

## 4. 관계 데이터베이스

# ❖ 관계 데이터 모델(1)

- ◆ 관계 데이터 모델(relational data model)의 탄생
  - 1970년대 IBM의 E. F. Codd에 의해 제안
- ◆ 관계 데이터 모델의 특성
  - 릴레이션(relation)과 수학적 이론에 기초
  - 일반 사용자는 테이블 형태로 생각
    - ◆ 통상적인 테이블의 개념과는 다름
    - ◆ 관계 데이터 모델의 직관적인 이해에 도움
    - ◆ 테이블의 열(column) = 필드(field) 혹은 아이템(item)
      - ≡ 관계 데이터 모델의 애트리뷰트(attribute)
    - ◆ 테이블의 행(row) = 레코드(record)
      - ≡ 관계 데이터 모델의 투플(tuple)

## ❖ 관계 데이터 모델(2)

학생(STUDENT) 테이블 : 릴레이션

학생  
(STUDENT)

학번 (Sno)	이름 (Sname)	학년 (Year)	학과 (Dept)
100	나 수 영	4	컴퓨터
200	이 찬 수	3	전기
300	정 기 태	1	컴퓨터
400	송 병 길	4	컴퓨터
500	박 종 화	2	산공

# ❖ 애트리뷰트와 도메인

- ◆ 테이블 ≡ 릴레이션
- ◆ 도메인(domain)
  - 하나의 애트리뷰트가 취할 수 있는 같은 타입의 모든 원자값(value)들의 집합
    - ◆ 같은 도메인의 값들끼리 비교가 허용됨
- ◆ 애트리뷰트(attribute)
  - 도메인의 역할 이름
  - 애트리뷰트 이름들은 모두 달라야 함
- ◆ 단순 도메인 (simple domain)
  - 단순 애트리뷰트 : 원자값
- ◆ 복합 도메인 (composite domain)
  - 복합 애트리뷰트 : 복합값
  - 연, 월, 일 ⇒ 날짜:<연,월,일>

## 👉 Note

- 도메인 : 애트리뷰트 (관계 데이터 모델) ≡ 데이터 타입 : 변수 (프로그래밍 언어)

# ❖ 릴레이션의 개념

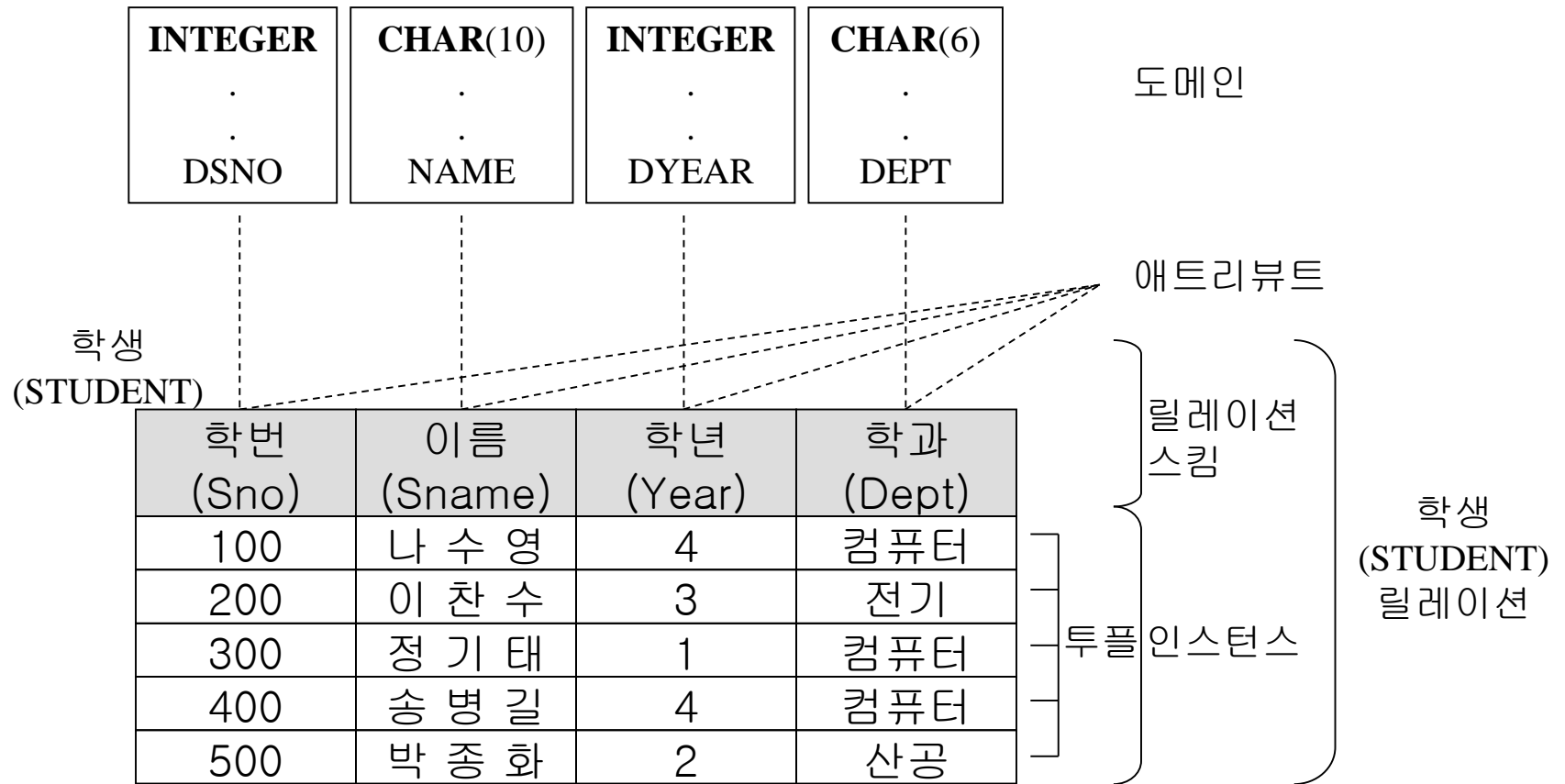
## ◆ 릴레이션 학생(STUDENT)의 정의

```
DCL DOMAIN DSNO INTEGER;  
DCL DOMAIN NAME CHAR(10);  
DCL DOMAIN DYEAR INTEGER;  
DCL DOMAIN DEPT CHAR(6);  
DCL RELATION STUDENT  
    (Sno      DOMAIN DSNO,  
     Sname    DOMAIN NAME,  
     Year     DOMAIN DYEAR,  
     Dept     DOMAIN DEPT);
```

