Databases – Introduction to Relational Model (Chapter 2)

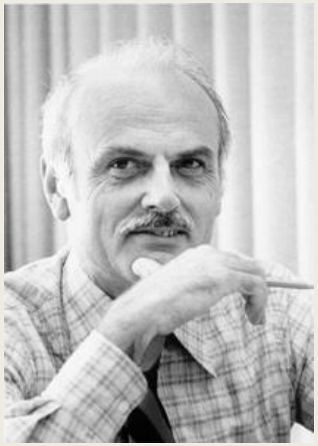
Jaeyong Choi
Dept. of AI-Software, Gachon University

Schedule

Week	Topic	Chapter	Note
1	Introduction to DBMS, Relational Model	1	
2	Relational Algebra : - Concept of Key - Relational algebra operators - Relational algebra expressions	2	추석
3	Introduction to SQL	3	
4	Advanced SQL : - Advanced expression of SQL - Nested SQL queries	4, 5	MOOC
5	Entity/Relationship Model	6	
6	Relational Database Design 1 Relational Database Design 2 (추석보강)	7	MOOC
7	Storage and File Structure	12, 13	MOOC
8	Mid-term Exam		

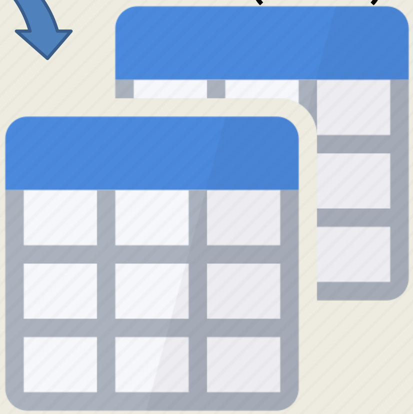
Relational Databases

- **What is Relation (table) Data Model:**
  - a collection of *table* / *relations*
    - Records with pre-defined columns
    - Assigned a unique name (e.g., instructor, department, ...)
  - Primary data model for commercial data-processing
  - Edgar F. Codd defines the Relational Data Model (1972)



Edger F. Codd

Relation (table)



Structure of Relational Databases

INT	CHAR(15)	CHAR(20)	INT

Domain

Schema (Relation Schema)

ID	*name*	*dept_name*	*salary*
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 (rows, records)

Instance (Relational Instance)

attributes (columns, fields)

Alternative Terminology

Table-oriented	Record-oriented	Relational DB
Table	File	
Row	Record	
Column	Field	

ID	*name*	*dept_name*	*salary*
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

Attribute types

- The set of allowed values for each attribute is called the *domain* of the attribute
  - Same data type
  - Order is not important (unordered)
- Attribute values (domains) are (normally) required to be atomic, i.e., indivisible
- The special value *null* is a member of every domain indicating that the value is “unknown” or “non-existent”
  - Causes complications in the definition of many operations

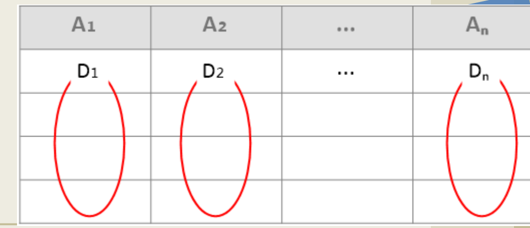
Result Grid | Filter Rows: | Edit

course_id	title	dept_name	credits
CS-111	db	comp. sci	2
CS-437	Database Systems	comp. sci	4
NULL	NULL	NULL	NULL

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

Database Schema

Relation (R)



Relation Schema and Relation Instances

- $A_1, A_2, \ldots, A_n$ are attributes
- $R\ (A_1, A_2, \ldots, A_n)$ is a *relation schema*
  - E.g., *instructor* (*ID*, *name*, *dept_name*, *salary*)
- The current values (*relation instance r*) of a relation are specified by a table form
  - The element *t* of *r* is a *tuple*, represented by a row in a table
  - A table *r* is a set of tuples *t*
- Formally, given domains $D_1, D_2, \ldots, D_n$, a relation R is a subset of $D_1 \times D_2 \times \ldots \times D_n$
  - $R \subseteq D_1 \times D_2 \times \ldots \times D_n$ *(Cartesian Product)*
  - A relation is a set of n-tuples $(a_1, a_2, \ldots, a_n)$ where $a_i \in D_i$
- Relation = Relation Schema + Relation Instances

R = Relation
A = Attribute
r = Relation Instance
t = tuple
D = Domain

Example

$R \subseteq D_1 \times D_2 \times \ldots \times D_n$ *(Cartesian Product)*

Student ID (D_1)

100
200

×

Class Number (D_2)

AI 101
AI 201
AI 301

=

Student ID x Class Number ($D_1 \times D_2$) = R

(100, AI 101)
(100, AI 201)
(100, AI 301)
(200, AI 101)
(200, AI 201)
(200, AI 301)

