Relational Data Model

A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system.

S. Ranbir Singh

CS344@iitg

Adapted from

Chapter 3: Database Management System, 3rd Ed. Ramakrishnan & Gehrke

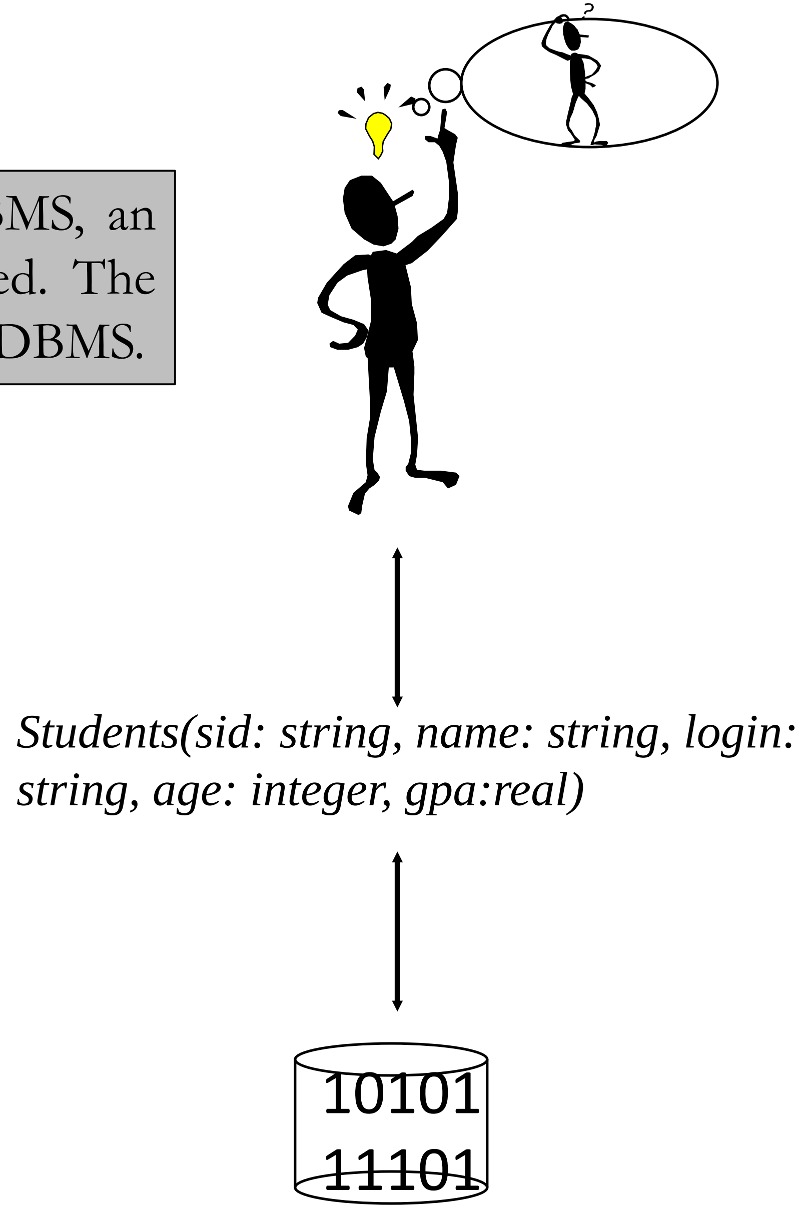
Ref:

Chapter 2: Database System Concept, Silberschatz, Korth, Sudarshan

Data Models

Before the data available in an enterprise can be put in a DBMS, an overall abstract view of the enterprise data must be developed. The view can then be translated into a form that is acceptable by the DBMS.

- DBMS models real world
- *Data Model* is link between user’s view of the world and bits stored in computer
- Many models exist
  - Hierarchical
  - Network
  - **Relational**
- We will concentrate on the Relational Model



Students(sid: string, name: string, login: string, age: integer, gpa:real)

Relational Database: Definitions

- *Relational database:* a set of *relations*.
- *Relation:* made up of 2 parts:
  - *Instance* : a *table,* with rows and columns.
  - *Schema* : specifies name of relation, plus name and type of each column.
    - E.g.

    Students(*sid*: string, *name*: string, *login*: string, *age*: integer, *gpa*: real)

Students

String	String	String	integer	real
sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Can think of a relation as a *set* of rows or *tuples*.
  - i.e., all rows are distinct
  - Some relational databases support duplicate records

In RDBMS, a relation is defined by a table with columns and rows.

Relational Database: Definitions

Attribute: An attribute is a named column of a relation.

Ex: sid, name, login, age, gpa

Domain: A domain is the set of allowable values for one or more attributes.

Ex: domain(sid) – String

domain(age) – integer

Cardinality of a relation: number of rows/tuples it contains

Tuple: A tuple is a row of a relation.

Degree/Arity of a relation: number of attributes it contains

Students

String	String	String	integer	real
sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Cardinality = 3
Degree =5
All rows distinct

Relational Database: A collection of normalized relations with distinct relation names.

