

데이터베이스

인공지능소프트웨어학과

담당교수: 김희숙
(jasmin11@hanmail.net)

데이터베이스

6주차

담당교수: 김희숙
(jasmin11@hanmail.net)

데이터베이스

6주차

담당교수: 김희숙
(jasmin11@hanmail.net)

Quiz

6주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[Quiz] 테이블 생성(테이블 1개)

데이터베이스: testdb

테이블: 사원

[실습 1-1] (테이블 1개) 사원 테이블 생성(데이터 입력, 널 값)

-- (MySQL)

drop table 사원;

-- 사원(사원번호,사원명,연락처,생일)

CREATE TABLE 사원 (

사원번호 char(4) ,

사원명 varchar(20) ,

연락처 char(13) ,

생일 varchar(15) ,

PRIMARY KEY(사원번호)

);

[Quiz] 테이블 생성(테이블 2개)

[실습 1-2] (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

[Quiz] 테이블 생성(테이블 3개)

데이터베이스: studydb

테이블: 학생, 수강, 과목

[실습 1-3] (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

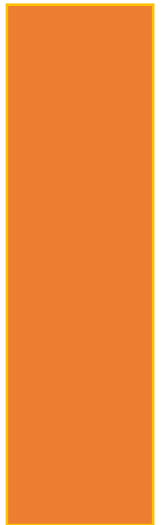
과목

과목번호	과목명
CS100	데이터베이스
CS101	운영체제
CS102	자료구조

❖ 수강 테이블

기본키:

외래키:



[실습] 테이블 생성(테이블 3개)

[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

[Quiz] 테이블 생성(테이블 2개)

[HW 1-1] (테이블 2개) 환자, 의사 테이블 생성(기본키, 외래키)

- ❖ 환자, 의사
테이블 생성 순서
테이블 삭제 순서
데이터 입력 순서

38 -- 데이터 조회

39 • `select * from 의사;`

Result Grid Filter Rows: <input type="text"/> Edit:				
	의사번호	의사이름	소속	근무연수
▶	D001	정지영	내과	5
	D002	김선주	피부과	10
	D003	정성호	정형외과	15
*	NULL	NULL	NULL	NULL

40 • `select * from 환자;`

Result Grid Filter Rows: <input type="text"/> Edit:				
	환자번호	환자이름	나이	담당의사
▶	P001	오우진	31	D002
	P002	채광주	50	D001
	P003	김용욱	43	D003

[실습] [Quiz] 테이블 생성(테이블 2개)

```
3  -- HW1-1 의사, 환자 테이블
4 • drop table 환자;
5 • drop table 의사;
6
7  -- 의사 테이블 생성
8 • CREATE TABLE 의사 (
9      의사번호      char(4)          NOT NULL ,
10     의사이름      varchar(20)      ,
11     소속          varchar(50)      ,
12     근무연수      int              ,
13     PRIMARY KEY(의사번호)
14 );
15
16 -- 환자 테이블 생성
17 • CREATE TABLE 환자 (
18     환자번호      char(4)          NOT NULL ,
19     환자이름      varchar(20)      ,
20     나이          int              ,
21     담당의사      char(4)          ,
22     PRIMARY KEY(환자번호) ,
23     FOREIGN KEY(담당의사) REFERENCES 의사(의사번호)
24 );
```

1] (테이블 2개) 환자, 의사 테이블 생성(기본키, 외래키)

```
26  -- 의사 데이터 입력
27  -- 의사(의사번호, 의사이름, 소속, 근무연수)
28 • INSERT INTO 의사 VALUES('D001', '정지영', '내과', 5);
29 • INSERT INTO 의사 VALUES('D002', '김선주', '피부과', 10);
30 • INSERT INTO 의사 VALUES('D003', '정성호', '정형외과', 15);
31
32  -- 환자 데이터 입력
33  -- 환자(환자번호, 환자이름, 나이, 담당의사)
34 • INSERT INTO 환자 VALUES('P001', '오우진', 31, 'D002');
35 • INSERT INTO 환자 VALUES('P002', '채광주', 50, 'D001');
36 • INSERT INTO 환자 VALUES('P003', '김용욱', 43, 'D003');
```

