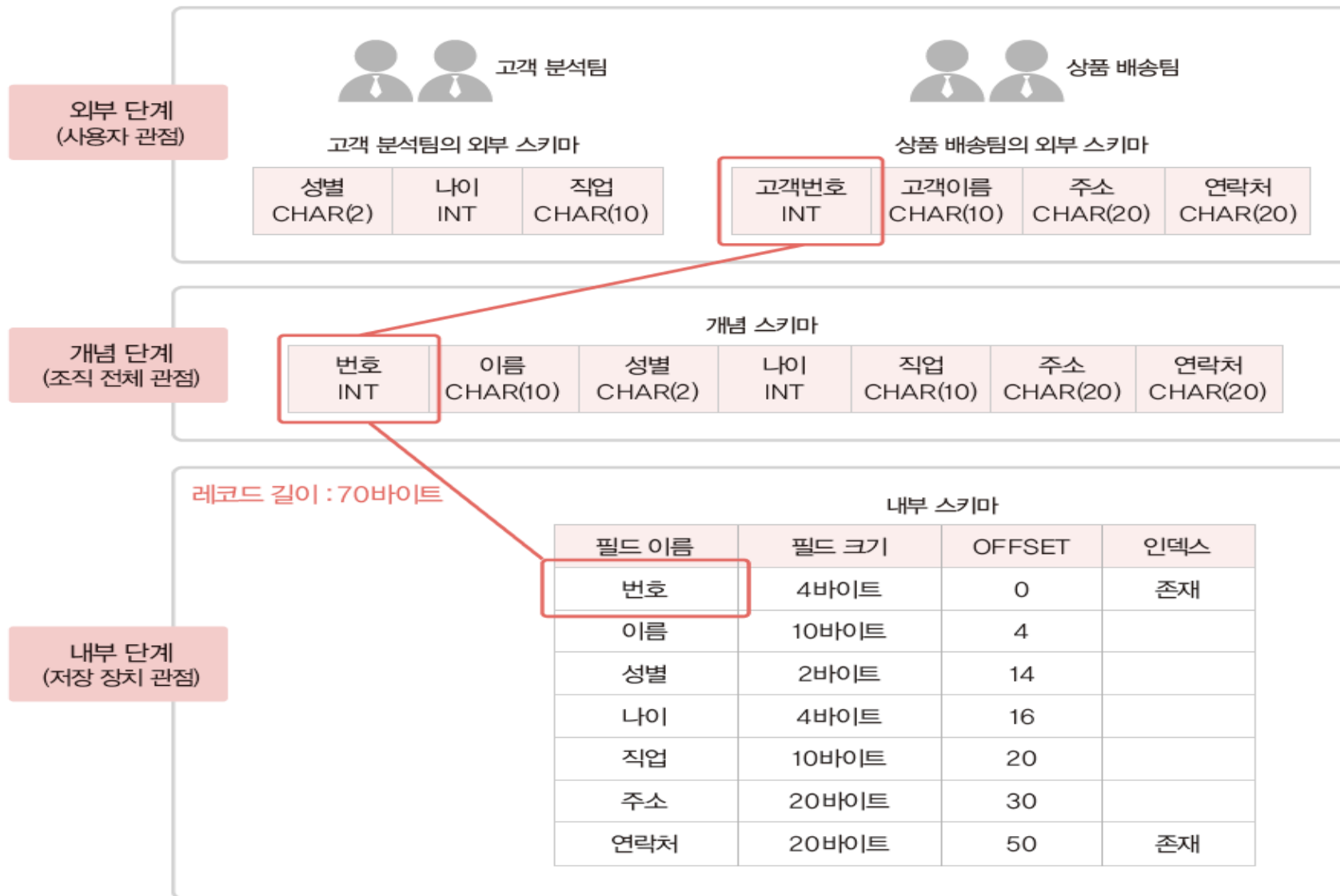
02 데이터베이스의 구조

- 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 저장 내용을 검색만 할 수 있음

02 데이터베이스의 구조

- **데이터 디렉터리(data directory)**
 - 데이터 사전에 있는 데이터에 실제로 접근하는 데 필요한 위치 정보를 저장하는 시스템 데이터베이스
 - 일반 사용자의 접근은 허용되지 않음
- **사용자 데이터베이스(user database)**
 - 사용자가 실제로 이용하는 데이터가 저장되어 있는 일반 데이터베이스

[실습] 데이터베이스 3단계 구조



[실습] 3단계 구조(외부스키마, 개념스키마, 내부스키마)

[실습] 외부 스키마(schema_view-ex.sql)

