Relational Model

Content

- Introduction to Relational Model
- Structure of Relational Databases
- Database Schema
- Keys
- Schema Diagrams
- Relational Query Languages
- Relational Operations
- Relational Algebra
- Operations of Relational Algebra

Introduction to Relational Model

- The relational model is today the primary data model for commercial data-processing applications.
- It attained its primary position because of its simplicity, which eases the job of the programmer, compared to earlier data models such as the network model or the hierarchical model.

Structure of Relational Databases

- A relational model can be represented as a table of rows and columns.
- A relational database has following major components:
  1. Table / Relation
  2. Record or Tuple
  3. Field or Column name or Attribute
  4. Domain
  5. Relation Instance

Components of Relational database

1. Table / Relation

- A table is a collection of data represented in rows and columns. Each table has a name in database.
- A relational database consists of a collection of tables, each of which is assigned a unique name.
- For example, the following table “STUDENT” stores the information of students in database.

Table: STUDENT

Student_Id	Student_Name	Student_Addr	Student_Age
101	Chaitanya	Dayal Bagh, Agra	27
102	Ajeet	Delhi	26
103	Rahul	Gurgaon	24

Components of Relational database

2. Record or Tuple

- Each row of a table is known as record. It is also known as tuple. For example, the following row is a record that we have taken from the above table.

102	Ajeet	Delhi	26

3. Field or Column name or Attribute

- The above table “STUDENT” has four fields (or attributes): Student_Id, Student_Name, Student_Addr & Student_Age.

Components of Relational database

4. Domain

- A domain is a set of permitted values for an attribute in table. For example, a domain of month-of-year can accept January, February,…December as values, a domain of dates can accept all possible valid dates etc.
- An attribute cannot accept values that are outside of their domains. For example, In the above table “STUDENT”, the Student_Id field has integer domain.

Components of Relational database

5. Relation Instance

- The term relation instance to refer to a specific instance of a relation, i.e., containing a specific set of rows.

Table: STUDENT

Student_Id	Student_Name	Student_Addr	Student_Age
101	Chaitanya	Dayal Bagh, Agra	27
102	Ajeet	Delhi	26
103	Rahul	Gurgaon	24

Database Schema

- Database schema is the logical design of the database.
- Database instance is a snapshot of the data in the database at a given instant in time.

Database Schema

- Consider the below relations

instructor 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

department relation

dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000

Keys in DBMS

Key plays an important role in relational database; it is used for identifying unique rows from table. It also establishes relationship among tables.

Types of keys in DBMS

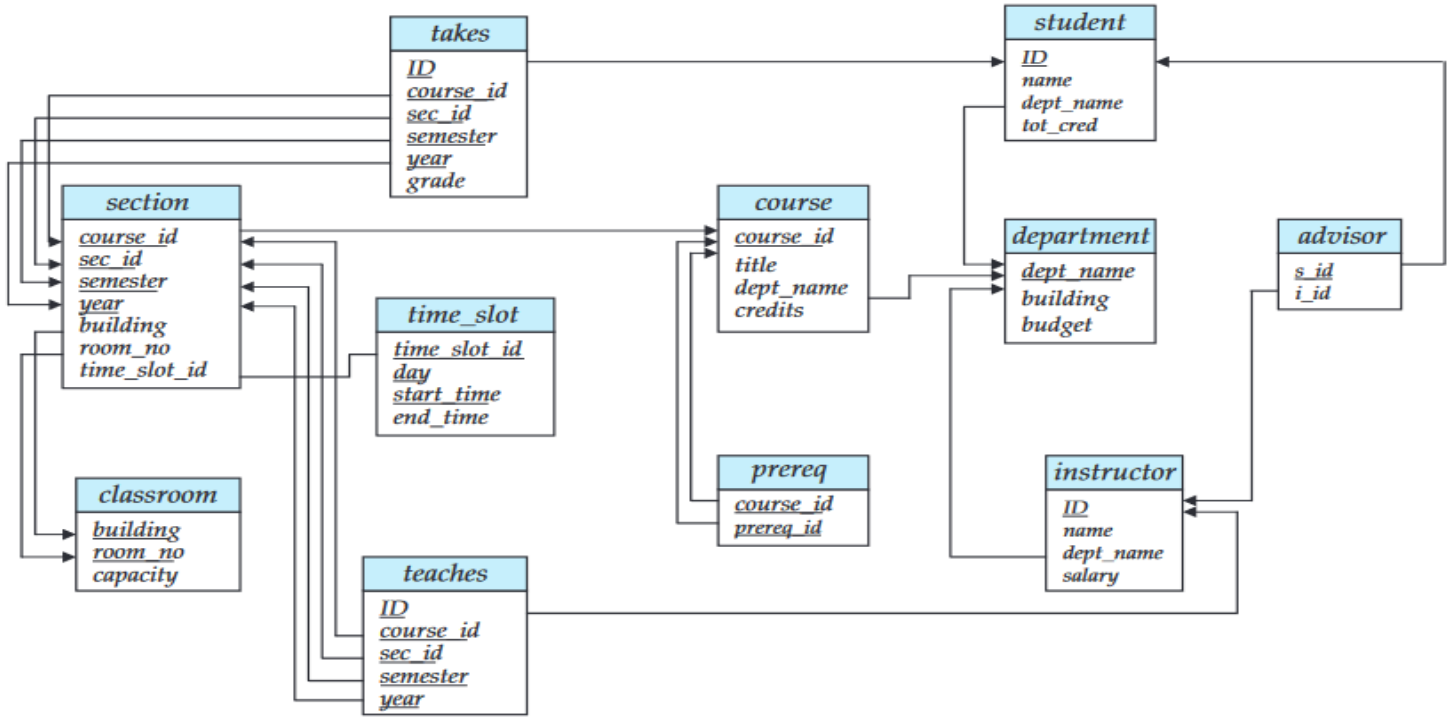
- **Primary Key** – A primary is a column or set of columns in a table that uniquely identifies tuples (rows) in that table.
- **Super Key –** A super key is a set of one of more columns (attributes) to uniquely identify rows in a table.
- **Candidate Key–** A super key with no redundant attribute is known as candidate key