[HW 1-1] (테이블 2개) 환자, 의사 테이블 생성(기본키, 외래키)

- 의사 테이블을 생성하는 SQL문을 작성하시오.
- <조건>
- 1-1) 소속 속성의 값을 입력하지 않으면 자동으로 내과가 지정
- 1-2) 근무연수는 1년 이상 50년 이하 범위의 값을 가지도록 지정

38 -- 데이터 조회

39 • `select * from 의사;`

< Result Grid | Filter Rows: Edit:

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

[HW 1-1] (테이블 2개) 환자, 의사 테이블 생성(기본키, 외래키)

- 2) 환자 테이블을 생성하는 SQL문을 작성하시오
- <조건>
- 2-1) 이름 속성은 널 값이 허용되지 않도록 지정
- 2-2) 담당의사 속성을 의사 테이블의 의사번호 속성을
- 참조하는 외래키 지정

40 • select * from 환자;

< Result Grid | Filter Rows: | Edit

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

[실습] 무결성 제약조건 (신입생)

[실습 1-02] DDL (무결성 제약조건 있는 경우)

- 다음 조건을 만족하는 데이터정의를 작성하시오
- (조건)
- 1) 기본키는 학번으로 지정
- 2) 주민등록번호는 중복된 값을 갖지 않는다
- 3) 학년은 기본값 1을 갖는다
- 4) 성별 입력값은 남, 여 로 값을 제한한다)

```
insert into 신입생(학번,이름,학년, 성별)
values('20211111','홍길동',3,'남');
```

```
insert into 신입생(학번,이름,성별)
values('20232222','조유나','여');
```

```
insert into 신입생(학번,이름,성별)
values('20233333','주민증','남자');
```

[실습 1-01] DDL (무결성 제약조건 없는 경우)

-- 신입생(학번,주민,이름,학년,성별,주소,학과명)

- 1) 제약조건 없는 경우
- ```
drop table 신입생;
```

```
CREATE TABLE 신입생(
 학번 char(8) ,
 주민등록번호 char(14) ,
 이름 varchar(20) ,
 학년 int ,
 성별 char(1) ,
 주소 varchar(20) ,
 학과명 varchar(20) ,
 PRIMARY KEY(학번)
);
```

- 테이블 구조 확인
- ```
DESC 신입생;
```

Quiz

6주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[Quiz] 기본키, 외래키

[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

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

[Quiz] 키(Key)

[2개 테이블 : 기본키, 외래키]

간호사

사번	성명	나이	업무
1111	홍길동	35	외래
2222	김윤식	46	병동
3333	이정진	32	종환자실
4444	홍진아	28	외래

연락처

사번	성별	연락처
1111	남	111-1111
2222	남	222-2222
4444	여	444-4444

[Quiz] 키(Key)

기본키:

인사기록
가족관계
거래처

인사기록

사원번호	성명	근무부서	직책	생년월일	주소	전화번호
A-1001	홍길동	기획전략실	대리	1979-08-04	서울시 종로구	010-0000-0000
A-1002	정명선	기획전략실	사원	1984-05-13	서울시 동대문구	010-1111-1111
A-1003	황성인	기획전략실	사원	1986-10-26	서울시 금천구	010-2222-2222
A-1004	방진석	기획전략실	과장	1973-02-09	서울시 성북구	010-3333-3333

가족관계

사원번호	이름	관계	직업	생년월일
A-1001	홍재민	부	회사원	1949-04-14
A-1001	소유진	모	가사	1954-09-21
A-1001	홍길순	여동생	회사원	1982-02-06
A-1002	이길자	모	가사	1958-11-17
A-1002	정명재	오빠	연구원	1982-06-28
A-1002	정명수	남동생	대학생	1989-03-09