Student ID	Class Number
100	AI 201
200	AI 101

$R_1 = D_1 \times D_2$

Domain example

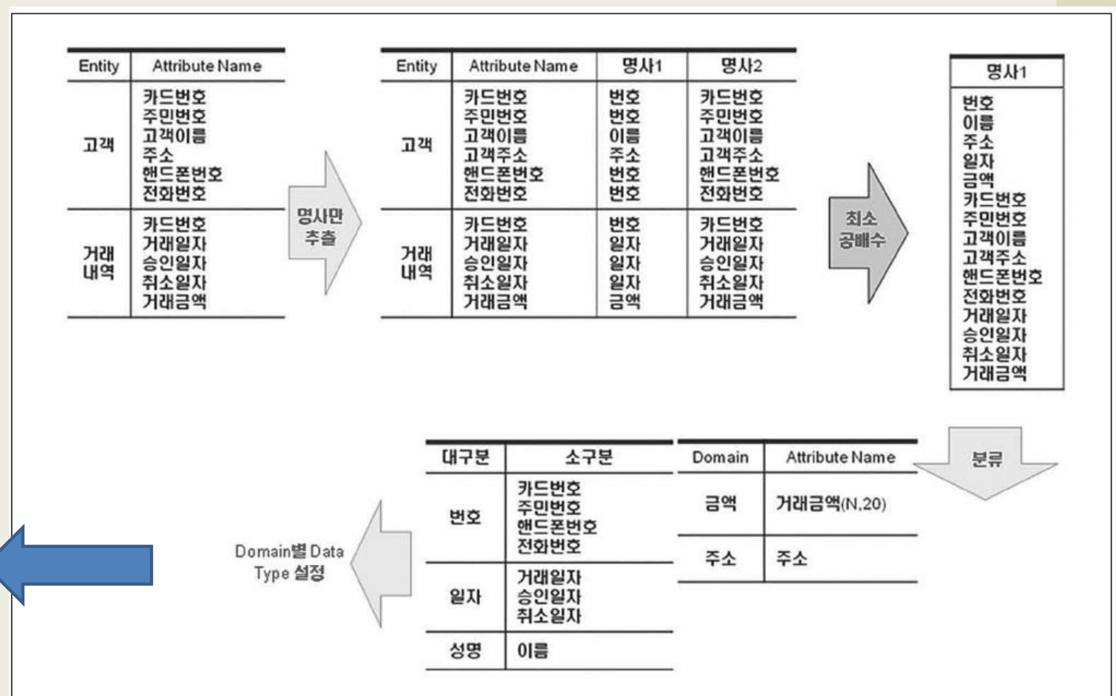
- A database *domain*, at its simplest, is the data type used by a column in a database. This data type can be a built-in type (such as an integer or a string) or a custom type that defines constraints on the data.

Customer

ID Number	Name	Age	Address
INT	CHAR(10)	INT	CHAR(20)

도메인그룹	도메인명	데이터 타입	설명
번호	전화번호	VARCHAR2(13)	
	우편번호	VARCHAR2(7)	
	비밀번호	VARCHAR2(10)	
	번호(PK)	NUMBER	시퀀스를 PK로 사용
금액	금액(N,13)	NUMBER(13)	
	금액(N,6)	NUMBER(6)	
명칭	이름	VARCHAR2(16)	
	제목	VARCHAR2(128)	
수량	주문수량	NUMBER	
여부	사용여부	VARCHAR2(1)	
날짜	일자	VARCHAR2(14)	YYYYMMDDHH24MISS
	월	VARCHAR2(2)	MM
	년도	VARCHAR2(4)	YYYY

〈표2〉 도메인 정의 예제



- Examples
  - *department* (*dept_name*, *building*, *budget*)
  - *teaches* (*ID*, *course_id*, *sec_id*, *semester*, *year*)

dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

Characteristics of Relation

- **One attribute contains same data type**
- **Relations are unordered**
  - Order of attributes / tuples is irrelevant
    - tuples may be stored in an arbitrary order
  - Example: *instructor* relation with unordered tuples

ID	*name*	*dept_name*	*salary*
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

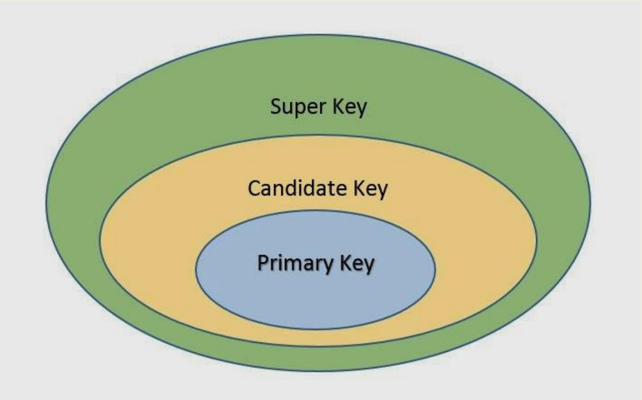
=

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

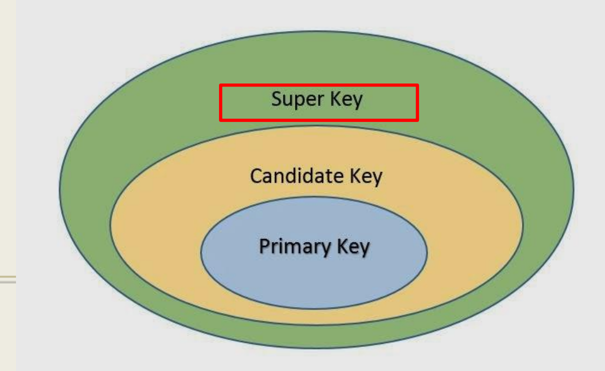
Keys

- Specify how tuples are distinguished

Student ID	Class Number	Grade
100	AI 101	A
200	AI 217	B
100	AI 314	B
200	AI 101	C