Keys in DBMS

- **Alternate Key** – Out of all candidate keys, only one gets selected as primary key, remaining keys are known as alternate or secondary keys.
- **Composite Key** – A key that consists of more than one attribute to uniquely identify rows (also known as records & tuples) in a table is called composite key.
- **Foreign Key** – Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables.

Schema Diagrams

A database schema, along with primary key and foreign key dependencies, can be depicted by **Schema diagrams**.



Relational Query Languages

- A **query language** is a language in which a user requests information from the database.
- Query languages can be categorized as either procedural or nonprocedural.
- In a **procedural** language, the user instructs the system to perform a sequence of operations on the database to compute the desired result.
- In a **nonprocedural** language, the user describes the desired information without giving a specific procedure for obtaining that information.

Relational Operations

- All procedural relational query languages provide a set of operations that can be applied to either a single relation or a pair of relations.
- These operations have the nice and desired property that their result is always a single relation.
- This property allows one to combine several of these operations in a modular way.
- Specifically, since the result of a relational query is itself a relation, relational operations can be applied to the results of queries as well as to the given set of relations.

Relational Operations

- Few relational operations include:
  - Join operation
  - Natural Join operation
  - Cartesian Product
  - Intersection
  - Union
  - Set difference

Relational Algebra

- Relational database systems are expected to be equipped with a query language that can assist its users to query the database instances. There are two kinds of query languages – relational algebra and relational calculus.
- Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output. It uses operators to perform queries. An operator can be either **unary** or **binary**. They accept relations as their input and yield relations as their output.

Operations of Relational Algebra

- Relational algebra is performed recursively on a relation and intermediate results are also considered relations.

The fundamental operations of relational algebra are as follows −

1. Selection
2. Projection
3. Union
4. Natural Join
5. Cartesian product

- Consider the below relations for example

instructor 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 Sald	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp.Sci	75000
58583	Califeri	History	62000
76543	Singh	Finance	80000
83821	Brandt	Comp.Sci	92000

department relation

dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

Operations of Relational Algebra

1. Selection Operation (σ)

- It selects tuples that satisfy the given predicate from a relation.

  Notation – $\sigma_p(r)$

  Where:

  σ is used for selection prediction

  r is used for relation

  p is used as a propositional logic formula which may use connectors like: AND OR and NOT. These relational can use as relational operators like =, ≠, ≥, <, >, ≤.

Operations of Relational Algebra

Example of Selection Operation (σ)

- **Notation** – $\sigma_{salary>=85000}(instructor)$

The above notation returns the records with the salary attribute value greater than 85000 from the instructor relation.

ID	name	dept_name	salary
12121	Wu	Finance	90000
22222	Einstein	Physics	95000
33456	Gold	Physics	87000
83821	Brandt	Comp.Sci	92000

Operations of Relational Algebra

2. Projection Operation (∏)

- It projects column(s) that satisfy a given predicate.

  Notation – $\prod_{A1, A2, An} (r)$

  Where A_1, A_2 , A_n are attribute names of relation **r**.

- Duplicate rows are automatically eliminated, as relation is a set.

Operations of Relational Algebra

Example of Projection Operation (∏)

- **Notation** – $\prod_{ID,\ salary}(instructor)$

The above notation returns the id and the salary attribute of the instructor relation.

id	salary
10101	65000
12121	90000
15151	40000
22222	95000
32343	60000
33456	87000
45565	75000
58583	62000
76543	80000
83821	92000

Operations of Relational Algebra

3. Union Operation (∪)

- It performs binary union between two given relations and is defined as −r ∪ s = { t | t ∈ r or t ∈ s}

  Notation − r U s

- A union operation must hold the following condition:
- R and S must have the attribute of the same number.
- Duplicate tuples are eliminated automatically.

Operations of Relational Algebra

Example of Union Operation (∪)

- **Notation** – instructor ∪ department { dept_name | dept_name ∈ instructor or dept_name ∈ department}

Output the union of tuples from the two relations, instructor and department.

id	name	dept_name	salary
10101	Srinivasan	Comp.Sci	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Sald	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp.Sci	75000
58583	Califeri	History	62000
76543	Singh	Finance	80000
83821	Brandt	Comp.Sci	92000

Operations of Relational Algebra

5. Cartesian Product (X)

- Combines information of two different relations into one.

  Notation – r X s

- Where **r** and **s** are relations and their output will be defined as
- r X s = { q t | q ∈ r and t ∈ s}

Operations of Relational Algebra

Example of Cartesian Product Operation (X)

Notation – instructor X department

Output all pairs of rows from the two input relations (regardless of whether or not they have the same values on common attributes)

THANK YOU