Introduction to Relational Model

Dr. Shrutilipi Bhattacharjee, Assistant Professor, Dept. of IT, NIT Karnataka, India

Example of a Relation

- The ***instructor*** relation

Attributes (or columns)

Tuples (or rows)

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

Relation Schema, Instance and Attributes

- $A_1, A_2, \ldots, A_n$ are *attributes*
- $R = (A_1, A_2, \ldots, A_n)$ is a *relation schema*
- Example: *instructor = (ID, name, dept_name, salary)*
- A relation instance *r* defined over schema *R* is denoted by *r*(*R*)
- The current values a relation are specified by a table
- An element ***t*** of relation ***r*** is called a *tuple* and is represented by a *row* in a table

- The set of allowed values for each attribute is called the **domain** of the attribute (alphanumeric string, alpha string, date, etc.)
- Attribute values are (normally) required to be **atomic**; that is, indivisible
- The special value ***null*** is a member of every domain, which indicates that the value is "unknown"
- The null value causes complications in the definition of many operations

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

- ***Database schema:*** It is the logical structure of the database
- ***Database instance:*** It is a snapshot of the data in the database at a given instant in time
- Example:
  - Schema: *instructor* (*ID, name, dept_name, salary*)
  - Instance:

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

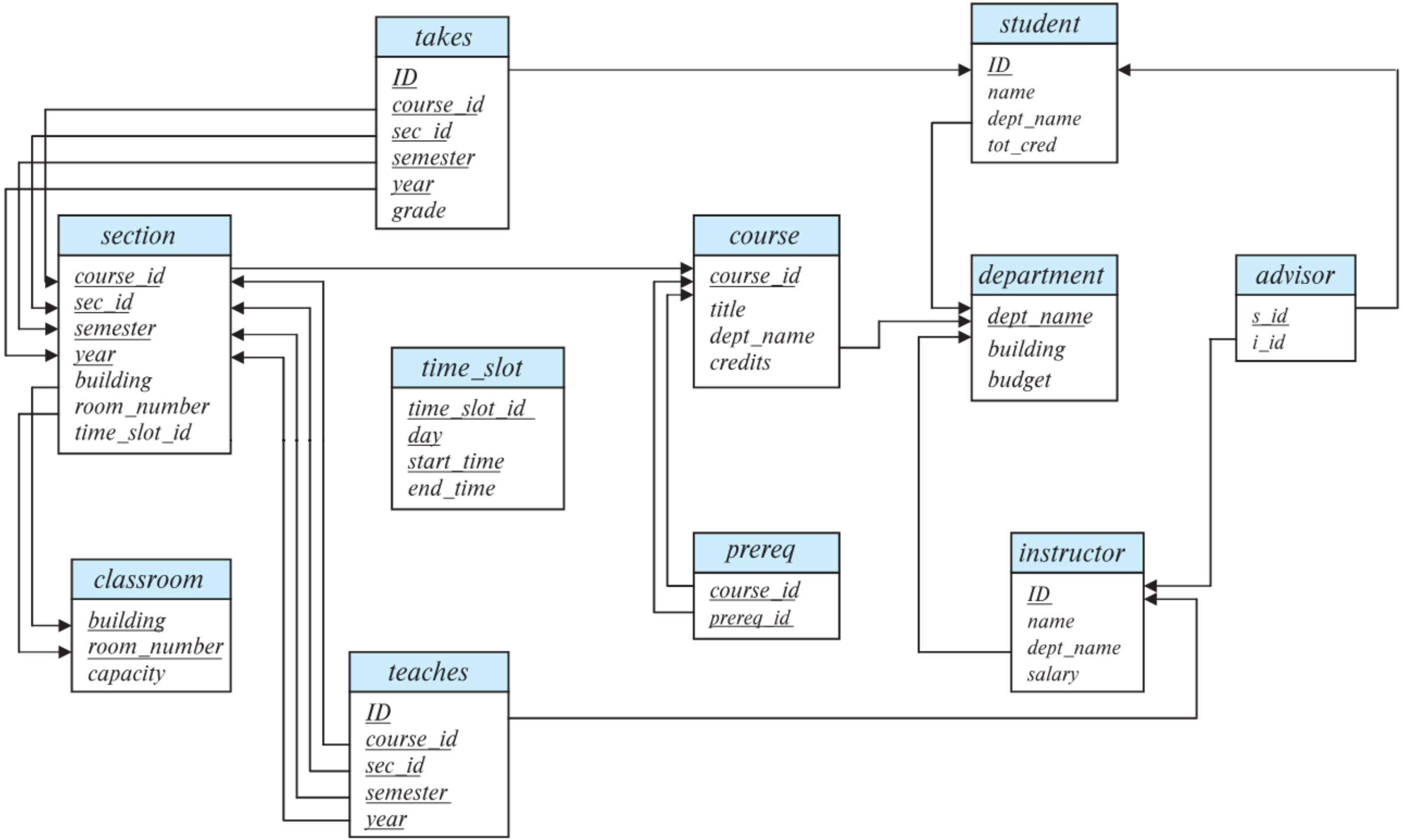
Keys

- 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? No key attribute can be a null-able field
- **Secondary/Alternate key**
- **Simple/Composite key**

Keys

- **Foreign key** constraint: Value in one relation must appear in another
  - **Referencing** relation
  - **Referenced** relation
  - Example: *dept_name* in ***instructor*** is a foreign key from ***instructor*** referencing ***department***
- A **compound key** consists of more than one attribute to uniquely identify an entity occurrence
  - Each attribute which form the key is a simple key in its own right
  - Generally, the components come from different tables
- A **surrogate key** (or **synthetic key**) in a database is a unique identifier for either an *entity* in the modeled world or an *object* in the database
  - The surrogate key is not derived from application data, unlike a natural (or business) key which is derived from application data

Schema Diagram for University Database



Relational Query Languages

- Procedural versus non-procedural or declarative
  - Procedural programing requires that the programmer tell the computer what to do
    - That is ***how*** to get the output for the range of required inputs
    - The programmer must know an appropriate algorithm
  - The declarative programming requires a more descriptive style
    - The programmer must know ***what*** relationship hold among various entities
- **Example:** Square root of N
  - Procedural
    - Guess x_0 (close to root of N)
    - $x_i = (x_i + N / x_i) / 2$
    - Repeat step 2 if $|x_i - x_{i-1}| > \delta$
  - Declarative
    - Root of N must be M such that $M^2 = N$

Relational Query Languages

- “Pure” languages:
  - Relational algebra
  - Tuple relational calculus
  - Domain relational calculus
- The above 3 pure languages are equivalent in computing power
- We will concentrate in this chapter on relational algebra
  - Not Turing-machine equivalent
    - Not all algorithms cab be expressed in RA
  - Consists of 6 basic operations

Next Lecture

Introduction to Relational Model

Thank you for your attention...

Any question?

Contact:
Department of Information Technology, NITK Surathkal, India
6th Floor, Room: 13
Phone: +91-9477678768
E-mail: shrutilipi@nitk.edu.in