

데이터베이스

인공지능소프트웨어학과

담당교수: 김희숙
(jasmin11@hanmail.net)

데이터베이스

5주차

담당교수: 김희숙
(jasmin11@hanmail.net)

실습

5주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[Quiz] 테이블 생성: (학원생, 학교)

부모 테이블:

자식 테이블:

학원생

학원생이름	<u>폰번호</u>	나이	외래키(FK) 학교이름	학년	반	반번호
홍길동	010-1237-6542	19	송원고	3	1	10
김하나	010-3218-8765	18	한빛고	2	2	7
홍길동	010-4587-9834	19	송원고	3	1	31
박순희	010-7789-6754	17	이슬고	1	3	16

참조

학교

<u>학교명</u>	분류	학생수	주소
송원고	3	435	경기 성남 분당구 황새울로 123
한빛고	2	377	경기 성남 분당구 판교역로 67
이슬고	1	507	경기 성남 분당구 미금로 567

기본키(PK)

1. 테이블 생성 순서
2. 테이블 삭제 순서
3. 데이터 입력 순서

1. 데이터베이스 생성: booksrdb
2. 테이블 생성: 학원생, 학교
3. 데이터 입력
4. 데이터 조회

select * from 학교;

select * from 학원생;

[Quiz 1] 테이블 생성

-- 직원(이름,주소,전화번호,연봉)

-- (MySQL)

DROP TABLE if exists 직원;

-- 직원(이름,주소,전화번호,연봉)

CREATE TABLE 직원 (

이름 VARCHAR(20) NOT NULL ,

주소 VARCHAR(100) ,

전화번호 CHAR(13) NOT NULL ,

연봉 INT DEFAULT 0 ,

PRIMARY KEY(이름)

);

데이터베이스: testdb

테이블: 직원

1. 데이터베이스 생성: testdb

2. 테이블 생성: 직원

3. 데이터 입력: 6건의 데이터를 입력하시오



이름	주소	전화번호	연봉
고진수	서울시 구로구	010-6666-6666	6000
김선우	서울시 구로구	010-2222-2222	2000
오형준	부산시 연산구	010-5555-5555	5000
우태하	서울시 서초구	010-1111-1111	1000
유희정	서울시 마포구	010-4444-4444	4000
이영지	서울시 마포구	010-3333-3333	3000

[Quiz 2] 테이블 생성

데이터베이스: studydb

테이블: dept, dmember

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	?	?	

[Quiz 2] 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 실습하세요

다음 빈칸을 완성하여 실습하시오
(MySQL)
테이블 생성,
데이터 입력(**본인 이름 1개 추가** 입력)
데이터 조회

* 참고) CONSTRAINT fk_dmember_dept 문법은 수업시간 설명합니다

김희숙

(복습)[예제] SQL: 테이블 생성(기본키, 외래키)

```
CREATE TABLE 학과 (  
    학과번호 int NOT NULL ,  
    학과명 varchar(50) ,  
    PRIMARY KEY(학과번호)  
);
```

학과

학과번호	학과명
1	컴퓨터소프트웨어공학과
2	컴퓨터정보공학과
3	인공지능소프트웨어과

학생

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

```
CREATE TABLE 학생 (  
    번호 int NOT NULL ,  
    이름 varchar(12) ,  
    학년 int ,  
    분반 char(2) ,  
    학과번호 int ,  
    PRIMARY KEY(번호) ,  
    FOREIGN KEY (학과번호)  
        REFERENCES 학과(학과번호)  
);
```


요약 (관계형 데이터 모델)

