The Relational Model

The *instructor* Table

attributes
(or columns)

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
(or rows)

- Is this table a relation?
- What are the differences?

Relation Schema

- $R = (A_1{:}D_1,\ A_2{:}D_2,\ \ldots,\ A_n{:}D_n)$ is called (relation) schema
  - $A_1, \ldots, A_n$ are called attributes
  - $D_1, \ldots, D_n$ are called domains

- Example: *instructor* = (*ID : char(5),*
  name : varchar(20),
  dept_name : varchar(20),
  salary : numeric(8))

- Usually domains are irrelevant for us, so we leave them out:
  - *instructor* = (*ID, name, dept_name, salary*)

Relation Schema

- A relation r is an instance of a schema R if
  - Its tupels have the same length as the schema
  - For each tupel in r, its i-th element is in the domain of the i-th attribute of R
- A relation with schema is a relation together with a schema such that the relation is an instance of the schema.
- A relation r with schema R is written r(R)
- We will often just say "relation" instead of "relation with schema".

The *course* Relation

course_id	*title*	*dept_name*	*credits*
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

The *section* Relation

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

The *teaches* 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

Superkeys

- Let *r* be a relation of schema *R* and let $K \subseteq R$.
- *K* is a **superkey of the relation** *r* if values for *K* are sufficient to identify a unique tuple of *r.*
- *K* is a **superkey of the schema** *R* if it is a superkey for each possible relation of the schema *R.*
- Example:
  1. Find all superkeys of the following relation
  2. Find all superkeys of the schema *instructor.*

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

Candidate Keys

- A minimal superkey (of a relation or schema) is a superkey such that, when one attribute is removed, it is no longer a superkey.

- A candidate key (of a relation or schema) is a minimal superkey.
  - Example: {*ID*} is a candidate key for instructor*, but* {*ID, name*} is not

- Task: Find all candidate keys for the schemas
  - *instructor*
  - *course*
  - *section*
  - *teaches*

Primary Keys

- Database designer chooses a primary means of identifying a tuple: the **primary key**.

- The primary key is part of the schema – every schema needs a primary key.
  Notation:
  instructor (<u>ID</u>, name, dept_name, salary)
  teaches (<u>ID</u>, <u>course_id</u>, <u>sec_id</u>, <u>semester</u>, <u>year</u>)

- The **primary key constraint** means two tuples can not have the same values for the primary key attributes
  - The database will reject adding a new instructor with an existing ID

How to Choose the Primary Key

- Choose a set of attributes which is
  1. a candidate key, and has
  2. stable values
- If there is no candidate key with stable values, and only then, add an additional attribute with a unique value per tuple, called surrogate key
- Choosing the primary key is an important and difficult decision: it is used in other parts of the database, often exposed to partners, hard to change
  - e.g. imagine choosing {semester, year, timeslot_id, building, room} as the primary key of section and later encountering a situation that requires that two sections happen in the same room

Foreign Keys

- A **foreign key** is a set of attributes in one relation which is the primary key of another relation
  - Example: the dept_name attribute in the instructor relation is the primary key of the department relation