Keys



❑ *Super Keys*

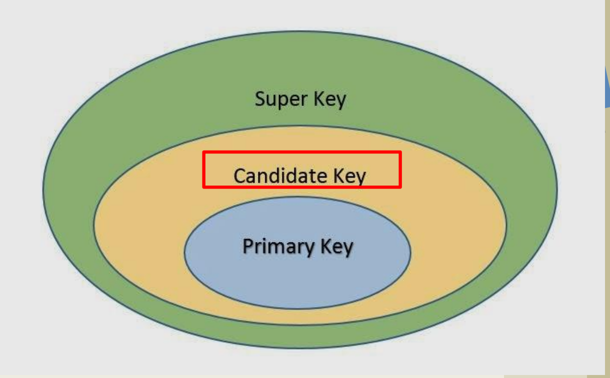
- The **set** of attributes which can *uniquely* identify a tuple
- *K* is a *superkey* of *R* ,if values for *K* are sufficient to identify a unique tuple of each possible relation *r*(*R*)
  - E.g., {*ID*} and {*ID*,*name*} are both superkeys of *instructor*
  - E.g., STUD_NO, {STUD_NO, STUD_NAME}
- Superkey *K* is a *candidate key* if *K* is minimal
  - E.g., {*ID*} is a candidate key for *instructor*

STUDENT

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY	STUD_AGE
1	RAM	9716271721	Haryana	India	20
2	RAM	9898291281	Punjab	India	19
3	SUJIT	7898291981	Rajsthan	India	18
4	SURESH		Punjab	India	21

Table 1

Keys



❑ *Candidate key*

- The *minimal* set of attribute which can uniquely identify a tuple is known as candidate key
  - The value is *unique* and *non-null* for every tuple.
  - There can be more than one candidate key in a relation.
    - E.g., STUD_NO
  - The candidate key can be only one attribute or composite as well.
    - E.g., {STUD_NO, COURSE_NO} is a composite candidate key for relation STUDENT_COURSE.

STUDENT

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY	STUD_AGE
1	RAM	9716271721	Haryana	India	20
2	RAM	9898291281	Punjab	India	19
3	SUJIT	7898291981	Rajsthan	India	18
4	SURESH		Punjab	India	21

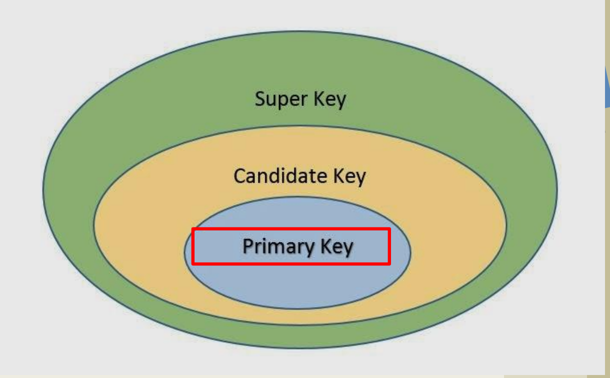
Table 1

STUDENT_COURSE

STUD_NO	COURSE_NO	COURSE_NAME
1	C1	DBMS
2	C2	Computer Networks
1	C2	Computer Networks

Table 2

Keys



❑ *Primary key*

- One of candidate keys is selected as a primary key by the database designer. Values are never, or rarely, changed
- List the primary key attributes before the other attributes with underline
  - *department* (*dept_name*, *building*, *budget*)
  - *course* (*course_id*, *title*, *dept_name*, *credits*)
  - student(stud_no, stud_name, stud_phone, stud_state, stud,countray,stud_age)