## ◆ 도메인 명세가 생략된 릴레이션 STUDENT의 정의

```
DCL RELATION STUDENT  
    (Sno      INTEGER,  
     Sname    CHAR(10),  
     Year     INTEGER,  
     Dept     CHAR(6));
```

# ▶ 릴레이션의 예



## ▶ 릴레이션 스키 (relation scheme)

- ◆ 릴레이션 스키마라고도 함  
릴레이션 이름 + 애트리뷰트 이름  
 $R(A_1, A_2, \dots, A_n), \quad A_i \Leftrightarrow D_i$   
 $\equiv R(\{A_1, A_2, \dots, A_n\})$
- ◆ 정적 성질
  - 시간에 무관
  - 릴레이션 타입

# ▶ 릴레이션 인스턴스 (relation instance)

## ◆ 릴레이션 R의 릴레이션 인스턴스

- 어느 한 시점에 릴레이션 R이 포함하고 있는 튜플들의 집합  
 $\{ \langle V_1, V_2, \dots, V_n \rangle \mid V_i \in D_i \}$
- 릴레이션의 내용, 상태

## ◆ $\{(\text{attr}_1=V_1, \text{attr}_2=V_2, \dots, \text{attr}_n=V_n)\}$

## ◆ 동적 성질

- 삽입, 삭제, 갱신으로 시간에 따라 변함
- 릴레이션 값



# ▶ 릴레이션(Relation) R

## i. 수학적 정의

릴레이션 R : 카티션 프로덕트(cartesian product)의 부분집합

$$R \subseteq D_1 \times D_2 \times \dots \times D_n$$

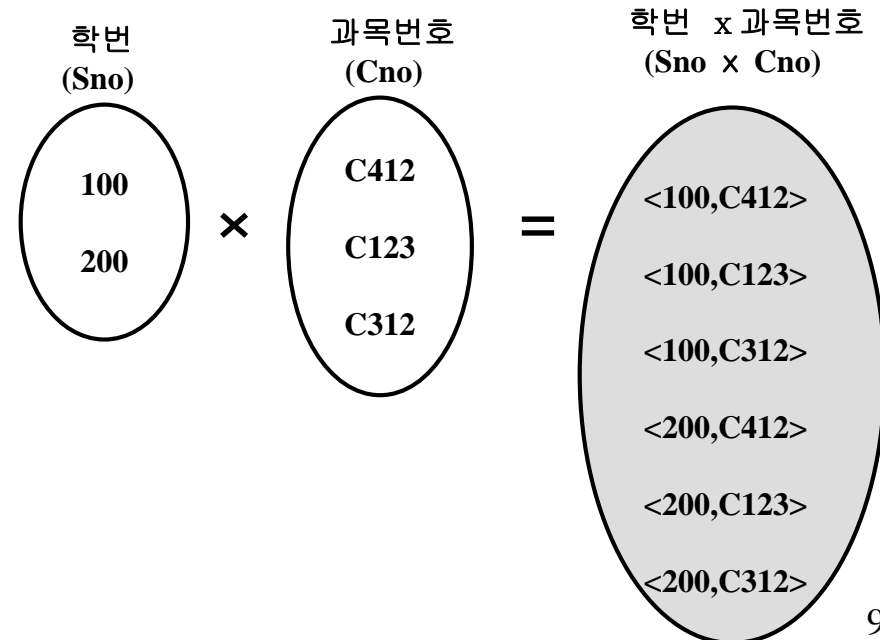
즉 n-튜플  $\langle d_1, d_2, \dots, d_n \rangle$ 의 집합

단  $D_i$  : i번째 도메인  $d_i \in D_i, i = 1, 2, \dots, n$

- n : R의 차수(degree : 일차, 이차, 삼차, ..., n차)
- 릴레이션에 포함되어 있는 튜플의 수 : 카디널리티(cardinality)

## ii. 개념적 정의

릴레이션 스키 + 릴레이션 인스턴스



## ❖ 릴레이션의 특성 (1)

### i. 튜플의 유일성

릴레이션 = 서로 다른 튜플들의 "집합"

### ii. 튜플의 무순서성

릴레이션 : 추상적 개념  $\Leftarrow$  튜플들의 집합

테이블 : 구체적 개념