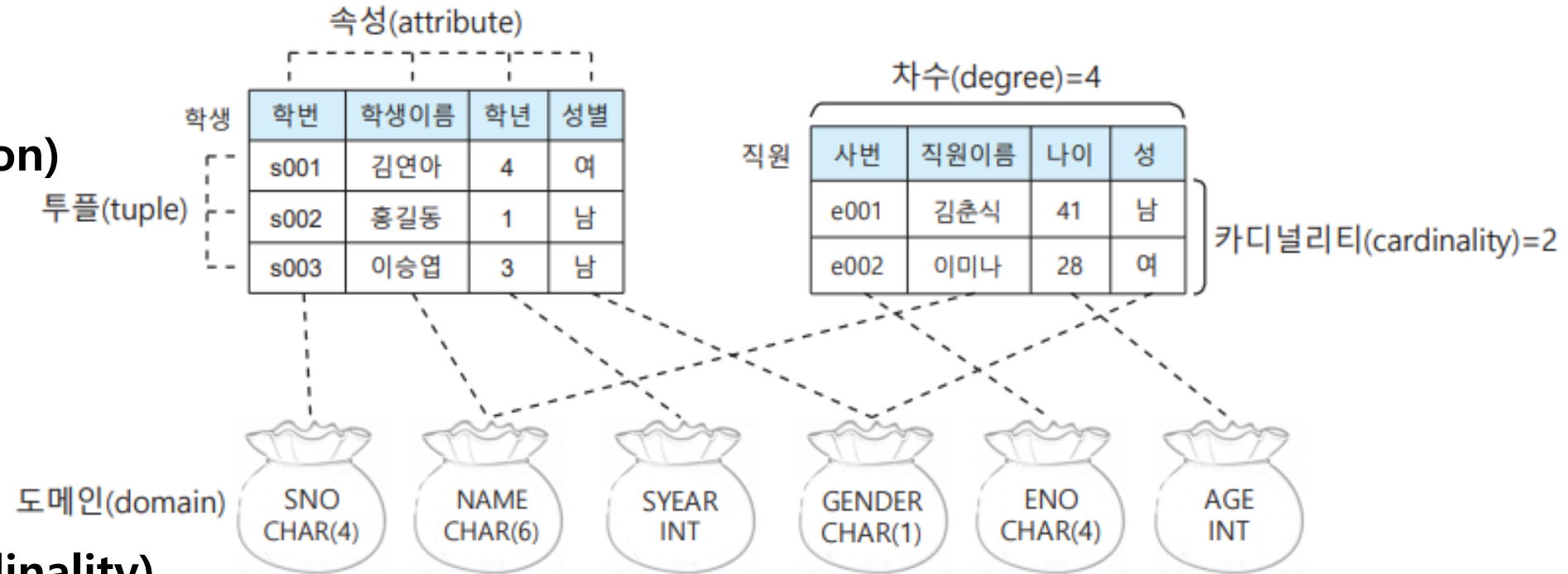
5주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[요약] 관계형 데이터 모델

□ 용어

- ✓ 릴레이션(Relation)
- ✓ 속성(attribute)
- ✓ 튜플(tuple)
- ✓ 도메인(domain)
- ✓ 차수(degree)
- ✓ 카디널리티(cardinality)



(그림 출처: "'SQL과 NoSQL 기반의데이터베이스 입문', 박성진, 생능, 2023)

[요약] 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)

