

# DB05

## Relational Model Concepts

**Database:** Relation의 집합

- Relation은 Table이다.
- Table의 집합이다.

**Relation:** Tuple의 집합

- 행의 집합

**Tuple:** 실세계의 Entity나 Relationship에 대응되는 사실

- Attribute의 집합이다.

**Attribute:** Tuple에 대응되는 각각의 Object의 성질을 정의한 것

**University Database**를 예시로 들 수 있다.

### STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

### COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

### PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

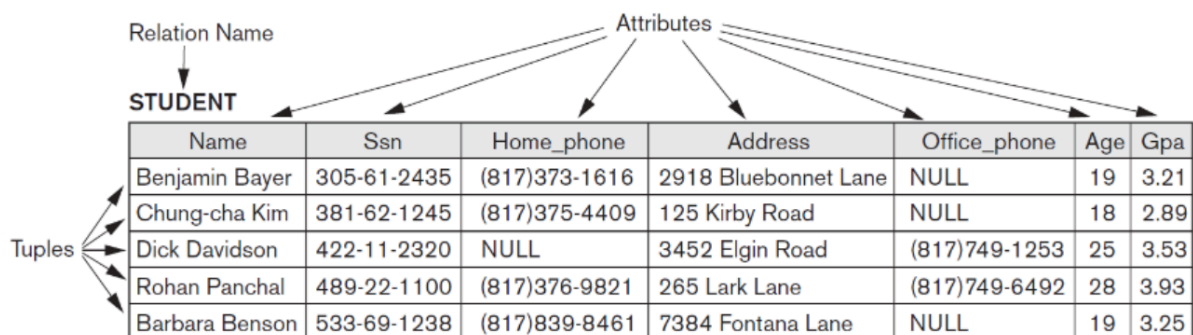
### SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

### GRADE\_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

- 위 전체가 Database이다.
- 각각의 STUDENT, SECTION 등은 Relation이다.
- 각 Table의 한 Row가 Tuple이다.
- STUDENT, COURSE, SECTION table의 각 Tuple은 Entity를 표현한다.
- PREREQUISTE, GRADE\_REPORT table의 각 Tuple은 Relationship을 표현한다.



- **Table name:** STUDENT
- **Attribute name:** Name, Ssn, Home\_phone, Address, Office\_phone, Age, Gpa
- Table(Relation) schema: **Table Name + Attribute Name**

- Benjamin Bayer ~ Barbara Benson까지의 Tuple의 집합이 **Relation instance**이다.

## Domain D

- **Atomic value**의 집합
- 어떤 Attribute가 가질 수 있는 값의 집합

## Attribute $A_i$

- 어떤 **Domain D**가 **Relation schema R**에서 어떤 역할을 하는지
- **dom( $A_i$ ): Domain of attribute  $A_i$  in relation schema R**
- EX) Relation schema R의 Employee\_ages attribute는 dom(Employee\_ages)를 domain으로 갖는다.

## Relation schema

- **Relation name + attribute의 리스트**
- $R(A_1, A_2, \dots, A_N)$ 으로 표현된다.
  - EX) STUDENT(NAME, SSN, HomePhone, Address, OfficePhone, Age, GPA)

## Degree of a relation

- **Relation schema(Table)의 attribute의 개수**
- 위에서 STUDENT relation의 degree는 7이다.

## Tuple t of relation R

- **n-tuple**이라고도 부른다.
  - 이때의 n은 attribute의 개수이다.
- $t = \langle v_1, v_2, \dots, v_n \rangle$ 으로 표현되며 **n개의 값의 순서가 있는 list**이다.
- 각  $v_i$ 는 **dom( $A_i$ )의 element**이다.

- $A_i$ 의 Domain 중 하나의 값이다.
- NULL 값을 가질 수도 있다.

### Relation instance $r(R)$

- **set of tuples**
    - $r(R) = t_1, t_2, \dots, t_m$
  - Relation을 정의하는 Domain의 **Subset of the Cartesian product**
    - $r(R) \subseteq (dom(A_1) \times dom(A_2) \times \dots \times dom(A_n))$
    - **Cartesian product**는 모든 가능한 Tuple 조합을 만드는 것이다.
- 

## Relations의 성질

### 1. Relation에서 Tuple간의 순서는 중요하지 않다

- $R_1 = \langle t_3, t_2, t_1 \rangle$ 과  $R_1 = \langle t_1, t_2, t_3 \rangle$ 는 동일하다.
- 집합은 순서가 상관없기 때문이다.

