

데이터베이스관리

5주차

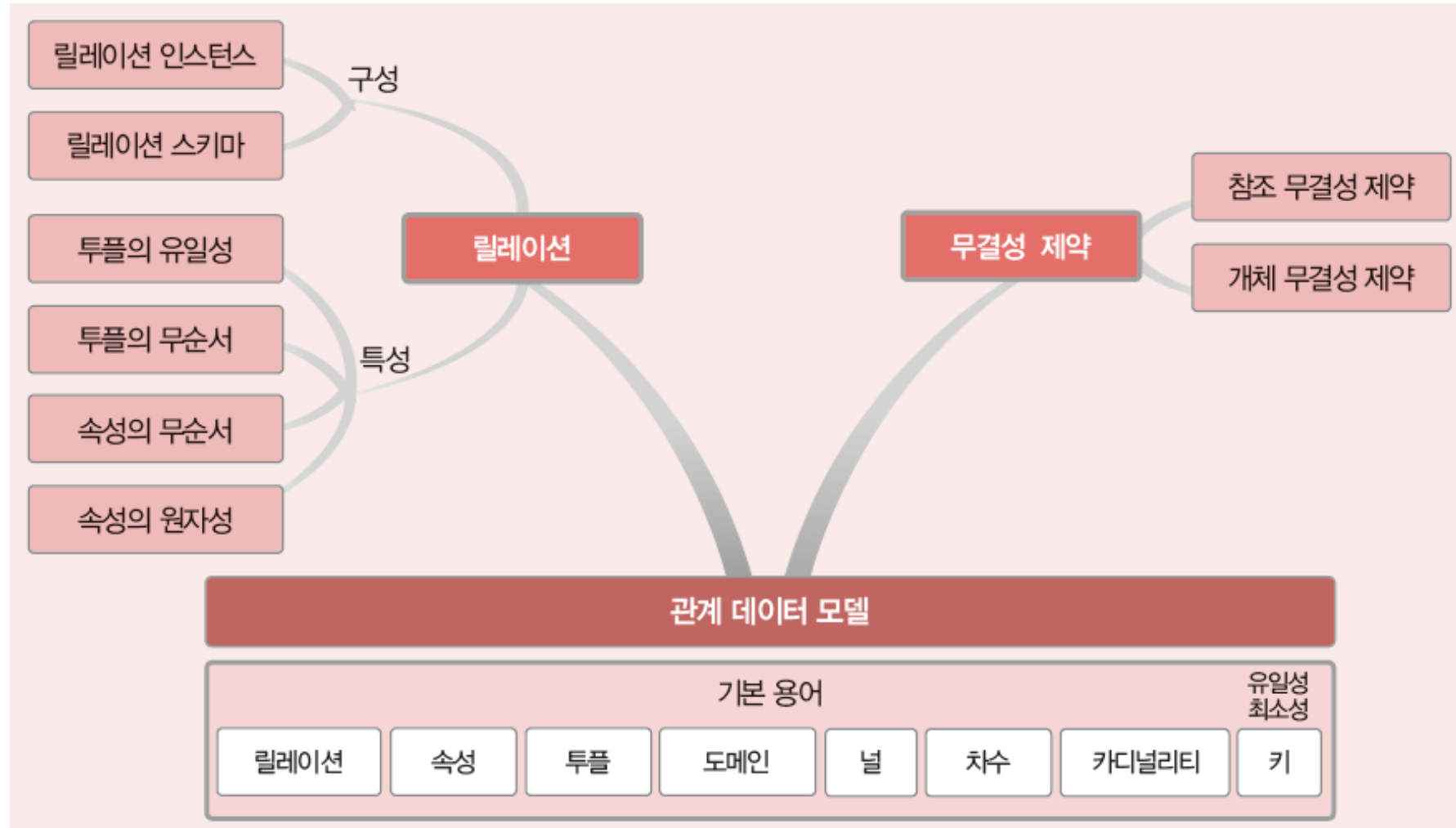
담당교수: 김희숙
(jasmin11@hanmail.net)

관계형 데이터베이스

5주차 5-01

담당교수: 김희숙
(jasmin11@hanmail.net)

[요약] 관계형 데이터베이스



01 관계 데이터 모델의 개념

- 관계 데이터 모델의 기본 개념

- 개념적 구조를 논리적 구조로 표현하는 논리적 데이터 모델
- 하나의 개체에 대한 데이터를 하나의 **릴레이션**에 저장

열(속성, 애트리뷰트)

| 고객아이디 | 고객이름 | 나이 | 등급 | 직업 | 적립금 |
|----------|----------|-----|----------|----------|------|
| CHAR(20) | CHAR(20) | INT | CHAR(10) | CHAR(10) | INT |
| apple | 김현준 | 20 | gold | 학생 | 1000 |
| banana | 정소화 | 25 | vip | 간호사 | 2500 |
| carrot | 원유선 | 28 | gold | 교사 | 4500 |
| orange | 정지영 | 22 | silver | 학생 | 0 |

도메인

행(튜플)

그림 5-1 릴레이션의 예 : 고객 릴레이션

01 관계 데이터 모델의 개념

- 관계 데이터 모델의 기본 용어

- 릴레이션(relation)**

- 하나의 개체에 관한 데이터를 **2차원 테이블**의 구조로 저장한 것
- 파일 관리 시스템 관점에서 파일(file)에 대응

- 속성(attribute)**

- 릴레이션의 **열, 애트리뷰트**
- 파일 관리 시스템 관점에서 **필드(field)**에 대응

- 투플(tuple)**

- 릴레이션의 **행**
- 파일 관리 시스템 관점에서 **레코드(record)**에 대응

| 고객아이디 | 고객이름 | 나이 | 등급 | 직업 | 적립금 |
|----------|----------|-----|----------|----------|------|
| CHAR(20) | CHAR(20) | INT | CHAR(10) | CHAR(10) | INT |
| apple | 김현준 | 20 | gold | 학생 | 1000 |
| banana | 정소화 | 25 | vip | 간호사 | 2500 |
| carrot | 원유선 | 28 | gold | 교사 | 4500 |
| orange | 정지영 | 22 | silver | 학생 | 0 |

그림 5-1 릴레이션의 예 : 고객 릴레이션

01 관계 데이터 모델의 개념

• 관계 데이터 모델의 기본 용어

• 도메인(domain)

- 하나의 속성이 가질 수 있는 모든 값의 집합
- 속성 값을 입력 및 수정할 때 적합성 판단의 기준이 됨
- 일반적으로 속성의 특성을 고려한 데이터 타입으로 정의

• 널(null)

- 속성 값을 아직 모르거나 해당되는 값이 없음을 표현

• 차수(degree)

- 하나의 릴레이션에서 속성의 전체 개수

• 카디널리티(cardinality)

- 하나의 릴레이션에서 튜플의 전체 개수

열(속성, 애트리뷰트)

| 고객아이디 | 고객이름 | 나이 | 등급 | 직업 | 적립금 |
|----------|----------|-----|----------|----------|------|
| CHAR(20) | CHAR(20) | INT | CHAR(10) | CHAR(10) | INT |
| apple | 김현준 | 20 | gold | 학생 | 1000 |
| banana | 정소화 | 25 | vip | 간호사 | 2500 |
| carrot | 원유선 | 28 | gold | 교사 | 4500 |
| orange | 정지영 | 22 | silver | 학생 | 0 |

행(튜플)

