ADITYA ENGINEERING COLLEGE (A)

DBMS
Relational Model

By

Dr. S Rama Sree
Professor in CSE & Dean(Academics)
Aditya Engineering College(A)
Surampalem.

CONTENTS

- Introduction to relational model
- Concepts of domain, attribute, tuple, relation
- Importance of null values
- Constraints (Domain, Key constraints, integrity constraints) and their importance

Relational Model Definition

Relational model is the most widely used data model. A database is a collection of one or more relations, where each relation is a table with rows and columns.

Simple tabular representation enable novice users to understand the contents of a database and ease with which complex queries can be expressed.

Introduction to the Relational Model

- The main construct for representing data in the relational model is a relation.
- A relation consists of a relation schema and a relation instance.
- The relation instance is a table, and the relation schema describes the column heads for the table.
- We first describe the relation schema and then the relation instance.
- The schema specifies the relation's name, the name of each field (or column, or attribute), and the domain of each field.
- A domain is referred to in a relation schema by the domain name and has a set of associated values.

- A sample Relation schema R(F1:D1,F2:D2,…..Fn:Dn)
- Example: The relation schema for Students

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

  Each Field name has a domain. The sid field has domain as integer.
- An instance of a relation is a set of tuples, also called records, in which each tuple has the same number of fields as the relation schema.
- A relation instance can be thought of as a table in which each tuple is a row, and all rows have the same number of fields. (The term relation instance means a relation.

SID	NAME	LOGIN	AGE
501	Ajay	ajay@cs	20
502	Ramya	ramya@ece	30
503	Rani	Rani@it	22

- A relation is defined to be a set of unique tuples or rows.
- The order of rows in relation is not important but order of fields is important.
- A relation schema specifies the domain of each field or column in the relation instance. These domain constraints in the schema specify an important condition that we want each instance of the relation to satisfy
- The Degree of a relation or arity of a relation is the number of fields

Ex:- In students relation, Degree=4(fields)

- The Cardinality of a relation is the number of tuples in the relation

Ex:- In students relation, Cardinality=3(rows)

- Let $R(f_1:D_1, ..., f_n:D_n)$ be a relation schema, and for each $f_1 \leq i \leq n$
- An instance of R that satisfies the domain constraints in the schema is a set of tuples with n fields:

$$\{ <f_I : d_I , \ldots , f_n : d_n > \; I \; d_I \; E \; Dom_i \; \ldots , d_n \; E \; Dom_n \}$$

Let Dom_i be the set of values associated with the domain named d_i.

The angular brackets < > identify the fields of a tuple.

The curly brackets { } identify the set of tuples.

- A relational database is a collection of relations with distinct relation names.
- The relational database schema is the collection of schemas for the relations in the database.
- An instance of a relational database is a collection of relation instances, one per relation schema in the database schema.

Creating & Modifying Relations using SQL

- The SQL language standard uses the word table to denote relation
- The CREATE TABLE statement is used to define a new table.
- Tuples are inserted ,using the INSERT command.
- We can delete tuples using the DELETE command.
- We can modify the column values in an existing row using the UPDATE command

Importance of NULL values in SQL

- It could be possible that at the time of data entry, information is not available.
- So SQL supports a special value known as NULL which is used to represent the values of attributes that may be unknown.
- SQL places a NULL value in the field in the absence of a user-defined value.
- Ex: The Apartment_number attribute of an address applies only to address that are in apartment buildings and not to other types of residences.
- **Importance of NULL value:**
- It is important to understand that a NULL value is different from a zero value.
- A NULL value is used to represent a missing value, but that it usually has one of three different interpretations:
  - The value is unknown (value exists but is not known)
  - Value is not available (exists but is purposely withheld)
  - Attribute is not applicable (undefined for this tuple)

Integrity Constraints over Relations

- Integrity constraint is a condition specified on a database schema and restricts the data that can be stored in an instance of the database.
- If a database instance satisfies all the integrity constraints specified on the database schema, it is a legal instance.
- A DBMS enforces Integrity Constraints(ICs), in that it permits only legal instances to be stored in the database.
- The integrity constraints must hold for the database in addition to the Domain Constraints(Not Null, Unique, Check, Default).

