Chapter 2: Intro to Relational Model

Chapter 2: Introduction to the Relational Model

- Structure of Relational Databases
  - relation
  - attributes
  - relation schema / relation instance
- Database Schema
- Keys
  - superkey / candidate key / primary key / foreign key
- Schema Diagrams
- Relational Query Languages
- Relational Operations
  - selection / projection / natural join / Cartesian product/ union
  - composition of operators

Example of a Relation

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)

Attributes

- The set of allowed values for each attribute is called the **domain** of the attribute
- Attribute values are (normally) required to be **atomic**; that is, indivisible
- The special value ***null*** is a member of every domain
- The null value causes complications in the definition of many operations

Relation Schema and Instance

- $A_1, A_2, \ldots, A_n$ are *attributes*

$R = (A_1, A_2, \ldots, A_n)$ is a **relation schema**

Example:

instructor = (*ID*, *name*, *dept_name*, *salary*)

- Formally, given sets $D_1, D_2, \ldots. D_n$ a **relation** ***r*** is a subset of

  $D_1 \times D_2 \times \ldots \times D_n$

  Thus, a relation is a set of n-tuples $(a_1, a_2, \ldots, a_n)$ where each $a_i \in D_i$

- The current values (**relation instance**) of a relation are specified by a table
- An element ***t*** of ***r*** is a *tuple*, represented by a *row* in a table

Relations are Unordered

- Order of tuples is irrelevant (tuples may be stored in an arbitrary order)
- Example: *instructor* relation with unordered tuples

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

- A database consists of multiple relations
- Information about an enterprise is broken up into parts

  instructor
  student
  advisor

- Bad design:

  univ (*instructor -ID, name, dept_name, salary, student_Id*, ..)

  results in
  - repetition of information (e.g., two students have the same instructor)
  - the need for null values (e.g., represent an student with no advisor)
- Normalization theory (Chapter 7) deals with how to design "good" relation schemas

Keys

- What is a key?

Keys

Dictionary

key [1] |kē|

noun (pl. **keys**)

1 a small piece of shaped metal with incisions cut to fit the wards of a particular lock, and that is inserted into a lock and turned to open or close it.

- a similar implement for operating a switch in the form of a lock, esp. one operating the ignition of a motor vehicle.
- short for KEY CARD .
- an instrument for grasping and turning a screw, peg, or nut, esp. one for winding a clock or turning a valve.
- a pin, bolt, or wedge inserted between other pieces, or fitting into a hole or space designed for it, so as to lock parts together.

2 one of several buttons on a panel for operating a typewriter, word processor, or computer terminal.

- a lever depressed by the finger in playing an instrument such as the organ, piano, flute, or concertina.
- a lever operating a mechanical device for making or breaking an electric circuit, for example, in telegraphy.

3 a thing that provides a means of gaining access to or understanding something : *the **key** to Jack's behavior may lie submerged in his unhappy past.*

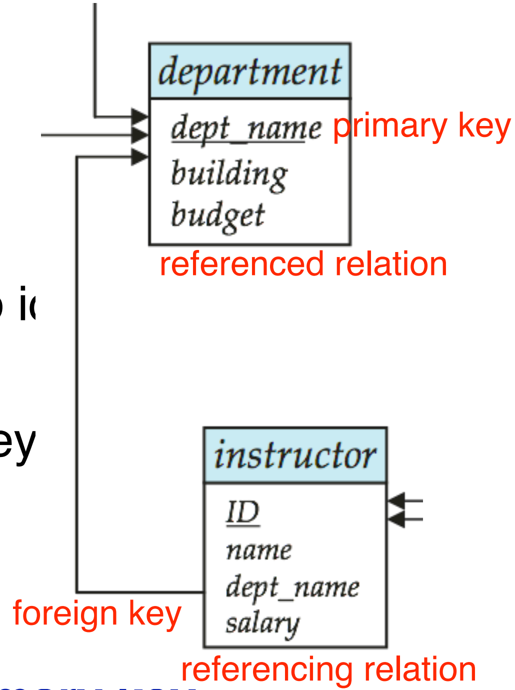
- an explanatory list of symbols used in a map, table, etc.
- a set of answers to exercises or problems.