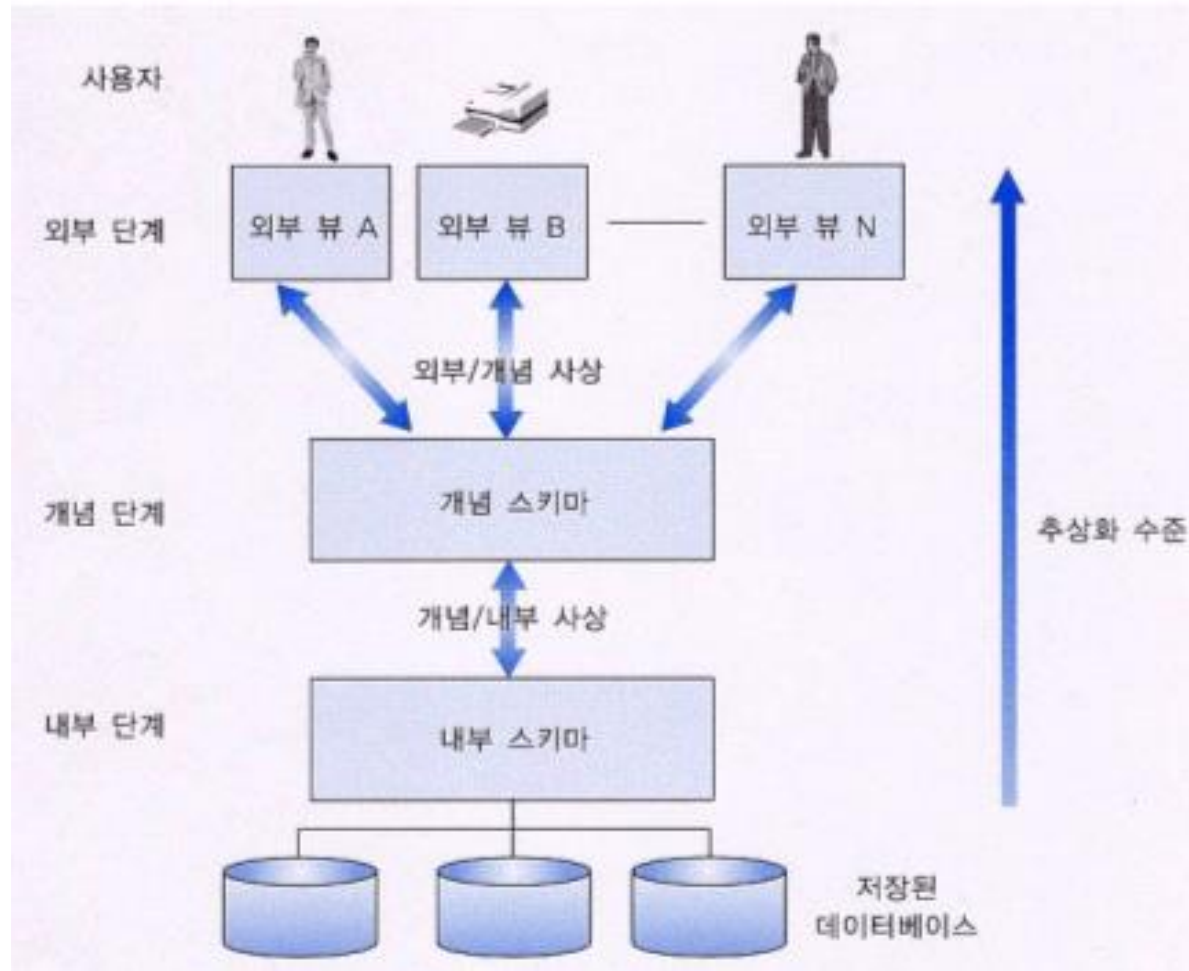
키	특성
수퍼키	유일성
후보키	유일성, 최소성
기본키	중복불가, 필수입력
대체키	
외래키	

SQL: 뷰(view)

담당교수: 김희숙
(jasmin11@hanmail.net)

[요약] 3단계 구조

ANSI/SPARC 아키텍처 (스키마 3단계 구조)



(그림 출처: "데이터베이스배움터", 홍의경 저, 생능, 2012)



[요약] 뷰(view) 생성



- 뷰(view):
 - 가상의 테이블(다른 테이블을 기반으로 만들어진 가상의 테이블)
 - 장점: **사용의 편의성, 보안**

```
-- 뷰 정의(뷰 생성)
CREATE VIEW 뷰이름
as
SELECT
FROM
```

```
DROP TABLE 테이블명;
```

```
DROP VIEW 뷰이름;
```

[실습] (MySQL) 뷰(view)

- 1단계: 데이터베이스 생성(스키마 생성)
- 사용할 데이터베이스 선택
- 2단계: 테이블 생성
- 3단계: 데이터 입력

-- 테이블
-- 직원(이름,주소,전화번호,연봉)