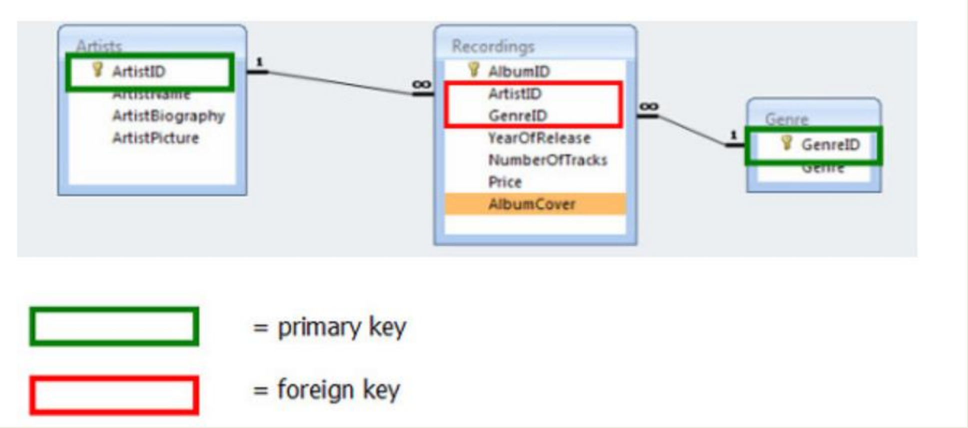
STUDENT

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY	STUD_AGE
1	RAM	9716271721	Haryana	India	20
2	RAM	9898291281	Punjab	India	19
3	SUJIT	7898291981	Rajsthan	India	18
4	SURESH		Punjab	India	21

Table 1

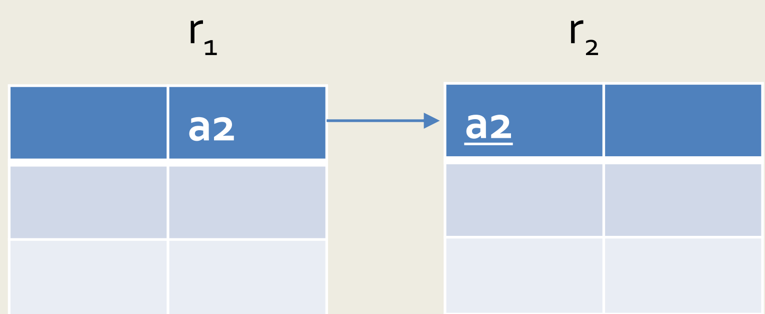
Keys

- ***Foreign key***
  - A key used to link two tables together
  - A FOREIGN KEY is a field in one table that refers to the **PRIMARY KEY** in another table.
  - The table containing the foreign key is called the child table, and the table containing the candidate key is called the referenced or parent table.



Keys

- ***Foreign key***
  - The primary key of r_2, $\boldsymbol{a_2}$ is called **a foreign key** from r_1, referencing r_2 When r_1 includes a_1,
    - r_1 – *referencing* relation, r_2 – *referenced* relation
  - Referential integrity constraint
    - Values of a_2 in r_1 must appear in r_2