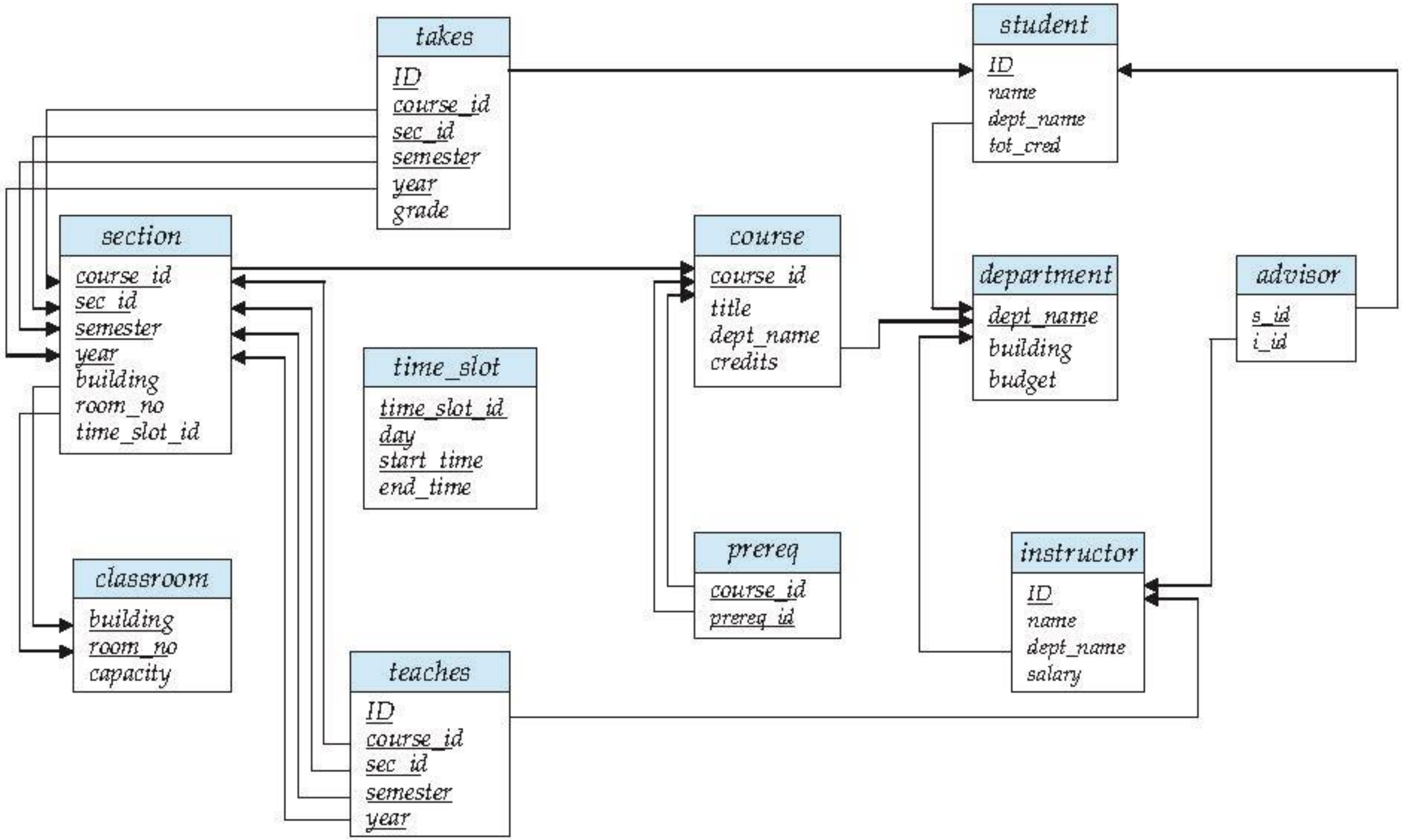
- Notation:

  instructor (<u>ID</u>, name, dept_name, salary)
  dept_name → department

  - instructor is called the referencing relation
  - department is called the referenced relation

- The **foreign key constraint** says that attribute values of the foreign key in the referencing relation have to occur in the referenced relation
  - The database will reject adding an instructor with a department which is not in the department relation

Schema Diagram for University Database



Relational Algebra I

Relational Algebra

- Six basic operators
  - select: σ
  - project: Π
  - union: $\cup$
  - set difference: –
  - Cartesian product: x
  - rename: ρ
- The operators take one or two relations as inputs and produce a new relation as a result.

Select Operation

- Notation: $\sigma_p(r)$
- p is called the **selection predicate**
- Defined as:

  $\sigma_p(\boldsymbol{r}) = \{t \mid t \in r \textbf{ and } p(t)\}$

  where p is a formula consisting of **terms** connected by : $\wedge$ (**and**), $\vee$ (**or**), $\neg$ (**not**)

  Each **term** is:

  <attribute> *op* <attribute> or

  <attribute> *op* <constant>

  and *op* is one of: $=, \neq, >, \geq, <, \leq$

r:

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

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

A	B	C	D
α	α	1	7
β	β	23	10

Select Operation

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

$\sigma_{\,dept\_name=\text{“Physics”}}(instructor)$

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

Project Operation

- Notation: $\Pi_{A_1, A_2, \ldots, A_k}(r)$

  where A_1, A_2 are attribute names and r is a relation name.