-- 뷰
-- 직원_뷰(이름, 전화번호)

[실습] 테이블 생성

-- (MySQL)

DROP TABLE if exists 직원;

-- 직원(이름,주소,전화번호,연봉)

CREATE TABLE 직원 (

이름 VARCHAR(20) NOT NULL ,

주소 VARCHAR(100) ,

전화번호 CHAR(13) NOT NULL ,

연봉 INT DEFAULT 0 ,

PRIMARY KEY(이름)

);

[실습]

-- 테이블

-- 직원(이름, 주소, 전화번호, 연봉)

-- 뷰

-- 직원_뷰(이름, 전화번호)

이름	주소	전화번호	연봉
고진수	서울시 구로구	010-6666-6666	6000
김선우	서울시 구로구	010-2222-2222	2000
오형준	부산시 연산구	010-5555-5555	5000
우태하	서울시 서초구	010-1111-1111	1000
유희정	서울시 마포구	010-4444-4444	4000
이영지	서울시 마포구	010-3333-3333	3000

[실습] 데이터 입력

-- (MySQL)

-- 직원(이름,주소,전화번호,연봉)

INSERT INTO 직원 VALUES('우태하','서울시 서초구','010-1111-1111',1000);

INSERT INTO 직원 VALUES('김선우','서울시 구로구','010-2222-2222',2000);

INSERT INTO 직원 VALUES('이영지','서울시 마포구','010-3333-3333',3000);

INSERT INTO 직원 VALUES('유희정','서울시 마포구','010-4444-4444',4000);

INSERT INTO 직원 VALUES('오형준','부산시 연산구','010-5555-5555',5000);


INSERT INTO 직원 VALUES('고진수','서울시 구로구','010-6666-6666',6000);

select * from 직원;

[실습] 뷰(view) 생성

```
-- 뷰 생성
CREATE VIEW 직원_뷰
as
    select 이름, 전화번호
    from 직원;

-- 뷰 조회
select * from 직원_뷰;
```




[실습]

```
-- 테이블
-- 직원(이름, 주소, 전화번호, 연봉)

-- 뷰
-- 직원_뷰(이름, 전화번호)
```

```
-- 뷰 데이터 입력

-- 직원_뷰(이름,전화번호)
insert into 직원_뷰(이름,전화번호)
values('오주원','010-9999-9999');
```



```
-- 뷰 데이터 조회
select * from 직원_뷰;

-- 테이블 조회
select * from 직원;
```

관계형 데이터 모델

담당교수: 김희숙
(jasmin11@hanmail.net)

데이터 모델

* 데이터모델 3가지 요소:

- 1) 구조(structure)
- 2) 연산(operation)
- 3) 제약조건(constraint)



• 데이터 모델:

- 데이터베이스 종류에 따라서 다양한 저장 구조를 갖는다
- 데이터베이스 **구조**를 명세하기 위한 개념

□ 데이터 모델 3가지 요소

✓ 데이터 구조(data structure)

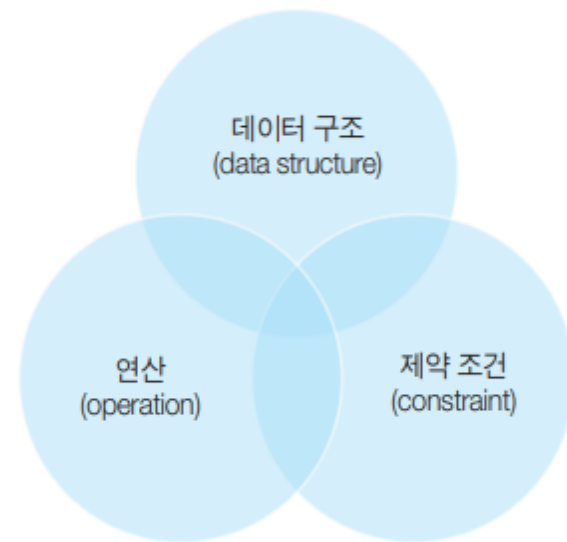
- ✓ 데이터를 어떤 형태로 저장하는지를 표현하는 추상적 표현

✓ 연산(operation)