Relational Database: Mathematical Definitions

Let D_1, D_2, …. D_n be n number of sets, then Cartesian product of these sets is

$$D = D_1 \text{ x } D_2 \text{ x} \dots \text{x } D_n = \{(a_1, a_2, \dots, a_n) \mid a_1 \in D_1, a_2 \in D_2 \dots a_n \in D_n\}$$

Any subset of D is a relation

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

Example: If

- *customer_name* = {Jones, Smith, Curry, Lindsay, …} /* Set of all customer names */
- *customer_street* = {Main, North, Park, …} /* set of all street names*/
- *customer_city* = {Harrison, Rye, Pittsfield, …} /* set of all city names */

Then *r* = { (Jones, Main, Harrison),
(Smith, North, Rye),
(Curry, North, Rye),
(Lindsay, Park, Pittsfield) }

is a relation over ***customer_name x customer_street x customer_city***

Relation Schema

Let a_1, a_2, …, a_n be attributes with domains D_1, D_2, …. D_n.

Then the set {a_1: D_1, a_2: D_2, … , a_n: D_n} is a relational schema.

Relational Schema: A named relation defined by a set of attribute and domain name pairs.

Ex.

Students(*sid*: string, *name*: string, *login*: string, *age*: integer, *gpa*: real)

A Relation is a set of tuples (a_1: d_1, a_2: d_2, … , a_n: d_n) such that $d_1 \in D_1$, $d_2 \in D_2$ … $d_n \in D_n$. We denote a relation r of a relation schema R by **r(R)** i.e. **s(Student)**

Each element in the *n*-tuple consists of an attribute and a value for that attribute.

Students					Relation Name
String	String	String	integer	real	Domain
sid	name	login	age	gpa	Attributes
53666	Jones	jones@cs	18	3.4	Relations Instance
53688	Smith	smith@eecs	18	3.2	
53650	Smith	smith@math	19	3.8	

Relation Schema

Let a_1, a_2, …, a_n be attributes with domains D_1, D_2, …. D_n.

Then the set {a_1: D_1, a_2: D_2, … , a_n: D_n} is a relational sche a.

Relational Schema: A n amed relation d by ttribute d domain name pairs.

Ex.

Students(*sid*: strir

Database Schema

R = { R_1, R_2, … , R_n}

A Relation is a set $d_2 \in D_2$ … $d_n \in D_n$

Each elem for that attribute.

Students					
String	String	S		Integer	real
sid	name	login		age	gpa
53666	Jones	jones@cs		18	3.4
53688	Smith	smith@eecs		18	3.2
53650	Smith	smith@math		19	3.8

Relation Name

Domain

Attributes

Relations Instance

Properties of Relations

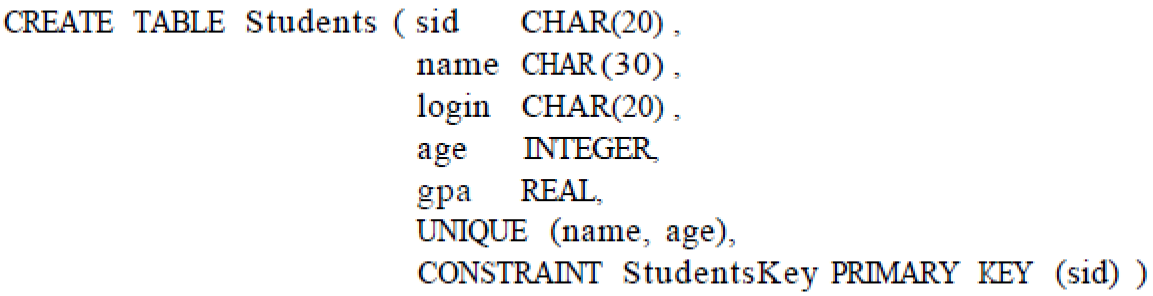
- Relation has a name that is distinct from all other relation names in the relational schema
- Each cell of the relation contains exactly one atomic (single) value (This is known as **First Normal Form**)
- Each attribute has a distinct name
- The values of an attribute are all from the same domain
- Each tuple is distinct; there are no duplicate tuples (Some RDBMS supports duplicates)
- The order of attributes has no significance
- The order of tuples has no significance.

INTEGRITY CONSTRAINTS OVER RELATIONS

An integrity constraint (IC) is a condition specified on a database schema and restricts the data that can be stored in an instance of the database.

Types

- Domain Constraints
- Key Constraints
- Referential Integrity



```
CREATE TABLE Students ( sid    CHAR(20) ,
                        name   CHAR(30) ,
                        login  CHAR(20) ,
                        age    INTEGER,
                        gpa    REAL,
                        UNIQUE (name, age),
                        CONSTRAINT StudentsKey PRIMARY KEY (sid) )
```

Relational Keys

Super key: An attribute, or set of attributes, that uniquely identifies a tuple within a relation.

Candidate Key: A superkey such that no proper subset is a **Superkey** within the relation.

