데이터베이스관리

4주차

담당교수: 김희숙
(jasmin11@hanmail.net)

테이블 생성

4주차 4-03

담당교수: 김희숙
(jasmin11@hanmail.net)

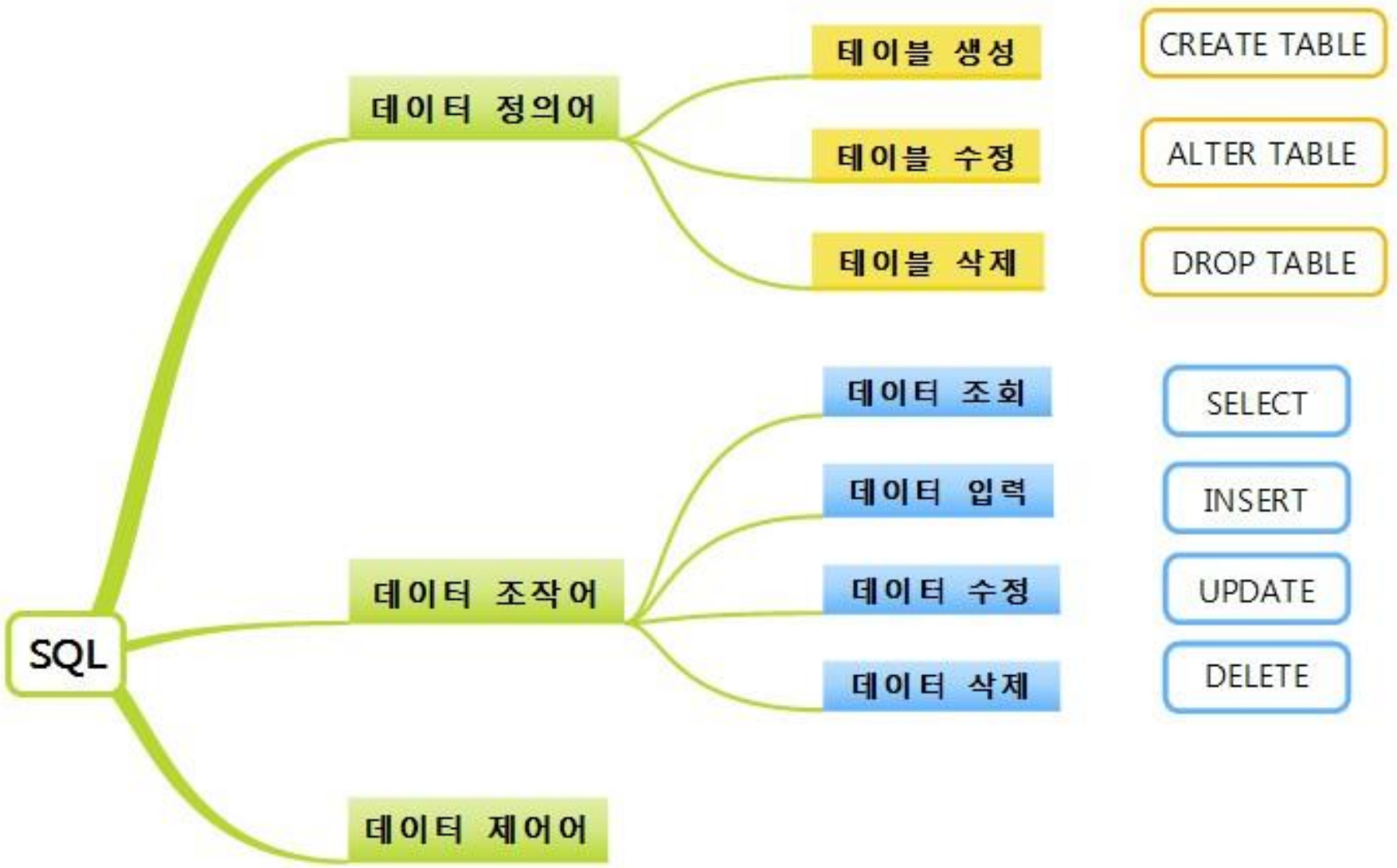
데이터베이스 언어(SQL)

❑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
데이터 정의어
데이터 조작어
데이터 제어어
테이블 생성
테이블 수정
테이블 삭제
데이터 조회
데이터 입력
데이터 수정
데이터 삭제
CREATE TABLE
ALTER TABLE
DROP TABLE
SELECT
INSERT
UPDATE
DELETE

[Quiz 1-1] 테이블 생성 (지금 이순간 작성)

[Quiz 1] 1) 스키마: studydb 생성

2) dept, dmember 테이블을 각각 생성하고 데이터 입력하시오 (dmemberdept-mysql.sql)

기본키, 외래키 주의사항

dept

dept_id	name
100	컴퓨터공학과
101	산업공학과

dmember

name	dept_id
김광식	100
김현정	101
조영수	101

1) 테이블 생성 순서는?