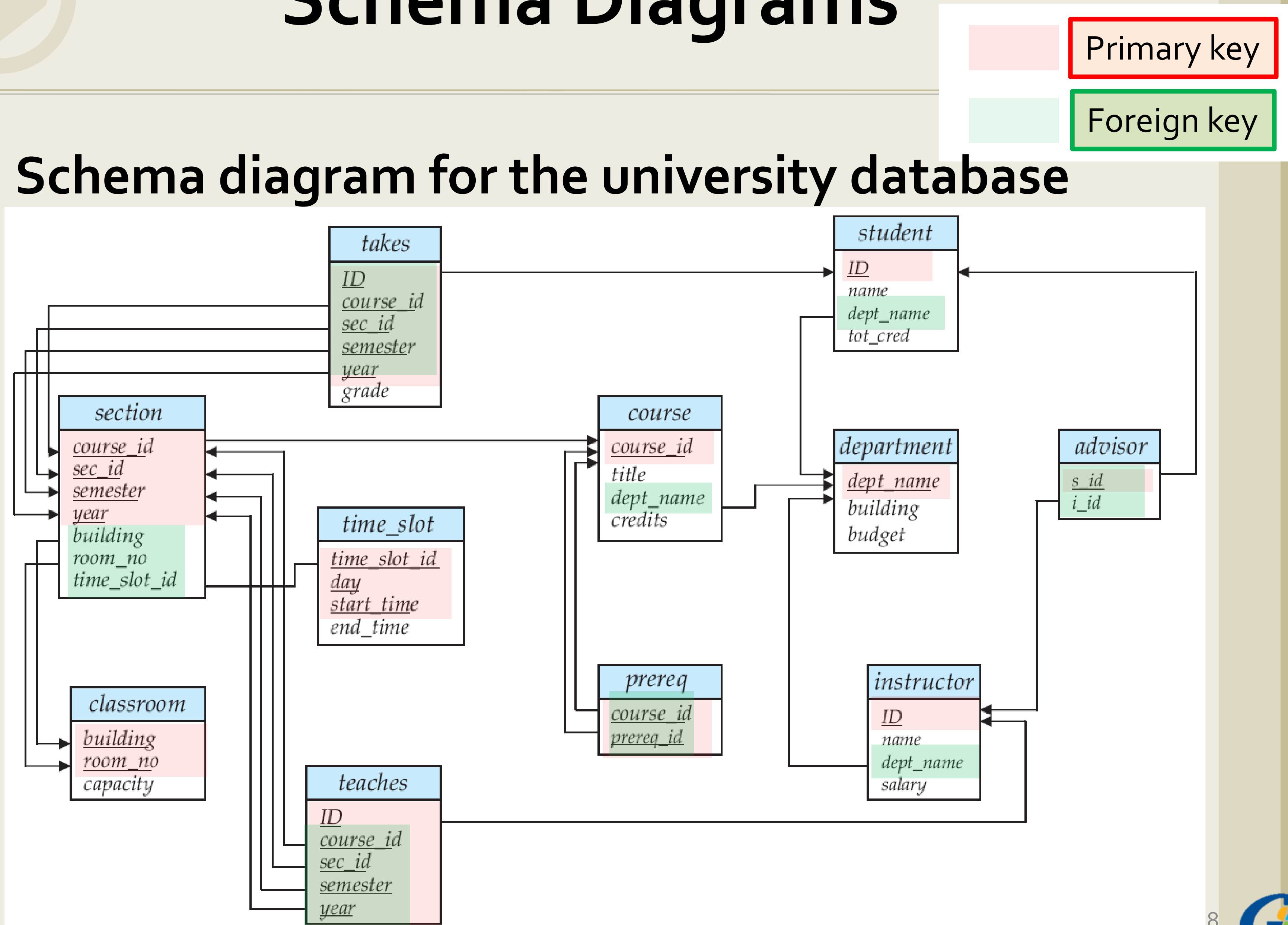


referencing relation (child)

referenced relation (parents)

Schema Diagrams

- Schema diagram for the university database



Example

Primary key
Foreign key

- Foreign Key in the same relation

FacSSN	FacFirstName	FacLastName	FacRank	RacSalary	FacSupervisor
598-76-5432	LEONARD	VINCE	ASST	$35,000	654-32-1098
543-21-0987	VICTORIA	EMMANUEL	PROF	$120,000	
654-32-1098	LEONARD	FIBON	ASSC	$70,000	543-21-0987
765-43-2109	NICKI	MACON	PROF	$65,000	
487-65-4321	JULIA	MILLS	ASSC	$75,000	765-43-2109

Languages of DBMS

- **Data Definition Language (DDL)**
  - define the *schema* and storage stored in a Data Dictionary
- **Data Manipulation Language (DML)**
  - Manipulative populate schema, update database
  - Retrieval querying content of a database
- **Data Control Language (DCL)**
  - permissions, access control etc...

Relational Query Languages

- **Relational query languages**
  - *Procedural* vs. *non-procedural* (*declarative*)
  - "Pure" languages:
    - **Relational algebra**
      - Theoretical basis of SQL query language
    - Tuple relational calculus
    - Domain relational calculus
  - We will concentrate in this chapter on relational algebra
    - Consists of 6 basic operations

Relational Algebra

- A **procedural language** consisting of **a set of operations** that take **one or two relations as input** and **produce a new relation as their result**.

- **Six basic operators**
  - select: σ
  - project: $\prod$
  - union: $\cup$
  - set difference: –
  - Cartesian product: x
  - rename: ρ

- **Additional operators**
  - intersection: $\cap$
  - project: $\prod$
  - join: $\bowtie$

Selection Operation

- ***Select*** – selection of rows (tuples)
  - Syntax: $\sigma_\theta(r)$ (θ: condition)

A	*B*	*C*	*D*
a	*a*	1	7
a	*b*	5	7
b	*b*	12	3
b	*b*	23	10

Relation *r*

A	*B*	*C*	*D*
a	*a*	1	7
b	*b*	23	10

$\sigma_{A=B \wedge D>5}(r)$

Conjunction (and): ∧

Disjunction (or): ∨

Negation (not): ¬

Implication (if..then): →

Equivalence (if and only if): ↔

Selection Operation

- We allow comparisons using
  - $=, \neq, >, \geq. <. \leq$ in the selection predicate.

- We can combine several predicates into a larger predicate by using the connectives:
  - $\wedge$ (and), $\vee$ (or), $\neg$ (not)

Selection Operation

- select those tuples of the instructor relation where the instructor is in the "Physics" department.
- Query

$$\sigma_{\theta}(r)$$

$$\sigma_{\text{dept\_name} = \text{"Physics"}}\ (\text{instructor})$$

- Result

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

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
33456	Gold	Physics	87000

Selection Operation

- Find the instructors in Physics with a salary greater \$90,000 $\sigma_{\text{dept\_name}=\text{"Physics"} \wedge \text{salary} > 90{,}000}(\text{instructor})$
- Find all departments whose name is the same as their building name: $\sigma_{\text{dept\_name}=\text{building}}(\text{department})$

ID	*name*	*dept_name*	*salary*
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

dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

Projection Operation

- ***Project*** – selection of columns (attributes)
  - Syntax: $\Pi_A(r)$ (*A*: attributes)
  - Deletes attributes that are not in *projection* list
    - Eliminate *duplicates*

A	B	C
a	10	1
a	20	1
b	30	1
b	40	2

Relation *r*

A	C
a	1
a	1
b	1
b	2

=

A	C
a	1
b	1
b	2

$\Pi_{A,C}(r)$

Projection Operation

- Eliminate the *dept_name* attribute of instructor
- Query: $\Pi_{\text{ID, name, salary}}$ (instructor)
- Result:

ID	*name*	*dept_name*	*salary*
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

ID	*name*	*salary*
10101	Srinivasan	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

Union operation

- ***Union*** of two relations

A	*B*
a	1
a	2
b	1

Relation *r*

A	*B*
a	2
b	3

Relation *s*

A	*B*
a	1
a	2
b	1
b	3

$r \cup s$

- *Union compatibility*
  - *r* and *s* are *union-compatible*, if they have the same # of attributes and each attribute is from the same domain

Union Operation

- Find all courses taught in the Fall 2017 semester, or in the Spring 2018 semester, or in both

$\sigma_{\theta}(r)$ $\Pi_{A}(r)$

$$\Pi_{course\_id}(\sigma_{semester=\text{"Fall"} \wedge year=2017}(section)) \cup$$
$$\Pi_{course\_id}(\sigma_{semester=\text{"Spring"} \wedge year=2018}(section))$$

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

Figure 2.6 The *section* relation.

course_id
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

Difference operation

- ***Difference*** of two relations

A	*B*
a	1
a	2
b	1

Relation *r*

A	*B*
a	2
b	3

Relation *s*

A	*B*
a	1
b	1

$r - s$

Intersection operation

- ***Intersection*** of two relations
  - Note: $r \cap s = r - (r - s)$

A	*B*
a	1
a	2
b	1

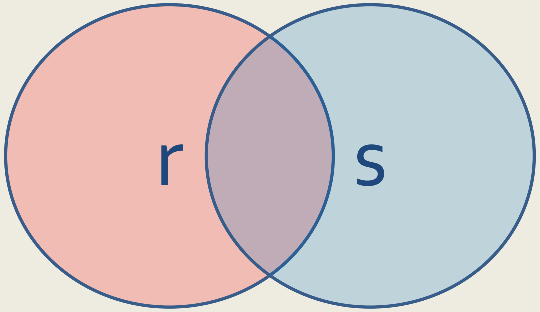
Relation *r*

A	*B*
a	2
b	3

Relation *s*

A	*B*
a	2

$r \cap s$



Set-Intersection Operation

- Find the set of all courses taught in both the Fall 2017 and the Spring 2018 semesters.

$$\Pi_{course\_id}\ (\sigma_{\ semester=\ \text{"Fall"}\ \ \land\ year=2017}\ (section))\ \cap$$
$$\Pi_{course\_id}\ (\sigma_{\ semester=\ \text{"Spring"}\ \ \land\ year=2018}\ (section))$$

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

Figure 2.6 The *section* relation.

course_id
CS-101

Cartesian product

- Joining two relations – ***Cartesian product***

A	*B*
a	1
b	2

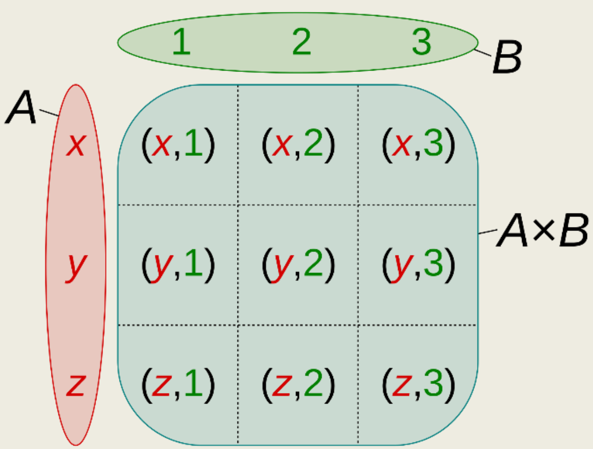
Relation *r*

C	*D*	*E*
a	10	*x*
b	10	*x*
b	20	*y*
c	10	*y*

Relation *s*

A	*B*	*C*	*D*	*E*
a	1	*a*	10	*x*
a	1	*b*	10	*x*
a	1	*b*	20	*y*
a	1	*c*	10	*y*
b	2	*a*	10	*x*
b	2	*b*	10	*x*
b	2	*b*	20	*y*
b	2	*c*	10	*y*

$r \times s$



Cartesian product

- Cartesian product – naming issue

A	*B*
a	1
b	2

Relation *r*

A	*D*	*E*
a	10	*x*
b	10	*x*
b	20	*y*
c	10	*y*

Relation *s*

r.A	*B*	*s.A*	*D*	*E*
a	1	*a*	10	*x*
a	1	*b*	10	*x*
a	1	*b*	20	*y*
a	1	*c*	10	*y*
b	2	*a*	10	*x*
b	2	*b*	10	*x*
b	2	*b*	20	*y*
b	2	*c*	10	*y*

$r \times s$

Renaming a table

- ***Renaming*** a table
  - Allows us to refer to a relation by more than one name
  - Syntax: $\rho_x(E)$ – returns the expression *E* under the name *X*

