

Introduction of Relational Model and Codd Rules in DBMS

proposes a relational database model developed by Dr. E.F. Codd, wherein data is presented in tabular form using the concept of relations; that is, two-dimensional tables. Its other key features are as follows:

- **Simplicity:** Conveying simplicity in implementation, simplifies the operations that one will want to perform on data.
- **Linking:** It uses primary and secondary keys to interlink two files or tables.
- **Normalization:** It utilizes the concept of normalization theory for designing non-redundant and efficient object-based data models.
- **Data Processing:** It utilizes Relational Algebra and Relational Calculus for manipulation of these relations. Many database languages have been developed having special features for handling relational data models.

Key elements of a relational database:

- **Relations (Tables):** It is the basic structure in which data is stored. Each relation is made up of rows and columns.

- Rows (Tuples): Each row is considered one record in the table.
- Columns (Attributes): Each column shows an attribute of the data.
- Primary Key: It is a unique identifier of each relation so that no two rows are identical. Structure: The relations comprise many columns or n-columns and rows or n-tuples, labeled using the names of the relations. Fields should be separated with a comma and enclosed in parentheses.

Example: Relational Model can be represented as shown below

STUDENT (StudNo, Sname, Special)

ENROLLMENT (StudNo, Subcode, marks)

SUBJECT (Subcode, Subname, Maxmarks, Faccode)

FACULTY (Faccode, Fname, Dept)

Terminologies المصطلحات

1. **Relational Schema**: Schema represents the structure of a relation.

Example: Relational Schema of STUDENT relation can be represented as STUDENT(STUD_NO, STUD_NAME, STUD_PHONE, STUD_STATE, STUD_COUNTRY, STUD_AGE).

2. **Relational Instance**: The set of values present in a relationship at a particular instance of time is known as a relational instance as shown in Table 1 and Table 2.

3. Attribute: Each relation is defined in terms of some properties, each of which is known as an attribute.

Example: STUD_NO, STUD_NAME, etc. are attributes of relation STUDENT.

4. The domain of an attribute: The possible values an attribute can take in a relation is called its domain.

Example: domain of STUD_AGE can be from 18 to 40.

5. Tuple: Each row of a relation is known as a tuple.

Example: STUDENT relation given below has 4 tuples.

6. NULL values: Values of some attribute for some tuples may be unknown, missing, or undefined which are represented by NULL. Two NULL values in a relationship are considered different from each other. Table 1 and Table 2 represent the relational model having two relations STUDENT and STUDENT_COURSE.

STUDENT TABLE

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY	STUD_AGE
1	RAM	9716271721	Haryana	India	20

STUD_NO	STUD_NAME	STUD_PHONE	STUD_STATE	STUD_COUNTRY	STUD_AGE
2	RAM	9898291281	Punjab	India	19
3	SUJIT	7898291981	Rajasthan	India	18
4	SURESH	9985286317	Punjab	India	21

STUDENT COURSE TABLE

STUD_NO	COURSE_NO	COURSE_NAME
1	C1	DBMS
2	C2	Computer Networks
1	C2	Computer Networks

RDBMS Vendors

There are several vendors that offer Relational Database Management Systems (RDBMS). Here are some of the most popular ones:

:

- **Oracle:** Oracle Database is one of the most widely used RDBMS products in the market. It is known for its robustness, scalability, and reliability. It is used by many large enterprises and is particularly well-suited for data warehousing and transaction processing.
- **Microsoft:** Microsoft SQL Server is a popular RDBMS used in Windows environments. It offers a range of features, including data mining, business intelligence, and reporting services.
- **IBM:** IBM DB2 is a popular RDBMS used in enterprise environments. It offers high availability, disaster recovery, and scalability features.
- **MySQL:** MySQL is an open-source RDBMS used by many small to medium-sized businesses. It is known for its ease of use, flexibility, and low cost.
- **PostgreSQL:** PostgreSQL is another popular open-source RDBMS. It is known for its scalability, reliability, and support for complex transactions.
- **SAP:** SAP HANA is an in-memory RDBMS that is designed for high-performance analytics and data processing. It is often used in enterprise environments for real-time reporting and business intelligence.

Relational Algebra

It is a procedural Language. It consists of a set of operators that can be performed on relations. [Relational Algebra](#) forms the basis for many other high-level data sub-languages like [SQL](#) and [QBE](#).

Relational algebra has mainly 9 types of operators.

- [UNION](#)
- [INTERSECTION](#)
- [MINUS](#)
- [TIMES](#)
- [SELECTION](#)
- [PROJECTION](#)
- [JOIN](#)
- [DIVISION](#)
- [RENAME](#)

1. UNION (U): A and B are two relations. It displays total values (Attributes) in both relations. It avoids duplicate values in both relations. U symbol can be used.

