**Relational Model**

Relational Model (RM) represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.

The table name and column names are helpful to interpret the meaning of values in each row. The data are represented as a set of relations. In the relational model, data are stored as tables. However, the physical storage of the data is independent of the way the data are logically organized.

**Some popular Relational Database management systems are:**

* DB2 and Informix Dynamic Server - IBM
* Oracle and RDB – Oracle
* SQL Server and Access - Microsoft

**Codd's Rule for Relational DBMS**

E. F Codd was a Computer Scientist who invented the **Relational Model** for Database management. Based on relational model, the **Relational Database** was created. Codd proposed 13 rules popularly known as **Codd's 12 rules** to test DBMS's concept against his relational model. Codd's rule actually define what quality a DBMS requires in order to become a Relational Database Management System (RDBMS). Till now, there is hardly any commercial product that follows all the 13 Codd's rules. Even **Oracle** follows only eight and half (8.5) out of 13. The Codd's 12 rules are as follows.

**Rule zero**

This rule states that for a system to qualify as an RDBMS, it must be able to manage database entirely through the relational capabilities.

**Rule 1: Information Rule**

All information (including metadata) is to be represented as stored data in cells of tables. The rows and columns have to be strictly unordered.

**Rule 2: Guaranteed Access**

Each unique piece of data (atomic value) should be accessible by: **Table Name + Primary Key (Row) + Attribute(column).**

**Rule 3: Systematic treatment of NULL**

Null has several meanings, it can mean missing data, not applicable or no value. It should be handled consistently. Also, Primary key must not be null, ever. Expression on NULL must give null.

**Rule 4: Active Online Catalog**

Database dictionary(catalog) is the structure description of the complete Database and it must be stored online. The Catalog must be governed by same rules as rest of the database. The same query language should be used on catalog as used to query database.

**Rule 5: Powerful and Well-Structured Language**

One well-structured language must be there to provide all manners of access to the data stored in the database. Example: SQL, etc. If the database allows access to the data without the use of this language, then that is a violation.

**Rule 6: View Updation Rule**

All the view that are theoretically updatable should be updatable by the system as well. A view is a virtual table in the database.

**Rule 7: Relational Level Operation**

There must be Insert, Delete, Update operations at each level of relations. Set operation like Union, Intersection and minus should also be supported.

**Rule 8: Physical Data Independence**

The physical storage of data should not matter to the system. If say, some file supporting table is renamed or moved from one disk to another, it should not affect the application.

**Rule 9: Logical Data Independence**

If there is change in the logical structure (table structures) of the database the user view of data should not change. Say, if a table is split into two tables, a new view should give result as the join of the two tables. This rule is most difficult to satisfy.

**Rule 10: Integrity Independence**

The database should be able to enforce its own integrity rather than using other programs. Key and Check constraints, trigger etc, should be stored in Data Dictionary. This also make RDBMS independent of front-end.

**Rule 11: Distribution Independence**

A database should work properly regardless of its distribution across a network. Even if a database is geographically distributed, with data stored in pieces, the end user should get an impression that it is stored at the same place. This lays the foundation of distributed database.

**Rule 12: No subversion Rule**

If low level access is allowed to a system, it should not be able to subvert or bypass integrity rules to change the data. This can be achieved by some sort of looking or encryption.

**Basic Relational DBMS Concepts**

A **Relational Database Management System** (RDBMS) is a database management system based on the relational model introduced by E.F Codd. In relational model, data is stored in **relations**(tables) and is represented in form of **tuples**(rows).

RDBMS is used to manage Relational database. **Relational Database** is a collection of organized set of tables related to each other, and from which data can be accessed easily. Relational Database is the most commonly used database these days.

* **What is Table?**

In Relational database model, a **table** is a collection of data elements organised in terms of rows and columns. A table is also considered as a convenient representation of **relations**. But a table can have duplicate row of data while a true relation cannot have duplicate data. Table is the simplest form of data storage. Below is an example of an Employee table.

**ID Name Age Salary**

1 Adam 34 13000

2 Alex 28 15000

3 Stuart 20 18000

4 Ross 42 19020

* **What is a Tuple?**

A single entry in a table is called a **Tuple or Record or Row**. A tuple in a table represents a set of related data. For example, the above Employee table has 4 tuples/records/rows.

Following is an example of single record or tuple.

1 Adam 34 13000

* **What is an Attribute?**

A table consists of several records(row), each record can be broken down into several smaller parts of data known as Attributes. The above Employee table consist of four attributes, ID, Name, Age and Salary.

* **Attribute Domain**

When an attribute is defined in a relation(table), it is defined to hold only a certain type of values, which is known as Attribute Domain.

Hence, the attribute Name will hold the name of employee for every tuple. If we save employee's address there, it will be violation of the Relational database model.