거래처

사원번호	사원명	거래처	담당자	거래처주소	거래처전화번호	거래체결날짜
A-1001	홍길동	동일물산	이효린	서울시 서초구	02-1111-0000	2000-06-05
A-1001	홍길동	백일상사	김병선	서울시 영등포구	02-1111-1111	2000-10-18
A-1002	정명선	선진기업	안지환	서울시 강서구	02-1111-2222	2000-03-25
A-1002	정명선	대성기획	나윤구	서울시 노원구	02-1111-3333	2001-08-02
A-1003	황성인	명진건설	장철수	서울시 종로구	02-1111-4444	2000-02-15
A-1003	황성인	제일설비	김혜성	서울시 양천구	02-1111-5555	2002-05-28

요약

6주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[요약] 관계형 데이터 모델

□ 용어

- ✓ 릴레이션(Relation), 테이블
 - ✓ 속성(attribute), 필드, 열(column)
 - ✓ 튜플(tuple), 레코드, 행(row)
 - ✓ 도메인(domain)
 - ✓ 차수(degree)
 - ✓ 카디널리티(cardinality)
-
- ✓ 릴레이션 스키마: 내포(intension), 정적
 - ✓ 릴레이션 인스턴스: 외연(extension), 동적

테이블
릴레이션

학생

필드 (속성, 열, 애트리뷰트) column

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

레코드
(튜플, 행)
row

사원

스키마	사원번호	사원명	연락처	생일
인스턴스	D001	정지영		NULL
	D002	김선주	010-1111-1111	NULL
	D003	정성호	NULL	10월04일

용어



- 릴레이션(Relation) : 행과 열로 구성된 2차원의 테이블
- 튜플(tuple) : 특정 인스턴스에 관한 사실(값)들의 모임
- 도메인(Domain) : 한 애트리뷰트(필드)에 나타날 수 있는 값들의 집합
- 차수(Degree) : 한 릴레이션에 나타날 수 있는 애트리뷰트의 수
- 카디날리티(Cardinality) : 튜플 수

키(Key) / 제약조건

□키

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

- ✓ 수퍼키(super key)
- ✓ 후보키(candidate key)
- ✓ 기본키(primary key): **PRIMARY KEY**
- ✓ 대체키(alternate key): **UNIQUE**
- ✓ 외래키(foreign key): **FOREIGN KEY**

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

[요약] 데이터 무결성(integrity)

- ◆ 무결성: 데이터의 정확성, 유효성
- ◆ 데이터가 입력되거나 수정, 삭제되어 질 때 잘못된 데이터가 존재하게 되는 경우를 방지하기 위한 것
- ◆ 데이터 정의어에서 기본키, 외래키를 정의하면 DBMS가 일관성 조건 검사

1. 무결성 제약조건

- 1) **개체 무결성 제약조건**: 기본키는 널 값을 가질 수 없다(기본키) **PRIMARY KEY**
- 2) **참조 무결성 제약조건**: (외래키) **FOREIGN KEY**
참조하는 테이블의 외래키 값은
참조되는 테이블의 기본키 값과 같다.
- 3) **키 제약조건**: 키는 중복된 값을 갖지 않는다 **UNIQUE**
- 4) **도메인 제약조건**: CHECK, DEFAULT 문법 **CHECK, DEFAULT**

[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

널 값(Null Value)

6주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[Quiz] 테이블 생성(테이블 1개)

데이터베이스: testdb

테이블: 사원

[실습 1-1] (테이블 1개) 사원 테이블 생성(데이터 입력, 널 값)

-- (MySQL)

drop table 사원;

-- 사원(사원번호,사원명,연락처,생일)

CREATE TABLE 사원 (

사원번호 char(4) ,

사원명 varchar(20) ,

연락처 char(13) ,

생일 varchar(15) ,

PRIMARY KEY(사원번호)

);

사원

사원번호	사원명	연락처	생일
D001	정지영		NULL
D002	김선주	010-1111-1111	NULL
D003	정성호	NULL	10월04일

[요약] 널 값(NULL)

❖ 널 값: 알려지지 않은 값(Unknown)

테이블 이름: 주소록

필드(속성, 열)

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

잘못된 입력된 값

널(NULL)이 입력된 필드

레코드(튜플, 행)

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

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

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

[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 입력하는 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'**);

데이터 조작어(DML)