A	*B*
a	1
b	2

Relation *r*

r.A	*r.B*	*s.A*	*s.B*
a	1	*a*	1
a	1	*b*	2
b	2	*a*	1
b	2	*b*	2

$r \times \rho_s(r)$

$$\rho_{(r.A, r.B, s.A, s.B)}(r \times s)$$

Composition Operation

- ***Composition*** of operations
  - Can build expressions using multiple operations
  - Note: the result of an operation is a table

r.A	B	s.A	D	E
a	1	a	10	x
a	1	b	10	x
a	1	b	20	y
a	1	c	10	y
b	2	a	10	x
b	2	b	10	x
b	2	b	20	y
b	2	c	10	y

$r \times s$

r.A	B	s.A	D	E
a	1	a	10	x
b	2	b	10	x
b	2	b	20	y

$\sigma_{r.A=s.A}(r \times s)$

Composition Operation

- Find the names of all instructors in the Physics department.

$$\Pi_{name}(\sigma_{dept\_name = \text{"Physics"}}(instructor))$$

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

Name
Einstein
Gold

Natural join

- Joining two relations – *Natural join*
  - Let r and s be relations on schemas R and S respectively
  - 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
    - t has the same value as t_r on R; t has the same value as t_s on S
  - Cf. theta join

Natural join

- Natural join example

A	*B*
a	1
b	2

Relation *r*

A	*D*	*E*
a	10	*x*
b	10	*x*
b	20	*y*
c	10	*y*

Relation *s*

A	*B*	*D*	*E*
a	1	10	*x*
b	2	10	*x*
b	2	20	*y*

$r \bowtie s$

$$r \bowtie s = \Pi_{A,B,D,E}\left(\sigma_{r.A=s.A}(r \times s)\right)$$

Natural join

- To get only those tuples of "instructor X teaches " that pertain to instructors and the courses that they taught

  $\sigma_{\text{instructor.id = teaches.id}}$ (instructor x teaches))

- Can equivalently be written as with natural join

  $instructor \bowtie_{Instructor.id\, =\, teaches.id} teaches.$

- The result of this expression, shown in the next slide *instructor*

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

Teaches

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

The *instructor* x *teaches* table

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	12121	FIN-201	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	15151	MU-199	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
12121	Wu	Finance	90000	10101	CS-101	1	Fall	2017
12121	Wu	Finance	90000	10101	CS-315	1	Spring	2018
12121	Wu	Finance	90000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
12121	Wu	Finance	90000	15151	MU-199	1	Spring	2018
12121	Wu	Finance	90000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
15151	Mozart	Music	40000	10101	CS-101	1	Fall	2017
15151	Mozart	Music	40000	10101	CS-315	1	Spring	2018
15151	Mozart	Music	40000	10101	CS-347	1	Fall	2017
15151	Mozart	Music	40000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
15151	Mozart	Music	40000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
22222	Einstein	Physics	95000	10101	CS-101	1	Fall	2017
22222	Einstein	Physics	95000	10101	CS-315	1	Spring	2018
22222	Einstein	Physics	95000	10101	CS-347	1	Fall	2017
22222	Einstein	Physics	95000	12121	FIN-201	1	Spring	2018
22222	Einstein	Physics	95000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

$\sigma_{instructor.id\ =\ teaches.id}$ *(instructor x teaches))* table

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
32343	El Said	History	60000	32343	HIS-351	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-101	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-319	1	Spring	2018
76766	Crick	Biology	72000	76766	BIO-101	1	Summer	2017
76766	Crick	Biology	72000	76766	BIO-301	1	Summer	2018
83821	Brandt	Comp. Sci.	92000	83821	CS-190	1	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-190	2	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-319	2	Spring	2018
98345	Kim	Elec. Eng.	80000	98345	EE-181	1	Spring	2017

÷ Division Operation Example

- Retrieve the studnos of students who are enrolled on all the courses that Capon lectures on
- Small_ENROL ÷ Capon_TEACH

Small_ENROL

studno	courseno
s1	cs250
s1	cs260
s1	cs280
s2	cs250
s2	cs270
s3	cs270
s4	cs280
s4	cs250
s6	cs250

÷

Capon_TEACH

courseno
cs250
cs280

result
s1
s4

- Notes on relational languages
  - Each query input is a table (or set of tables)
  - Each query output is a table
  - All data in the output table appears in one of the input tables

Equivalent Queries

- There is more than one way to write a query in relational algebra.
- Example: Find instructors in the Physics department with salary greater than 90,000

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

Equivalent Queries

- There is more than one way to write a query in relational algebra.
- Example: Find information about courses taught by instructors in the Physics department
- Query 1

$\sigma_{dept\_name= \text{"Physics"}}(instructor) \bowtie_{instructor.ID = teaches.ID} teaches$

- Query 2

$\sigma_{dept\_name= \text{"Physics"}}(instructor \bowtie_{instructor.ID = teaches.ID} teaches)$

1 $\sigma_{\text{dept\_name= "Physics"}}$ (instructor) $\bowtie_{\text{instructor.ID = teaches.ID}}$ teaches

2 $\sigma_{\text{dept\_name= "Physics"}}$ (instructor $\bowtie_{\text{instructor.ID = teaches.ID}}$ teaches)

ID	*name*	*dept_name*	*salary*
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

Figure 2.1 The *instructor* relation.