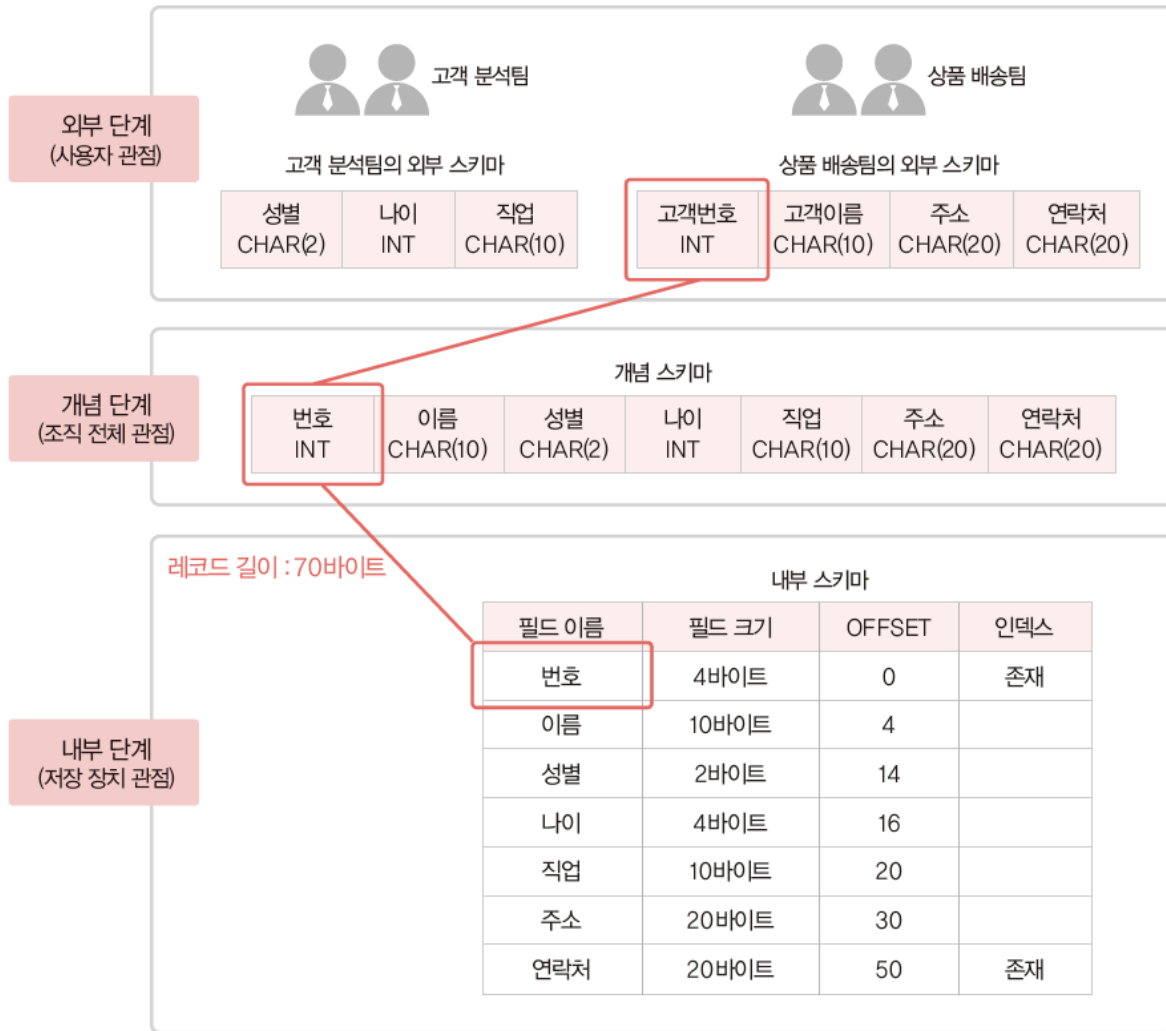


그림 3-5 3단계 데이터베이스 구조의 예

외부 뷰(view)

35 • `select * from 고객분석팀;`

성별	나이	직업
남	20	학생
남	22	학생
여	20	학생

외부 뷰(view)

47 • `select * from 상품배송팀;`

고객 번호	고객 이름	주소	연락처
1	홍길동	서울	NULL
2	임꺽정	인천	010-1111-1111
3	신아로미	서울	010-2222-2222

23 • `select * from 가입고객;`

번호	이름	성별	나이	직업	주소	연락처
1	홍길동	남	20	학생	서울	NULL
2	임꺽정	남	22	학생	인천	010-1111-1111
3	신아로미	여	20	학생	서울	010-2222-2222
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```

1  -- (schema_view-ex.sql)
2  -- 3단계 스키마 구조
3  • use studydb;
4
5  • drop table 가입고객;
6
7  • CREATE TABLE 가입고객 (
8      번호 int NOT NULL,
9      이름 char(10),
10     성별 char(2) ,
11     나이 int ,
12     직업 char(10),
13     주소 char(20) ,
14     연락처 char(20) ,
15     PRIMARY KEY(번호)
16 );
17
18 • INSERT INTO 가입고객 VALUES (1, '홍길동', '남', 20, '학생', '서울', NULL) ,
19                                (2, '임꺽정', '남', 22, '학생', '인천', '010-1111-1111') ,
20                                (3, '신아로미', '여', 20, '학생', '서울', '010-2222-2222')
21 ;
22
23 • select * from 가입고객;
    
```

번호	이름	성별	나이	직업	주소	연락처
1	홍길동	남	20	학생	서울	NULL
2	임꺽정	남	22	학생	인천	010-1111-1111
3	신아로미	여	20	학생	서울	010-2222-2222

가입고객 4 x

```

-- 3단계 스키마 구조
use studydb;
    
```

```
drop table 가입고객;
```

```

CREATE TABLE 가입고객 (
    번호 int NOT NULL,
    이름 char(10),
    성별 char(2) ,
    나이 int ,
    직업 char(10),
    주소 char(20) ,
    연락처 char(20) ,
    PRIMARY KEY(번호)
);
    
```

```

INSERT INTO 가입고객 VALUES (1, '홍길동', '남', 20, '학생', '서울', NULL) ,
                              (2, '임꺽정', '남', 22, '학생', '인천', '010-1111-1111') ,
                              (3, '신아로미', '여', 20, '학생', '서울', '010-2222-2222')
;
    
```

```
select * from 가입고객;
```

```
-- 가입고객(번호,이름,성별,나이,직업,주소,연락처)
-- 고객분석팀(성별,나이,직업)
-- 상품배송팀(고객번호,고객이름,주소,연락처)
```



-- 뷰 생성

```
CREATE VIEW 고객분석팀
as
```

```
    SELECT 성별, 나이, 직업
    FROM   가입고객;
```

```
select * from 고객분석팀;
```

```
25 -- 가입고객(번호,이름,성별,나이,직업,주소,연락처)
26 -- 고객분석팀(성별,나이,직업)
27 -- 상품배송팀(고객번호,고객이름,주소,연락처)
28
29 -- 뷰 생성
30 • CREATE VIEW 고객분석팀
31   as
32     SELECT 성별, 나이, 직업
33     FROM   가입고객;
34
35 • select * from 고객분석팀;
```

Result Grid			
Filter Rows: <input type="text"/>			
Export:  Wrap Cell Content: 			
	성별	나이	직업
▶	남	20	학생
	남	22	학생
	여	20	학생

```
-- 가입고객(번호,이름,성별,나이,직업,주소,연락처)
-- 고객분석팀(성별,나이,직업)
-- 상품배송팀(고객번호,고객이름,주소,연락처)
```

-- 뷰 생성

CREATE VIEW 상품배송팀(고객번호,고객이름,주소,연락처)

as