도메인

그림 5-1 릴레이션의 예 : 고객 릴레이션

01 관계 데이터 모델의 개념

- 릴레이션의 구성

- 릴레이션 **스키마(relation schema)**
 - 릴레이션의 논리적 구조
 - 릴레이션의 이름과 릴레이션에 포함된 모든 속성 이름으로 정의
 - 예) **고객(고객아이디, 고객이름, 나이, 등급, 직업, 적립금)**
 - 릴레이션 **내포(relation intension)**라고도 함
 - **정적**인 특징이 있음
- 릴레이션 **인스턴스(relation instance)**
 - 어느 한 시점에 릴레이션에 존재하는 튜플들의 집합
 - 릴레이션 **외연(relation extension)**이라고도 함
 - **동적**인 특징이 있음

01 관계 데이터 모델의 개념

- 릴레이션의 구성

| 고객아이디 | 고객이름 | 나이 | 등급 | 직업 | 적립금 |
|--------|------|----|--------|-----|------|
| apple | 김현준 | 20 | gold | 학생 | 1000 |
| banana | 정소화 | 25 | vip | 간호사 | 2500 |
| carrot | 원유선 | 28 | gold | 교사 | 4500 |
| orange | 정지영 | 22 | silver | 학생 | 0 |

릴레이션 스키마

릴레이션 인스턴스

그림 5-2 릴레이션 구성의 예 : 고객 릴레이션

01 관계 데이터 모델의 개념

- 데이터베이스의 구성

- 데이터베이스 스키마(database schema)
 - 데이터베이스의 전체 구조
 - 데이터베이스를 구성하는 릴레이션 스키마의 모음
- 데이터베이스 인스턴스(database instance)
 - 데이터베이스를 구성하는 릴레이션 인스턴스의 모음

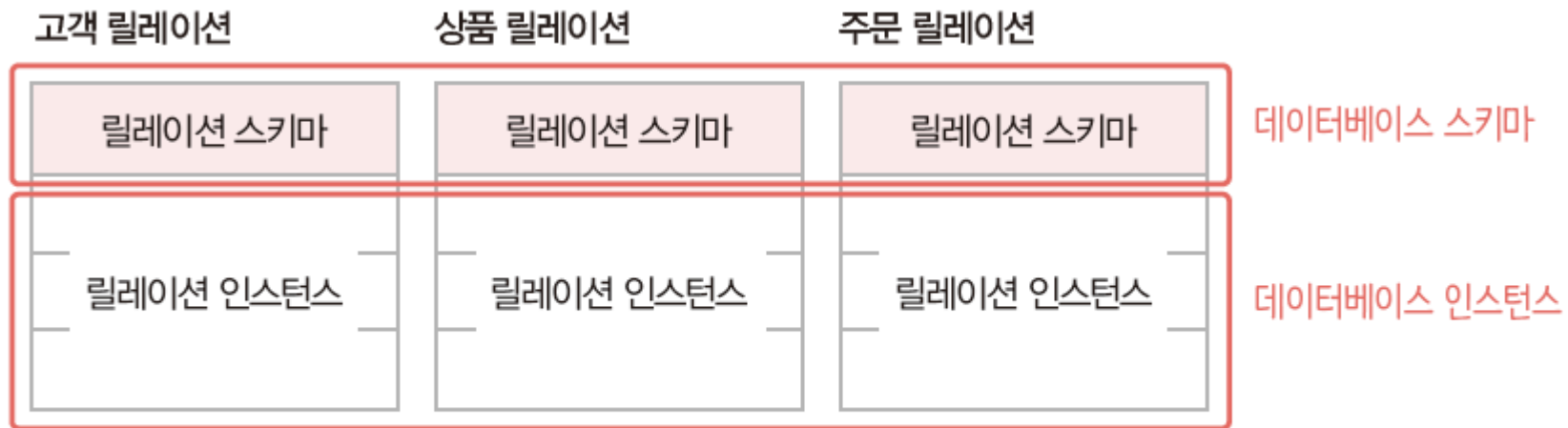


그림 5-3 데이터베이스 구성의 예 : 쇼핑몰 데이터베이스

01 관계 데이터 모델의 개념

- 릴레이션의 특성

- 튜플의 유일성
 - 하나의 릴레이션에는 동일한 튜플이 존재할 수 없다.
- 튜플의 무순서
 - 하나의 릴레이션에서 튜플 사이의 순서는 무의미하다.
- 속성의 무순서
 - 하나의 릴레이션에서 속성 사이의 순서는 무의미하다.
- 속성의 원자성
 - 속성 값으로 원자 값만 사용할 수 있다.

01 관계 데이터 모델의 개념

- 릴레이션의 특성

| 고객아이디 | 고객이름 | 나이 | 등급 | 직업 | 적립금 |
|--------|------|----|--------|---------|------|
| apple | 김현준 | 20 | gold | 학생 | 1000 |
| banana | 정소화 | 25 | vip | 간호사 | 2500 |
| carrot | 원유선 | 28 | gold | 교사 | 4500 |
| orange | 정지영 | 22 | silver | 회사원, 학생 | 0 |

그림 5-4 다중 값 속성을 포함하는 릴레이션의 예 : 고객 릴레이션

01 관계 데이터 모델의 개념

- 키(key)