2) 테이블 삭제 순서는?

3) 데이터 입력 순서는?

테이블명세서

dept, dmember 테이블 생성

테이블명	열 이름	데이터 형식	NULL 유무	기본키	외래키	FK 테이블명	FK 열 이름	비고
dept	dept_id	char(3)	NOT NULL	PK				
	name	varchar(20)	NULL					

테이블명	열 이름	데이터 형식	NULL 유무	기본키	외래키	FK 테이블명	FK 열 이름	비고
dmember	name	varchar(10)	NOT NULL	PK				
	dept_id	?	NULL		FK	?	?	

[실습] [Quiz 1-1] 테이블 생성 (테이블 2개)

(지금 이순간 작성)

동양미래대학교 DONGYANG MIRAE UNIVERSITY

[Quiz 1-1] SQL 문법을 사용하여 테이블 생성 하고 데이터 입력하시오 (dmemberdept-mysql.sql)

```
-- dept, dmember 테이블 생성 (dmemberdept-mysql.sql)
use studydb;

DROP TABLE if exists 1) ;
DROP TABLE if exists 2) ;

3) TABLE dept (
  dept_id   char(3)                NOT NULL ,
  name      varchar(20) ,
  4) KEY(dept_id)
);

   TABLE dmember (
  name      varchar(10)            NOT NULL ,
  dept_id   5)                     ,
   KEY(name)     ,
  CONSTRAINT fk_dmember_dept 6)
);
```

```
INSERT INTO dept VALUES('100', '컴퓨터공학과');
INSERT INTO dept VALUES('101', '산업공학과');

insert into dmember values('김광식','100');
insert into dmember values('김현정','101');
insert into dmember values('조영수','101');
-- 본인이름으로 레코드 1개 입력 추가하여 제출
select * from dept;
select * from dmember;
```

name	dept_id
김광식	100
김희숙	100
김현정	101
조영수	101

지금 제출하세요

다음 빈칸을 완성하여 실습하시오
MySQL 에서 작성하고
데이터조회 결과를 화면캡처하여
네이버카페 [Quiz1-1제출]
게시글에 **댓글로 제출**

[요약] 기본키 설정

```
--부서 테이블 생성   (기본키 설정 방법)
--방법1
CREATE TABLE 부서1(
 부서코드      char(2)       NOT NULL    PRIMARY KEY,
 부서명        varchar(10)
);
--방법2 : 복합키인 경우 방법2 사용해야 한다.
CREATE TABLE 부서2(
 부서코드      char(2)       NOT NULL,
 부서명        varchar(10),
 PRIMARY KEY(부서코드)
);
--방법3
CREATE TABLE 부서3(
 부서코드      char(2)       NOT NULL   CONSTRAINT PK_부서3_부서코드 PRIMARY KEY,
 부서명        varchar(10)
);
--방법4
CREATE TABLE 부서4(
 부서코드      char(2)       NOT NULL,
 부서명        varchar(10),
 CONSTRAINT PK_부서4_부서코드 PRIMARY KEY(부서코드)
);
```

부서(부서코드, 부서명)

부서코드	부서명
AA	총무부
BB	영업부
CC	기획부

```
drop table 부서;

-- 부서 테이블 생성
CREATE TABLE 부서 (
        부서코드            char(2)  NOT NULL,
        부서명              varchar(10),
        PRIMARY KEY (부서코드)
);

-- 부서 데이터입력
insert into 부서 values('AA','총무부','서울');
insert into 부서 values('BB','영업부','대전');
insert into 부서 values('CC','기획부','서울 ' );

select * from 부서;
```

[요약] 기본키 설정

[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

--수강 테이블 생성 | 수강(학번, 과목번호, 학점)

```
CREATE TABLE 수강 (
    학번 char(5),
    과목번호 char(5),
    학점 char(2),
    PRIMARY KEY(학번, 과목번호),
    FOREIGN KEY (학번) REFERENCES 학생(학번),
    FOREIGN KEY (과목번호) REFERENCES 과목(과목번호)
);
```

기본키가 복합키 인 경우,

[요약] 외래키 설정

[2개 테이블 : 기본키, 외래키]