```
SELECT 번호, 이름, 주소, 연락처
FROM 가입고객;
```

```
select * from 상품배송팀;
```

```
36
37 -- 가입고객(번호,이름,성별,나이,직업,주소,연락처)
38 -- 고객분석팀(성별,나이,직업)
39 -- 상품배송팀(고객번호,고객이름,주소,연락처)
40
41 -- 뷰 생성
42 • CREATE VIEW 상품배송팀(고객번호,고객이름,주소,연락처)
43   as
44     SELECT 번호, 이름, 주소, 연락처
45     FROM 가입고객;
46
47 • select * from 상품배송팀;
```

Result Grid

	고객 번호	고객 이름	주소	연락처
▶ 1		홍길동	서울	NULL
2		임궽정	인천	010-1111-1111
3		신아로미	서울	010-2222-2222

데이터베이스 사용자

6주차 6-02

담당교수: 김희숙
(jasmin11@hanmail.net)



[요약] 데이터베이스 용어

* 데이터베이스 사용자:

데이터베이스 관리자(DBA), 응용프로그램 개발자, 최종 사용자

데이터베이스 관리자(Database Administrator: DBA)
(데이터 정의어와 데이터 제어어를 사용)

* 데이터베이스 언어(SQL):

데이터 정의어(DDL)

데이터 조작어(DML)

데이터 제어어(DCL)

03 데이터베이스 사용자

- 데이터베이스 사용자

- 데이터베이스를 이용하기 위해 접근하는 모든 사람
- 데이터베이스 관리자, 최종 사용자, 응용 프로그래머

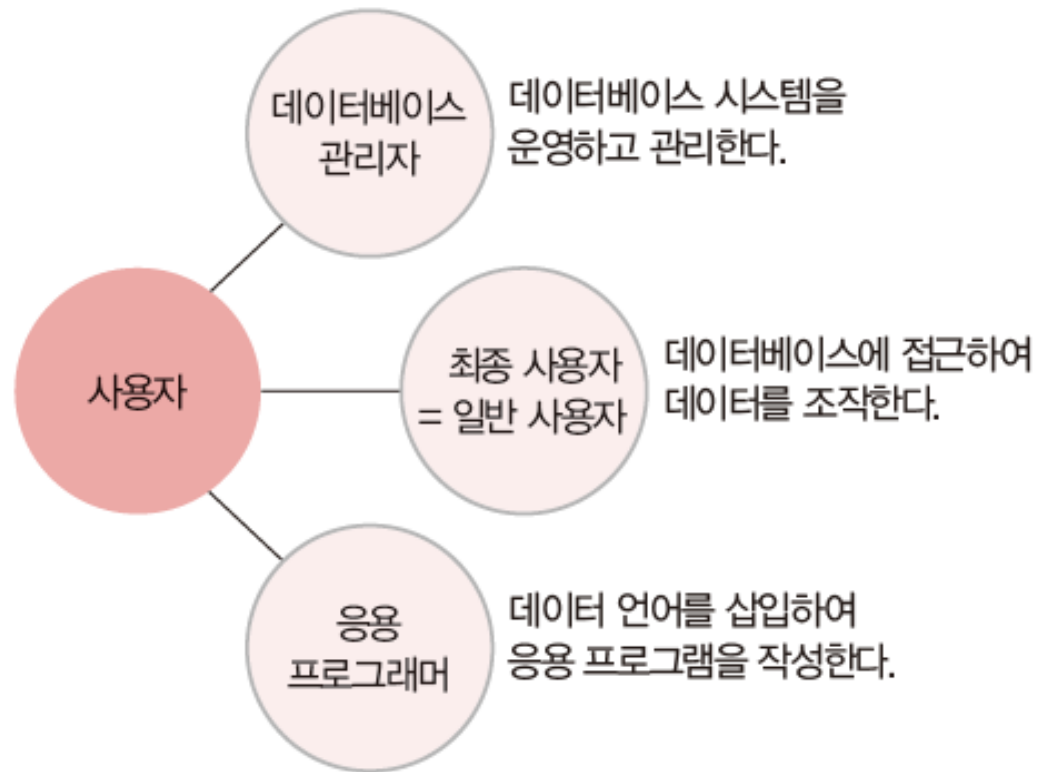


그림 3-7 데이터베이스 사용자

03 데이터베이스 사용자

- 데이터베이스 관리자(DBA; DataBase Administrator)
 - 데이터베이스 시스템을 운영 및 관리하는 사람
 - 주로 데이터 정의어와 데이터 제어어를 이용
 - 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
- 데이터 조작어를 사용
- 캐주얼 사용자와 초보 사용자로 구분

- 응용 프로그래머(application programmer)

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



그림 3-8 최종 사용자와 응용 프로그래머의 예

04 데이터 언어

- 데이터 언어

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

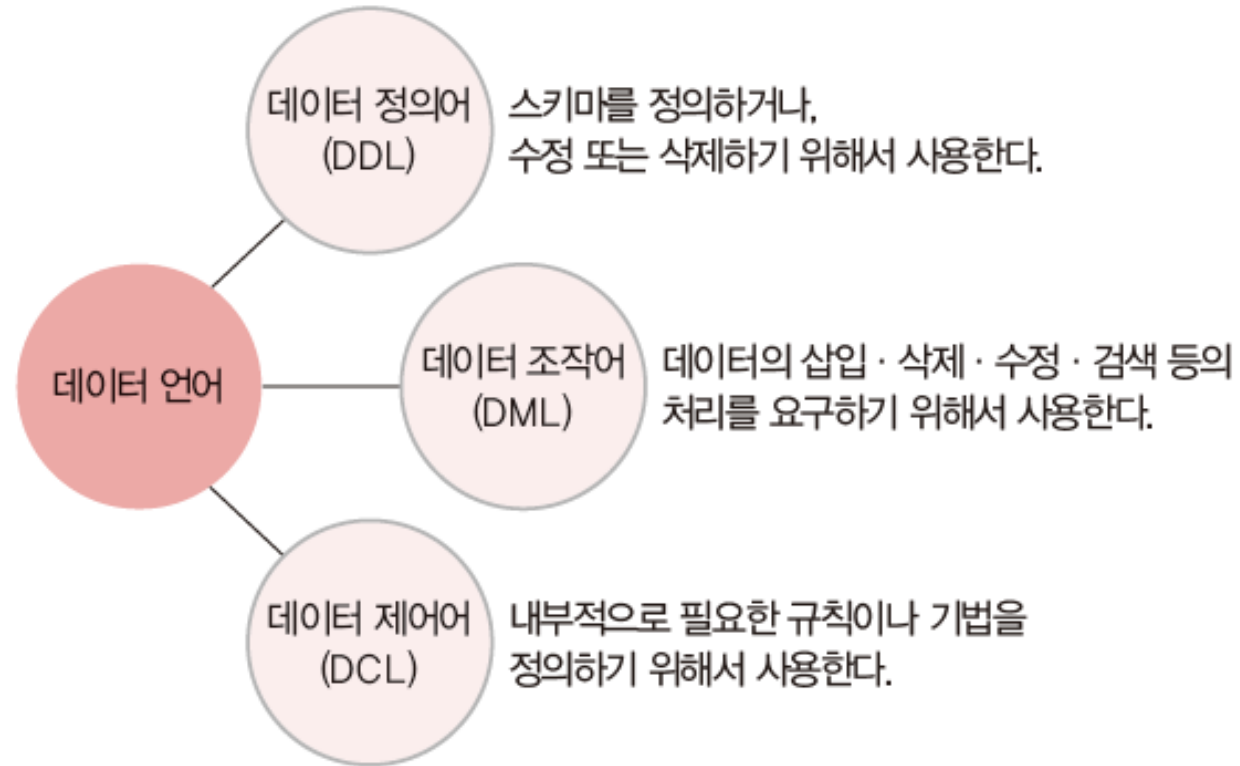


그림 3-9 데이터 언어의 종류와 용도

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 조작어 vs 비절차적 데이터 조작어
 - 절차적 데이터 조작어(procedural DML)
 - 사용자가 어떤(what) 데이터를 원하고 그 데이터를 얻기 위해 어떻게(how) 처리해야 하는지도 설명
 - 비절차적 데이터 조작어(nonprocedural DML)