Keys

- Let $K \subseteq R$
- *K* is a **superkey** of *R* if values for *K* are sufficient to identify a *unique* tuple of each possible relation *r(R)*
  - Example: {*ID*} and {ID, name} are both superkeys of *instructor.*
- Superkey *K* is a **candidate key** if *K* is *minimal*

  Example: {*ID*} is a candidate key for *Instructor*
- One of the candidate keys is selected to be the **primary key**.
  - which one?
- A relation r_1 may have attributes that correspond to the primary key of another relation r_2. The attribute is called a **foreign key**.
  - r_1: **referencing** relation
  - r_2: **referenced** relation
  - **Foreign key** constraint: foreign key values in the referencing relation must appear in the referenced relation.

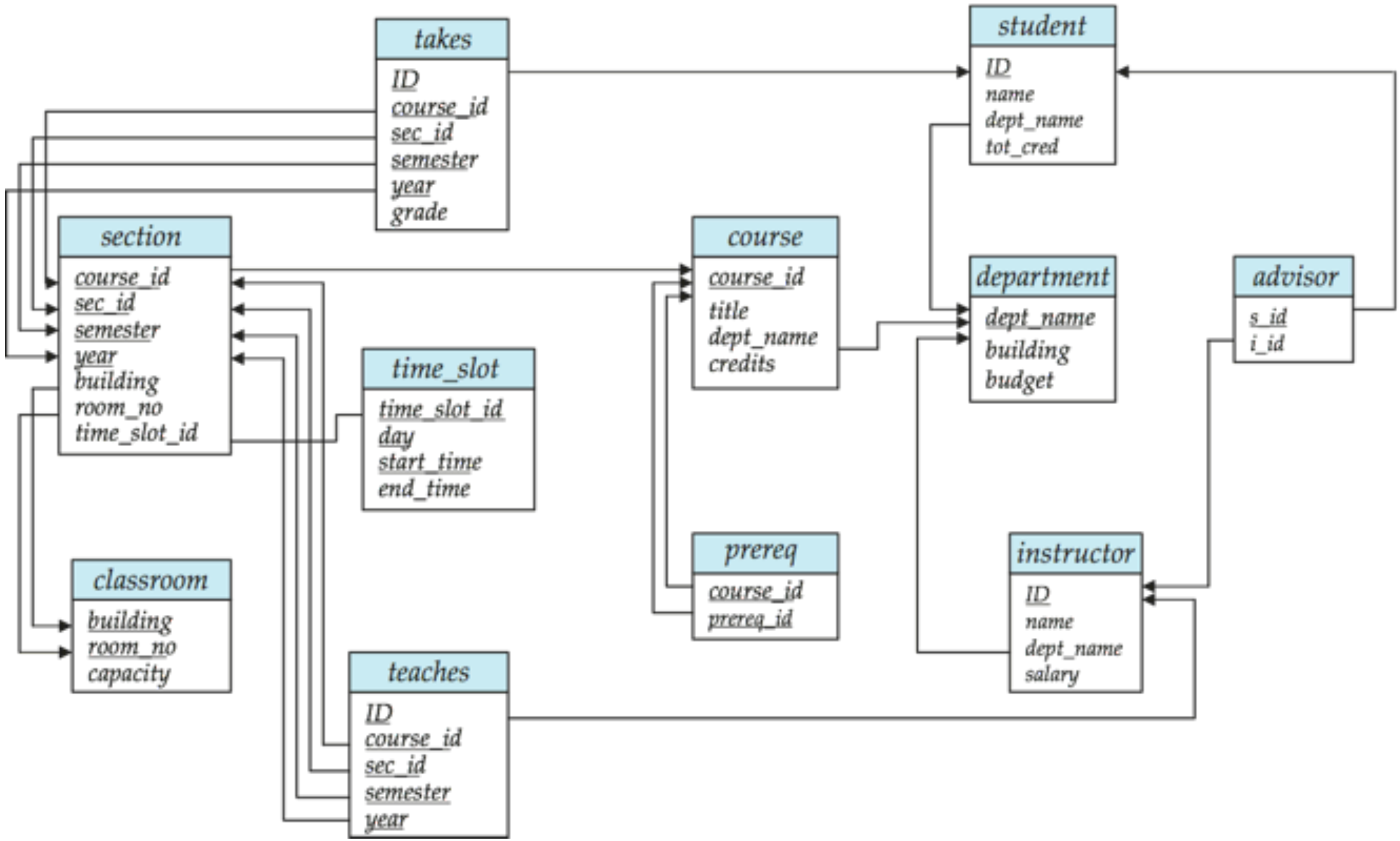
Keys



- Let K ⊆ R
- *K* is a **superkey** of *R* if values for *K* are sufficient to i
  tuple of each possible relation *r(R)*
  - Example: {*ID*} and {ID, name} are both superkey
- Superkey *K* is a **candidate key** if *K* is *minimal*
  Example: {*ID*} is a candidate key for *Instructor*
- One of the candidate keys is selected to be the **primary key**.
  - which one?
- A relation r_1 may have attributes that correspond to the primary key of another relation r_2. The attribute is called a **foreign key**.
  - r_1: **referencing** relation
  - r_2: **referenced** relation
  - **Foreign key** constraint: foreign key values in the referencing relation must appear in the referenced relation.

Thursday, January 31, 2013

Schema Diagram for University Database



Relational Query Languages

- Procedural vs. non-procedural, or declarative
- “Pure” languages:
  - Relational algebra (Section 6.1)
  - Tuple relational calculus (Section 6.2)
  - Domain relational calculus (Section 6.3)
- Relational operators
  - selection
  - projection
  - natural join
  - Cartesian product
  - union

Selection of tuples

- Relation r

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

- Select tuples with A=B and D > 5
- $\sigma_{A=B \text{ and } D > 5}(r)$

A	B	C	D
α	α	1	7
β	β	23	10

Selection of Columns (Attributes)

- Relation *r*:

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

- Select A and C
  - Projection
  - $\Pi_{A, C}(r)$