6주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[실습 01] (주소록): 데이터 입력

-- 주소록(이름, 전화번호, 주소, 생일)

use studydb;

DROP TABLE 주소록;

-- 주소록 테이블 생성

CREATE TABLE 주소록(

이름 char(10) NOT NULL ,

전화번호 char(13) ,

주소 varchar(10) ,

생일 varchar(11) ,

PRIMARY KEY(이름)

);

자동증가 없는 경우,
주소록

이름	전화번호	주소	생일
홍길동	010-1234-5678	서울	3월 15일
이몽룡	010-3343-5643	부산	12월 14일
최용만	321-2346	대전	5월 8일
이건우	010-2132-2345	NULL	NULL

[실습 01] (주소록): 데이터 입력

-- 데이터 입력

-- 주소록(이름, 전화번호, 주소, 생일)

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

```
INSERT INTO 주소록  
VALUES('이몽룡','010-3354-5643','부산','12월 14일');
```

```
INSERT INTO 주소록  
VALUES('최용만','321-2345','대전','5월 8일');
```

-- 다음과 같이 널 값 데이터 입력 하시오

	이름	전화번호	주소	생일
▶	유정두	010-9999-9999	NULL	NULL
	이건우	010-2132-2345	NULL	NULL
	이몽룡	010-3354-5643	부산	12월 14일
	최용만	321-2345	대전	5월 8일
	홍길동	010-1234-5678	서울	3월 15일
*	NULL	NULL	NULL	NULL

-- 데이터 조회

```
select * from 주소록;
```

[실습 01] (주소록): 널 값 입력

[실습] 널(NULL) 값 입력하는 방법

-- 데이터 입력 방법

INSERT INTO 테이블명 VALUES(데이터값, ...);

INSERT INTO 테이블명(필드명) VALUES(데이터값, ...);

-- 필드명 생략 한 경우

-- 필드명 생략하지 않은 경우

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		

--널 값 데이터 입력하는 방법

INSERT INTO 주소록(이름, 전화번호, 주소, 생일)
VALUES('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);

INSERT INTO 주소록(이름, 전화번호) VALUES('유정두','010-9999-9999');

[실습] (MySQL): 입력,수정,삭제

❖ (MySQL): Safe_mode 해제

1) MySQL Workbench

[Edit]-[Preferences]-SQL Editor-Safe Updates 체크 해제

2) 쿼리: safe_mode 체크 해제

```
SET SQL_SAFE_UPDATES = 0;
```

```
SET SQL_SAFE_UPDATES = 0;
```


키의 종류

6주차

담당교수: 김희숙
(jasmin11@hanmail.net)

키(Key)