 - 사용자가 어떤(what) 데이터를 원하는지만 설명
 - 선언적 언어(declarative language)라고도 함



그림 3-10 절차적 데이터 조작어와 비절차적 데이터 조작어의 이해

04 데이터 언어

- 데이터 제어어(DCL; Data Control Language)

- 내부적으로 필요한 규칙이나 기법을 정의하기 위해 사용
- 사용 목적
 - **무결성** : 정확하고 유효한 데이터만 유지
 - **보안** : 허가 받지 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 주요 구성요소:

- 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 데이터베이스와 데이터 사전을 관리하고 접근함

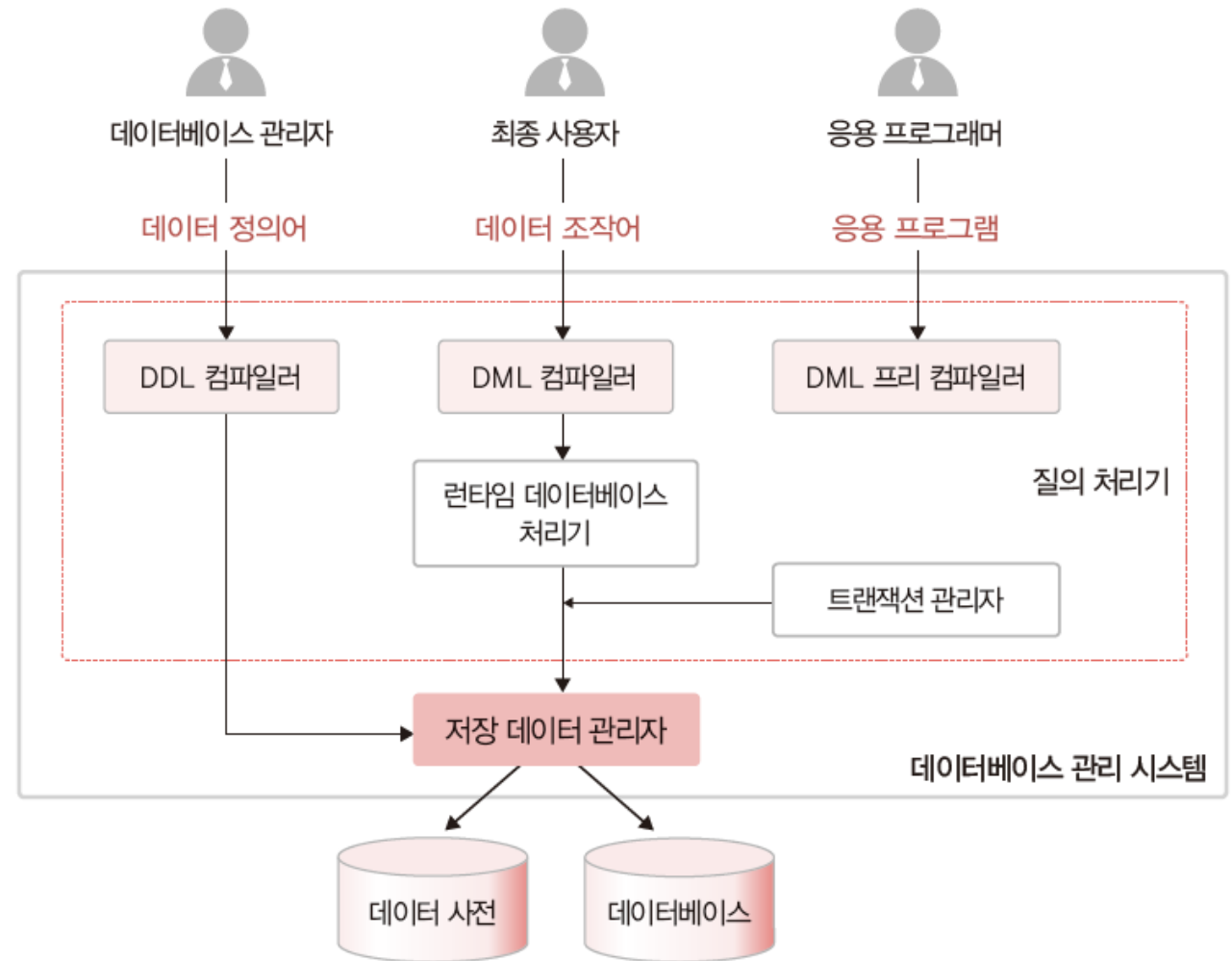


그림 3-11 데이터베이스 관리 시스템의 구성

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 ㉠과 ㉡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, 저장인터페이스라고도 한다.

11. 다음 설명에서 ㉠과 ㉡가 각각 무엇인지 답하시오.

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

(A) - 데이터 사전

(B) - 데이터 디렉토리

12. 다음 설명에서 ㉠과 ㉡가 각각 무엇인지 답하시오.

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

(A) - 질의 처리기

(B) - 저장 데이터 관리자

SQL: SELECT 문법(SELECT 기초)

6주차 6-03

담당교수: 김희숙
(jasmin11@hanmail.net)

데이터베이스 언어(SQL)

□ 데이터베이스 언어(SQL)

- ✓ 데이터 정의어(DDL)
- ✓ 데이터 조작어(DML)
- ✓ 데이터 제어어(DCL)

SQL	명령어
DDL	CREATE ALTER DROP
DML	INSERT UPDATE DELETE SELECT
DCL	GRANT REVOKE

SQL



SQL



❖ SQL(Structured Query Language) : 데이터베이스 언어

종류	약어	명령어	비고
데이터 정의어	DDL (Data Definition Language)	CREATE ALTER DROP	객체 생성 객체 수정 객체 삭제
데이터 조작어	DML (Data Manipulation Language)	SELECT INSERT UPDATE DELETE	데이터 검색 데이터 입력 데이터 수정 데이터 삭제
데이터 제어어	DCL (Data Control Language)	GRANT REVOKE	권한부여 권한취소
	TCL (Transaction Control Language)	COMMIT; ROLLBACK; SAVEPOINT;	작업완료 작업취소 저장포인트

[예제] SELECT (데이터 조회)