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

Figure 2.7 The *teaches* relation.

ID	Name	Dept_name	Salary
22222	Einstein	Physics	95000
33456	Gold	Physics	87000

ID	Name	Dept_name	Salary	Course_id	Sec_id	semester	Year
22222	Einstein	Physics	95000	PHY-101	1	Fall	2009

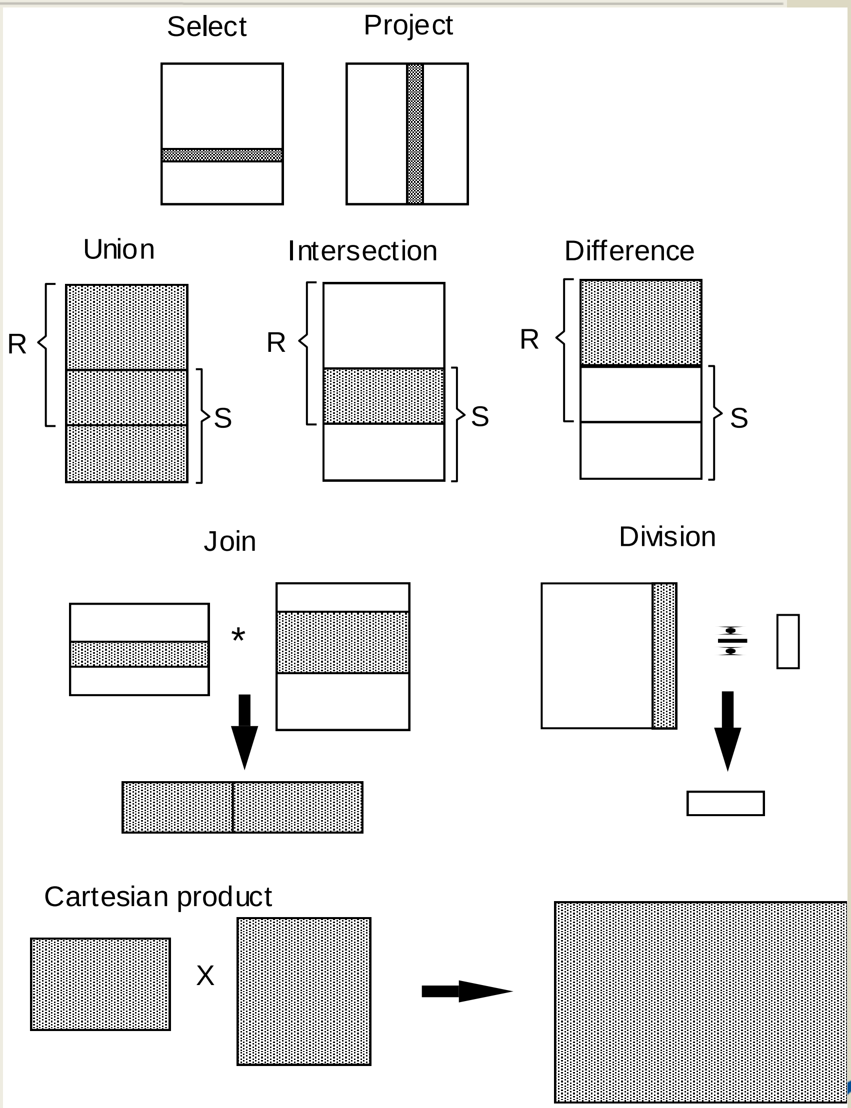
1 $\sigma_{dept\_name= \text{“Physics”}} (instructor) \bowtie_{instructor.ID = teaches.ID} teaches$

2 $\sigma_{dept\_name= \text{“Physics”}} (instructor \bowtie_{instructor.ID = teaches.ID} teaches)$

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
32343	El Said	History	60000	32343	HIS-351	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-101	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-319	1	Spring	2018
76766	Crick	Biology	72000	76766	BIO-101	1	Summer	2017
76766	Crick	Biology	72000	76766	BIO-301	1	Summer	2018
83821	Brandt	Comp. Sci.	92000	83821	CS-190	1	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-190	2	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-319	2	Spring	2018
98345	Kim	Elec. Eng.	80000	98345	EE-181	1	Spring	2017

Summary

Symbol (Name)	Example of Use
σ (Selection)	$\sigma_{\text{salary}>=85000}(instructor)$
	Return rows of the input relation that satisfy the predicate.
Π (Projection)	$\Pi_{ID,salary}(instructor)$
	Output specified attributes from all rows of the input relation. Remove duplicate tuples from the output.
$\bowtie$ (Natural join)	$instructor \bowtie department$
	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)	$instructor \times department$
	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)	$\Pi_{name}(instructor) \cup \Pi_{name}(student)$
	Output the union of tuples from the two input relations.



Assignment #2 (150pt)

- Do Exercises (p. 62):
  - 2.10, 2.11, 2.12, 2.13, 2.15, 2.18
- Due: Before the next lecture
  - 09/14 (Wed.)
- Method: upload your report in Cyber Campus
  - Questions are uploaded in Assignment 2 folder