- The result is the relation obtained by erasing the columns that are not listed

- Duplicate rows are removed from the result, since relations are sets

r :

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

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

Project Operation

$\Pi_{ID,\ name,\ salary}\ (instructor)$

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

- Defined as:

  $r \cup s = \{t \mid t \in r \text{ or } t \in s\}$

- For $r \cup s$ to exist, r,s must be **compatible**:
  - *r, s* must have the *same* **arity** (the same size of tupels)
  - the attributes must match: the i-th attribute of r must have the same name and domain as i-th attribute of s.

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

A	B
α	1
α	2
β	1
β	3

$r \cup s$

Union Operation

Find all courses taught in the Fall 2009 semester, or in the Spring 2010 semester, or in both (note composition of operations):

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

$$\Pi_{course\_id} (\sigma_{semester=\text{“Fall”} \ \wedge \ year=2009} (section)) \ \cup$$
$$\Pi_{course\_id} (\sigma_{semester=\text{“Spring”} \ \wedge \ year=2010} (section))$$

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

Set Difference Operation

- Defined as:

  $$r - s = \{t \mid t \in r \textbf{ and } t \notin s\}$$

- r and s must be **compatible**

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

A	B
α	1
β	1

$r - s$

Set Difference Operation

Example: Find all courses taught in the Fall 2009 semester, but not in the Spring 2010 semester

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

$$\Pi_{course\_id} (\sigma_{semester=\text{“Fall”} \land year=2009} (section)) -$$
$$\Pi_{course\_id} (\sigma_{semester=\text{“Spring”} \land year=2010} (section))$$

course_id
CS-347
PHY-101

Rename Operation

- Allows us to rename attributes of relations

- If E has arity n, then

$$\rho_{A_1, A_2, \ldots, A_n}(E)$$

returns the result of expression E with the attributes renamed to A_1, A_2, …., A_n.

r :

A	B
1	2
3	4

$\rho_{C,D}(r)$:

C	D
1	2
3	4

Cartesian-Product Operation

- Defined as:

  $$r \times s = \{t\, q \mid t \in r \text{ and } q \in s\}$$

- Generally assumes that attributes of r(R) and s(S) are disjoint:
  $R \cap S = \varnothing$.

- If attributes of are not disjoint, prefix the common attributes with the relation name:
  - *instructor(ID,..) x student(ID,…)*

    results in schema:

    (instructor.ID, …, student.ID, …)

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

r x *s*

Relational Algebra Expressions

- A basic expression in the relational algebra is either:
  - a relation in the database, like *instructor*, or
  - a constant relation, like {(1,2), (3,4)}.

- Let E_1 and E_2 be relational-algebra expressions; the following are all relational-algebra expressions:
  - $\sigma_p (E_1)$, p is a predicate on attributes in E_1
  - $\prod_S(E_1)$, S is a list of attributes in E_1
  - $E_1 \cup E_2$
  - $E_1 - E_2$
  - $E_1 \times E_2$
  - $\rho_S (E_1)$, S is a list of attributes

Example: Find the Courses Given by Instructors from the Physics Dept.

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	*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

Two Equivalent Queries

- Find the Courses by Instructors from the Physics Dept.

  - Query 1

    $$\Pi_{course\_id}\left(\sigma_{dept\_name = \text{“Physics”}}\left(\sigma_{instructor.ID = teaches.ID}(instructor \times teaches)\right)\right)$$

  - Query 2

    $$\Pi_{course\_id}\left(\sigma_{instructor.ID = teaches.ID}\left(\sigma_{dept\_name = \text{“Physics”}}(instructor) \times teaches\right)\right)$$

Example: Find the Largest Salary

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

Example: Largest Salary

- Find the largest salary in the university
  - Step 1: find instructor salaries that are less than some other instructor salary (i.e. not maximum)
    - $\Pi_{salary}(\sigma_{salary < s}(instructor \times \rho_{i,n,d,s}(instructor)))$
  - Step 2: Find the largest salary
    - $\Pi_{salary}(instructor) -$
      $\Pi_{salary}(\sigma_{salary < s}(instructor \times \rho_{i,n,d,s}(instructor)))$
- In reality, we would use aggregation (which we see later) for this query – this is just to exercise renaming and the cartesian product of a relation with itself

Example: Largest Salary

$$\Pi_{salary}\,(\sigma_{\,salary\,<\,s}\,(instructor\ x\ \rho_{i,n,d,s}(instructor)))$$

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

salary
65000
90000
40000
60000
87000
75000
62000
72000
80000
92000

salary
95000