```
CREATE TABLE customer (
  cno    char(4)      NOT NULL ,
  cname  varchar(10)  NOT NULL ,
  city   varchar(20) ,
  point  int ,
  CONSTRAINT pk_customer_cno PRIMARY KEY(cno)
);
```

테이블 이름	필드 이름	데이터 형식	NULL 유무	기본키
customer	cno	char(4)	NOT NULL	PK
	cname	varchar(10)	NOT NULL	
	city	varchar(20)		
	point	int		

Result Grid				
	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
●	NULL	NULL	NULL	NULL

```
insert into customer values('c101','홍길동','서울',500);
insert into customer values('c102','임꺽정','인천',300);
insert into customer values('c103','박찬호','안양',800);
insert into customer values('c204','신동엽','과천',350);
insert into customer values('c205','정진우','고양',400);
```

```
/* 새 레코드를 추가하고 SELECT문 예제 실습하시오 */
insert into customer values('c307','정동우','서울', NULL);
```

[실습1] SELECT (데이터 조회)

[실습] (customer.sql)

[실습] (SELECT 기초, NULL, LIKE, Order by)

```
/* customer(cno, cname, city, point) */
```

-- 고객(고객번호, 고객명, 거주지, 포인트)

-- 1-1) 테이블의 모든 열을 검색하라

-- 1-2) 테이블의 모든 열을 검색(필드명 사용)

-- 1-3) 고객의 고객명, 거주지를 검색하라(테이블의 특정 열을 검색)

-- 1-4) cname 은 성명, city는 거주지로 출력하라

-- (화면에 표시되는 열 이름 변경하여 검색)

-- 1-5) customer 테이블에서 거주지를 검색하라

-- 1-6) 거주지를 검색하는데 중복 행을 제거하여 한 번씩만 검색하라

Result Grid				
Filter Rows:				
	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
●	NULL	NULL	NULL	NULL

[실습1] SELECT (데이터 조회)

[실습 1]

-- customer(cno, cname, city, point)

-- 1-1) 테이블의 모든 열을 검색하라

```
SELECT *  
FROM customer;
```

[실습] (customer.sql)

Result Grid				
Filter Rows: <input type="text"/>				
	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL

[실습1] SELECT (데이터 조회)

[실습 1]

- customer(cno, cname, city, point)
- 1-2) 테이블의 모든 열을 검색(필드명 사용)

```
SELECT cno, cname, city, point
FROM customer;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL

[실습1] SELECT (데이터 조회)

[실습 1]

- customer(cno, cname, city, point)
- 1-3) 고객의 고객명, 거주지를 검색하라(테이블의 특정 열을 검색)

```
SELECT  cname, city
FROM    customer;
```

[실습] (customer.sql)

Result Grid					Filter Rows:
	cno	cname	city	point	
▶	c101	홍길동	서울	500	
	c102	임꺽정	인천	300	
	c103	박찬호	안양	800	
	c204	신동엽	과천	350	
	c205	정진우	고양	400	
	c307	정동우	서울	NULL	
•	NULL	NULL	NULL	NULL	



	cname	city
▶	홍길동	서울
	임꺽정	인천
	박찬호	안양
	신동엽	과천
	정진우	고양
	정동우	서울

[실습1] SELECT (데이터 조회)

[실습 1]

- customer(cno, cname, city, point)
- 1-4) cname 은 성명, city는 거주지로 출력하라
- (화면에 표시되는 열 이름 변경하여 검색)

```
SELECT  cname 성명, city 거주지
FROM    customer;
```

[실습] (customer.sql)

Result Grid					Filter Rows:
	cno	cname	city	point	
▶	c101	홍길동	서울	500	
	c102	임꺽정	인천	300	
	c103	박찬호	안양	800	
	c204	신동엽	과천	350	
	c205	정진우	고양	400	
	c307	정동우	서울	NULL	
•	NULL	NULL	NULL	NULL	