### 2. 하나의 Tuple 내에서 Attribute의 순서는 중요하다

- **n-Tuple**  $t = \langle v_1, v_2, \dots, v_n \rangle$ 의 정의가 **Ordered list of n values**이기 때문이다.
- Attribute의 순서가 다른 두 Tuple은 서로 다른 Tuple로 봐야한다.

하지만 실제로 Tuple을 `{Name: 'Smith', Student_number: 17}` 와 같이 **(Attribute, value)**의 쌍으로 본다면 **(Attribute, value)**의 쌍의 순서는 중요하지 않다.

- Attribute에 대응되는 value를 찾으면 되기 때문이다.

### 3. Tuple의 각 Value는 Atomic하다.

- **Atomic:** 더 나눌 수 없다 / **Value**의 일부만을 사용하면 검색이 불가하고 아무 의미도 없다.
  - EX) 주소에서 서울시 성동구 중에 서울시만 사용하면 아무 의미도 없다.
  - **Compostie and multivalued attributes**가 허용되지 않는다.
  - **Table**이 **First Normal Form**을 만족함을 가정한다.
- **NULLs:** **Attribute**에 대응되는 값이 존재하지 않거나 적용이 가능하지 않는 경우에 사용하는 값
  - **Not Applicable** (적용 불가능한 값): **EMPLOYEE** 테이블에 **Bonus** 열이 있을 때, 정규직원은 보너스 금액(숫자)이 들어가지만, 보너스 지급 대상이 아닌 인턴사원의 **Bonus** 열은 NULL이다.

---

## Relational Model Natation

$t[A_i]$ : Tuple  $t$ 에서 **Attribute**  $A_i$ 에 대응되는 **value**  $v_i$

$t[A_u, A_w, \dots, A_z]$ : Tuple  $t$ 에서 특정 **Attribute list**에 대응되는 **Value**  $< v_u, v_w, \dots, v_z >$ 의 **Sub-tuple**

---

## Relational Model Constraints

- **Relation schema**의 실제 값에 적용되는 **Restrictions**
- 각 **Relation**의 모든 **Instance**는 **Constraint**를 따라야 한다.
- **Mini-world**의 규칙으로부터 정해진다.

아래 세 가지 종류의 **Constraint**가 존재한다.

### 1. Key Constraints

- **Relation(Table)**은 반드시 **Candidate key**를 가져야 한다.

### Superkey of relational schema R

- **Relational instance**에 속하는 어떤 서로 다른 두 Tuple은 같은 **Superkey** 값을 가지면 안된다.
  - $t_1[SK] \neq t_2[SK]$
- Relation R에 해당되는 **Attribute의 집합**
- **Uniqueness constraint**라고도 한다.

### Candidate key $K$

- 아래 두 가지 조건을 만족해야 한다.
  - 우선 **Relation R의 Superkey**여야 한다.
  - **Minimal Superkey**여야 한다.
    - Superkey에서 **하나의 Attribute만 제거해도 그 Attribute의 집합이 Superkey가 되지 않는 경우**
- 하나의 Relation schema에서 **두 개 이상의 Candidate key**가 존재할 수 있다.

### CAR

<u>License_number</u>	Engine_serial_number	Make	Model	Year
Texas ABC-739	A69352	Ford	Mustang	02
Florida TVP-347	B43696	Oldsmobile	Cutlass	05
New York MPO-22	X83554	Oldsmobile	Delta	01
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	04
Texas RSK-629	U028365	Jaguar	XJS	04

- 위 상황에서 **5가지의 Attribute의 value가 모두 동일한 Tuple**은 없다.
  - {License\_number, Engine\_serial\_number, Make, Model, Year}은 **Superkey**이다.

- 그러나 {License\_number, Engine\_serial\_number, Make, Model, Year}은 **Candidate key**는 아니다.
  - 위에서 Make나 Model을 제외해도 여전히 Superkey이기 때문이다.

