



# **Chpater 2 Review**

## **(Introduction to Relational Model)**



# Structure of Relational Database (Cont.)

## ■ Relation

- Refer to a table

## ■ Attribute

- Refer to a column of a table

## ■ Tuple (or record)

- Refer to a row in a table

attributes  
(or columns)

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

tuples  
(or rows)

**Tuple**



## 2.2 Database Schema

- A database consists of multiple relations
- Database schema
  - Logical design of the database
- Database instance
  - A snapshot of the data in the database at a given instant in time
- Relation schema
  - Logical design of a table
  - A list of attributes and their corresponding domains
- Relation instance
  - A snapshot of the data in a table at a given instant in time



## 2.3 Keys

- Type of Keys: **Superkey, Candidate key, Primary key, Foreign key**
- **Superkey (슈퍼키):**  $K$  is a **superkey** of  $R$  if values for  $K$  are sufficient to identify a unique tuple of each possible relation  $r(R)$

튜플을 식별할 수 있는 하나의 속성(Attributes) 또는 속성들의 집합

ID	이름	주민번호	주소	핸드폰
C1	박지성	810101-1111111	영국 맨체스터	000-5000-0001
C2	김연아	900101-2222222	한국 서울	000-6000-0001
C3	장미란	830101-2333333	한국 강원도	000-7000-0001
C4	추신수	820101-1444444	미국 클리브랜드	000-8000-0001

- **슈퍼키**

- {주민번호}, {주민번호, 이름}, {주민번호, 이름, 주소}, {주민번호, 이름, 핸드폰}  
{ID}, {ID, 이름, 주소}, {ID, 이름, 주민번호, 주소, 핸드폰} 등....



## 2.3 Keys

- **Candidate key(후보키):** **Candidate Key** is minimal superkey for which no proper subset is a superkey

튜플을 식별할 수 있는 슈퍼키들의 최소 집합

ID	이름	주민번호	주소	핸드폰
C1	박지성	810101-1111111	영국 맨체스터	000-5000-0001
C2	김연아	900101-2222222	한국 서울	000-6000-0001
C3	장미란	830101-2333333	한국 강원도	000-7000-0001
C4	추신수	820101-1444444	미국 클리브랜드	000-8000-0001

- 후보키
  - {ID}, {주민번호}
- 슈퍼키
  - {주민번호}, {주민번호, 이름}, {주민번호, 이름, 주소}, {주민번호, 이름, 핸드폰}
  - {ID}, {ID, 이름, 주소}, {ID, 이름, 주민번호, 주소, 핸드폰} 등....



## 2.3 Keys

- **Primary key(주 키, 기본키):** One of the candidate keys is selected to be the **primary key**.

여러 후보키들 중 하나를 선정하여 대표로 삼는 키

기본키 조건

- 튜플을 식별할 수 있는 고유의 값을 가져야함 (슈퍼키, 후보키)
- 최대한 적은 수의 속성을 가져야 함 (후보키)
- 키 값의 변동이 일어나면 안됨
- Null 값이 없어야함





## 2.3 Keys

- **Foreign key** (외래키): Value in one relation must appear in another
  - **Referencing** relation
  - **Referenced** relation

하나의 릴레이션에서 다른 릴레이션의 속성(attribute)을 참조하는 속성  
(일반적으로 외래키는 다른 릴레이션의 기본키를 참조)

ID	이름	주민번호
C1	박지성	810101-1111111
C2	김연아	900101-2222222
C3	장미란	830101-2333333
C4	추신수	820101-1444444

기본키

주문번호	ID	도서
O1	C1	축구의역사
O2	C1	맨유의 전술
O3	C2	피겨 교본
O4	C3	역도의 이해

기본키

외래키

참조



## 2.5 Relational Query Languages

- Procedural vs. non-procedural, or declarative
- “Pure” languages:
  - Relational algebra
  - Tuple relational calculus
  - Domain relational calculus
- Relational operators (Symbol)
  - Selection (  $\sigma$  )
  - Project (  $\Pi$  )
  - Union (  $\cup$  )
  - Difference (  $-$  )
  - Intersection (  $\cap$  )
  - Join
    - ▶ Cartesian product (  $\times$  )
    - ▶ Natural Join (  $\bowtie$  )





# Joining two relations – Cartesian Product

## ■ Cartesian Product

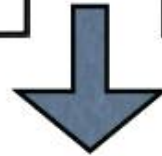
- Combines all rows in the two tables

Product code	Product name	Unit price
101	Melon	800G
102	Strawberry	150G
103	Apple	120G



Export dest. code	Export dest. name
12	The Kingdom of Minanmi
23	Alpha Empire
25	The Kingdom of Ritol

3 rows



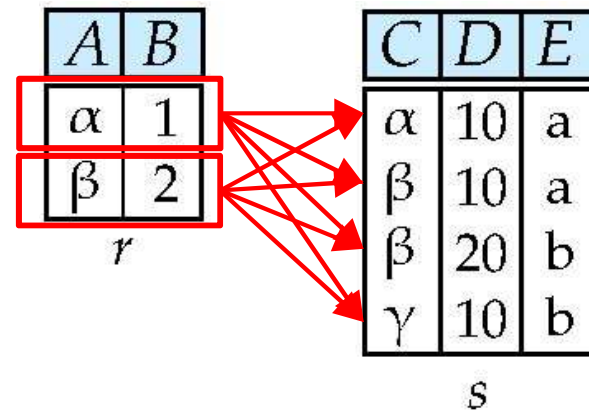
Product code	Product name	Unit price	Export dest. code	Export dest. name
101	Melon	800G	12	The Kingdom of Minanmi
101	Melon	800G	23	Alpha Empire
101	Melon	800G	25	The Kingdom of Ritol
102	Strawberry	150G	12	The Kingdom of Minanmi
102	Strawberry	150G	23	Alpha Empire
102	Strawberry	150G	25	The Kingdom of Ritol
103	Apple	120G	12	The Kingdom of Minanmi
103	Apple	120G	23	Alpha Empire
103	Apple	120G	25	The Kingdom of Ritol