### iii. 애트리뷰트의 무순서성

릴레이션 스키마  $\rightarrow$  애트리뷰트들의 "집합"

튜플 :  $\langle \text{attr:value} \rangle$ 쌍의 집합

## ❖ 릴레이션의 특성 (2)

### iv. 애트리뷰트의 원자성

- 애트리뷰트 값 = 원자 값(atomic value)
  - ◆ 논리적으로 분해 불가능
- 정규화 릴레이션 (normalized relation)
  - ◆ 비정규화 릴레이션은 분해로 정규화
- 널 값(null value) = 원자 값
  - ◆ unknown, not applicable
- 도메인
  - ◆ 단순 도메인
  - ◆ 복합 도메인 : 값을 하나의 단위로 취급

# ▶ 릴레이션의 정규화

- 정규화 릴레이션 : 반복 그룹을 애트리뷰트 값으로 허용하지 않는 릴레이션

등록1 (ENROL1)

학번 (Sno)	과목성적 (Cgrade)	
	과목번호 (Cno)	성적 (Grade)
100	C413	A
	E412	A
200	C123	B
300	C312	A
	C324	C
	C413	A
400	C312	A
	C324	A
	C413	B
	E412	C
500	C312	B

(a) 비정규 릴레이션

등록 (ENROL)

학번 (Sno)	과목번호 (Cno)	성적 (Grade)
100	C413	A
100	E412	A
200	C123	B
300	C312	A
300	C324	C
300	C413	A
400	C312	A
400	C324	A
400	C413	B
400	E412	C
500	C312	B

(b) 정규 릴레이션

# ❖ 관계 데이터베이스

- ◆ 관계 데이터베이스
  - 데이터베이스를 시간에 따라 그 내용(상태)이 변할 수 있는 테이블 형태로 표현
- ◆ 관계 데이터베이스 스키마 = {릴레이션 스키마} + 무결성 제약조건
- ◆ 관계 데이터 모델  $\Leftrightarrow$  프로그래밍 시스템
  - 릴레이션  $\Leftrightarrow$  화일
  - 튜플  $\Leftrightarrow$  레코드 (레코드 어커런스)
  - 애트리뷰트  $\Leftrightarrow$  필드(필드 타입)

# ※ example

## ◆ 대학(University) 관계 데이터베이스

학생  
(STUDENT)

<u>학번</u> (Sno)	이름 (Sname)	학년 (Year)	학과 (Dept)
100	나 수 영	4	컴퓨터
200	이 찬 수	3	전기
300	정 기 태	1	컴퓨터
400	송 병 길	4	컴퓨터
500	박 종 화	2	산공