- ✓ 개념적인 데이터 구조 안의 데이터를 어떤 방식으로 처리하는지를 표현

✓ 제약 조건(constraint)

- ✓ 데이터 구조 안에 데이터를 저장할 때의 구조적 제약 사항과 연산을 적용할 때의 행위적 제약 사항을 표현





테이블(릴레이션)

- 테이블(table): 행과 열로 구성된 2차원의 표 형태

테이블 릴레이션

학생

필드 (속성, 열, 애트리뷰트) column

번호	이름	학년	분반	학과번호
1	하지혜	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

학생(번호, 이름, 학년, 분반, 학과번호)



[요약] 테이블(릴레이션)

-테이블 (릴레이션)

학생 테이블 《출력 형태》	데이터베이스																																						
<div><div>테이블 릴레이션</div><div>학생</div><div>필드 (속성, 열, 애트리뷰트) column</div><table border="1"><thead><tr><th>번호</th><th>이름</th><th>학년</th><th>분반</th><th>학과번호</th></tr></thead><tbody><tr><td>1</td><td>하지혜</td><td>1</td><td>YB</td><td>1</td></tr><tr><td>2</td><td>이정우</td><td>1</td><td>YA</td><td>1</td></tr><tr><td>3</td><td>오지영</td><td>2</td><td>J1</td><td>2</td></tr><tr><td>4</td><td>강재미</td><td>1</td><td>YB</td><td>1</td></tr><tr><td>5</td><td>박철호</td><td>2</td><td>J1</td><td>2</td></tr></tbody></table><div>레코드 (튜플, 행) row</div></div>	번호	이름	학년	분반	학과번호	1	하지혜	1	YB	1	2	이정우	1	YA	1	3	오지영	2	J1	2	4	강재미	1	YB	1	5	박철호	2	J1	2	<p>테이블(table) : 릴레이션(Relation) 필드(field) : 속성, 열, 애트리뷰트 column 레코드(record) : 튜플(tuple), 행 row</p> <p>학과 테이블 《출력 형태》</p> <p>학과</p> <table border="1"><thead><tr><th>학과번호</th><th>학과명</th></tr></thead><tbody><tr><td>1</td><td>컴퓨터소프트웨어공학과</td></tr><tr><td>2</td><td>컴퓨터정보공학과</td></tr><tr><td>3</td><td>정보통신과</td></tr></tbody></table>	학과번호	학과명	1	컴퓨터소프트웨어공학과	2	컴퓨터정보공학과	3	정보통신과
번호	이름	학년	분반	학과번호																																			
1	하지혜	1	YB	1																																			
2	이정우	1	YA	1																																			
3	오지영	2	J1	2																																			
4	강재미	1	YB	1																																			
5	박철호	2	J1	2																																			
학과번호	학과명																																						
1	컴퓨터소프트웨어공학과																																						
2	컴퓨터정보공학과																																						
3	정보통신과																																						

학생(번호, 이름, 학년, 분반, 학과번호)