학과

학과코드	학과명
C1	소프트웨어정보
C2	인터넷정보
E3	정보통신
E4	정보전자

학생 fk

학번	학생명	학년	학과코드
1111	홍길동	1	C1
2222	김윤식	3	E3
3333	이정진	2	E4
4444	홍진아	1	C1

학과 테이블
기본키 :
외래키 :

학생 테이블
기본키 :
외래키 :

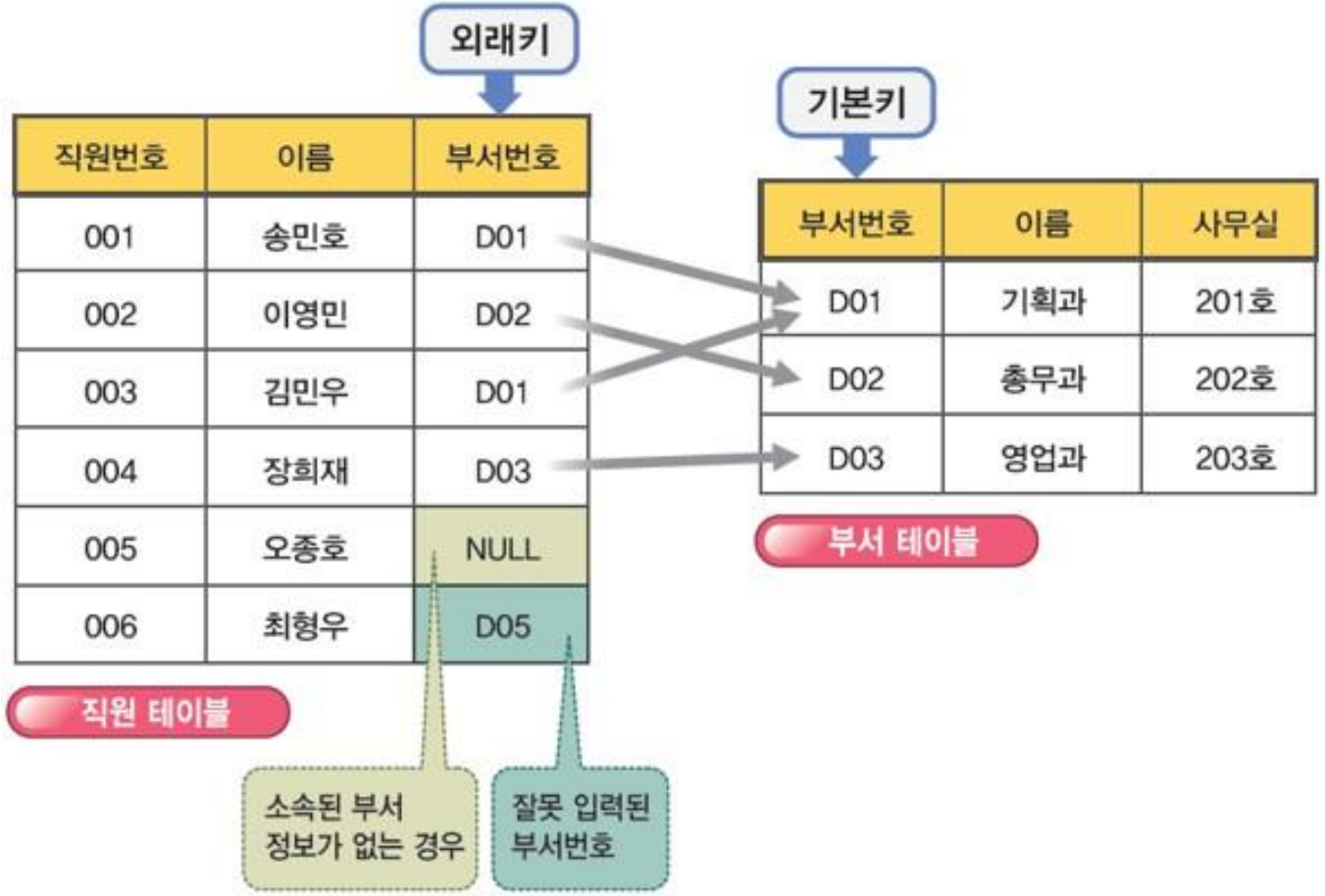
의사

의사번호	의사이름	소속	근무연수
D001	정지영	내과	5
D002	김선주	피부과	10
D003	정성호	정형외과	15

환자 외래키

사원번호	환자이름	나이	담당의사
P001	오우진	31	D002
P002	채광주	50	D001
P003	김용욱	43	D003

[요약] 외래키 설정



❖ 외래키(참조무결성 제약조건)

