Chapter 2: Intro to Relational Model

Database System Concepts, 6th Ed.

Example of a Relation

Table *instructor*

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)

Attribute Types

- The set of allowed values for each attribute is called the **domain** of the attribute (denoted as *D*)
- 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*
  - with the corresponding domains $D_1, D_2, \ldots. D_n$

- $R = (A_1, A_2, \ldots, A_n)$ is a *relation schema*
  Example: *INSTRUCTOR* = (*ID, name, dept_name, salary*)

- Formally, given sets $D_1, D_2, \ldots. D_n$ corresponding to attributes in schema *R* a **relation *r*** is a subset of

$$r \subseteq D_1 \times D_2 \times \ldots \times D_n$$

  Thus, a relation is a set of *n*-tuples $(a_1, a_2, \ldots, a_n)$ where each $a_i \in D_i$
  - An element ***t*** of ***r*** is a *tuple*, represented by a *row* in a table
  - Notation: instructor(INSTRUCTOR) or
    instructor(*ID, name, dept_name, salary*)
- The current values (**relation instance**) of a relation are specified by 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

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

  instructor - ID, name , salary, department
  student - ID, name, program, credits, advisor
  advisor - ID, name, topic

- Bad design:

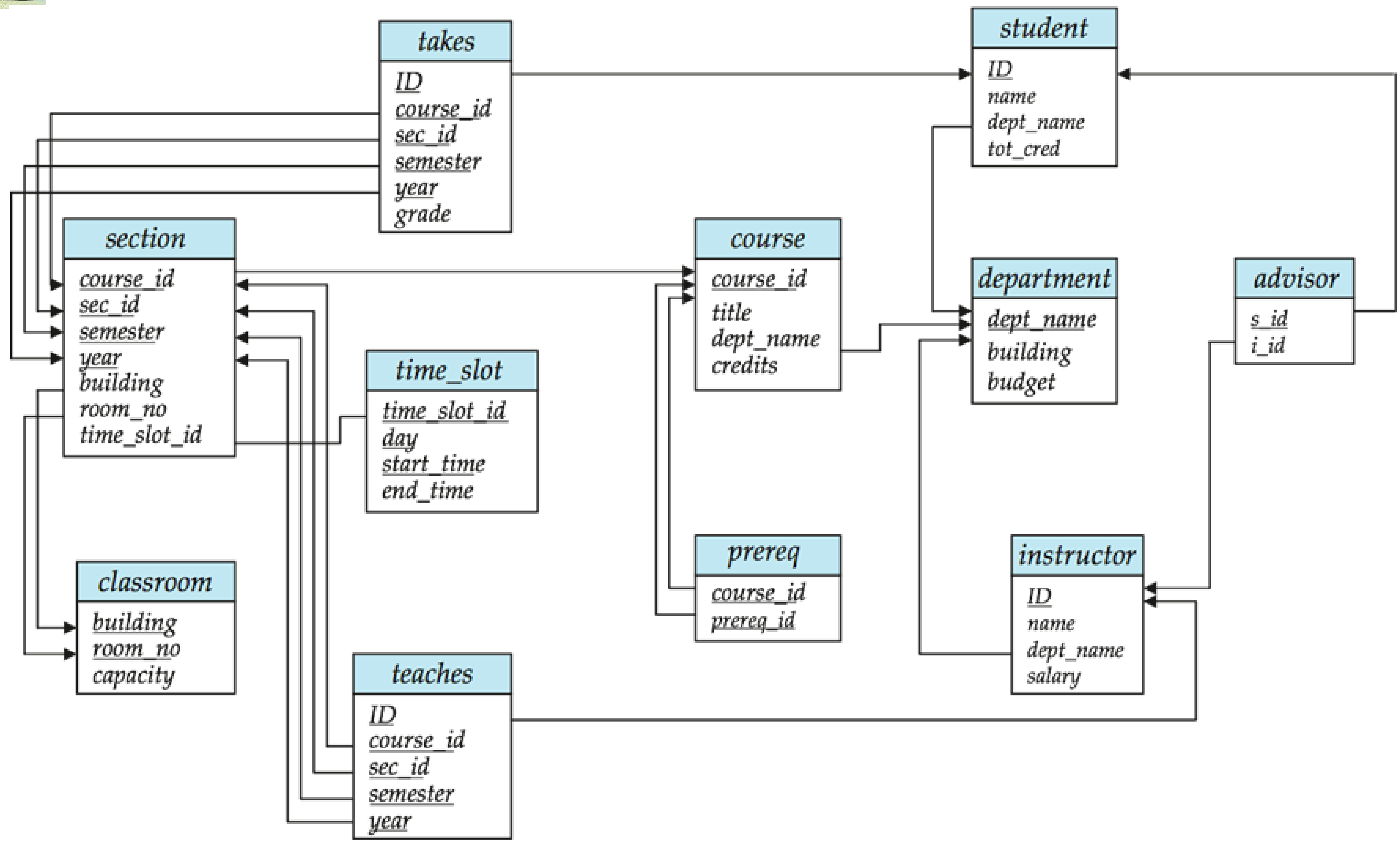
  university (*instructor_ID, name, dept_name, salary, student_ID*, …)

  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” relational schemas

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: instructor(*ID, name, dept_name, salary*)
    - Some superkeys: {*ID*} {*ID*, *name*} {*name*, *dept_name*}
- 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?
- **Foreign key** constraint: Value in one relation must appear in another
  - **Referencing** relation
  - **Referenced** relation

Schema Diagram for University Database



Foreign keys are emphasized by arrows.

Relational Query Languages

- Language in which user requests information from the database.
- Procedural vs. non-procedural (or declarative)
- “Pure” languages:
  - Relational algebra – procedural language
  - Tuple relational calculus – declarative language
  - Domain relational calculus – declarative language
- Pure languages form underlying basis of query languages that people use.
  - E.g. SQL

Relational Query Languages

- Why relational algebra?
  - It is simple
    - Few operators only …
  - An expression specifies the procedure to evaluate it
    - We know what to do first, second, …
- SQL
  - can be complicated (many keywords, …)
  - does not give the recipe

End of Chapter 2

Database System Concepts, 6th Ed.