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

Joining two relations – Cartesian Product

- Relations *r*, *s*:

A	B
α	1
β	2

r

C	D	E
α	10	a
β	10	a
β	20	b
γ	10	b

s

- *r* x *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

Union of two relations

- Relations *r, s:*

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- r ∪ s:

A	B
α	1
α	2
β	1
β	3

Set difference of two relations

- Relations *r*, *s*:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- *r* – *s*:

A	B
α	1
β	1

Set Intersection of two relations

- Relation *r*, *s*:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

- $r \cap s$

A	B
α	2

Natural Join of Two Relations

- Relation *r* on relation schema *R*, relation *s* on relation schema *S*:

instructor_id	department_id
A	1
B	1
C	2

department_id	college
1	X
2	Y

- Natural Join
  - r ⋈ s

instructor_id	department_id	college
A	1	X
B	1	X
C	2	Y

- find each pair of **matching tuples**

  (i.e., tuples t_r from *r* and t_s from *s* such that

  t_r and t_s have the same value on each **common attribute** in $R \cap S$
- output the concatenation of matching tuples

Relational Operations

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.
⋈ (Natural Join)	*instructor* ⋈ *department*
	Output pairs of rows from the two input relations that have the same value on all attributes that have the same name.
× (Cartesian Product)	*instructor* × *department*
	Output all pairs of rows from the two input relations (regardless of whether or not they have the same values on common attributes)
∪ (Union)	$\Pi_{name}(instructor) \cup \Pi_{name}(student)$
	Output the union of tuples from the two input relations.

Composition of Operations

- Can build expressions using multiple operations
- Example: $\sigma_{A=C}(r \; x \; s)$
- *r x 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

- $\sigma_{A=C}(r \; x \; s)$

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

Summary

- Structure of Relational Databases
  - relation
  - attributes
  - relation schema / relation instance
- Database Schema
- Keys
  - superkey / candidate key / primary key / foreign key
- Schema Diagrams
- Relational Query Languages
- Relational Operations
  - selection / projection / natural join / Cartesian product/ union
  - composition of operators