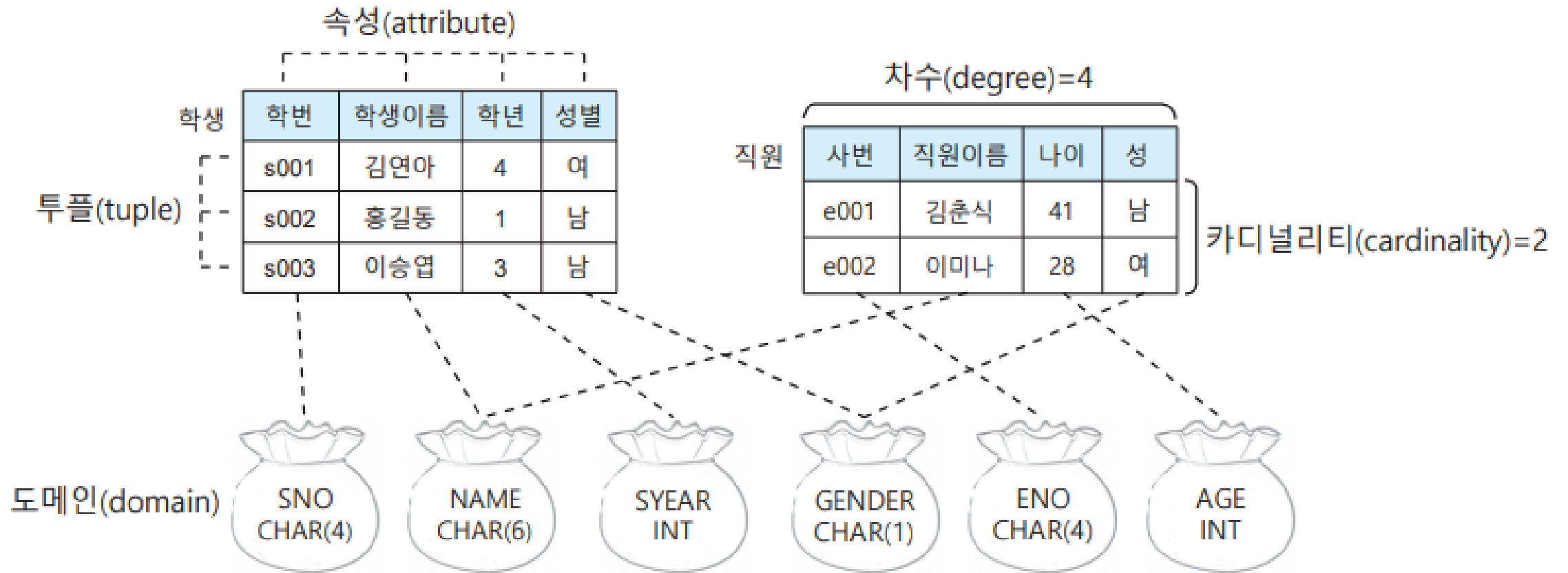
학과(학과번호, 학과명)

용어



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

용어



(그림 출처: ""SQL과 NoSQL 기반의데이터베이스 입문", 박성진, 생능, 2023)

스키마(schema)



□스키마

- ✓데이터베이스를 구성하는 데이터의 구조와 제약조건에 대한 명세를 구체적으로 기술한 것

릴레이션		학생				릴레이션 스키마 (relation schema) 릴레이션 인스턴스 (relation instance)	
		학번	이름	학년	성별		
		s001	김연아	4	여		
		s002	홍길동	1	남		
		s003	이승엽	3	남		

(그림 출처: "'SQL과 NoSQL 기반의데이터베이스 입문', 박성진, 생능, 2023)

스키마 vs. 인스턴스



* 릴레이션 스키마와 인스턴스

-데이터베이스 스키마(Schema) : 데이터베이스의 구조와 제약조건에 대하여 기술한 것

-데이터베이스 상태(State) : 특정시점의 데이터베이스 내용



[Quiz] 신입생

1. 신입생 테이블의 스키마 표기법을 적으시오
2. 차수(degree) 는?
3. 카디날리티(cardinality) 는?
4. 기본키(Primary 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

릴레이션의 특성

□ 튜플의 유일성

- ✓ 하나의 릴레이션에는 동일한 튜플이 존재할 수 없다.

□ 튜플의 무순서성

- ✓ 하나의 릴레이션에서 튜플 사이의 순서는 무의미하다.

□ 속성의 무순서성

- ✓ 하나의 릴레이션에서 속성 사이의 순서는 무의미하다.

□ 속성의 무순서성

- ✓ 속성 값으로 **원자 값**만 사용할 수 있다.