Syntax:

A UNION B (or) A U B

Example:

A = { clerk, manager, salesman }

B = { president, clerk, manager }

A UNION B = { clerk, manager, salesman, president }

2. INTERSECTION ($\cap$): A and B are two relations. It displays common elements in both relations. “ $\cap$ ” symbol can be used.

Syntax:

A INTERSECT B (or) $A \cap B$

Example:

$A = \{ \text{clerk, manager, salesman} \}$

$B = \{ \text{president, clerk, manager} \}$

$A \text{ INTERSECT } B = \{ \text{clerk, manager} \}$

3. DIFFERENCE ($-$): A and B are two relations. It displays elements in relation A not in relation B.

Syntax:

A MINUS B (OR) $A - B$

Example:

$A = \{ \text{clerk, manager, salesman} \}$

$B = \{ \text{president, clerk, manager} \}$

$A \text{ MINUS } B = \{ \text{salesman} \}$

4. CARTESIAN PRODUCT(X): A and B are two relations. It has a new relation consisting of all pair wise combinations of all elements in A and B. The relation A has “m” elements and relation B has “n” elements, then the resultant relation will be “ $m * n$ ”.

Syntax:

A TIMES B (OR) $A \times B$

Example:

$A = \{ \text{clerk, manager, salesman} \}$

$B = \{ \text{president, clerk, manager} \}$

$A \text{ TIMES } B = \{ (\text{clerk, president}),$
 $(\text{clerk, clerk}), (\text{clerk, manager}),$
 $(\text{manager, president}), (\text{manager, clerk}),$
 $(\text{manager, manager}), (\text{salesman, president}),$
 $(\text{salesman, clerk}), (\text{salesman, manager}) \}$

5. SELECTION (σ): It displays all the attributes or columns of all the rows in a relation. “ σ ” operator can be used to select the attributes in a relation.

In general SELECT operation is denoted by

$(\sigma)(R)$

(σ) (Sigma): SELECT operator

θ : Selection condition

R: Relation or relational algebra expression.

In general the select condition is a Boolean condition (i.e. an expression using logical connective) of terms that have the form attribute OP constant 1 OP attribute2 where OP is the comparison operators <, >, =, >= etc.

Syntax:

σ condition (relation name)

6. PROJECTION (π): It displays some specified columns in a relation. “ π ” operator can be used to select some specified columns in a relation. It selects tuples that satisfy the given predicate from a relation. It displays some specified columns by using some conditions.

Syntax:

Syntax: $\pi(\text{col1, col2...}) \text{ Relation Name}$

Example:

$\pi(\text{sno, sname, total}) \text{ MARKS}$

7. JOIN(): It combines two or more relations. Symbols can be used. It can be mainly divided into mainly 4 types. These are mainly

- [Inner Join](#)
- [Outer Join](#)
- [Left Outer Join](#)
- [Right Outer Join](#)

8. DIVIDE ($\div$): It divides the tuple from one relation to another relation

Syntax:

A DIVIDE B (OR) $A \div B$

Example:

$A = \{\text{clerk, manager, salesman}\}$

$B = \{\text{clerk, manager}\}$

$A \text{ DIVIDE } B = \{\text{salesman}\}$

9. RENAME(ρ): It gives another name to the relation.

Syntax:

$\rho(\text{OLD RELATION, NEW RELATION})$

Example:

$\rho(\text{STUDENT, MARKS})$

It changes the “student” relation to “Marks” relation.

It also renames the specified column.

It changes the old-column name to new-column name.

Features of the Relational Model and Codd's Rules

- **Tables/Relations:** The basic building block of the relational model is the table or relation, which represents a collection of related data. Each table consists of columns, also known as attributes or fields, and rows, also known as tuples or records.
- **Primary Keys:** In the relational model, each row in a table must have a unique identifier, which is known as the primary key. This ensures that each row is unique and can be accessed and manipulated easily.
- **Foreign Keys:** Foreign keys are used to link tables together and enforce referential integrity. They ensure that data in one table is consistent with data in another table.
- **Normalization:** The process of organizing data into tables and eliminating redundancy is known as normalization. Normalization is important in the relational model because it helps to ensure that data is consistent and easy to maintain.
- **Codd's Rules:** Codd's Rules are a set of 12 rules that define the characteristics of a true relational DBMS. These rules ensure that the DBMS is consistent, reliable, and easy to use.
- **Atomicity, Consistency, Isolation, Durability (ACID):** The ACID properties are a set of properties that ensure that transactions are processed reliably in the relational model. Transactions are sets of operations that are executed as a single unit, ensuring that data is consistent and accurate.