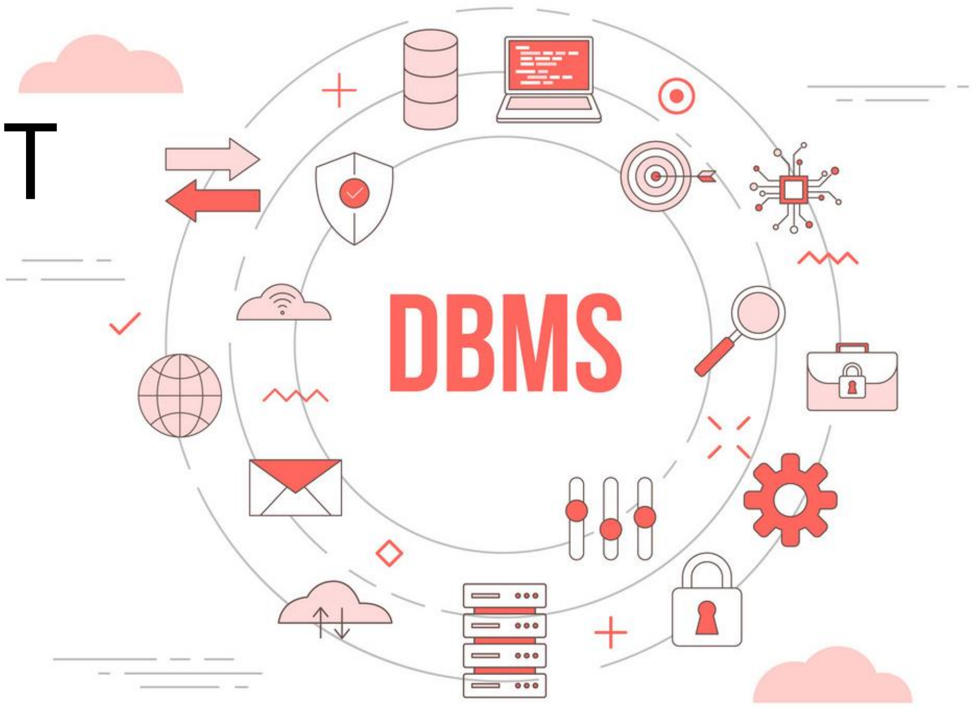
DATABASE MANAGEMENT SYSTEMS

Relational Model



Dr. Jay Sarraf
School of Computer Engineering
KIIT Deemed to be University

Overview

What is a Relational Model?

A database management system (DBMS) uses an abstract model called the relational model to organize and manage the data it stores.

The Relational Data Model was invented in 1970 by computer scientist Edgar F. Codd.

The relational data model is a paradigm that uses a collection of tables or relations to describe information about entities and connections between various entity sets. It makes use of the mathematical relational and set theory concepts and is employed in RDBMS.

Businesses were tremendously assisted in improving data usability by the Relational Data Model in RDBMS. In order to express and store data in tables, logical relations are used. SQL makes it simpler to construct sophisticated queries and to access and query data.

Relational Data Model

- An association between a group of values is represented by a row in a table. As a result, a row is an entity and a table is an entity set.
- The attributes refer to the columns or characteristics, while the domain refers to the range of acceptable values for each attribute. More than one attribute may share the same domain.
- The attribute values must be atomic, i.e. indivisible, where Degree is the number of attributes in the relation or table and Cardinality is the number of tuples or rows in the relation or table.

Primary Key

Domain
(Not Null constraint)

CID	CName	Status
C1	Addy	Active
C2	Bobby	Inactive
C3	Cyril	Active

Tuple / Row
Cardinality is the total no. of Rows.

Column / Attributes
Degree is the total no. of Columns.

Cont…

- Assume that the domains are D1, D2, and D3. A 3-tuple (v1, v2, v3) makes up any row of the table, where v1 ∈ D1, v2 ∈ D2
- and v3 ∈ D3. Therefore, just a portion of the set of all potential rows will be present in the table. The table is therefore a subset of D1 x D2 x D3.
- Every relation's attribute has a specific name.
- The database's logical layout is defined by its schema. If (a1, a2 ...an) be the attributes, then the relation schema will be R=(a1,a2 ...an).
- Relation is indicated by lower case names, and Relation Schema is the name beginning with an uppercase letter. • A database instance is a snapshot of the data in the database at a certain moment in time.

Relation (R)

D1	D2	D3
v1	v2	V3

Cont…

Advantages of Relational Model

- The correctness and consistency of the data.
- There is no redundant data.
- Constraints that permit validation before data entry and access provide access control and integrity.
- Offers very high security.
- Supports for any form of storage (numbers, characters, date, images, audio, text files).
- Multiple people can manage and use data simultaneously.
- Information may be transferred between different platforms.

Relational Data Integrity

Integrity restrictions are a set of guidelines. It serves to preserve the integrity of the data. Integrity restrictions guarantee that data insertion, updating, and other procedures must be carried out without compromising data integrity.

There are 4 types of integrity constraints:

1. Key constraints
2. Domain Constraints.
3. Entity Integrity Constraints.
4. Referential Integrity Constraints

Key Constraints

The constraint that uniquely identifies the tuple of a relation is known as a key constraint. A Relation must have at least one key constraint. Each tuple of a key attribute has a distinct value if that attribute is specified as a key constraint.