✓ 외래키가 설정된 필드에는 반드시 참조되는 부모테이블의 기본키 필드값에 있는 것만 입력가능

✓ 외래키 설정된 필드는 반드시 부모테이블의 기본키 필드값의 데이터형식과 동일해야 한다(도메인이 같아야)

(그림 출처: "Understanding of Database", 이상구외 공저, 이한, 2012)

[요약] 테이블 생성과 삭제

❖ 테이블 생성과 테이블 삭제

- ✓ 테이블 생성시, 부모 테이블 먼저 생성
- ✓ 테이블 삭제시, 자식 테이블 먼저 삭제

부모테이블

부서

부서코드	부서명
AA	총무부
BB	영업부
CC	기획부

자식테이블

사원

외래키

사원번호	이름	부서코드
1111	홍길동	AA
2222	임꺽정	AA
3333	박찬호	BB
4444	선동열	BB
5555	차두리	AA
6666	신동엽	BB

[실습] 테이블 생성

[실습 1-1] SQL 문법을 사용하여 테이블 생성 하고 데이터 입력하시오 (empdept-non.sql)

(empdept-non.sql)

```
drop table 사원;
drop table 부서;

CREATE TABLE 부서 (
        부서코드 char(2)      NOT NULL ,
        부서명   varchar(10) ,
        PRIMARY KEY(부서코드)
);

CREATE TABLE 사원 (
        사원번호 char(4)      NOT NULL ,
        이름     varchar(12) ,
        부서코드 char(2) ,
        PRIMARY KEY(사원번호) ,
        FOREIGN KEY (부서코드) REFERENCES 부서(부서코드)
);
```

[실습] 테이블 생성(제약조건 설정) CONSTRAINT

[실습 1-2] SQL 문법을 사용하여 테이블 생성 하고 데이터 입력하시오 (empdept.sql)

(empdept.sql)

```
drop table 사원;
drop table 부서;

CREATE TABLE 부서 (
        부서코드 char(2)      NOT NULL ,
        부서명   varchar(10) ,
        constraint pk_부서_부서코드 PRIMARY KEY(부서코드)
);

CREATE TABLE 사원 (
        사원번호 char(4)      NOT NULL ,
        이름     varchar(12) ,
        부서코드 char(2) ,
        constraint pk_사원_사원번호 PRIMARY KEY(사원번호) ,
        constraint fk_사원_부서코드 FOREIGN KEY (부서코드) REFERENCES 부서(부서코드)
);
```

키의 종류

4주차 4-02

담당교수: 김희숙
(jasmin11@hanmail.net)

키(Key)

❑기본키(Primary Key)

- ✓각 투플을 유일하게 구별할 수 있는 하나 이상의 속성의 집합
- ✓중복불가
- ✓필수입력

❑외래키(Foreign Key)

- ✓어떤 릴레이션의 기본키를 참조하는 키

학과 **PK**

학과번호	학과명
1	컴퓨터소프트웨어공학과
2	컴퓨터정보공학과
3	정보통신과

학생 **PK** **FK**

번호	이름	학년	분반	학과번호
1	한지혜	1	YB	1
2	이정우	1	YA	1
3	오지영	2	J1	2
4	강재미	1	YB	1
5	박철호	2	J1	2

키(Key)

❑키

- 각 투플을 고유하게 식별할 수 있는 하나 이상의 애트리뷰트 들의 모임

✓ 수퍼키(super key)
✓ 후보키(candidate key)
✓ 기본키(primary key)
✓ 대체키(alternate key)
✓ 외래키(foreign key)

키	특성
수퍼키	유일성
후보키	**유일성, 최소성**
기본키	중복불가, 필수입력
대체키	
외래키	

[요약] 기본키 설정 (후보키)

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
1292303	910715-1524390	박광수	광주	산업공학과
1292305	921011-1809003	김우주	부산	산업공학과
1292501	900825-1506390	박철수	대전	전자공학과
1292502	911011-1809003	백태성	서울	전자공학과

❖ 신입생

후보키: {학번}, {주민등록번호}

기본키: {학번}

대체키: {주민등록번호}

수퍼키는?

후보키의 성질: 유일성, 최소성

[요약] 기본키 설정 (후보키)

[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

❖ 수강

후보키: {학번, 과목번호}

기본키:

대체키:

수퍼키는?