2.3 릴레이션의 특성 (stu)

2.3 릴레이션의 특성 (stu)

2.3 릴레이션의 특성 (stu)

2.3 릴레이션의 특성 (stu)

[참고] 관계 데이터 모델 개념 (stu)

- 릴레이션의 특성: 속성의 원자성

키의 종류

담당교수: 김희숙
(jasmin11@hanmail.net)

키(Key)

□기본키(Primary Key)

- ✓ 각 튜플을 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**

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

FK

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] 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

기본키 설정 (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

키(Key)



□기본키(Primary Key)

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

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

키(Key)



□외래키(Foreign Key)

- ✓한 릴레이션의 기본키를 참조하는 키
- ✓참조되는 릴레이션의 기본키 값에 반드시 존재해야 한다

(외래키는 널 값 입력허용)

- ✓참조무결성 제약조건

고객 테이블

고객 ID	이름	주민등록번호	전화번호
1	홍길동	111111-111111	011-234-5678
2	원도우	222222-222222	012-546-8526
3	김한빛	333333-333333	014-654-8752

주문 테이블

물품 ID	물품 이름	고객 ID
1	A	1
2	B	1
3	C	2

키(Key): 기본키

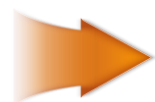
[동일한 이름 입력]

직원

성명	직급	급여	부서명
김사랑	사원	300	인사부
한예슬	대리	250	영업부
오세오	과장	500	영업부
구병서	부장	600	인사부

데이터입력

오세오	사원	300	경리부
-----	----	-----	-----



[기본키]

직원

사번	성명	직급	급여	부서명
111	김사랑	사원	300	인사부
222	한예슬	대리	250	영업부
333	오세오	과장	500	영업부
444	구병서	부장	600	인사부

데이터입력

555	오세오	사원	300	경리부
-----	-----	----	-----	-----

키(Key): 기본키

[복합키]

판매 실적

도시	지점명	판매액
서울	강남지점	330
서울	강북지점	330
광주	광주지점	190
서울	강서지점	150
서울	강동지점	180
대전	대전지점	240



키(Key): 외래키

[중복 데이터 발생]

거래처직원

사번	성명	직급	급여	부서명	위치
111	김사랑	사원	300	인사부	수원
222	한예슬	대리	250	영업부	용인
333	오세오	과장	500	영업부	용인
444	구병서	부장	600	인사부	수원

[중복 데이터 분리] 기본키, 외래키

거래처직원

사번	성명	직급	급여	부서
111	김사랑	사원	300	10
222	한예슬	대리	250	20
333	오세오	과장	500	20
444	구병서	부장	600	10

거래처부서

부서번호	부서명	위치
10	인사부	수원
20	영업부	용인
30	총무부	서울

[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) (stu)

키(Key)

--주소록(이름, 전화번호, 주소, 생일)				--주소록(번호, 이름, 전화번호, 주소, 생일)				
이름	전화번호	주소	생일	번호	이름	연락처	거주지	생일
이몽룡	010-3354-5643	부산	12월 14일	1	홍길동	010-1234-5678	서울	1990-03-15
최용만	321-2345	대전	5월 8일	2	이몽룡	010-3354-5643	부산	1994-12-14
홍길동	010-1234-5678	서울	3월 15일	3	최용만	011-321-2345	대전	1994-05-08
				* 번호 필드 추가, 생일 필드에 날짜형식(date)으로 입력				

[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

수퍼키의 성질: 유일성

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

Quiz

5주차

담당교수: 김희숙
(jasmin11@hanmail.net)

[Quiz] 테이블 생성: (학원생, 학교)

부모 테이블:
자식 테이블:

학원생

학원생이름	폰번호	나이	학교이름	학년	반	반번호
홍길동	010-1237-6542	19	송원고	3	1	10
김하나	010-3218-8765	18	한빛고	2	2	7
홍길동	010-4587-9834	19	송원고	3	1	31
박순희	010-7789-6754	17	이슬고	1	3	16

외래키(FK)

참조

학교

학교명	분류	학생수	주소
송원고	3	435	경기 성남 분당구 황새울로 123
한빛고	2	377	경기 성남 분당구 판교역로 67
이슬고	1	507	경기 성남 분당구 미금로 567

기본키(PK)

1. 학원생 테이블의 스키마 표기법을 작성하시오
2. 차수(degree) 는?
3. 카디날리티(cardinality) 는?
4. 기본키(Primary key) 는?
5. 외래키(Foreign key) 는?

