Database Management Systems

Module 06: Introduction to Relational Model/1

Partha Pratim Das

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

ppd@cse.iitkgp.ac.in

Week Recap

- The proliferation of DBMS in wide range of applications provide motivation to study the subject
- Know Your Course provided information about prerequisites, outline and text book
- The specific need for a DBMS discussed in contrast to a file system based application using a programming language like Python
- Basic notions of a DBMS are introduced

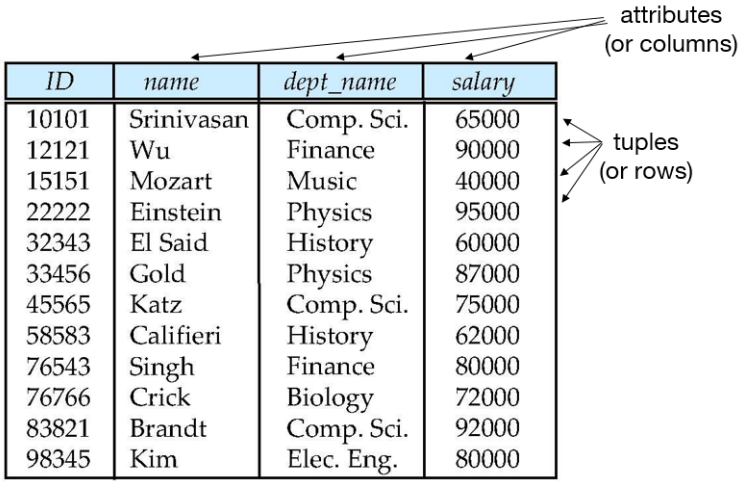
Module Objectives

- To understand attributes and their types
- To understand the mathematical structure of relational model
  - Schema
  - Instance
  - Keys
- To familiarize with different types of relational query languages

Module Outline

- Attribute Types
- Relation Schema and Instance
- Keys
- Relational Query Languages

Example of a Relation



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

Attributes

Attribute Types

- Consider
  $Students = Roll\#, First\ Name, Last\ Name, DoB, Passport\#, Aadhaar\#, Department$
  relation
- The set of allowed values for each attribute is called the **domain** of the attribute
  - **Roll #:** Alphanumeric string
  - **First Name, Last Name:** Alpha String
  - **DoB:** Date
  - **Passport #:** String (Letter followed by 7 digits) – nullable (optional)
  - **Aadhaar #:** 12-digit number
  - **Department:** Alpha String
- Attribute values are (normally) required to be **atomic**; that is, indivisible
- The special value **null** is a member of every domain. Indicates that the value is *unknown*
- The null value may cause complications in the definition of many operations

Attribute Types

- For
  $Students = Roll\#, First\ Name, Last\ Name, DoB, Passport\#, Aadhaar\#, Department$
- And **domain** of the attributes as:
  - **Roll #:** Alphanumeric string
  - **First Name, Last Name:** Alpha String
  - **DoB:** Date
  - **Passport #:** String (Letter followed by 7 digits) – nullable (optional)
  - **Aadhaar #:** 12-digit number
  - **Department:** Alpha String

Roll #	First Name	Last Name	DoB	Passport #	Aadhaar #	Department
15CS10026	Lalit	Dubey	27-Mar-1997	L4032464	1728-6174-9239	Computer
16EE30029	Jatin	Chopra	17-Nov-1996	null	3917-1836-3816	Electrical

Schema and Instance

Relation Schema and Instance

- $A_1, A_2, \cdots, A_n$ are *attributes*
- $R = (A_1, A_2, \cdots, A_n)$ is a *relation schema*
  Example: $instructor = (ID, name, dept\_name, salary)$
- Formally, given sets $D_1, D_2, \cdots, D_n$ a **relation** r is a subset of

$$D_1 \times D_2 \times \cdots \times D_n$$

  Thus, a relation is a set of n-tuples $(a_1, a_2, \cdots, a_n)$ 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
- Example:
  $instructor \equiv (String(5) \times String \times String \times Number+)$, where $ID \in String(5)$, $name \in String$, $dept\_name \in String$, and $salary \in Number+$

Relations are Unordered with Unique Tuples

- **Order of tuples / rows is irrelevant** (tuples may be stored in an arbitrary order)
- **No two tuples / rows may be identical**
- 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

Keys

Keys

- Let $K \subseteq R$, where R is the set of attributes in the relation
- 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 **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

