

RELATIONAL MODEL

The relational model is based on the mathematical concept of a **relation**, which is physically represented as a **table**.

RELATIONAL DATA STRUCTURE

Relation. A relation is a table with columns and rows. A relation is also referred to as a **file**.

Attribute. An attribute is a named column of a relation. An attribute is also referred to as a **field**.

Domain. A domain is the set of allowable values for one or more attributes. Every attribute in a relation is defined on a **domain**. Domains may be distinct for each attribute, or two or more attributes may be defined on the same domain.

Tuple. A tuple is a row of a relation. The elements of a relation are the rows or **tuples** in the table. A tuple is also referred to as a **record**.

Degree. The degree of a relation is the number of attributes it contains.

Cardinality. The cardinality of a relation is the number of tuples it contains.

DATABASE RELATIONS

Relation schema. A named relation defined by a set of attribute and domain name pairs.

The relation instance for Branch relation can be defined as:

{(branchNo: B005, street: 22Deer Rd, city: London, postcode: SW1 4EH)}

Also,

{(branchNo: B007, street: 22 Argyll St, city: Aberdeen, postcode: AB2 3SU)}

Properties of Relations

A relation has the following properties:

- The 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;
- 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;
- the order of attributes has no significance;
- the order of tuples has no significance.

RELATIONAL KEYS

One or more attributes that uniquely identifies each tuple in a relation.

Taking an example of Employee table:

Employee (EmployeeID, FullName, SSN, DeptID)

Candidate key. Are individual columns in a table that qualifies for uniqueness of all the rows. In Employee table EmployeeID and SSN are candidate keys.

Primary key. The column that is chosen to maintain uniqueness in a table. In Employee table, you can choose either EmployeeID or SSN columns. EmployeeID is preferable choice, as SSN is a secure value.

Alternate keys. Candidate column other than the Primary column, like if EmployeeID is the PK, then SSN would be the alternate key.

Super key. If you add any other column/attribute to a primary key, then it becomes a super key, e.g EmployeeID + FullName is a super key.

Foreign key. An attribute, or set of attributes, within one relation that matches the candidate key of some relation.

INTEGRITY CONSTRAINTS

Entity Integrity

The first integrity rule applies to the primary keys of base relations.

Entity integrity. In a base relation, no attribute of a primary key can be null. A primary key is a minimal identifier that is used to identify tuples uniquely.

Referential Integrity

The second integrity rule applies to foreign keys.

Referential integrity. 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.

VIEWS

View is the dynamic result of one or more relational operations operating on the base relations to produce another relation. A view is a *virtual relation* that does not necessarily exist in the database but can be produced upon request by a particular user, at the time of request.

Purpose of Views

- It provides a powerful and flexible security mechanism by hiding parts of the database from certain users. Users are not aware of the existence of any attributes or tuples that are missing from the view.

- It permits users to access data in a way that is customized to their needs, so that the same data can be seen by different users in different ways, at the same time.
- It can simplify complex operations on the base relations. For example, if a view is defined as a combination of two relations, users may perform more simple operations on the view, which will be translated by the DBMS into equivalent operations on the join.

RELATIONAL DATABASE LANGUAGE

Structured Query Language (SQL)

Structured Query Language (SQL) is a language that emerged from the development of the relational model.

SQL components:

- Data Definition Language (DDL) - Includes commands to create database objects such as tables, indexes, and views, as well as commands to define access rights to those database objects i.e DDL provides commands for defining relation schemes, deletion relation, creating indices and modifying relation schemers
- Data Manipulation Language (DML) for retrieving and updating data. It includes commands to insert tuples into, delete tuples from and modify tuples in the database.

SQL data definition commands

COMMAND OR OPTION	DESCRIPTION
CREATE DATABASE	- Creates a database schema
CREATE TABLE	- Creates a new table in the user's database schema
NOT NULL	- Ensures that a column will not have null values
PRIMARY KEY	- Defines a primary key for a table
FOREIGN KEY	- Defines a foreign key for a table
CREATE VIEW	- Creates a dynamic subset of rows/columns from one or more tables
DROP TABLE	- Permanently deletes a table (and its data)
DROP VIEW	- Permanently deletes a view

SQL data manipulation commands

COMMAND OR OPTION	DESCRIPTION
INSERT	- Inserts row(s) into a table
SELECT	- Selects attributes from rows in one or more tables or views
WHERE	- Restricts the selection of rows based on a conditional expression
GROUP BY	- Groups the selected rows based on one or more attributes

HAVING - Restricts the selection of grouped rows based on a condition

ORDER BY - Orders the selected rows based on one or more attributes

UPDATE - Modifies an attribute's values in one or more table's rows

DELETE - Deletes one or more rows from a table

COMPARISON OPERATORS

=, <, >, <=, >=, <> - Used in conditional expressions

LOGICAL OPERATORS

AND/OR/NOT - Used in conditional expressions

SPECIAL OPERATORS Used in conditional expressions

BETWEEN - Checks whether an attribute value is within a range

IS NULL - Checks whether an attribute value is null

IN - Checks whether an attribute value matches any value within a value list

AGGREGATE FUNCTIONS Used with SELECT to return mathematical summaries on columns

COUNT - Returns the number of rows with non-null values for a given column

MIN - Returns the minimum attribute value found in a given column

MAX - Returns the maximum attribute value found in a given column

SUM - Returns the sum of all values for a given column

AVG - Returns the average of all values for a given column