A candidate key ***K*** for a relation ***R*** has two properties:

- **Uniqueness:** In each tuple of *R*, the values of *K* uniquely identify that tuple.
- **Irreducibility:** No proper subset of *K* has the uniqueness property.

Composite key: A key with more than one attribute

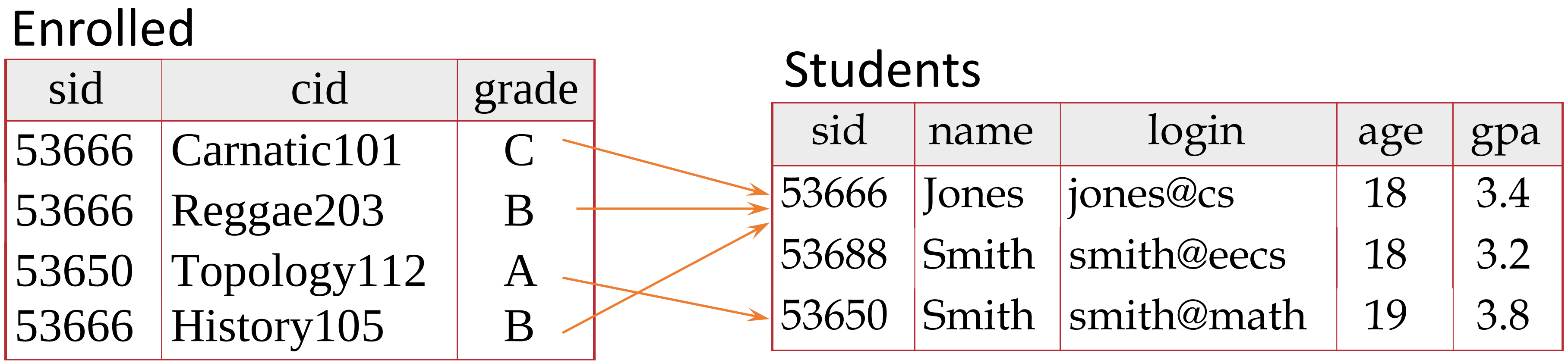
Candidate Key is a **Superkey.** But a **Superkey** may not be a **Candidate key.**

Primary Key: The candidate key that is selected to identify tuples uniquely within the relation.

Foreign Key: An attribute, or set of attributes, within one relation that matches the candidate key of some (possibly the same) relation.

Foreign Keys (Referential Integrity)

- A Foreign Key is a field whose values are Primary keys in another relation.
- Set of fields in one relation that is used to `refer’ to a tuple in another relation



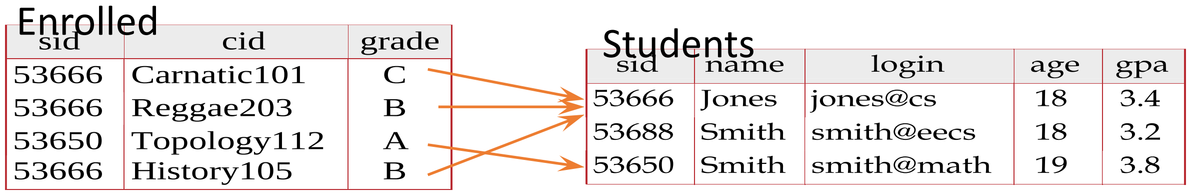
Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

```
CREATE TABLE Enrolled
   (sid CHAR(20), cid CHAR(20), grade CHAR(2),PRIMARY KEY (sid,cid),
    FOREIGN KEY (sid) REFERENCES Students )
```



Enrolled

sid	cid	grade
53666	Carnatic101	C
53666	Reggae203	B
53650	Topology112	A
53666	History105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Enforcing Referential Integrity

- Remember Students and Enrolled; *sid* in Enrolled is a foreign key that references Students.
- What should be done if an Enrolled tuple with a non-existent student id is inserted?
  - (*Reject it!*)
- What should be done if a Students tuple is deleted?
  - Also delete all Enrolled tuples that refer to it.
  - Disallow deletion of a Students tuple that is referred to.
  - Set sid in Enrolled tuples that refer to it to a *default sid*.

Integrity Constraints (ICs)

- A data model has two other parts:
  - a manipulative part, defining the types of operation that are allowed on the data
  - **a set of integrity constraints**, which ensure that the data is accurate.

- Integrity Constraints: condition that must be true for *any* instance of the database. ICs are specified when schema is defined. ICs are checked when relations are modified.
  - **Domain Constraint**
  - **Referential Constraint**: If a foreign key exists in a relation, either the foreign key value must match a candidate key value of some tuple in its home relation or the foreign key value must be wholly null.
  - **Entity Integrity**: In a **base relation**, no attribute of a primary key can be null. However, a candidate key may be NULL
  - Multiplicity Constraint
  - **General Constraints:** Additional rules specified by the users or database administrators of a database that define or constrain some aspect of the enterprise.

NULL: Represents a value for an attribute that is currently unknown or is not applicable for this tuple. Nulls are a way to deal with incomplete or exceptional data. A null represents the absence of a value.