후보키의 성질: 유일성, 최소성

데이터 입력

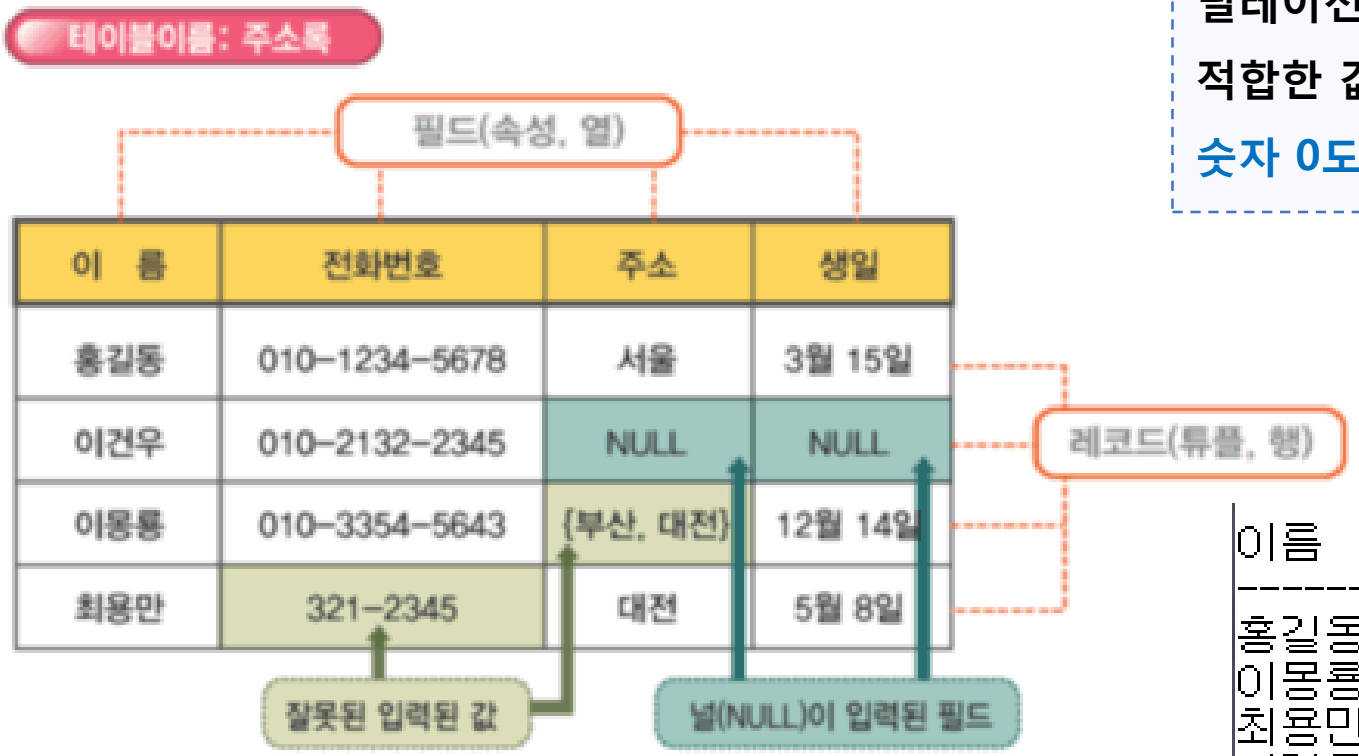
4주차 4-03

담당교수: 김희숙
(jasmin11@hanmail.net)

[요약] 널 값(NULL)

❖ **널 값**: 알려지지 않은 값(Unknown)

릴레이션에 있는 특정 투플의 속성값을 모르거나
적합한 값이 없는 경우
숫자 0도 아니고 공백도 아니다



(그림 출처: "Understanding of Database", 이상구외 공저, 이한, 2012)

이름	전화번호	주소	생일
홍길동	010-1234-5678	서울	3월 15일
이몽룡	010-3354-5643	부산	12월 14일
최용만	321-2345	대전	5월 8일
이건우	010-2132-2345	NULL	

[실습] (MySQL) 널 값 입력

[실습] 널(NULL) 값 입력하는 방법

```
-- 주소록(이름, 전화번호, 주소, 생일)
CREATE TABLE 주소록(
  이름        char(10) NOT NULL ,
  전화번호     char(13) ,
  주소        varchar(10) ,
  생일        varchar(11) ,
  PRIMARY KEY(이름)
);
```

```
INSERT INTO 주소록
VALUES('홍길동','010-1234-5678','서울','3월 15일');
```