- 릴레이션에서 튜플들을 유일하게 구별하는 속성 또는 속성들의 집합

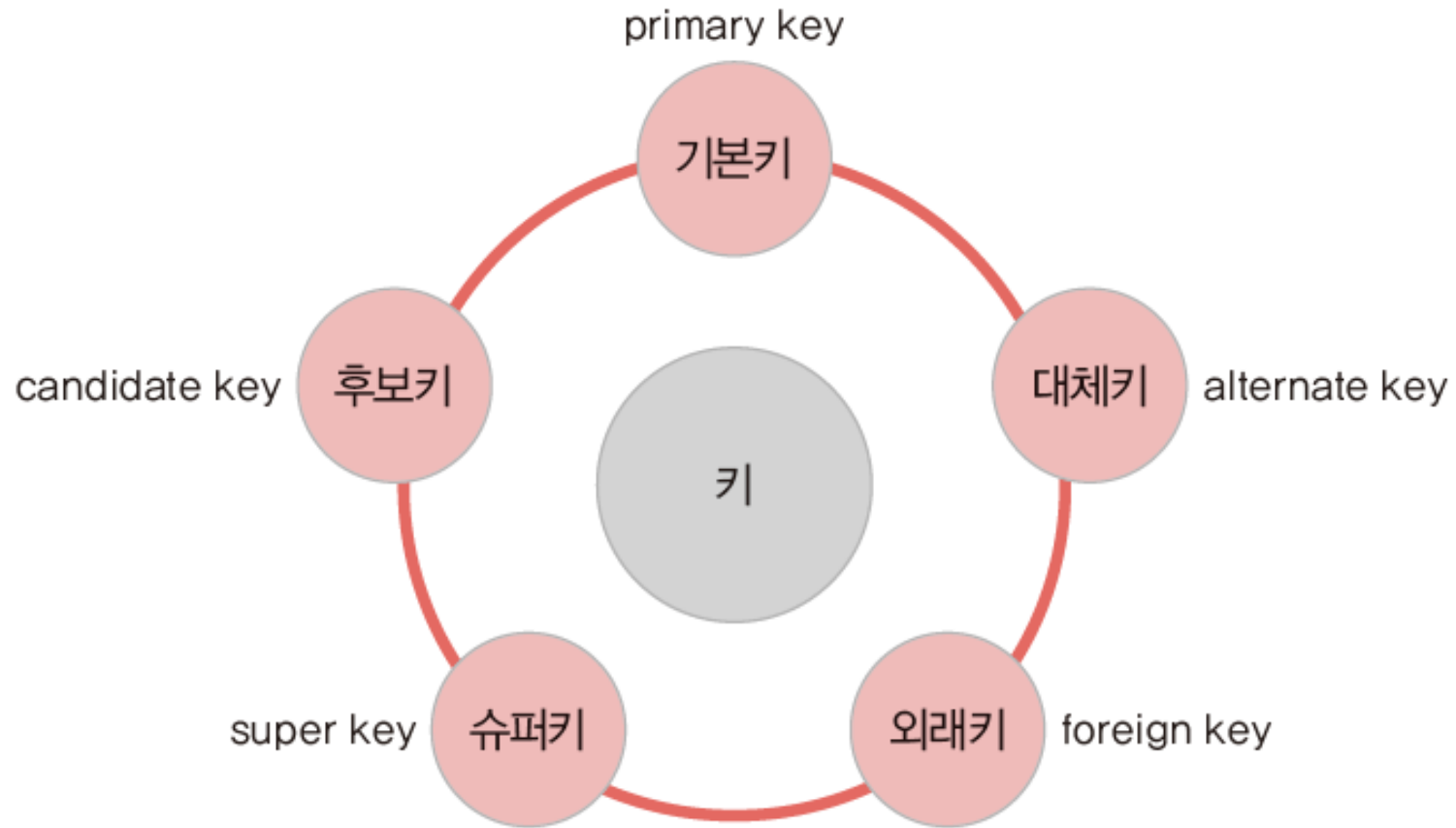


그림 5-5 키의 종류

01 관계 데이터 모델의 개념

- 키의 특성

- 유일성(uniqueness)
 - 하나의 릴레이션에서 모든 튜플은 서로 다른 키 값을 가져야 함
- 최소성(minimality)
 - 꼭 필요한 최소한의 속성들로만 키를 구성함

01 관계 데이터 모델의 개념

- 키의 종류

- 수퍼키(super key)

- 유일성을 만족하는 속성 또는 속성들의 집합
 - 예) 고객 릴레이션의 수퍼키 :
고객아이디, (고객아이디, 고객이름), (고객이름, 주소) 등

- 후보키(candidate key)

- 유일성과 최소성을 만족하는 속성 또는 속성들의 집합
 - 예) 고객 릴레이션의 후보키 : **고객아이디**, (**고객이름**, **주소**) 등

- 기본키(primary key)

- 후보키 중에서 기본적으로 사용하기 위해 선택한 키
 - 예) 고객 릴레이션의 기본키 : 고객아이디

01 관계 데이터 모델의 개념

- 키의 종류

- 대체키(alternate key)
 - 기본키로 선택되지 못한 후보키
 - 예) 고객 릴레이션의 대체키 : (고객이름, 주소)

기본키
↓

| <u>고객아이디</u> | 고객이름 | 나이 | 등급 | 직업 | 적립금 | 주소 |
|--------------|------|----|--------|-----|------|-----------------|
| apple | 김현준 | 20 | gold | 학생 | 1000 | 서울시 구로구 고척로 27 |
| banana | 정소화 | 25 | vip | 간호사 | 2500 | 부천시 원미구 석천로 58 |
| carrot | 원유선 | 28 | gold | 교사 | 4500 | 서울시 영등포구 대림로 73 |
| orange | 정지영 | 22 | silver | 학생 | 0 | 서울시 마포구 토정로 56 |

그림 5-7 주소 속성이 추가된 릴레이션의 예 : 고객 릴레이션

01 관계 데이터 모델의 개념

- 키의 종류

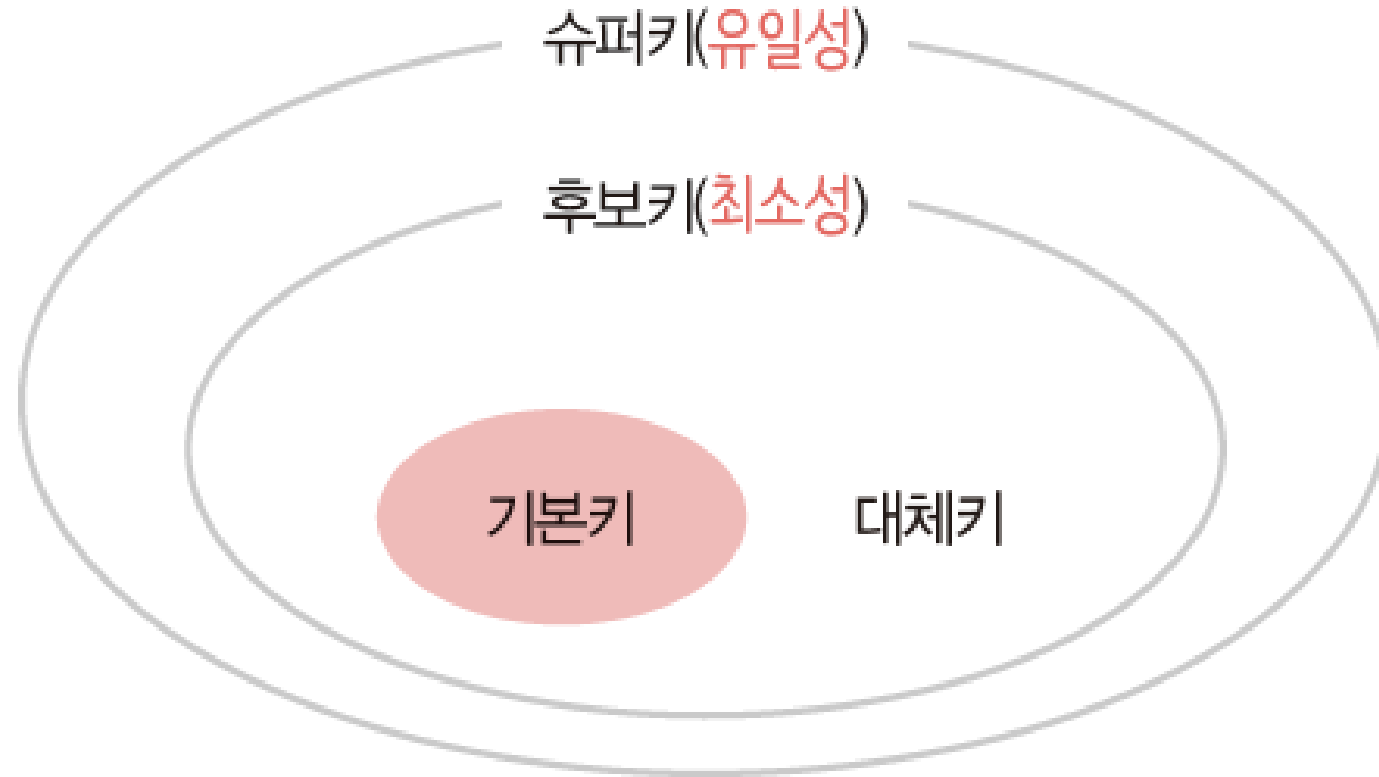


그림 5-8 키의 관계

01 관계 데이터 모델의 개념

• 키의 종류

- 외래키(foreign key)
 - 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합
 - 릴레이션들 간의 관계를 표현
 - 참조하는 릴레이션 : 외래키를 가진 릴레이션
 - 참조되는 릴레이션 : 외래키가 참조하는 기본키를 가진 릴레이션