□키

- 각 튜플을 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	

[Quiz] 기본키, 외래키

[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

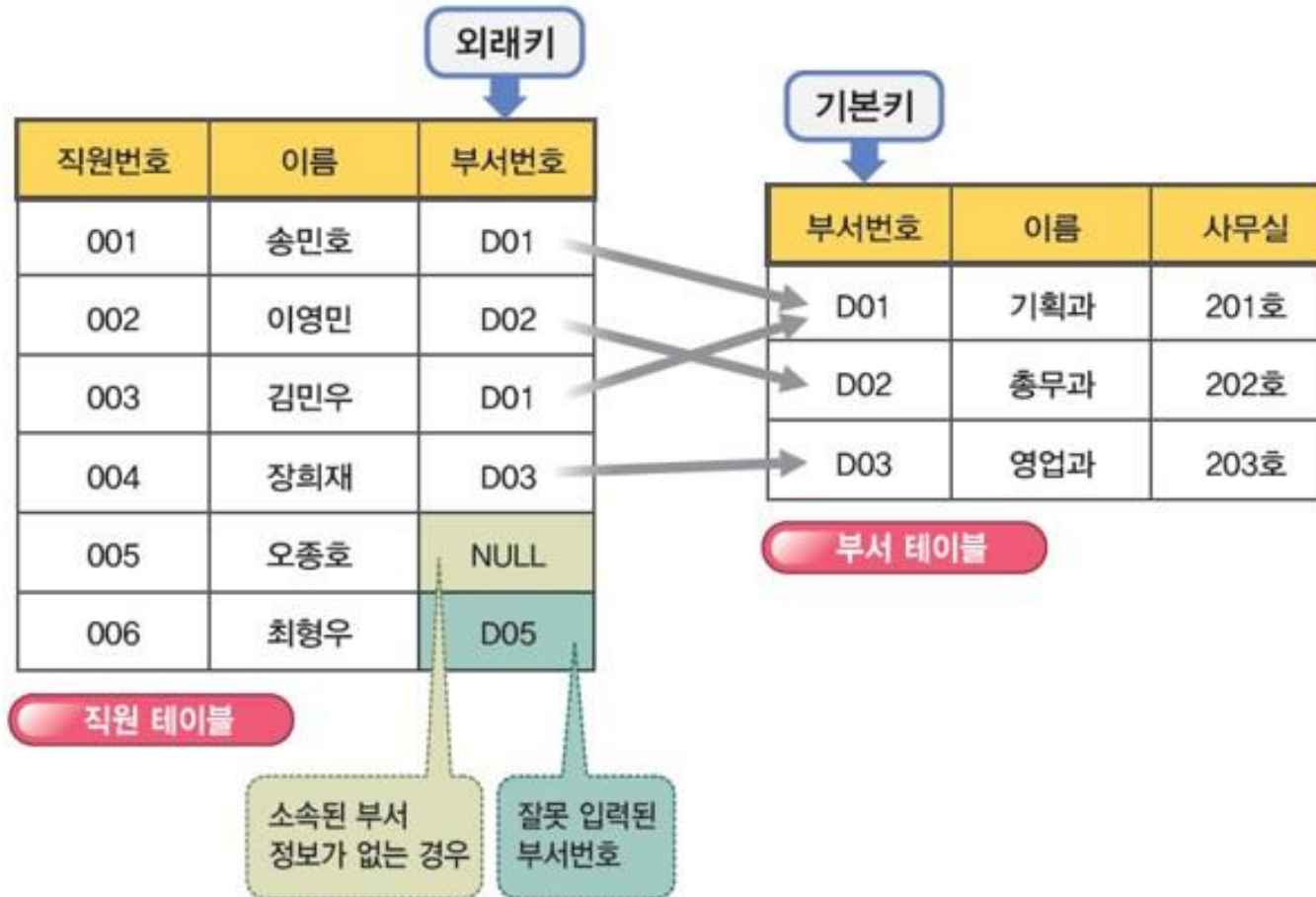
의사번호	의사이름	소속	근무연수
D001	정지영	내과	5
D002	김선주	피부과	10
D003	정성호	정형외과	15

환자

외래키

환자번호	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)

키(Key): 기본키, 외래키, 후보키

[3개 테이블 : 기본키, 외래키]

학생

학번	학생명	학년
1111	홍길동	1
2222	김문식	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

[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(학번, 과목번호, 성적)

CREATE TABLE 수강 (

학번 char(5) ,

과목번호 char(5) ,

성적 int ,

PRIMARY KEY(학번, 과목번호) ,

FOREIGN KEY(학번) REFERENCES 학생(학번) ,

FOREIGN KEY(과목번호) REFERENCES 과목(과목번호)

);

--수강 테이블 생성	수강(학번, 과목번호, 학점)
<pre> 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)); </pre>	

기본키가 복합키 인 경우,

[Quiz] 신입생

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

❖ 신입생

후보키: {학번}, {주민등록번호}

기본키: {학번}

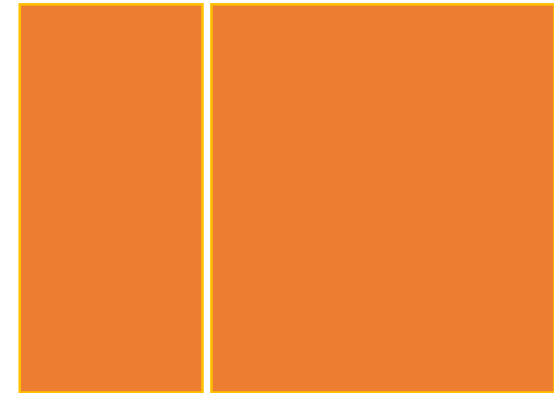
대체키: {주민등록번호}

수퍼키는?