$3 \times 3 =$   
9 rows



# Joining two relations – Cartesian Product

■ Relations  $r$ ,  $s$ :



■  $r \times s$ :

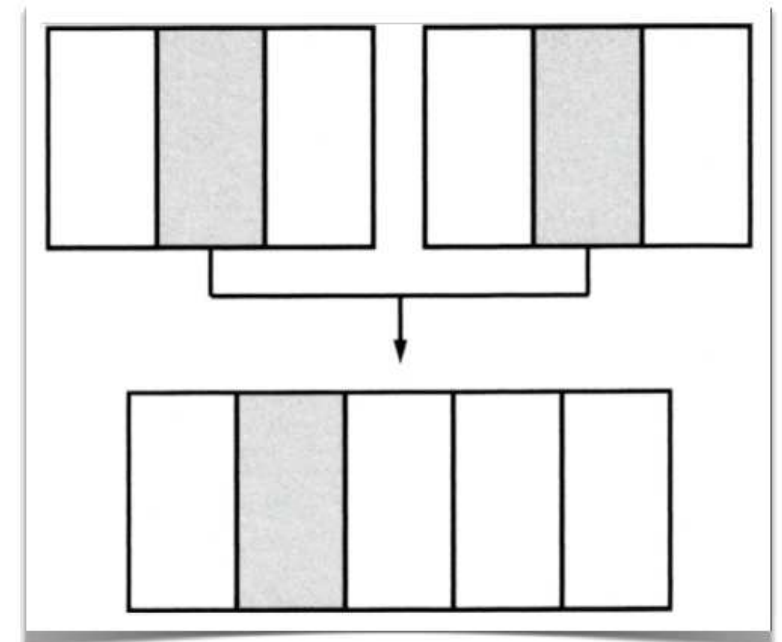
A	B	C	D	E
$\alpha$	1	$\alpha$	10	a
$\alpha$	1	$\beta$	10	a
$\alpha$	1	$\beta$	20	b
$\alpha$	1	$\gamma$	10	b
$\beta$	2	$\alpha$	10	a
$\beta$	2	$\beta$	10	a
$\beta$	2	$\beta$	20	b
$\beta$	2	$\gamma$	10	b



# Joining two relations – Natural Join

- Let  $r$  and  $s$  be relations on schemas  $R$  and  $S$  respectively. Then, the “natural join” of relations  $R$  and  $S$  is a relation on schema  $R \cup S$  obtained as follows:
  - Consider each pair of tuples  $t_r$  from  $r$  and  $t_s$  from  $s$ .
  - If  $t_r$  and  $t_s$  have the same value on each of the attributes in  $R \cap S$ , add a tuple  $t$  to the result, where
    - ▶  $t$  has the same value as  $t_r$  on  $r$
    - ▶  $t$  has the same value as  $t_s$  on  $s$

자연조인이란 릴레이션  $R$ 과  $S$ 에 공통적으로 존재하는 속성들을 이용하여 공통 속성들의 값들이 서로 같은 튜플들을 조인하는 것이다.





# Natural Join Example

■ Relations  $r, s$ :

$A$	$B$	$C$	$D$		$B$	$D$	$E$
$\alpha$	<u>1</u>	$\alpha$	<u>a</u>		<u>1</u>	<u>a</u>	$\alpha$
$\beta$	2	$\gamma$	a		<u>3</u>	<u>a</u>	$\beta$
$\gamma$	4	$\beta$	b		<u>1</u>	<u>a</u>	$\gamma$
$\alpha$	1	$\gamma$	a		<u>2</u>	<u>b</u>	$\delta$
$\delta$	2	$\beta$	b		<u>3</u>	<u>b</u>	$\epsilon$

$r$   $s$

■ Natural Join

■  $r \bowtie s$

$A$	$B$	$C$	$D$	$E$
$\alpha$	1	$\alpha$	a	$\alpha$
$\alpha$	1	$\alpha$	a	$\gamma$
$\alpha$	1	$\gamma$	a	$\alpha$
$\alpha$	1	$\gamma$	a	$\gamma$
$\delta$	2	$\beta$	b	$\delta$



# Relational Algebra

기억할필요없고 표형식으로 어떻게 표현되는지 기억

Symbol (Name)	Example of Use
$\sigma$ (Selection)	$\sigma_{\text{salary} \geq 85000}(\text{instructor})$
	Return rows of the input relation that satisfy the predicate.
$\Pi$ (Projection)	$\Pi_{ID, salary}(\text{instructor})$
	Output specified attributes from all rows of the input relation. Remove duplicate tuples from the output.
$\bowtie$ (Natural Join)	$\text{instructor} \bowtie \text{department}$
	Output pairs of rows from the two input relations that have the same value on all attributes that have the same name.
$\times$ (Cartesian Product)	$\text{instructor} \times \text{department}$
	Output all pairs of rows from the two input relations (regardless of whether or not they have the same values on common attributes)
$\cup$ (Union)	$\Pi_{name}(\text{instructor}) \cup \Pi_{name}(\text{student})$
	Output the union of tuples from the two input relations.