**Name**

Adam

Alex

Stuart - 9/401, OC Street, Amsterdam

Ross

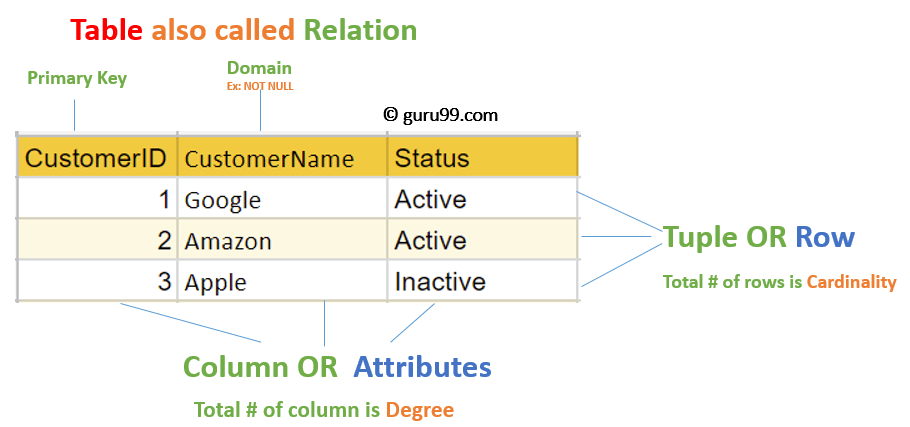
* **What is a Relation Schema?**

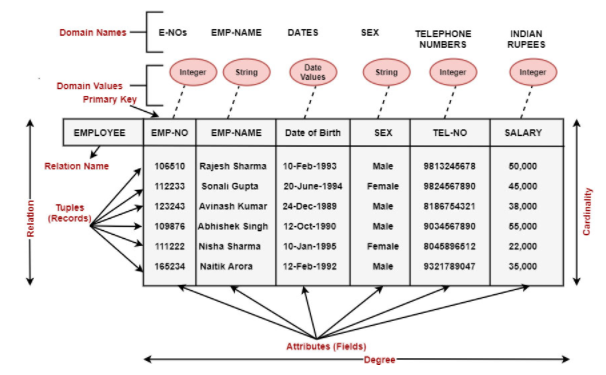
A relation schema describes the structure of the relation, with the name of the relation (name of table), its attributes and their names and type.

* **What is a Relation Key?**

A relation key is an attribute which can uniquely identify a particular tuple(row) in a relation(table).

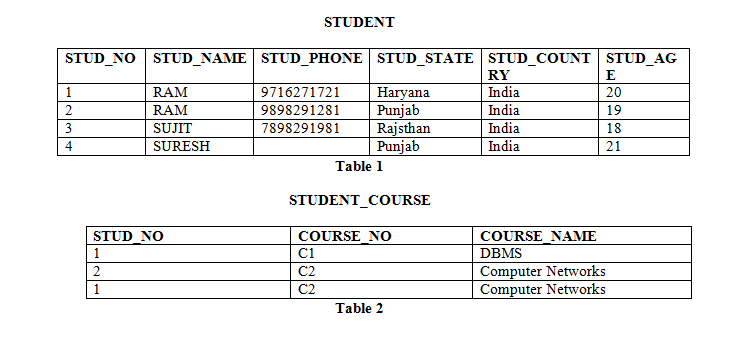
* **Degree:** The total number of attributes which in the relation is called the degree of the relation.
* **Cardinality:** Total number of rows present in the Table.
* **Column:** The column represents the set of values for a specific attribute.
* **Relation instance:** Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.

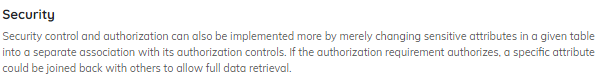




* **NULL values:** Values of some attribute for some tuples may be unknown, missing or undefined which are represented by NULL. Two NULL values in a relation are considered different from each other.

Table 1 and Table 2 represent relational model having two relations STUDENT and STUDENT\_COURSE.



* **Properties of Relations**
* Name of the relation is distinct from all other relations.
* Each relation cell contains exactly one atomic (single) value
* Each attribute contains a distinct name
* Tuple has no duplicate value
* Order of tuple can have a different sequence
* **Advantages of using Relational Model**
* **Simplicity:** A Relational data model in DBMS is simpler than the hierarchical and network model.
* **Structural Independence:** The relational database is only concerned with data and not with a structure. This can improve the performance of the model.
* **Easy to use:** The Relational model in DBMS is easy as tables consisting of rows and columns are quite natural and simple to understand
* **Query capability:** It makes possible for a high-level query language like SQL to avoid complex database navigation.
* **Data independence:** The Structure of Relational database can be changed without having to change any application.
* **Scalable:** Regarding a number of records, or rows, and the number of fields, a database should be enlarged to enhance its usability.
* 
* **Disadvantages of using Relational Model**
* Few relational databases have limits on field lengths which can't be exceeded.
* Relational databases can sometimes become complex as the amount of data grows, and the relations between pieces of data become more complicated.
* Complex relational database systems may lead to isolated databases where the information cannot be shared from one system to another.
* 