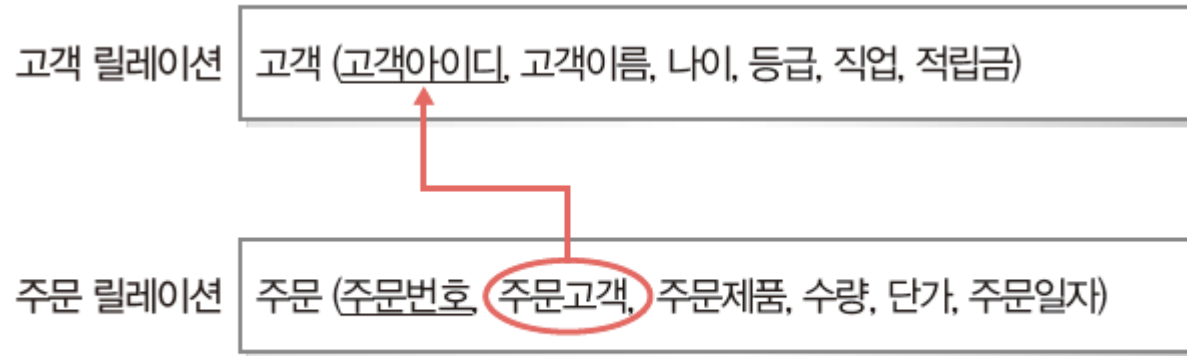


그림 5-9 고객 릴레이션과 주문 릴레이션의 스키마

01 관계 데이터 모델의 개념



그림 5-10 외래키의 예 : 고객 릴레이션과 주문 릴레이션

외래키 속성과 그것이 참조하는 기본키 속성의 이름은 달라도 되지만 **도메인은 같아야** 한다

01 관계 데이터 모델의 개념

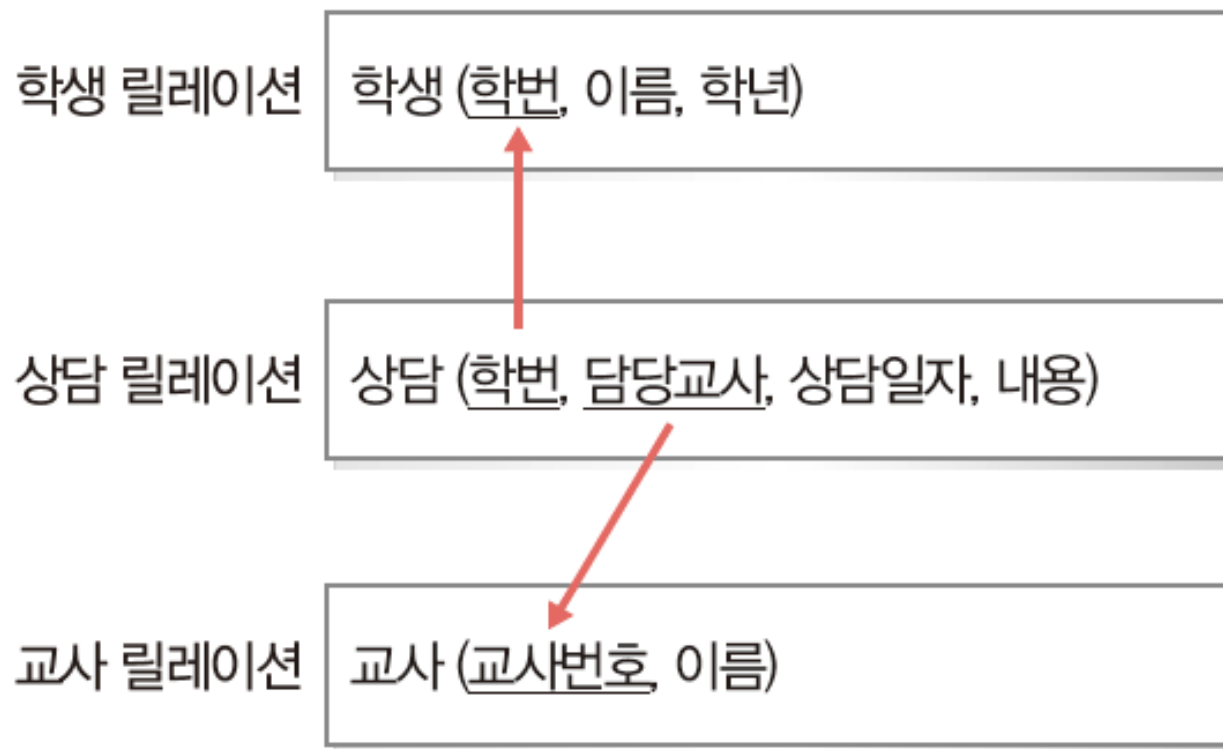


그림 5-11 학생 상담 데이터베이스 스키마

하나의 릴레이션에는 외래키가 여러 개 존재할 수도 있고
외래키를 기본키로 사용할 수도 있다.

01 관계 데이터 모델의 개념

고객 릴레이션의 기본키

| 고객아이디 | 고객이름 | 나이 | 등급 | 직업 | 적립금 | 추천고객 |
|--------|------|----|--------|-----|------|--------|
| apple | 김현준 | 20 | gold | 학생 | 1000 | orange |
| banana | 정소화 | 25 | vip | 간호사 | 2500 | orange |
| carrot | 원유선 | 28 | gold | 교사 | 4500 | apple |
| orange | 정지영 | 22 | silver | 학생 | 0 | NULL |

고객 릴레이션의 외래키

그림 5-12 기본키와 외래키의 관계가 함께 정의된 릴레이션의 예 : 고객 릴레이션

같은 릴레이션의 기본키를 참조하는 외래키도 정의할 수 있다.
그리고 외래키 속성은 널 값을 가질 수도 있다

02 관계 데이터 모델의 제약

- 무결성 제약조건(integrity constraint)
 - 데이터의 무결성을 보장하고 일관된 상태로 유지하기 위한 규칙
 - 무결성 : 데이터를 결함이 없는 상태, 즉 정확하고 유효하게 유지하는 것