이름	전화번호	주소	생일
홍길동	010-1234-5678	서울	3월 15일
이몽룡	010-3354-5643	부산	12월 14일
최용만	321-2345	대전	5월 8일
이건우	010-2132-2345	NULL	NULL

```
--널 값 데이터 입력하는 방법
INSERT INTO 주소록(이름, 전화번호, 주소, 생일)
VALUES('이건우','010-2132-2345', NULL, NULL);
```

-- **널 값 데이터 입력**하는 방법

INSERT INTO 주소록(이름, 전화번호, 주소, 생일)
VALUES('이건우','010-2132-2345', **NULL, NULL**);

INSERT INTO 주소록**(이름, 전화번호)**
VALUES('이건우','010-2132-2345');

[실습] (MySQL) 자동증가: auto_increment

[실습] 자동증가 없음

```
-- 주소록(이름, 전화번호, 주소, 생일)
CREATE TABLE 주소록(
  이름        char(10) NOT NULL ,
  전화번호     char(13) ,
  주소        varchar(10) ,
  생일        varchar(11) ,
  PRIMARY KEY(이름)
);
```

[실습] 자동증가 있음

```
-- 주소록(번호, 이름, 전화번호, 주소, 생일)
CREATE TABLE 주소록(
  번호        int       AUTO_INCREMENT ,
  이름        char(10) NOT NULL ,
  전화번호     char(13) ,
  주소        varchar(10) ,
  생일        varchar(11) ,
  PRIMARY KEY(번호)
```

```
INSERT INTO 주소록(이름, 전화번호, 주소, 생일)
VALUES('홍길동','010-1234-5678','서울','3월 15일');
```

[실습] (MySQL) 자동증가: auto_increment


```
-- 주소록(이름, 전화번호, 주소, 생일)

-- 주소록 테이블 생성
CREATE TABLE 주소록(
  이름        char(10) NOT NULL ,
  전화번호     char(13) ,
  주소        varchar(10) ,
  생일        varchar(11) ,
  PRIMARY KEY(이름)
);

-- 데이터 입력
-- 주소록(이름, 전화번호, 주소, 생일)

INSERT INTO 주소록(이름, 전화번호, 주소, 생일)
VALUES('홍길동','010-1234-5678','서울','3월 15일');

INSERT INTO 주소록(이름, 전화번호, 주소, 생일)
VALUES('이몽룡','010-3354-5643','부산','12월 14일');

INSERT INTO 주소록(이름, 전화번호, 주소, 생일)
VALUES('최용만','321-2345','대전','5월 8일');

-- 데이터 조회
select * from 주소록;
```

[실습] (MySQL) 자동증가: auto_increment


```
-- MySQL 실습: 자동증가 필드 추가

use studydb;

-- 주소록(이름, 전화번호, 주소, 생일)

drop table 주소록;

-- 번호 필드 추가한다(자동증가: auto_increment)
-- 주소록(번호, 이름, 전화번호, 주소, 생일)

-- 주소록 테이블 생성
CREATE TABLE 주소록(
  번호        int       AUTO_INCREMENT,
  이름        char(10) NOT NULL ,
  전화번호     char(13) ,
  주소        varchar(10) ,
  생일        varchar(11) ,
  PRIMARY KEY(번호)
);

-- 데이터 입력
-- 주소록(번호, 이름, 전화번호, 주소, 생일)

-- 입력 가능
INSERT INTO 주소록(이름, 전화번호, 주소, 생일)
VALUES('홍길동','010-1234-5678','서울','1990-03-15');

INSERT INTO 주소록(이름, 전화번호, 주소, 생일)
VALUES('이몽룡','010-3354-5643','부산','1994-12-14');

INSERT INTO 주소록(이름, 전화번호, 주소, 생일)
VALUES('최용만','321-2345','대전','1994-05-08');

-- 데이터 조회
select * from 주소록;
```

무결성 제약조건

4주차 4-04

담당교수: 김희숙
(jasmin11@hanmail.net)

[요약] 데이터 무결성(integrity)

◆ **무결성: 데이터의 정확성, 유효성**

◆ 데이터가 입력되거나 수정, 삭제되어 질 때 잘못된 데이터가 존재하게 되는 경우를 방지하기 위한 것

◆ **데이터 정의어**에서 기본키, 외래키를 정의하면 **DBMS**가 **일관성 조건 검사**

1. 무결성 제약조건

1) 개체 무결성 제약조건: 기본키는 널 값을 가질 수 없다(기본키)

2) 참조 무결성 제약조건: (외래키)

참조하는 테이블의 외래키 값은

참조되는 테이블의 기본키 값과 같다.