수퍼키의 성질: 유일성

후보키의 성질: 유일성, 최소성

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] 키(Key)

[2개 테이블 : 기본키, 외래키]

간호사

사번	성명	나이	업무
1111	홍길동	35	외래
2222	김윤식	46	병동
3333	이정진	32	종환자실
4444	홍진아	28	외래

연락처

사번	성별	연락처
1111	남	111-1111
2222	남	222-2222
4444	여	444-4444

[Quiz] 키(Key)

기본키:

인사기록
가족관계
거래처

인사기록

사원번호	성명	근무부서	직책	생년월일	주소	전화번호
A-1001	홍길동	기획전략실	대리	1979-08-04	서울시 종로구	010-0000-0000
A-1002	정명선	기획전략실	사원	1984-05-13	서울시 동대문구	010-1111-1111
A-1003	황성인	기획전략실	사원	1986-10-26	서울시 금천구	010-2222-2222
A-1004	방진석	기획전략실	과장	1973-02-09	서울시 성북구	010-3333-3333

가족관계

사원번호	이름	관계	직업	생년월일
A-1001	홍재민	부	회사원	1949-04-14
A-1001	소유진	모	가사	1954-09-21
A-1001	홍길순	여동생	회사원	1982-02-06
A-1002	이길자	모	가사	1958-11-17
A-1002	정명재	오빠	연구원	1982-06-28
A-1002	정명수	남동생	대학생	1989-03-09

거래처

사원번호	사원명	거래처	담당자	거래처주소	거래처전화번호	거래체결날짜
A-1001	홍길동	동일물산	이효린	서울시 서초구	02-1111-0000	2000-06-05
A-1001	홍길동	백일상사	김병선	서울시 영등포구	02-1111-1111	2000-10-18
A-1002	정명선	선진기업	안지환	서울시 강서구	02-1111-2222	2000-03-25
A-1002	정명선	대성기획	나윤구	서울시 노원구	02-1111-3333	2001-08-02
A-1003	황성인	명진건설	장철수	서울시 종로구	02-1111-4444	2000-02-15
A-1003	황성인	제일설비	김혜성	서울시 양천구	02-1111-5555	2002-05-28

[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

무결성 제약조건

6주차

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 정의어에서 기본키, 외래키를 정의하면 DBMS가 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**: CHECK, DEFAULT 문법

[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

연습문제

3판 p151 26번
2판 p156 20번

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

	INSERT	UPDATE	DELETE
외래키에 의해 참조되는 기본키			
외래키			

[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

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

Quiz

6주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[실습] DML (사원)

[실습] 데이터 입력/수정/삭제/조회

[Quiz 1-01] DML: (사원)

- 1. 정지영 연락처를 010-9999-9999, 생일을 10월11일 로 수정하라
- 2. 김선주 생일을 10월11일로 수정하라
- 3. 정성호 레코드를 삭제하라

사원

사원번호	사원명	연락처	생일
D001	정지영		NULL
D002	김선주	010-1111-1111	NULL
D003	정성호	NULL	10월04일

[실습] 무결성 제약조건 (신입생)

[실습 1-02] DDL (무결성 제약조건 있는 경우)

- 다음 조건을 만족하는 데이터정의어를 작성하시오
- (조건)
- 1) 기본키는 학번으로 지정
- 2) 주민등록번호는 중복된 값을 갖지 않는다
- 3) 학년은 기본값 1을 갖는다
- 4) 성별 입력값은 남, 여 로 값을 제한한다)

```
insert into 신입생(학번,이름,학년, 성별)
values('20211111','홍길동',3,'남');
```

```
insert into 신입생(학번,이름,성별)
values('20232222','조유나','여');
```

```
insert into 신입생(학번,이름,성별)
values('20233333','주민증','남자');
```

[실습 1-01] DDL (무결성 제약조건 없는 경우)

-- 신입생(학번,주민,이름,학년,성별,주소,학과명)