	성명	거주지
▶	홍길동	서울
	임꺽정	인천
	박찬호	안양
	신동엽	과천
	정진우	고양
	정동우	서울

[실습1] SELECT (데이터 조회)

[실습 1]

- customer(cno, cname, city, point)
- 1-5) customer 테이블에서 거주지를 검색하라

```
SELECT city
FROM customer;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	city
▶	서울
	인천
	안양
	과천
	고양
	서울

[실습1] SELECT (데이터 조회)

[실습 1]

-- customer(cno, cname, city, point)

-- 1-6) 거주지를 검색하는데 중복 행을 제거하여 한 번씩만 검색하라

```
SELECT DISTINCT city
FROM customer;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



	city
▶	서울
	인천
	안양
	과천
	고양

[실습1] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 1]

-- customer(cno, cname, city, point)

-- 1-6) 거주지를 검색하는데 **중복 행을 제거**하여 한 번씩만 검색하라

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL

```
SELECT city
FROM customer;
```



```
SELECT DISTINCT city
FROM customer;
```

	city
▶	서울
	인천
	안양
	과천
	고양
	서울

	city
▶	서울
	인천
	안양
	과천
	고양

[실습2] SELECT (데이터 조회)

[실습] (customer.sql)

[실습] (SELECT 기초, NULL, LIKE, Order by)

```
/* customer(cno, cname, city, point) */
```

-- 고객(고객번호, 고객명, 거주지, 포인트)

-- 2-1) 고객번호가 c101 인 고객의 모든 정보를 검색하라

-- 2-2) 포인트가 400 이하인 고객의 모든 정보를 검색하라

-- 2-3) 거주지가 서울 이면서 포인트가 500 이상인 고객의 이름, 거주지, 포인트를 검색하라

-- 2-4) 거주지가 서울 이거나 포인트가 500 이상인 고객의 이름, 거주지, 포인트를 검색하라

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
●	NULL	NULL	NULL	NULL

❖ 관계 연산자

> >= = <> <= <

❖ 논리 연산자

NOT

AND

OR

[실습2] SELECT (데이터 조회)

[실습 2]

-- customer(cno, cname, city, point)

-- 2-1) 고객번호가 c101 인 고객의 모든 정보를 검색하라

```
SELECT *  
FROM customer  
WHERE cno = 'c101';
```

[실습] (customer.sql)

Result Grid					Filter Rows:
	cno	cname	city	point	
▶	c101	홍길동	서울	500	
	c102	임걱정	인천	300	
	c103	박찬호	안양	800	
	c204	신동엽	과천	350	
	c205	정진우	고양	400	
	c307	정동우	서울	NULL	
*	NULL	NULL	NULL	NULL	



	cno	cname	city	point
▶	c101	홍길동	서울	500

[실습2] SELECT (데이터 조회)

[실습 2]

-- customer(cno, cname, city, point)

-- 2-2) 포인트가 400 이하인 고객의 모든 정보를 검색하라

```
SELECT *  
FROM customer  
WHERE point <= 400;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c102	임꺽정	인천	300
	c204	신동엽	과천	350
	c205	정진우	고양	400

[실습2] SELECT (데이터 조회)

[실습 2]

-- customer(cno, cname, city, point)

-- 2-3) 거주지가 서울 이면서 포인트가 500 이상인 고객의 이름, 거주지, 포인트를
검색하라

[실습] (customer.sql)

Result Grid					Filter Rows:
	cno	cname	city	point	
▶	c101	홍길동	서울	500	
	c102	임걱정	인천	300	
	c103	박찬호	안양	800	
	c204	신동엽	과천	350	
	c205	정진우	고양	400	
	c307	정동우	서울	NULL	
✱	NULL	NULL	NULL	NULL	



```
SELECT  cname, city, point
FROM    customer
WHERE   city = '서울' AND point >= 500;
```

	cname	city	point
▶	홍길동	서울	500

[실습2] SELECT (데이터 조회)

[실습 2]

-- customer(cno, cname, city, point)

-- 2-4) 거주지가 서울 이거나 포인트가 500 이상인 고객의 이름, 거주지, 포인트를
검색하라

```
SELECT  cname, city, point
FROM    customer
WHERE   city = '서울' OR point >= 500;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cname	city	point
▶	홍길동	서울	500
	박찬호	안양	800
	정동우	서울	NULL

[실습2] SELECT (데이터 조회)

[실습] (SELECT 기초, NULL, LIKE, Order by)

```
/* customer(cno, cname, city, point) */
```

-- 고객(고객번호, 고객명, 거주지, 포인트)

-- 2-5) 포인트가 350 부터 500 사이인 고객이름, 거주지, 포인트를 검색하라

-- 부등호 사용

-- BETWEEN ... AND 사용

-- 2-6) 거주지가 서울 이거나 안양인 고객번호, 이름, 거주지를 검색하라

-- 부등호 사용

-- IN 사용

-- 2-7) 거주지가 서울이 아니거나 안양이 아닌 고객번호, 이름, 거주지를 검색하라

-- 부등호 사용

-- NOT IN 사용

[실습] (customer.sql)

Result Grid					Filter Rows:
	cno	cname	city	point	
▶	c101	홍길동	서울	500	
	c102	임걱정	인천	300	
	c103	박찬호	안양	800	
	c204	신동엽	과천	350	
	c205	정진우	고양	400	
	c307	정동우	서울	NULL	
•	NULL	NULL	NULL	NULL	

❖ SELECT 문법

BETWEEN ... AND

IN

[실습2] SELECT (데이터 조회)

[실습 2]

- customer(cno, cname, city, point)
- 2-5) 포인트가 350 부터 500 사이인 고객이름, 거주지, 포인트를 검색하라
- 부등호 사용
- BETWEEN ... AND 사용

```
SELECT  cname, city, point
FROM    customer
WHERE   point >= 350 AND point <= 500;
```

```
SELECT  cname, city, point
FROM    customer
WHERE   point BETWEEN 250 AND 500;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cname	city	point
▶	홍길동	서울	500
	신동엽	과천	350
	정진우	고양	400

[실습2] SELECT (데이터 조회)

[실습 2]

- customer(cno, cname, city, point)
- 2-6) 거주지가 서울 이거나 안양인 고객번호, 이름, 거주지를 검색하라
- 부등호 사용
- IN 사용