Example:
Employee ID is a key attribute in the following Employee table. No two values for this characteristic may share the same name. There can be no Null values for this property.

Key Constraint → EID

EMPLOYEE

EID	**CName**	**Salary**
E1	Addy	10000
E2	Bobby	20000
E3	Cyril	30000

Domain Constraints

The constraint known as a "domain constraint" establishes the set of guidelines for a "Relation" attribute. It states that the field will only take values that fall inside the given domain. The user-defined data types, such as Date, String, Integer, Currency, Character, etc., are domain constraints.

Example:
If we wish to enter salaries as whole numbers in the following Employee table, we may declare the domain Integer on the Salary attribute.

As a result, the attribute only allows whole numbers. Values like 10,000.11, "twenty-five thousand," etc. cannot be accepted.

EMPLOYEE

EID	**CName**	**Salary**
E1	Addy	10000
E2	Bobby	20000
E3	Cyril	30000

Entity Integrity Constraints

A restriction on entity integrity stipulates that the main key attribute's value cannot be NULL. It is impossible for the main key property to be NULL since it uniquely identifies each tuple in the relation; if any of the key attribute's cells have a NULL value, it will be difficult to retrieve that row.

Example:
The main key in the following table is EID. Because we cannot access the information of an employee whose EID is NULL, this constraint indicates that EID cannot be NULL.

Entity Integrity Constraints → EID

NULL not allowed → empty EID cell (Diva row)

EMPLOYEE

EID	**CName**	**Salary**
E1	Addy	10000
E2	Bobby	20000
E3	Cyril	30000
	Diva	40000

Referential Integrity Constraints

A sort of constraint called a referential integrity constraint exists between two tables in order to preserve data consistency in both tables.This particular integrity restriction is based on the idea of a foreign key.This constraint indicates that each value of the foreign key must be present in the Primary key attribute if the foreign key of one table is related to the primary key of another table. Additionally, any foreign key value may be NULL.

Example: The two tables, titled Employee and Department, are listed below. A foreign key in the Employee database is DID (Foreign Key).

Referential Integrity Constraints

EMPLOYEE

EID	CName	Salary	DID
E1	Addy	10000	D1
E2	Bobby	20000	D1
E3	Cyril	30000	D2

DEPARTMENT

DID	DName	Location
D1	Addy	L1
D2	Bobby	L2
D3	Cyril	L3

CODD’s Rule

Rule 0: The Fundamental Principle
A relational database must be used. so that the system's relational skills can manage the database.

Rule 1: Information Rule
A database contains a variety of data, and each cell of a table in the form of rows and columns must be used to store this data.

Rule 2: Assurance of Access
A relational database's primary key value, table name, and column name can be used in conjunction to logically access every single or precise piece of data (atomic value).

Rule 3: Treat Null Values Systematically
This rule outlines how database entries with Null values should be handled systematically. In a database, the word "null" can signify a number of things, including "missing data," "no value in a cell," "inappropriate information," "unknown data," and "the main key shouldn't be empty."

CODD's Rule

Rule 4: Relational-based Active/Dynamic Online Catalogue
A database dictionary is a representation of the whole logical structure of the descriptive database that has to be kept online. It uses a comparable query language to access the database and grants users permission to do so.

Rule 5: Comprehensive Data SubLanguage Rule
The relational database supports a number of different languages, but if we want to access the database, the language must be explicit, linear or well-defined in terms of syntax, character strings, and support a wide range of operations, including data definition, view definition, data manipulation, and integrity constraints. It is a breach of the database's rules if access to the data is permitted without the use of any language.

Rule 6: View Updating Rule for
The database systems must realistically update all views tables, which can potentially be modified.

CODD’s Rule

Rule 7: High-Level Insert, Update, and Delete at the Relational Level,
High-level relational actions like insert, update, and delete should be followed by a database system at each level or for a single row. Additionally, the database system provides union, intersection, and minus operations.

Rule 8: The Independence of Physical Data,
To access a database, all data must be physically independent and be stored in the application or database. Each piece of data shouldn't be dependent on any other pieces or programmes. External applications that access the database's data will not be impacted by data updates or changes to the database's physical structure.

CODD’s Rule

Rule 9: Independent Logical Data Rule,
It is comparable to the independence of physical facts. This means that modifications to the logical level (table structures) shouldn't have an impact on the user's view (application). For instance, if a table is divided into two tables or two tables are joined to form one table, the user view application shouldn't be affected by these modifications.

Rule 10: Independence, Integrity Rule,
When utilizing the SQL query language to put data into table cells, a database must guarantee integrity independence. The integrity of all entered values should never be compromised by outside influences or applications. Additionally, it aids in separating each front-end application's database needs.

CODD's Rule

Rule 11 The : Distribution Independence Rule,
Distribution independence rule depicts a database that must function properly even if it is utilized by various end users and stored in various places. Let's say a person uses an application to access the database; in such instance, they shouldn't be aware that another user also utilizes certain data and that the data they always obtain is only available on one site. The database is accessible to end users, and the access data must be separate for each user to execute SQL queries.

Rule 12: Prohibition on Subversion
RDBMS is a SQL language that is used to store and modify data in databases, according to the non-submersion rule. A system shouldn't subvert or bypass integrity to change data if it uses a different or low-level language in addition to SQL to access the database system.