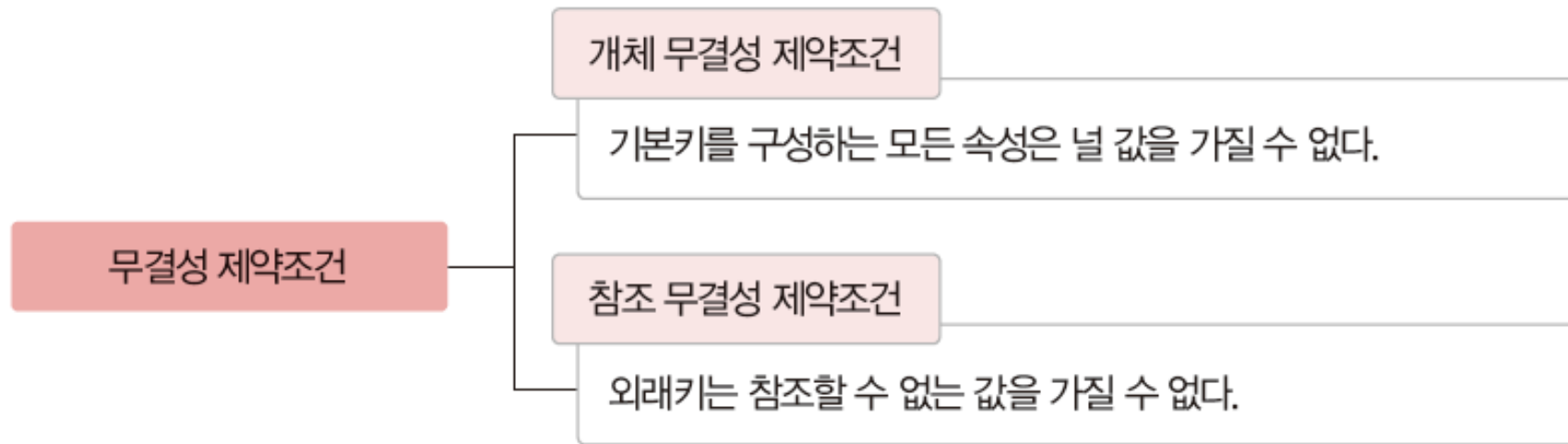


그림 5-13 관계 데이터 모델의 무결성 제약조건

02 관계 데이터 모델의 제약

- 개체 무결성 제약조건(entity integrity constraint)
 - 기본키를 구성하는 모든 속성은 널 값을 가질 수 없는 규칙

| <u>고객아이디</u> | 고객이름 | 나이 | 등급 | 직업 | 적립금 |
|--------------|------|----|--------|-----|------|
| apple | 김현준 | 20 | gold | 학생 | 1000 |
| NULL | 정소화 | 25 | vip | 간호사 | 2500 |
| carrot | 원유선 | 28 | gold | 교사 | 4500 |
| NULL | 정지영 | 22 | silver | 학생 | 0 |

그림 5-14 개체 무결성 제약조건을 위반한 릴레이션의 예 : 고객 릴레이션

02 관계 데이터 모델의 제약

- 참조 무결성 제약조건(referential integrity constraint)
 - 외래키는 참조할 수 없는 값을 가질 수 없는 규칙



02 관계 데이터 모델의 제약



그림 5-16 외래키가 널 값인 릴레이션의 예 : 주문 릴레이션

외래키 속성이 널 값을 가진다고 해서 참조 무결성 제약조건을 위반한 것은 아니다.

[요약] 키

* 키(key): 각 튜플을 고유하게 식별할 수 있는 하나 이상의 애트리뷰트 들의 모임

1) **수퍼키**(Super Key): (유일성)

2) **후보키**(Candidate Key): (유일성, 최소성)

각 튜플을 고유하게 식별하는 최소한의 애트리뷰트 들의 모임

3) **기본키**(Primary Key):

릴레이션에 존재하는 N개의 튜플들을 유일하게 구별할 수 있는 후보키 중에서 선택된 하나의 키

4) **대체키**(Alternate Key):

하나의 릴레이션에 존재하는 후보키들 중에서 기본키로 선택되지 않은 키

5) **외래키**(Foreign Key):

개체와 개체를 상호 연결하거나 참조할 수 있는 키로서 한 릴레이션의 외래키는 참조되는 릴레이션의 기본키와 상호 대응된다.

* 6) **대리키**(Surrogate Key):

식별자가 너무 길거나 여러 개의 속성으로 구성되어 있는 경우에 인위적으로 추가한 식별자

[요약] 기본키 설정 (대리키)

| course_id | year | semester | division | prof_id | classroom | enroll |
|-----------|------|----------|----------|---------|-----------|--------|
| C101 | 2012 | 1 | A | 92301 | 301호 | 40 |
| C102 | 2012 | 1 | A | 92001 | 209호 | 30 |
| C103 | 2012 | 1 | A | 92501 | 208호 | 30 |
| C103 | 2012 | 1 | B | 92301 | 301호 | 30 |
| C501 | 2012 | 1 | A | 92501 | 103호 | 45 |
| C501 | 2012 | 1 | B | 92502 | 204호 | 25 |
| C301 | 2012 | 2 | A | 92502 | 301호 | 30 |
| C302 | 2012 | 2 | A | 92501 | 209호 | 45 |
| C502 | 2012 | 2 | A | 92001 | 209호 | 30 |
| C502 | 2012 | 2 | B | 92301 | 103호 | 26 |

| class_id | course_id | year | semester | division | prof_id | classroom | enroll |
|----------|-----------|------|----------|----------|---------|-----------|--------|
| C101-01 | C101 | 2012 | 1 | A | 92301 | 301호 | 40 |
| C102-01 | C102 | 2012 | 1 | A | 92001 | 209호 | 30 |
| C103-01 | C103 | 2012 | 1 | A | 92501 | 208호 | 30 |
| C103-02 | C103 | 2012 | 1 | B | 92301 | 301호 | 30 |
| C501-01 | C501 | 2012 | 1 | A | 92501 | 103호 | 45 |
| C501-02 | C501 | 2012 | 1 | B | 92502 | 204호 | 25 |
| C301-01 | C301 | 2012 | 2 | A | 92502 | 301호 | 30 |
| C302-01 | C302 | 2012 | 2 | A | 92501 | 209호 | 45 |
| C502-01 | C502 | 2012 | 2 | A | 92001 | 209호 | 30 |
| C502-02 | C502 | 2012 | 2 | B | 92301 | 103호 | 26 |

[Quiz] 신입생

1. 기본키(Primary key) 는?
2. 수퍼키(Super key) 는?
3. 후보키(Candidate key) 는?
4. 대체키(Alternate key) 는?

| 학번 | 주민등록번호 | 이름 | 주소 | 학과명 |
|---------|----------------|-----|----|--------|
| 1292001 | 900424-1825409 | 김광식 | 서울 | 컴퓨터공학과 |
| 1292002 | 900305-1730021 | 김정현 | 서울 | 컴퓨터공학과 |
| 1292003 | 891021-2308302 | 김현정 | 대전 | 컴퓨터공학과 |
| 1292301 | 890902-2704012 | 김현정 | 대구 | 산업공학과 |

[Quiz] 개설강좌

1. 기본키(Primary key) 는?

개설강좌

| 과목코드 | 연도 | 학기 | 분반 | 담당교수 | 강의실 | 수강인원 |
|------|------|----|----|-------|------|------|
| C101 | 2012 | 1 | A | 92301 | 301호 | 40 |
| C102 | 2012 | 1 | A | 92001 | 209호 | 30 |
| C103 | 2012 | 1 | A | 92501 | 208호 | 30 |
| C103 | 2012 | 1 | B | 92301 | 301호 | 30 |

개설강좌

| 강좌코드 | 과목코드 | 연도 | 학기 | 분반 | 담당교수 | 강의실 | 수강인원 |
|---------|------|------|----|----|-------|------|------|
| C101-01 | C101 | 2012 | 1 | A | 92301 | 301호 | 40 |
| C102-01 | C102 | 2012 | 1 | A | 92001 | 209호 | 30 |
| C103-01 | C103 | 2012 | 1 | A | 92501 | 208호 | 30 |
| C103-02 | C103 | 2012 | 1 | B | 92301 | 301호 | 30 |

[Quiz] 개설강좌

1. 기본키(Primary key) 는?