- 1) 제약조건 없는 경우
- ```
drop table 신입생;
```

```
CREATE TABLE 신입생(
 학번 char(8) ,
 주민등록번호 char(14) ,
 이름 varchar(20) ,
 학년 int ,
 성별 char(1) ,
 주소 varchar(20) ,
 학과명 varchar(20) ,
 PRIMARY KEY(학번)
);
```

- 테이블 구조 확인
- ```
DESC 신입생;
```

[실습] 무결성 제약조건 (신입생) (ans)

[실습 1-01] DDL (무결성 제약조건 없는 경우)

-- 신입생(학번,주민,이름,학년,성별,주소,학과명)

-- 1) 제약조건 없는 경우
drop table 신입생;

```
CREATE TABLE 신입생(  
    학번          char(8) ,  
    주민등록번호  char(14) ,  
    이름          varchar(20) ,  
    학년          int ,  
    성별          char(1) ,  
    주소          varchar(20) ,  
    학과명        varchar(20) ,  
    PRIMARY KEY(학번)  
);
```

-- 테이블 구조 확인
DESC 신입생;

-- 다음 조건을 만족하는 데이터 정의어 작성

-- (조건)

- 1) 기본키는 학번으로 지정
- 2) 주민등록번호는 중복된 값을 갖지 않는다
- 3) 학년은 기본값 1을 갖는다
- 4) 성별 입력값은 남, 여 로 값을 제한한다)

[실습 1-02] DDL (무결성 제약조건 있는 경우)

-- 신입생(학번,주민,이름,학년,성별,주소,학과명)

-- 2) 제약조건 있는 경우
drop table if exists 신입생;

```
CREATE TABLE 신입생(  
    학번          char(8) ,  
    주민등록번호  char(14)  UNIQUE ,  
    이름          varchar(20) ,  
    학년          int  DEFAULT 1 ,  
    성별          char(1)  CHECK (성별 IN('남','여')) ,  
    주소          varchar(20) ,  
    학과명        varchar(20) ,  
    PRIMARY KEY(학번)  
);
```

[실습] 무결성 제약조건 (과목) (stu)

[실습 2-01] DDL (무결성 제약조건 없는 경우)

-- 과목(과목번호,이름,강의실,개설학과, 시수)

-- 1) 제약조건 없는 경우
drop table 과목;

```
CREATE TABLE 과목 (  
    과목번호 char(4)      NOT NULL ,  
    이름      VARCHAR(20) ,  
    강의실    CHAR(3)     ,  
    개설학과  VARCHAR(20) ,  
    시수      INT  
);
```

-- 테이블 구조 확인
DESC 과목;

[실습 2-02] DDL (무결성 제약조건 있는 경우)

-- 과목(과목번호,이름,강의실,개설학과, 시수)

-- 2) 제약조건 있는 경우
drop table if exists 과목;



-- 다음 조건을 만족하는 데이터 정의어
작성

<조건>

- 1) 과목번호,이름,강의실,개설학과, 시수 로 구성된 과목 테이블을 생성
- 2) 기본키는 과목번호로 설정한다

[실습] 무결성 제약조건 (학생) (stu)

[실습 3-01] DDL (무결성 제약조건 없는 경우)

-- 학생(학번, 이름, 주소, 학년, 나이, 휴대폰번호, 소속학과)

-- 1) 제약조건 없는 경우
drop table 학생;

```
CREATE TABLE 학생 (  
    학번      CHAR(4)      NOT NULL ,  
    이름      VARCHAR(20)  NOT NULL ,  
    주소      VARCHAR(50)  ,  
    학년      INT          ,  
    나이      INT          ,  
    성별      CHAR(1)      ,  
    휴대폰번호 CHAR(14)    ,  
    소속학과  VARCHAR(20)  ,  
);
```

-- 테이블 구조 확인
DESC 학생;

[실습 3-02] DDL (무결성 제약조건 있는 경우)

-- 학생(학번, 이름, 주소, 학년, 나이, 휴대폰번호, 소속학과)

-- 2) 제약조건 있는 경우
drop table if exists 학생;