```
SELECT cno, cname, city
FROM customer
WHERE city = '서울' OR city = '안양';
```

```
SELECT cno, cname, city
FROM customer
WHERE city IN ('서울', '안양');
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cno	cname	city
▶	c101	홍길동	서울
	c103	박찬호	안양
	c307	정동우	서울

[실습2] SELECT (데이터 조회)

[실습 2]

- customer(cno, cname, city, point)
- 2-6) 거주지가 서울 이거나 안양인 고객번호, 이름, 거주지를 검색하라
- 부등호 사용
- IN 사용

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임궽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



```
SELECT cno, cname, city
FROM customer
WHERE city <> '서울' AND city <> '안양';
```

```
SELECT cno, cname, city
FROM customer
WHERE city NOT IN ('서울', '안양');
```

	cno	cname	city
▶	c102	임궽정	인천
	c204	신동엽	과천
	c205	정진우	고양

[실습3] SELECT (데이터 조회)

[실습] (SELECT 기초, NULL, LIKE, Order by)

```
/* customer(cno, cname, city, point) */
```

-- 고객(고객번호, 고객명, 거주지, 포인트)

- 3-1) 정씨 성을 가진 고객의 모든 열을 검색하라
- 3-2) 이름에 '동' 자가 들어가는 고객의 모든 열을 검색하라
- 3-3) 이름의 세번째 글자가 '우' 자가 들어가는 고객의 모든 열을 검색하라
- 3-4) 성이 홍씨, 박씨, 정씨인 고객을 검색하라
- 성이 홍씨, 박씨, 정씨가 아닌 고객을 검색하라
- 3-5) 포인트가 없는 고객의 번호, 이름, 포인트를 검색하라
- 포인트가 있는 고객의 번호, 이름, 포인트를 검색하라

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
●	NULL	NULL	NULL	NULL

❖ 문자열 검색

% 0개 이상 검색

_ 1개 검색

❖ NULL 값 검색

IS NULL

IS NOT NULL

[실습3] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 3]

-- customer(cno, cname, city, point)

-- 3-1) 정씨 성을 가진 고객의 모든 열을 검색하라

Result Grid					Filter Rows:
	cno	cname	city	point	
▶	c101	홍길동	서울	500	
	c102	임걱정	인천	300	
	c103	박찬호	안양	800	
	c204	신동엽	과천	350	
	c205	정진우	고양	400	
	c307	정동우	서울	NULL	
•	NULL	NULL	NULL	NULL	



SELECT *

FROM customer

WHERE cname LIKE '정%';

	cno	cname	city	point
▶	c205	정진우	고양	400
	c307	정동우	서울	NULL

[실습3] SELECT (데이터 조회)

[실습 3]

-- customer(cno, cname, city, point)

-- 3-2) 이름에 '동' 자가 들어가는 고객의 모든 열을 검색하라

[실습] (customer.sql)

Result Grid					Filter Rows:
	cno	cname	city	point	
▶	c101	홍길동	서울	500	
	c102	임걱정	인천	300	
	c103	박찬호	안양	800	
	c204	신동엽	과천	350	
	c205	정진우	고양	400	
	c307	정동우	서울	NULL	
*	NULL	NULL	NULL	NULL	



SELECT *

FROM customer

WHERE cname LIKE '%동%';

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c204	신동엽	과천	350
	c307	정동우	서울	NULL

[실습3] SELECT (데이터 조회)

[실습 3]

-- customer(cno, cname, city, point)

-- 3-3) 이름의 세번째 글자가 '우' 자가 들어가는 고객의 모든 열을 검색하라

[실습] (customer.sql)

Result Grid					Filter Rows:
	cno	cname	city	point	
▶	c101	홍길동	서울	500	
	c102	임걱정	인천	300	
	c103	박찬호	안양	800	
	c204	신동엽	과천	350	
	c205	정진우	고양	400	
	c307	정동우	서울	NULL	
•	NULL	NULL	NULL	NULL	



```
SELECT *  
FROM customer  
WHERE cname LIKE '__우';
```

	cno	cname	city	point
▶	c205	정진우	고양	400
	c307	정동우	서울	NULL

[실습3] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 3]

- customer(cno, cname, city, point)
- 3-4) 성이 홍씨, 박씨, 정씨인 고객을 검색하라
- 성이 홍씨, 박씨, 정씨가 아닌 고객을 검색하라

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



```
SELECT *  
FROM customer  
WHERE cname LIKE '홍%' or cname LIKE '박%' or cname LIKE '정%';
```

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c103	박찬호	안양	800
	c205	정진우	고양	400
	c307	정동우	서울	NULL

```
SELECT *  
FROM customer  
WHERE cname NOT LIKE '홍%' AND cname NOT LIKE '박%' AND cname NOT LIKE '정%';
```

	cno	cname	city	point
▶	c102	임꺽정	인천	300
	c204	신동엽	과천	350

[실습3] SELECT (데이터 조회)

[실습] (customer.sql)

[실습 3]

- customer(cno, cname, city, point)
- 3-5) 포인트가 없는 고객의 번호, 이름, 포인트를 검색하라
- 포인트가 있는 고객의 번호, 이름, 포인트를 검색하라

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



```
SELECT cno, cname, point
FROM customer
WHERE point IS NULL;
```

	cno	cname	point
▶	c307	정동우	NULL

```
SELECT cno, cname, point
FROM customer
WHERE point IS NOT NULL;
```

	cno	cname	point
▶	c101	홍길동	500
	c102	임꺽정	300
	c103	박찬호	800
	c204	신동엽	350
	c205	정진우	400

[요약] SELECT (정렬)

❖ 정렬: **ORDER BY**

오름차순 (Ascending order) **ASC**

내림차순 (Descending order) **DESC**

SELECT
FROM
ORDER BY

SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY

[실습4] SELECT (데이터 조회)

[실습] (customer.sql)