STUDENTS RELATION

Let us consider the following instance of the students relation

sid	*name*	*login*	*age*	*gpa*
50000	Dave	dave@cs	19	3.2
53666	Jones	jones@cs	18	3.3
53688	Smith	smith@ee	18	3.2
53650	Smith	smith@math	19	3.7
53831	Madayan	madayan@music	11	1.8
53832	Guldu	guldu@music	12	2.0

Key Constraints

- A Key Constraint is a statement that a certain minimal subset of the fields of a relation is a unique identifier for a tuple.
- Ex: The Students relation and the constraint that no two students have the same student id is a key constraint.
- A set of fields that uniquely identifies a tuple according to a key constraint is called a candidate key for the relation; we often refer this as key.
- In the case of the Students relation, the (set of fields containing just the) sid field is a candidate key

- There are two parts to the definition of Candidate Key:

1. Two distinct tuples in a legal instance (an instance that satisfies all ICs, including the key constraint) cannot have identical values in all the fields of a key.
2. No subset of the set of fields in a key is a unique identifier for a tuple

- A relation may have several candidate keys.
- The set of fields {sid, name} is not a key for Students, because this set properly contains the key {sid}.
- By stating that {login, age} is a key, the user is declaring that two students may have the same login or age, but not both
- Out of all the available candidate keys, a database designer can identify a primary key.
- **A primary key is the column or columns that contain values that uniquely identify each row in a table**. These column(s) cannot have Null values.
- Intuitively, a tuple can be referred to from elsewhere in the database by storing the values of its primary key fields.

Specifying Key Constraints in SQL

- CREATE TABLE Students (

  sid int,

  name CHAR (30) ,

  login CHAR(20) ,

  age int,

  gpa float,

  UNIQUE (name, age),

  CONSTRAINT StudentsKey PRIMARY KEY (sid)) ;

- This definition says that sid is the primary key and the combination of name and age is also a key.
- The definition of the primary key also illustrates how we can name a constraint by preceding it with CONSTRAINT constraint-name. If the constraint is violated, the constraint name is returned and can be used to identify the error

Foreign Key Constraints

- Sometimes the information stored in a relation is linked to the information stored in another relation.
- If one of the relations is modified, the other must be checked, and perhaps modified, to keep the data consistent.
- An IC involving both relations must be specified if a DBMS is to make such checks. The most common IC involving two relations is a foreign key constraint.
- Suppose that, in addition to Students, we have a second relation:

  Enrolled(studid: string, cid: string, grade: string)
- Any value that appears in the studid field of an instance of the Enrolled relation should also appear in the sid field of some tuple in the Students relation.
- The studid field of Enrolled is called a foreign key and refers to Students.
- The foreign key in the referencing relation Enrolled must match the primary key of the referenced relation Students. The column names may be different but the data types must be compatible.
- The FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.