위의 **Universal database**를 통한 예시로 살펴볼 수 있다.

## Example: STUDENT relational schema

- {SSN, Name, Age}: is a superkey / *not a candidate key*
- {SSN}: is a superkey / is a candidate key

### Primary key(PK)

- **Database designer**가 하나의 **Relation**에 두 개 이상의 **Candidate key**가 있을 때 그 중 하나를 정해 **Primary key**로 지정한다.
- Primary key에 속하는 Attribute에는 밑줄이 쳐진다.
- 서로 다른 두 Tuple을 구별하는데 사용된다.

**Company database**의 예시를 통해 살펴보자.

## Example: DEPARTMENT relational schema

- DepartmentID, DepartmentName

- DepartmentID와 DepartmentName 모두 Candidate key이다.
- 이 중, Database designer가 DepartmentID를 Primary key로 지정한 것이다.
- 둘 다 Candidate key일 때, **DepartmentID**를 PK로 정한 이유는 ID의 길이가 고정적이기 때문에 **Primary key**를 고정적인 길이를 value로 갖는 Attribute로 설정하는 것이 **Database 설계 관점에서** 유리하기 때문이다.

## 2. Entity integrity constraint

각 Relation schema  $R$ 의 Primary key 내의 Attribute의 Value는 NULL이 아니어야 한다.

- NULL이라면 구분이 불가능하기 때문이다.
- $t[PK] \neq NULL$  for every tuple  $t$  in  $r(R)$

## 3. Referential integrity constraint

2개의 Tables/Relations에 대한 Constraint

- 두 Table 간의 관계를 설명하는 Constraint이다.
- Key constraints와 Entity integrity constraints가 각 Relation에 적용되는 것과 다르다.

Foreign key를 명시하는 Constraint이다.

두 Relation에서 Tuple간의 Consistency (일관성)을 유지하기 위해 사용된다.

한 Relation의 한 Tuple이 다른 Relation의 Tuple을 참조할 때, 존재하는 Tuple을 참조해야 한다.

Example: DNO attribute in EMPLOYEE should refer to *existing value* for DNumber attribute in DEPARTMENT

Foreign key(FK)

- Relation schema  $R_1, R_2$ 가 존재할 때, FK의 집합은 아래 규칙을 만족한다.
  - $R_1$ 이  $R_2$ 를 참조하는 경우를 생각해보자.



- $t_1$  refers to  $t_2$ 라고 한다.
  - $FK(R_1)$ 과  $PK(R_2)$ 는 같은 Domain을 가져야 한다.
  - $FK(R_1)$ 의 값은 실제로 다른 Table에 존재하는  $PK(R_2)$  값을 갖거나 NULL 값을 가져야 한다.
  - $t_1[FK] = t_2[PK] / NULL$
- 

## Relational database schema

Database에 포함된 Relation schema의 집합

- $S = R_1, R_2, \dots, R_m$

Integrity constraints (IC)의 집합을 포함한다.

- **IC**: Relation schema의 Constraint 집합
  - Key constraint도 속한다.

Company database에 대한 ER Model을 Relational database로 표현한 결과이다.

### EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

### DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

### DEPT\_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

### PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

### WORKS\_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

### DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

# Relational Database Instance

같은 Database에 속하는 Relation instance의 집합

- $r_1, r_2, \dots, r_m$
- $r_i$ 는 IC내의 Integrity constraint를 만족하는  $R_i$ 의 Instance이다.
  - DBMS가 각 Relation Instance가 Integrity constraint를 따르도록 강제한다.

Relational database schema에 Relational Database Instance를 추가하는 것은 다음과 같다.

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT\_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
1	Houston
4	Stafford
5	Bellaire
5	Sugarland

WORKS\_ON

<u>Essn</u>	<u>Pno</u>	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

<u>Pname</u>	<u>Pnumber</u>	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Relational database schema에서 참조하는 것은 화살표로 나타낸다.

참조하는 쪽: FK of  $R_1$ 에서 참조당하는 쪽: PK of  $R_2$ 으로 화살표를 그린다.

## EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

## DEPARTMENT

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

## DEPT\_LOCATIONS

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

## PROJECT

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

## WORKS\_ON

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

## DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