3) 키 제약조건: 키는 중복된 값을 갖지 않는다

4) 도메인 제약조건:

[요약] 무결성 제약조건(integrity Constraint)

1) 개체 무결성 제약조건(엔티티 무결성 제약조건)

기본키는 널 값을 가질 수 없다

2) 참조 무결성 제약조건

참조하는 테이블의 외래키 값은 참조되는 테이블의 기본키 값에 반드시 존재해야 한다

제약조건	부모 테이블	자식 테이블
입력	**제약 없음**	부모테이블에 데이터가 존재하는지 검증
수정	수정하려는 데이터를 자식테이블에서 참조하고 있는지를 검증	부모테이블에 존재하는 다른 데이터로 변경가능
삭제	삭제하려는 데이터를 자식테이블에서 참조하고 있는지를 검증	**제약 없음**

[Quiz] 무결성 제약조건

(문제) 참조 무결성 제약 조건 만족 여부를 검사할 필요가 있는 항목에 O 표시를 하시오

	INSERT	UPDATE	DELETE
외래키에 의해 참조되는 기본키			
외래키			

[실습] (MySQL) 무결성 제약조건


```
-- empdept.sql
-- DDL (CREATE TABLE 실습)

-- 부서(부서코드, 부서명)
-- 사원(사원번호, 이름, 부서코드)

drop table 사원;
drop table 부서;

CREATE TABLE 부서 (
        부서코드 char(2)      NOT NULL ,
        부서명    varchar(10) ,
        constraint pk_부서_부서코드 PRIMARY KEY(부서코드)
);

CREATE TABLE 사원 (
        사원번호 char(4)      NOT NULL ,
        이름      varchar(12) ,
        부서코드 char(2) ,
        constraint pk_사원_사원번호 PRIMARY KEY(사원번호) ,
        constraint fk_사원_부서코드 FOREIGN KEY (부서코드) REFERENCES 부서(부서코드)
);
```

테이블명	열 이름	데이터 형식	NULL 유무	기본키	외래키	FK 테이블명	FK 열이름	비고
부서	부서코드	char(2)	NOT NULL	PK				
	부서명	varchar(10)	NULL					

테이블명	열 이름	데이터 형식	NULL 유무	기본키	외래키	FK 테이블명	FK 열이름	비고
사원	사원번호	char(4)	NOT NULL	PK				
	사원명	varchar(12)	NULL					
	부서코드	char(2)	NULL		FK	부서	부서코드	

```
insert into 부서(부서코드, 부서명) values('AA','총무부');
insert into 부서(부서코드, 부서명) values('BB','영업부');
insert into 부서(부서코드, 부서명) values('CC','기획부');

insert into 사원 values('1111','홍길동','AA');
insert into 사원 values('2222','임꺽정','AA');
insert into 사원 values('3333','박찬호','BB');
insert into 사원 values('4444','선동열','BB');
insert into 사원 values('5555','차두리','AA');
insert into 사원 values('6666','신동엽','BB');

select * from 부서;
select * from 사원;
```

[실습] (MySQL) 무결성 제약조건

부서코드	부서명
AA	총무부
BB	영업부
CC	기획부

사원번호	이름	부서코드
1111	홍길동	AA
2222	임꺽정	AA
3333	박찬호	BB
4444	선동열	BB
5555	차두리	AA
6666	신동엽	BB

```
--[Quiz] 다음 문법은 실행하라.  실행 가능한가? 그 이유는?
-- 2-1) 다음 문법을 실행하라.  실행 가능한가?  그 이유는?
INSERT INTO 사원 VALUES('7777','이승엽','BB');

-- 2-2) 다음 문법의 실행결과는? 그 이유는?
UPDATE 부서 SET  부서코드='DD'    WHERE   부서코드='BB';

-- 2-3) 다음 문법의 실행결과는? 그 이유는?
INSERT INTO 부서 VALUES('DD','개발부');
```

❖ 참조 무결성 제약조건

제약조건	부모 테이블	자식 테이블
입력	제약 없음	부모테이블에 데이터가 존재하는지 검증
수정	수정하려는 데이터를 자식테이블에서 참조하고 있는지를 검증	부모테이블에 존재하는 다른 데이터로 변경가능
삭제	삭제하려는 데이터를 자식테이블에서 참조하고 있는지를 검증	제약 없음