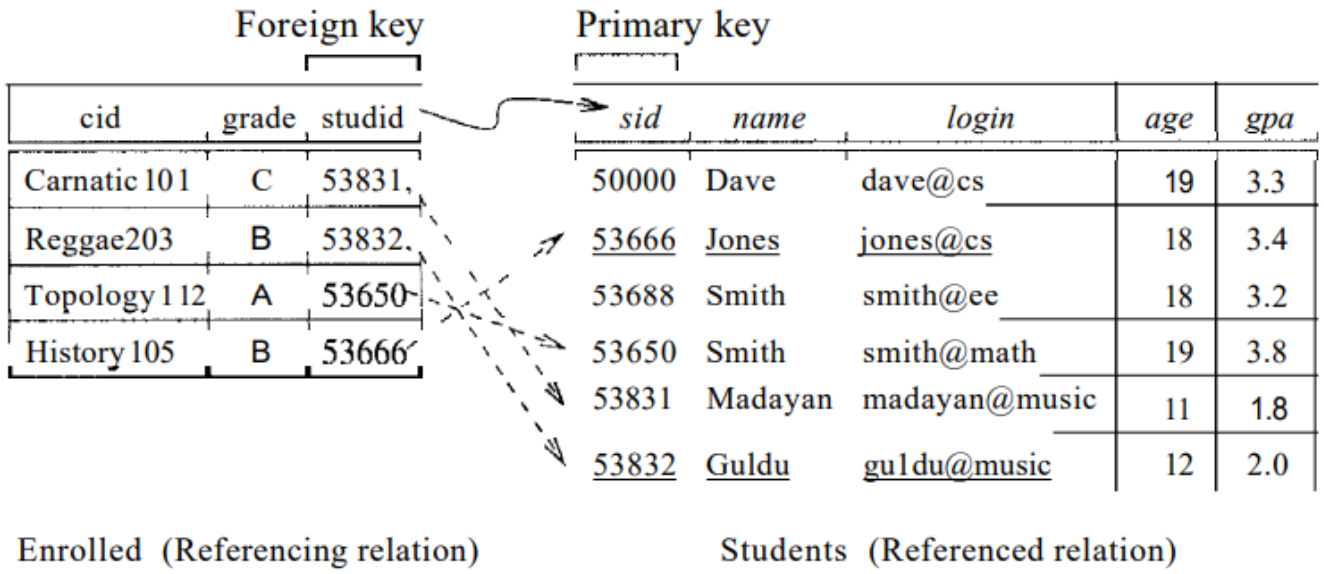

Figure 3.4 Referential Integrity

- ***Referential integrity* constraint** requires that a foreign key must have a matching primary key.
- If we try to insert the tuple (55555, Artl04, A) into Enrolled, the IC is violated because there is no tuple in Students with sid 55555; the database system should reject such an insertion.
- Similarly, if we delete the tuple (53666, Jones, jones@cs, 18, 3.4) from Students, we violate the foreign key constraint. So the database system will reject the deletion

Specifying Foreign Key Constraints in SQL

- CREATE TABLE Enrolled (

  studid int ,

  cid int,

  grade CHAR(10),

  PRIMARY KEY (studid, cid),

  FOREIGN KEY (studid) REFERENCES Students) ;

- The foreign key constraint states that every studid value in Enrolled must also appear in Students, that is, studid in Enrolled is a foreign key referencing Students.
- The primary key constraint for Enrolled states that a student has exactly one grade for each course he or she is enrolled in.

Enforcing Integrity Constraints

ICs are specified when a relation is created and enforced when a relation is modified.

- Insertion that violates the primary key constraint will not be allowed by DBMS.

  Insert into Students values(53688,'Mike', 'mike@ee',17,3.4); // duplicate sid

  Insert into Students values(null,'Mike', 'mike@ee',17,3.4); // null sid

- Updation that violates the primary key constraint will not be allowed by DBMS.

  Update Students set sid=50000 where sid=53688; //duplicate value

- Foreign Key Violations are also possible

  insert into Enrolled (cid,grade,studid) values('Hindi101', 'B',51111); // studid is not in Students table.

- What happens when the foreign key is violated
- The options are:
  1. Delete all Enrolled rows that refer to the deleted Students row.
  2. Disallow the deletion of the Students row if an Enrolled row refers to it.
  3. Set the studid column to the sid of some (existing) 'default' student, for every Enrolled row that refers to the deleted Students row.
  4. For every Enrolled row that refers to it, set the studid column to null

- The solution by DBMS is the use of ON DELETE CASCADE(Option 1), ON DELETE NO ACTION (Option 2), ON DELETE SET DEFAULT (Option 3), ON DELETE SET NULL (Option 4)
- CREATE TABLE Enrolled (studid CHAR(20) , cid CHAR(20) , grade CHAR(10),

```
PRIMARY KEY (studid, dd),
FOREIGN KEY (studid) REFERENCES Students
ON DELETE CASCADE
ON UPDATE NO ACTION);
```

Querying Relational data

- A query is a request for data or information from a database table or combination of tables.
- Find all the details of students who are younger than 19

  Select * from students s where s.age <19;

- Find the name and login of students who are younger than 19.

  Select s.name, s.login from students s where s.age <19.

- Find the names and courseid of all students who obtained grade A

  Select s.name, c.cid from students s,enrolled c where s.sid=c.studid and c.grade='A';

References

Text Books:

- Database Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke, TMH.
- Database System Concepts,5/e, Silberschatz, Korth, TMH.

Reference Books:

- Introduction to Database Systems, 8/e C J Date, PEA.
- Database Management System, 6/e Ramez Elmasri, Shamkant B. Navathe, PEA
- Database Principles Fundamentals of Design Implementation and Management, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

Web Links:

- https://nptel.ac.in/courses/106/105/106105175/
- https://www.geeksforgeeks.org/introduction-to-nosql/
- https://beginnersbook.com/2015/05/normalization-in-dbms/


Thank
You