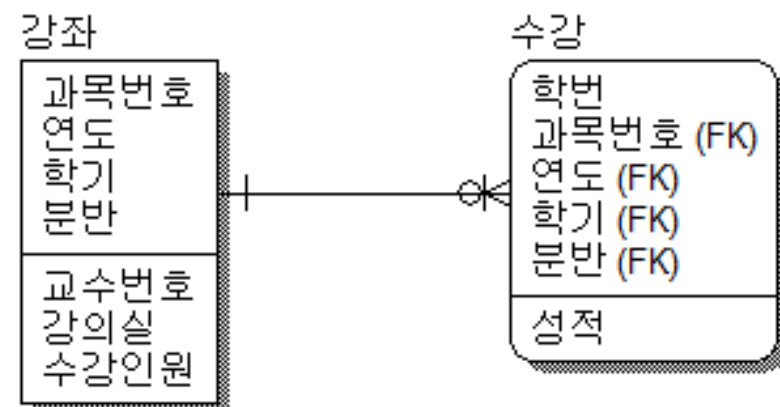
개설강좌

| 과목코드 | 연도 | 학기 | 분반 | 담당교수 | 강의실 | 수강인원 |
|------|------|----|----|-------|------|------|
| C101 | 2012 | 1 | A | 92301 | 301호 | 40 |
| C102 | 2012 | 1 | A | 92001 | 209호 | 30 |
| C103 | 2012 | 1 | A | 92501 | 208호 | 30 |
| C103 | 2012 | 1 | B | 92301 | 301호 | 30 |

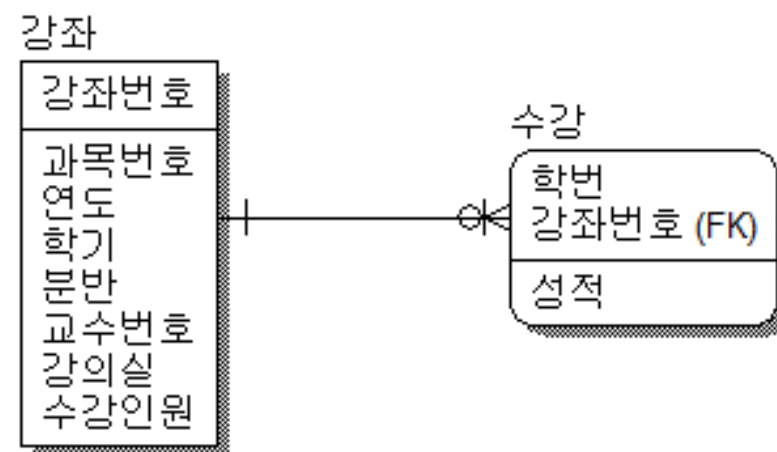
개설강좌

| 강좌코드 | 과목코드 | 연도 | 학기 | 분반 | 담당교수 | 강의실 | 수강인원 |
|---------|------|------|----|----|-------|------|------|
| C101-01 | C101 | 2012 | 1 | A | 92301 | 301호 | 40 |
| C102-01 | C102 | 2012 | 1 | A | 92001 | 209호 | 30 |
| C103-01 | C103 | 2012 | 1 | A | 92501 | 208호 | 30 |
| C103-02 | C103 | 2012 | 1 | B | 92301 | 301호 | 30 |

1) 기본키가 복합키인 경우.



2) 기본키가 대리키인 경우.



SQL: CREATE 문법

5주차 5-02

담당교수: 김희숙
(jasmin11@hanmail.net)

[요약] 데이터 형식

❖ 데이터 형식(Data types)

| -MS SQL Server | | -Oracle | | -MySQL | |
|----------------|-----------------------|-------------|--------------|--------------|---------------|
| tinyint | 1 byte (0~255) | number(3) | -999~999 | tinyint | -128~+127 |
| smallint | 2 byte (-32768~32767) | number | | smallint | 2 byte |
| int | 4 byte | int | | int | 4 byte |
| bigint | 8 byte | | | bigint | 8 byte |
| float | 8 byte | number(4,1) | -999.9~999.9 | float | 예) float(7,4) |
| real | 8 byte | | | double(4,1) | -999,9999 |
| decimal(4,1) | -999.9~999.9 | | | decimal(4,1) | |
| numeric(4,1) | | | | | |
| char() | 1 byte | char() | 최대 2000 byte | char() | 0~255 |
| varchar() | 최대 8000 byte | varchar2() | 최대 4000 byte | varchar() | 고정길이문자 |
| nchar() | | nchar() | | tinytext | 0~65526 가변길이 |
| nvarchar() | 최대 4000 byte | nvarchar2() | | text | 0~255 문자열 |
| | | | | | 0~ 약 7만개 문자열 |
| datetime | | date | | date | YYYY-MM-DD |
| | | timestamp | | datetime | YYYY-MM-DD |
| | | | | | HH:MM:SS |

[요약] 데이터 형식

문자형 데이터 형식

고정문자열

char(8)

| | | | | | | | |
|---|---|---|---|--|--|--|--|
| d | a | t | a | | | | |
|---|---|---|---|--|--|--|--|

가변문자열

varchar(8)

| | | | |
|---|---|---|---|
| d | a | t | a |
|---|---|---|---|

[요약] 데이터 형식

숫자형 데이터 형식 (MySQL)

```
CREATE TABLE zerofill_test (  
  A int(11) ,  
  B int(11) unsigned zerofill ,  
  C int(4) ,  
  D int(4) unsigned zerofill  
);
```

```
-- 1, 123, 1234567890
```

```
INSERT INTO zerofill_test VALUES(1,1,1,1);
```

```
INSERT INTO zerofill_test VALUES(123,123,123,123);
```

```
INSERT INTO zerofill_test VALUES(1234567890,1234567890,1234567890,1234567890);
```

```
select * from zerofill_test;
```

| | A | B | C | D |
|---|------------|--------------|------------|------------|
| ▶ | 1 | 000000000001 | 1 | 0001 |
| | 123 | 00000000123 | 123 | 0123 |
| | 1234567890 | 01234567890 | 1234567890 | 1234567890 |

zerofill: 나머지 자릿수를 0으로 채운다

[실습] 테이블 생성(테이블 2개)

[실습 1-1] (테이블 2개) 학과, 학생 테이블 생성(기본키, 외래키)

```
/* 2개 테이블 (기본키, 외래키실습) : 학과, 학생 (studept-1.sql) */
/* -----*/
```

--테이블 삭제

```
drop table 학생;
[ ] 학과;
```

--학과(학과코드, 학과명)

```
[ ] TABLE 학과 (
  학과코드 char(2) NOT NULL ,
  학과명 varchar(30) ,
  [ ] KEY ( [ ])
);
```

1. 부모 테이블

--학생(학번, 학생명, 학년, 학과코드)

```
[ ] TABLE 학생 (
  학번 char(4) NOT NULL ,
  학생명 varchar(12) ,
  학년 int ,
  학과코드 [ ] ,
  PRIMARY KEY ( [ ]) ,
  FOREIGN KEY ( [ ]) REFERENCES [ ](학과코드)
);
```

2. 자식 테이블

❖ 학과, 학생

테이블 생성 순서

테이블 삭제 순서

데이터 입력 순서

[실습] 테이블 생성(테이블 3개)

[실습 1-2] (테이블 3개) 학생, 수강, 과목 테이블 생성(기본키, 외래키)

[3개 테이블 : 기본키, 외래키]

학생

| 학번 | 학생명 | 학년 |
|------|-----|----|
| 1111 | 홍길동 | 1 |
| 2222 | 김윤식 | 3 |
| 3333 | 이정진 | 2 |
| 4444 | 홍진아 | 1 |

수강

| 학번 | 과목번호 | 성적 |
|------|-------|----|
| 1111 | CS100 | 98 |
| 1111 | CS102 | 88 |
| 2222 | CS102 | 90 |
| 3333 | CS100 | 92 |

과목

| 과목번호 | 과목명 |
|-------|--------|
| CS100 | 데이터베이스 |
| CS101 | 운영체제 |
| CS102 | 자료구조 |