1. 학교 테이블의 스키마 표기법을 작성하시오
2. 학교 테이블의 인스턴스 는?
3. 학교 테이블의 학교명 도메인(domain) 은?

[Quiz] 테이블 생성: (환자, 의사)

부모 테이블:

자식 테이블:

-- 환자(환자번호,환자이름,나이,담당의사)

-- 의사(의사번호,의사이름,소속,근무연수)

환자

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

의사

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

1) 환자 테이블의
기본키는? 외래키는?

2) 의사 테이블의
기본키는? 외래키는?

1. 환자, 학생 테이블을
각각 작성하시오(기본키, 외래키 설정)

2. 환자, 학생 테이블에
데이터를 각각 입력하시오

2. 환자, 학생 테이블을
각각 조회하시오

[과제01] (데이터베이스)

테이블 2개 생성(기본키, 외래키)

담당교수: 김희숙
(jasmin11@hanmail.net)

[과제01] (MySQL)

SQL작성(MySQL)
실습하기
(실습 화면캡처하여
워드에 작성)

□[과제01] **MySQL 프로그램 사용**하여 테이블 각각 작성하고
화면캡처하여 워드에 작성하시오

1) 제출방법: eClass-과제

첨부파일로 제출

2) 파일명: 과제01-분반_이름_학번-작성날짜.doc (또는 hwp)

(예. 과제01-QC_홍길동_24229999-제출날짜)

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

제출방법:
eClass [과제] 제출

[과제01] (실습)

SQL작성(MySQL) 실습하기 (화면캡처 워드작성)

[과제01] 다음을 SQL문법으로 작성하여 실습하시오(MySQL 실습)

* (문제파일: HW1-MySQL-테이블2개-stu.pdf)

[과제 1-1] 다음 테이블 2개 생성(환자, 의사)
(기본키, 외래키 작성) 하고 데이터 입력

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

SQL 작성

[과제 1-2] 테이블 생성(사원)

- 1-1) 테이블 생성 → 다음 테이블을 생성하는 SQL 문법 작성(기본키 설정)
- 1-2) 데이터 입력 → 본인 이름을 입력한 레코드를 입력하시오
- 1-3) 데이터 조회 → 테이블 조회한 결과를 화면캡처하여 작성하시오
- 1-4) 차수(degree)는 얼마인가?
- 1-5) 카디널리티(cardinality)는 얼마인가?
- 1-6) 사원이름 속성은 기본키로 적합하지 않다. 그 이유는 무엇인가?
- 1-7) 기본키로 적합한 속성은 무엇인가?

사원번호	사원이름	나이	주소	직급
E001	홍준화	30	서울시 마포구	대리
E002	김연주	28	서울시 영등포구	사원
E003	이명기	32	서울시 강남구	사원

(답안)[Quiz] SQL: 테이블 2개 생성(dept,dmember)

```
-- dept, dmember 테이블 생성  
use studydb;
```

```
DROP TABLE if exists dmember;  
DROP TABLE if exists dept;
```

```
CREATE TABLE dept (  
    dept_id char(3) NOT NULL ,  
    name varchar(20) ,  
    PRIMARY KEY(dept_id)  
);
```

```
CREATE TABLE dmember (  
    name varchar(10) NOT NULL ,  
    dept_id char(3) ,  
    PRIMARY KEY(name) ,  
    CONSTRAINT fk_dmember_dept  
        foreign key(dept_id)  
        references dept(dept_id)  
);
```