Keys

- *Students* = *Roll#*, *First Name*, *Last Name*, *DoB*, *Passport#*, *Aadhaar#*, *Department*
- **Super Key:** Roll #, {Roll #, DoB}
- **Candidate Keys:** Roll #, {First Name, Last Name}, Aadhaar#
  - Passport # cannot be a key. Why?
  - Null values are allowed for Passport # (a student may not have a passport)
- **Primary Key:** Roll #
  - Can Aadhaar# be a key?
  - It may suffice for unique identification. But Roll# may have additional useful information. For example: 14CS92P01
    - Read 14CS92P01 as 14-CS-92-P-01
    - 14: Admission in 2014
    - CS: Department = CS
    - 92: Category of Student
    - P: Type of admission: *Project*
    - 01: Serial Number

- **Secondary / Alternate Key:** {First Name, Last Name}, Aadhaar #
- **Simple Key:** Consists of a *single attribute*
- **Composite Key:** {First Name, Last Name}
  - Consists of more than one attribute to uniquely identify an entity occurrence
  - One or more of the attributes, which make up the key, are not simple keys in their own right

Roll #	First Name	Last Name	DoB	Passport #	Aadhaar #	Department
15CS10026	Lalit	Dubey	27-Mar-1997	L4032464	1728-6174-9239	Computer
16EE30029	Jatin	Chopra	17-Nov-1996	null	3917-1836-3816	Electrical
15EC10016	Smriti	Mongra	23-Dec-1996	G5432849	2045-9271-0914	Electronics
16CE10038	Dipti	Dutta	02-Feb-1997	null	5719-1948-2918	Civil
15CS30021	Ramdin	Minz	10-Jan-1997	X8811623	4928-4927-5924	Computer

Keys

- **Foreign key constraint:** Value in one relation must appear in another
  - **Referencing** relation
    - Enrolment: Foreign Keys – Roll #, Course #
  - **Referenced** relation
    - Students, Courses
- A **compound key** consists of *more than one attribute* to uniquely identify an entity occurrence
  - Each attribute, which makes up the key, is a simple key in its own right
  - {Roll #, Course #}

Students

Roll #	First Name	Last Name	DoB	Passport #	Aadhaar #	Department

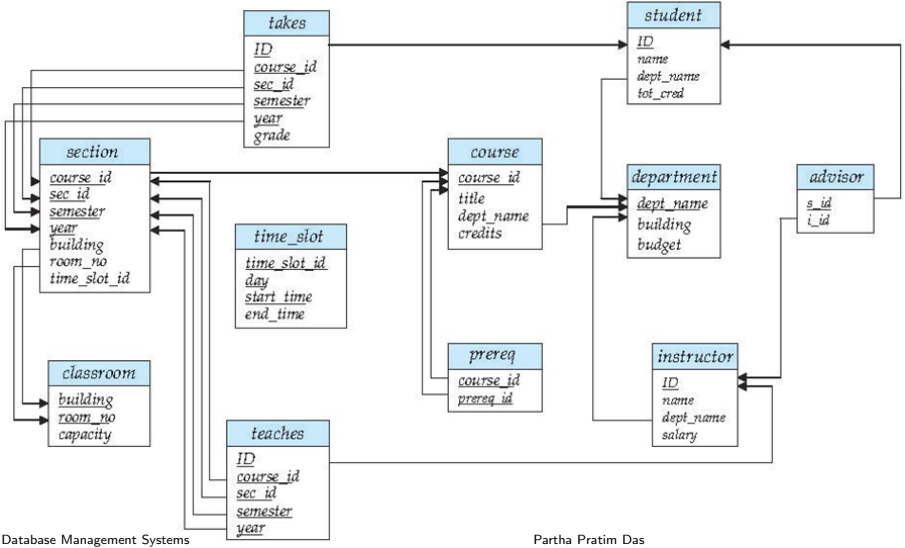
Courses

Course #	Course Name	Credits	L-T-P	Department

Enrolment

Roll #	Course #	Instructor ID

Schema Diagram for University Database



Relational Query Languages

Procedural viz-a-viz Non-procedural or Declarative Paradigms

- **Procedural programming** 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
- **Declarative programming** requires a more descriptive style
  - The programmer must know *what* relationships hold between various entities

Procedural vs. Non-procedural or Declarative Paradigms

- **Example: Square root of n**
  - Procedural
    a) Guess x_0 (close to root of n)
    b) $i \leftarrow 0$
    c) $x_{i+1} \leftarrow (x_i + n/x_i)/2$
    d) Repeat Step 2 if $|x_{i+1} - x_i| > delta$
  - Declarative
    - ▷ Root of n is m such that $m^2 = n$

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

Module Summary

- Introduced the notion of attributes and their types
- Taken an overview of the mathematical structure of relational model – schema and instance
- Introduced the notion of keys – primary as well as foreign

Slides used in this presentation are borrowed from http://db-book.com/ with kind permission of the authors.
Edited and new slides are marked with "PPD".