[실습] (MySQL) 무결성 제약조건

부서코드	부서명
AA	총무부
BB	영업부
CC	기획부

사원번호	이름	부서코드
1111	홍길동	AA
2222	임꺽정	AA
3333	박찬호	BB
4444	선동열	BB
5555	차두리	AA
6666	신동엽	BB

```
--[Quiz] 다음 문법은 실행하라.  실행 가능한가? 그 이유는?
-- 2-4) 다음 문법의 실행결과는? 그 이유는?
DELETE  FROM 부서 WHERE 부서코드='DD';

-- 2-5) 다음 문법의 실행결과는? 그 이유는?
UPDATE 사원 SET  부서코드='AA'    WHERE   사원번호='7777';

-- 2-6) 다음 문법의 실행결과는? 그 이유는?
UPDATE 사원 SET  부서코드='FF'    WHERE   사원번호='7777';

-- 2-7) 다음 문법을 실행하라.  실행 가능한가?  그 이유는?
DELETE  FROM 사원 WHERE 사원번호='7777';
```

❖ 참조 무결성 제약조건

제약조건	부모 테이블	자식 테이블
입력	제약 없음	부모테이블에 데이터가 존재하는지 검증
수정	수정하려는 데이터를 자식테이블에서 참조하고 있는지를 검증	부모테이블에 존재하는 다른 데이터로 변경가능
삭제	삭제하려는 데이터를 자식테이블에서 참조하고 있는지를 검증	제약 없음

[실습] [Quiz] (MySQL) 데이터 입력


```
--clubmember(mno, mname, mbirth, mreg_date, mgpa)
CREATE TABLE clubmember (
    mno         int          NOT NULL ,                  -- 회원번호
    mname       varchar(20)  NOT NULL ,                  -- 회원명
    mbirth      date ,                                   -- 생일
    mreg_date   TIMESTAMP    DEFAULT NOW()  ,            -- 등록일자 (MySQL)
    mgpa        decimal(3, 1) ,                          -- 평점
    PRIMARY KEY (mno)
);
```

TIMESTAMP DEFAULT NOW()

데이터 값 입력 하지 않으면
현재 날짜시간이 자동으로 입력

datetime DEFAULT CURRENT_TIMESTAMP

[실습] (MySQL) 데이터 입력


```
--clubmember(mno, mname, mbirth, mreg_date, mgpa)
CREATE TABLE clubclubmember (
   mno          int          NOT NULL ,                  -- 회원번호
   mname        varchar(20)   NOT NULL ,                 -- 회원명
   mbirth       date ,                                   -- 생일
   mreg_date    TIMESTAMP    DEFAULT NOW()  ,    -- 등록일자 (MySQL)
   mgpa         decimal(3, 1) ,                          -- 평점
   PRIMARY KEY (mno)
);
```

TIMESTAMP DEFAULT NOW()

데이터 값 입력 하지 않으면
현재 날짜시간이 자동으로 입력

datetime DEFAULT CURRENT_TIMESTAMP

```
-- [Quiz] 다음 데이터 입력을 실행하고 데이터 조회한 결과를 작성하라
-- clubmember(mno, mname, mbirth, mreg_date, mgpa)

-- 1-1) 필드개수와 입력할 값의 개수가 동일할 때는 필드 부분 생략가능
INSERT INTO clubmember VALUES(121,'홍길동','19900315','140320',4.3);

-- 1-2) NULL 값 입력 방법
INSERT INTO clubmember VALUES(123,'이정진', NULL, '140425', NULL);
```

[실습] (MySQL) 데이터 입력


```
--clubmember(mno, mname, mbirth, mreg_date, mgpa)
CREATE TABLE clubclubmember (
   mno          int          NOT NULL ,                  -- 회원번호
   mname        varchar(20)   NOT NULL ,                 -- 회원명
   mbirth       date ,                                   -- 생일
   mreg_date    TIMESTAMP    DEFAULT NOW()  ,     -- 등록일자 (MySQL)
   mgpa          decimal(3, 1) ,                         -- 평점
   PRIMARY KEY (mno)
);
```

TIMESTAMP DEFAULT NOW()

데이터 값 입력 하지 않으면
현재 날짜시간이 자동으로 입력

datetime DEFAULT CURRENT_TIMESTAMP

```
-- 3) 입력할 필드순서 변경가능(단, 필드명을 입력할 값의 순서대로 작성)
INSERT INTO clubmember(mno,mbirth,mreg_date,mname,mgpa)
VALUES(510,'19841210','140816','이정무',3.8);

-- 4) 입력하고자 하는 필드명만 작성한 경우
INSERT INTO clubmember(mno,mname,mbirth,mreg_date) VALUES(210,'김윤식','1994-05-08','140912');

-- 5) DEFAULT 값 입력
INSERT INTO clubmember(mno,mname,mbirth) VALUES(325,'홍진아','1994-12-14');
```