과목  
(COURSE)

<u>과목번호</u> (Cno)	과목이름 (Cname)	학점 (Credit)	학과 (Dept)	담당교수 (PRname)
C123	프로그래밍	3	컴퓨터	김성국
C312	자료 구조	3	컴퓨터	황수관
C324	화일 구조	3	컴퓨터	이규찬
C413	데이터베이스	3	컴퓨터	이일로
E412	반 도 체	3	전자	홍봉진

## ※ example

### ◆ 대학(University) 관계 데이터베이스(cont'd)

등록  
(ENROL)

<u>학번</u> (Sno)	<u>과목번호</u> (Cno)	성적 (Grade)	중간성적 (Midterm)	기말성적 (Final)
100	C413	A	90	95
100	E412	A	95	95
200	C123	B	85	80
300	C312	A	90	95
300	C324	C	75	75
300	C413	A	95	90
400	C312	A	90	95
400	C324	A	95	90
400	C413	B	80	85
400	E412	C	65	75
500	C312	B	85	80

# ❖ 기본 키(Primary key)

## ◆ 키(key)

- 튜플을 유일하게 식별할 수 있는 애트리뷰트 집합

## ◆ 후보 키(candidate key)

- 릴레이션  $R(A_1, A_2, \dots, A_n)$ 에 대한 애트리뷰트 집합  $K = \{ A_i, A_j, \dots, A_k \}$ 로서 아래 성질을 만족하면 후보키

### ① 유일성(uniqueness)

각 튜플의  $K (= \{ A_i, A_j, \dots, A_k \})$ 의 값  $\langle v_i, v_j, \dots, v_k \rangle$ 은 유일

### ② 최소성(minimality)

$K$ 는 튜플을 유일하게 식별하기 위해 필요한 애트리뷰트로만 구성



## ❖ 기본 키 (2)

### ◆ 슈퍼 키 (super key)

- 유일성(uniqueness)은 만족하지만 최소성(minimality)이 만족되는 않는 애트리뷰트의 집합

### ◆ 기본 키 (primary key)

- 후보 키(candidate key) 중에서 데이터베이스 설계자가 지정한 하나의 키
- 각 튜플에 대한 기본 키 값은 항상 유효(no null value)
- 값에 기반한 내용 중심 주소법 제공

### ◆ 대체 키 (alternate key)

- 후보 키 중에 기본 키를 제외한 나머지 후보 키

## ❖ 외래 키(Foreign key)

### ◆ 외래 키(foreign key)

- 릴레이션 R에 속한 애트리뷰트 집합 FK가 릴레이션 S의 기본 키일 때 FK는 R의 외래 키이다.
- (FK의 도메인) = (S의 기본 키의 도메인)
- FK의 값은 S에 존재하는 값이거나 null
- R을 참조 릴레이션(referencing relation), S를 피참조 릴레이션(referenced relation)이라 함

## ❖ 외래 키 (2)

### ◆ $R \neq S$ 인 경우

- R의 외래키가 반드시 R의 기본키에 포함될 필요는 없음

교수 (교수번호, 교수이름, 학과번호, 직급)

학과 (학과번호, 학과이름, **학과장교수번호**, 학생수)

PK

**FK**

학생 (학번, 이름, 학년, 학과)

과목 (과목번호, 과목이름, 학점, 학과, 담당교수)

등록 (**학번**, **과목번호**, 성적)

**FK**

**FK**

### ◆ $R = S$ 인 경우

- R과 S가 반드시 다른 릴레이션일 필요는 없음

교수1 (교수번호, 교수이름, 학과번호, **학장교수번호**)

PK

**FK**

## ❖ 무결성 제약(Integrity Constraint)

### (1) 개체 무결성(entity integrity)

- 기본 키 값은 언제 어느 때고 null값을 가질 수 없다.

### (2) 참조 무결성(referential integrity)

- 외래 키의 값은 참조된 릴레이션의 기본 키 값이거나 null이다.

- ◆ 데이터베이스 상태(database state)가 항상 만족시켜야 하는 기본 규칙

## ❖ 무결성 제약(Integrity Constraint)(2)

- ◆ 데이터베이스 상태 (database state)
  - 어느 한 시점에 데이터베이스에 저장된 데이터 값
  - 데이터베이스 인스턴스
  - 데이터베이스 상태 변화: 삽입, 삭제, 변경 연산
  - DBMS는 데이터베이스 상태의 변화에도 항상 무결성 제약을 만족시키도록 해야 함