❖ 학과, 학생

테이블 생성 순서

테이블 삭제 순서

데이터 입력 순서



[실습 1-2] (테이블 3개) 학생, 수강, 과목 테이블 생성(기본키, 외래키)

❖ 기본키가 복합키인 경우

```
drop table ;
drop table ;
drop table 학생;
```

```
--학생 (학번, 학생명, 학년)
 학생 (
    학번 char(4) NOT NULL,
    학생명 varchar(12),
    학년 int,
     KEY(학번)
);
```

```
--과목 (과목번호, 과목명)
 과목 (
    과목번호 char(5),
    과목명 varchar(30),
     KEY(과목번호)
);
```

```
--수강 (학번, 과목번호, 성적)
 수강 (
    학번 ,
    과목번호 char(5),
    성적 int,
     KEY() ,
     KEY (학번) REFERENCES 
     KEY (과목번호) REFERENCES 
);
```

[3개 테이블 : 기본키, 외래키]

학생

| 학번 | 학생명 | 학년 |
|------|-----|----|
| 1111 | 홍길동 | 1 |
| 2222 | 김윤식 | 3 |
| 3333 | 이정진 | 2 |
| 4444 | 홍진아 | 1 |

수강

| 학번 | 과목번호 | 성적 |
|------|-------|----|
| 1111 | CS100 | 98 |
| 1111 | CS102 | 88 |
| 2222 | CS102 | 90 |
| 3333 | CS100 | 92 |

과목

| 과목번호 | 과목명 |
|-------|--------|
| CS100 | 데이터베이스 |
| CS101 | 운영체제 |
| CS102 | 자료구조 |

[실습] 테이블 생성(자체 릴레이션)

[실습 1-3] (테이블 1개) emp 테이블 생성(자체 릴레이션)

(테이블 1개)

emp

외래키

| empno | empname | manager | dno |
|-------|---------|---------|-----|
| 1003 | 조민희 | 3011 | 1 |
| 2106 | 김창섭 | 3426 | 2 |
| 3011 | 이수민 | | 1 |
| 3426 | 박영권 | 3011 | 3 |
| 3427 | 최종철 | 2106 | 3 |

| 테이블명 | 열 이름 | 데이터 형식 | NULL 유무 | 기본키 | 외래키 | FK 테이블명 | FK 열 이름 | 비고 |
|------|---------|-------------|----------|-----|-----|---------|---------|----|
| emp | empno | char(4) | NOT NULL | PK | | | | |
| | empname | varchar(10) | | | | | | |
| | manager | ? | | | ? | ? | ? | |
| | dno | char(1) | | | | | | |

[실습] 테이블 생성(자체 릴레이션)

[실습 1-3] (테이블 1개) emp 테이블 생성

| 테이블명 | 열 이름 | 데이터 형식 | NULL 유무 | 기본키 | 외래키 | FK 테이블명 | FK 열 이름 | 비고 |
|------|---------|-------------|----------|-----|-----|---------|---------|----|
| emp | empno | char(4) | NOT NULL | PK | | | | |
| | empname | varchar(10) | | | | | | |
| | manager | ? | | | ? | ? | ? | |
| | dno | char(1) | | | | | | |

```

1 • use studydb;
2
3 • drop table if exists emp;
4
5 • create table emp (
6     empno char(4) PRIMARY KEY ,
7     empname varchar(10) ,
8     manager char(4) ,
9     dno char(1) ,
10    [ ] KEY (manager) REFERENCES [ ]([ ])
11 );
12
13 • insert into emp values('3011','이수민',NULL,'1');
14 • insert into emp values('3426','박영권','3011','3');
15 • insert into emp values('1003','조민희','3011','1');
16 • insert into emp values('2106','김창섭','3426','2');
17 • insert into emp values('3427','최종철','2106','3');
18

```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: [A]

| empno | empname | manager | dno |
|-------|---------|---------|------|
| 1003 | 조민희 | 3011 | 1 |
| 2106 | 김창섭 | 3426 | 2 |
| 3011 | 이수민 | NULL | 1 |
| 3426 | 박영권 | 3011 | 3 |
| 3427 | 최종철 | 2106 | 3 |
| NULL | NULL | NULL | NULL |

레코드간 간격 조정

[Quiz 2] 테이블 생성 (지금 이순간 작성)

다음 테이블을 생성하고 데이터를 입력하시오(자체 릴레이션)

5주차 영상 확인

네이버카페 [Quiz2 제출]
게시글에 댓글로 제출

-- 본인이름으로 레코드 1개 입력 추가하여 제출

-- 주의) 데이터 입력 순서

지금 제출하세요

다음을 실습하시오
MySQL 에서 작성하고
데이터조회 결과를 화면캡처하여
네이버카페 [Quiz2 제출]
게시글에 댓글로 제출



SQL: ALTER 문법

5주차 5-03

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



[실습] ALTER 문법

ALTER: 테이블의 구조를 변경 (필드 추가/수정/삭제)

```
CREATE TABLE project (  
  projno int NOT NULL ,  
  projname char(20) ,  
  period int  
);  
  
INSERT INTO project VALUES(111,'작전중',2);  
INSERT INTO project VALUES(222,'특공대',3);  
INSERT INTO project VALUES(444,'다모여',1);  
INSERT INTO project VALUES(333,'유레카',2);  
  
select * from project;
```

| projno | projname | period |
|--------|----------|--------|
| 111 | 작전중 | 2 |
| 222 | 특공대 | 3 |
| 444 | 다모여 | 1 |
| 333 | 유레카 | 2 |