[실습] (SELECT 기초, NULL, LIKE, Order by)

```
/* customer(cno, cname, city, point) */
```

```
-- 고객(고객번호, 고객명, 거주지, 포인트)
```

```
-- 4-1) 고객 테이블에서 이름을 오름차순 정렬하라
```

```
-- 4-2) 거주지가 서울인 고객의 모든 데이터를 검색하는데, 이름의 오름차순 정렬하여 출력하라
```

```
-- 4-3) 거주지의 오름차순으로 정렬하고, 거주지가 같으면 포인트의 내림차순으로 정렬하라
```

```
-- 4-4) 포인트가 많은 순으로(내림차순) 먼저 정렬하고, 같은 포인트는 이름의 오름차순으로 정렬하고
```

```
-- 이름이 같으면 거주지의 오름차순으로 정렬하여 검색하라
```

```
-- 4-5) 다음의 의미는?
```

```
SELECT      cno, cname, city, point
```

```
FROM        customer
```

```
ORDER BY 3;
```

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
●	NULL	NULL	NULL	NULL

[실습4] SELECT (정렬)

[실습 4]

-- customer(cno, cname, city, point)

-- 4-1) 고객 테이블에서 이름을 오름차순 정렬하라

```
SELECT *  
FROM customer  
ORDER BY cname ASC;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c102	임꺽정	인천	300
	c307	정동우	서울	NULL
	c205	정진우	고양	400
	c101	홍길동	서울	500

[실습4] SELECT (정렬)

[실습 4]

-- customer(cno, cname, city, point)

-- 4-2) 거주지가 서울인 고객의 모든 데이터를 검색하는데,
이름의 오름차순 정렬하여 출력하라

```
SELECT *  
FROM customer  
WHERE city = '서울'  
ORDER BY cname;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임걱정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c307	정동우	서울	NULL
	c101	홍길동	서울	500

[실습4] SELECT (정렬)

[실습 4]

-- customer(cno, cname, city, point)

-- 4-3) 거주지의 오름차순으로 정렬하고, 거주지가 같으면
포인트의 내림차순으로 정렬하라

```
SELECT *  
FROM customer  
ORDER BY city ASC, point DESC;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c205	정진우	고양	400
	c204	신동엽	과천	350
	c101	홍길동	서울	500
	c307	정동우	서울	NULL
	c103	박찬호	안양	800
	c102	임꺽정	인천	300

[실습4] SELECT (정렬)

[실습 4]

- customer(cno, cname, city, point)
- 4-4) 포인트가 많은 순으로(내림차순) 먼저 정렬하고,
- 같은 포인트는 이름의 오름차순으로 정렬하고
- 이름이 같으면 거주지의 오름차순으로 정렬하여 검색하라

```
SELECT *  
FROM customer  
ORDER BY point DESC, cname ASC, city ASC;
```

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
*	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c103	박찬호	안양	800
	c101	홍길동	서울	500
	c205	정진우	고양	400
	c204	신동엽	과천	350
	c102	임꺽정	인천	300
	c307	정동우	서울	NULL

[실습4] SELECT (정렬)

[실습 4]

-- customer(cno, cname, city, point)

-- 4-5) 다음의 의미는?

SELECT cno, cname, city, point

FROM customer

ORDER BY 3;

SELECT cno, cname, city, point
FROM customer
ORDER BY 3;

[실습] (customer.sql)

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임꺽정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
•	NULL	NULL	NULL	NULL



	cno	cname	city	point
▶	c205	정진우	고양	400
	c204	신동엽	과천	350
	c101	홍길동	서울	500
	c307	정동우	서울	NULL
	c103	박찬호	안양	800
	c102	임꺽정	인천	300

[과제02] (데이터베이스 관리)

6주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[과제02] (MySQL)

(MySQL)

테이블 생성(자유주제)

SELECT문제 만들기
(화면캡처 본문작성)

□ [과제02] **MySQL 프로그램 사용**하여 테이블 작성하고
SELECT 문제 만들고 실습하시오

- 1) 제출방법: 네이버카페-게시판제출: 과제
게시글에 **답글**로 작성하고 **게시글 본문**에 화면캡처 첨부
- 2) 테이블 생성(자유주제: 단, 학과,학생,부서,사원은 제외)
SELECT 문제, SQL 답안, 실행결과 화면캡처
- 3) 제출마감일: 20/10/19 (월) 23:55 까지

* (작성한 SQL파일은 저장해 두세요: 각자 별도로 보관해 두시기 바랍니다)

[과제02] SELECT 문제 작성

다음은 샘플예제입니다(동일한 내용 작성하는 것 아닙니다)

[테이블]

	번호	이름	생년월일	키	몸무게
▶	1	최민식	1962-04-27	177	70
	2	류승룡	1970-11-29	NULL	NULL
	4	한석규	1964-11-03	NULL	NULL
	5	송강호	1967-01-17	180	80
	6	이병헌	NULL	177	72
	7	한효주	NULL	172	48
	12	차태현	1976-03-25	175	65
	NULL	NULL	NULL	NULL	NULL

배우 1 ×

[문제]

- 문제 예시
- 1) 이름을 오름차순으로 정렬하라
- 2) 키가 175~177 인 배우이름을 검색하라
- 3) 생년월일이 입력되지 않은 배우의 이름, 생년월일을 검색하라
- 4) 한씨 성을 배역의 배우번호, 배역을 검색하라
- 5) 번호가 2 또는 4 또는 6번의 이름, 생년월일을 검색하라

[테이블]

Result Grid

Filter Rows:

	cno	cname	city	point
▶	c101	홍길동	서울	500
	c102	임격정	인천	300
	c103	박찬호	안양	800
	c204	신동엽	과천	350
	c205	정진우	고양	400
	c307	정동우	서울	NULL
	NULL	NULL	NULL	NULL

[문제]

- 문제 예시
- 1) 포인트가 400 이하인 고객의 모든 정보 검색
- 2) 포인트가 350 부터 500 사이인 고객이름, 거주지, 포인트를 검색하라
- 3) 포인트가 없는 고객의 번호, 이름, 포인트를 검색하라
- 4) 정씨 성을 가진 고객의 모든 열을 검색하라
- 5) 고객 테이블에서 이름을 오름차순 정렬하라