-- 다음 조건을 만족하는 데이터 정의어 작성

<조건>

- 1) 기본키는 학번으로 설정한다
- 2) 휴대폰번호를 대체키로 지정한다
- 3) 주소 에는 기본값을 설정한다 (기본값: 미정)

[실습] 무결성 제약조건 (수강) (stu)

[실습 4-01] DDL (무결성 제약조건 없는 경우)

-- 수강(학번,과목번호,신청날짜,중간성적,기말성
적,평가학점)

-- 1) 제약조건 없는 경우
drop table 수강;

```
CREATE TABLE 수강 (  
    학번      char(6)      NOT NULL ,  
    과목번호  CHAR(4)      NOT NULL ,  
    신청날짜  DATE         ,  
    중간성적  INT          ,  
    기말성적  INT          ,  
    평가학점  CHAR(1) ,  
    PRIMARY KEY(학번, 과목번호)  
);
```

-- 테이블 구조 확인
DESC 수강;

[실습 4-02] DDL (무결성 제약조건 있는 경우)

-- 수강(학번,과목번호,신청날짜,중간성적,기말
성적,평가학점)

-- 2) 제약조건 있는 경우
drop table if exists 수강;



-- 다음 조건을 만족하는 데이터 정의어 작성

<조건>

- 1) 기본키는 (학번,과목번호) 로 설정한다
- 2) 학번 은 외래키로 학생 테이블의 학번을 참조한다
- 3) 과목번호는 외래키로 과목 테이블의 과목번호를 참조
한다
- 4) 중간성적, 기말성적에는 기본값을 설정한다(기본값: 0)

[실습] [Quiz] 의사, 환자

- 환자(환자번호,환자이름,나이,담당의사)
- 의사(의사번호,의사이름,소속,근무연수)

환자 테이블

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

의사 테이블

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

- 의사 테이블을 생성하는 SQL문을 작성하시오.
- <조건>
- 1-1) 소속 속성의 값을 입력하지 않으면 자동으로 내과가 지정
- 1-2) 근무연수는 1년 이상 50년 이하 범위의 값을 가지도록 지정

- 2) 환자 테이블을 생성하는 SQL문을 작성하시오
- <조건>
- 2-1) 이름 속성은 널 값이 허용되지 않도록 지정
- 2-2) 담당의사 속성을 의사 테이블의 의사번호 속성을 참조하는 외래키 지정

[실습] [Quiz] 의사, 환자 (답안)

- 환자(환자번호,환자이름,나이,담당의사)
- 의사(의사번호,의사이름,소속,근무연수)

환자 테이블

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

의사 테이블

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

- 의사 테이블을 생성하는 SQL문을 작성하시오.
- <조건>
- 1-1) 소속 속성의 값을 입력하지 않으면 자동으로 내과가 지정
- 1-2) 근무연수는 1년 이상 50년 이하 범위의 값을 가지도록 지정

```
CREATE TABLE 의사 (
    의사번호    char(4)    NOT NULL ,
    의사이름    varchar(20) ,
    소속        varchar(30) DEFAULT '내과' ,
    근무연수    int        CHECK (근무연수 >= 1 AND 근무연수 <= 50),
    PRIMARY KEY(의사번호)
);
```


[실습] [Quiz] 의사, 환자 (답안)

- 환자(환자번호,환자이름,나이,담당의사)
- 의사(의사번호,의사이름,소속,근무연수)

환자 테이블

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

의사 테이블

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

- 2) 환자 테이블을 생성하는 SQL문을 작성하시오
- <조건>
- 2-1) 이름 속성은 널 값이 허용되지 않도록 지정
- 2-2) 담당의사 속성을 의사 테이블의 의사번호 속성을 참조하는 외래키 지정

```
CREATE TABLE 환자 (  
    환자번호      char(4)      NOT NULL ,  
    환자이름      varchar(20)  NOT NULL ,  
    나이          int ,  
    담당의사      char(4)      ,  
    PRIMARY KEY(환자번호) ,  
    FOREIGN KEY(담당의사) REFERENCES 의사(의사번호)  
);
```