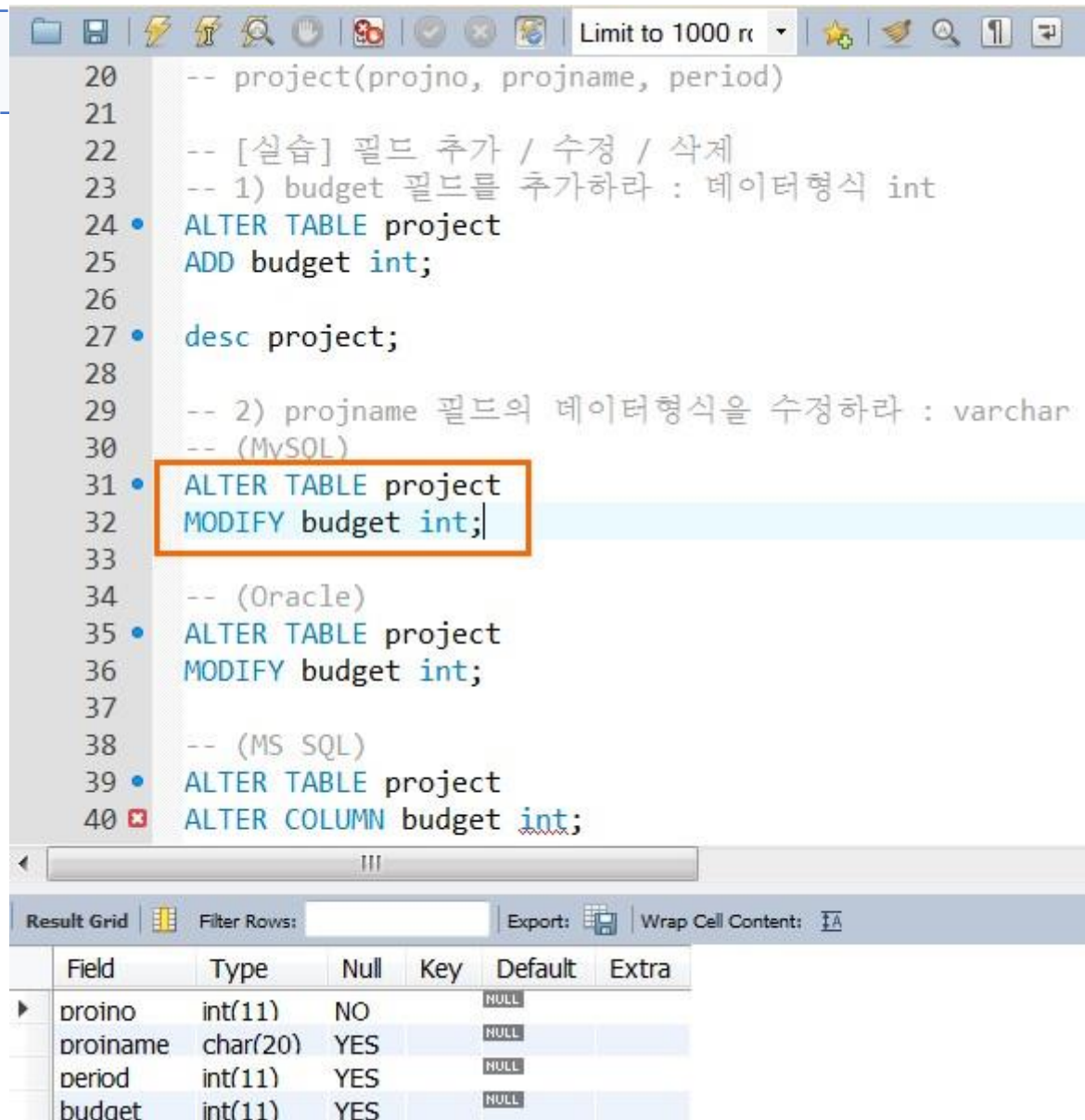
- [실습] 필드 추가 / 수정 / 삭제
- 1) budget 필드를 추가하라 : 데이터형식 int
- 2) projname 필드의 데이터형식을 수정하라 : varchar
- 3) budget 필드를 삭제하라

[실습] ALTER 문법

[실습] 필드 추가 / 수정 / 삭제 (project.sql)

| | |
|---|---|
| <pre>/* 필드 추가: 필드 맨 끝에 추가된다 */ ALTER TABLE project ADD budget int; /* 특정 필드 뒤에 추가 */ ALTER TABLE project ADD branch varchar(10) AFTER projname;</pre> | <pre>--데이터 조회 mysql>select * from project; --테이블 구조 mysql>desc project;</pre> |
| <pre>/* 필드 수정: 데이터형식만 수정 */ ALTER TABLE project MODIFY projname varchar(30); /* 필드 수정: 필드명, 데이터형식 수정가능 */ ALTER TABLE project CHANGE projname proj_name char(30);</pre> | <pre>/* 테이블 생성 후에 기본키 추가 */ ALTER TABLE project ADD CONSTRAINT pk_project_projno PRIMARY KEY(proj_no); /* 테이블 생성 후에 기본키 삭제 */ ALTER TABLE project DROP PRIMARY KEY;</pre> |
| <pre>/* 필드 삭제: budget */ ALTER TABLE project DROP budget;</pre> | |
| <pre>/* 테이블 이름 변경 */ ALTER TABLE project RENAME project_info;</pre> | |

[실습] 필드 추가 / 수정 / 삭제 (project.sql)



```
20 -- project(projno, projname, period)
21
22 -- [실습] 필드 추가 / 수정 / 삭제
23 -- 1) budget 필드를 추가하라 : 데이터형식 int
24 • ALTER TABLE project
25   ADD budget int;
26
27 • desc project;
28
29 -- 2) projname 필드의 데이터형식을 수정하라 : varchar
30 -- (MySQL)
31 • ALTER TABLE project
32   MODIFY budget int;
33
34 -- (Oracle)
35 • ALTER TABLE project
36   MODIFY budget int;
37
38 -- (MS SQL)
39 • ALTER TABLE project
40   ALTER COLUMN budget int;
```

Result Grid

| | Field | Type | Null | Key | Default | Extra |
|---|----------|----------|------|-----|---------|-------|
| ▶ | proino | int(11) | NO | | NULL | |
| | proiname | char(20) | YES | | NULL | |
| | period | int(11) | YES | | NULL | |
| | budaet | int(11) | YES | | NULL | |

[실습] 자체 릴레이션(self_prof.sql)

| 교수 번호 | 주민등록번호 | 이 름 | 학과명 | 학과장 |
|----------|----------------|--------|--------|-------|
| 92001 | 590327-1839240 | 이태규 | 컴퓨터공학과 | 92001 |
| 92002 | 690702-1350026 | 고희석 | 컴퓨터공학과 | 92001 |
| 92301 | 741011-2765501 | 최성희 | 산업공학과 | 92302 |
| 92302 | 750728-1102458 | 김태석 | 산업공학과 | 92302 |
| 92501 | 620505-1400546 | 박철재 | 전자공학과 | NULL |
| 92502 | 740101-1830264 | 장민석 | 전자공학과 | NULL |

```
-- 교수(교수번호,주민등록번호,이름,학과명,학과장)
select * from 교수;

-- 오류(
insert into 교수 values('99999','190101-399999','홍길동','컴퓨터공학과','11111');

-- 잘못 입력된 데이터 삭제
delete
from 교수
where 교수번호 = '99999';

-- 외래키 추가
ALTER TABLE 교수
ADD CONSTRAINT fk_교수_학과장
FOREIGN KEY (학과장) REFERENCES 교수(교수번호);

-- 입력 되지 않음
insert into 교수 values('99999','190101-399999','홍길동','컴퓨터공학과','11111');

select * from 교수;
```

[실습] 자체 릴레이션(self_prof.sql)

| 교수 번호 | 주민등록번호 | 이 름 | 학과명 | 학과장 |
|----------|----------------|--------|--------|-------|
| 92001 | 590327-1839240 | 이태규 | 컴퓨터공학과 | 92001 |
| 92002 | 690702-1350026 | 고희석 | 컴퓨터공학과 | 92001 |
| 92301 | 741011-2765501 | 최성희 | 산업공학과 | 92302 |
| 92302 | 750728-1102458 | 김태석 | 산업공학과 | 92302 |
| 92501 | 620505-1400546 | 박철재 | 전자공학과 | NULL |
| 92502 | 740101-1830264 | 장민석 | 전자공학과 | NULL |

```
drop table 교수;
```

```
CREATE TABLE 교수 (
    교수번호      char(5)      NOT NULL ,
    주민등록번호  char(14)     NOT NULL ,
    이름          varchar(20)  NOT NULL ,
    학과명        varchar(20)  NOT NULL ,
    학과장        char(5)      ,
    CONSTRAINT pk_교수_교수번호 PRIMARY KEY(교수번호)
);
```

```
insert into 교수 values('92001','590327-1839240','이태규','컴퓨터공학과','92001');
insert into 교수 values('92002','690702-1350026','고희석','컴퓨터공학과','92001');
insert into 교수 values('92301','741011-2765501','최성희','산업공학과','92302');
insert into 교수 values('92302','750728-1102458','김태석','산업공학과','92302');
insert into 교수 values('92501','620505-1400546','박철재','전자공학과',NULL);
insert into 교수 values('92502','740101-1830264','장민석','전자공학과',NULL);
```