**Relational Data Model in DBMS: Concepts, Constraints, Example**

**What is 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

In this tutorial, you will learn

* [Relational Model Concepts](https://www.guru99.com/relational-data-model-dbms.html#2)
* [Relational Integrity constraints](https://www.guru99.com/relational-data-model-dbms.html#3)
* [Operations in Relational Model](https://www.guru99.com/relational-data-model-dbms.html#4)
* [Best Practices for creating a Relational Model](https://www.guru99.com/relational-data-model-dbms.html#5)
* [Advantages of using Relational model](https://www.guru99.com/relational-data-model-dbms.html#6)
* [Disadvantages of using Relational model](https://www.guru99.com/relational-data-model-dbms.html#7)

**Relational Model Concepts**

* Relations are basically tables of data
* Each row represents a record in the relation. A relational database is

a set of relations

Collection of Relations: Student is Relation, Course is Relation, Enrollment is a Relation

Student\_Database = {Student, Course, Enrollment}

* Each relation has a unique name in the database
* Each row in the table specifies a relationship between the values in that row

|  |  |  |
| --- | --- | --- |
| Acc\_ID | Branch\_Name | Balance |
| A\_301 | Lahore | 23000 |
| A\_307 | Multan | 34000 |
| A\_318 | Islamabad | 20000 |

*Account relation*

The account ID “A-307”, branch name “Multan”, and balance “34000” are all related to each other

1. **Attribute:** Each column in a Table. Attributes are the properties which define a relation. e.g., Student\_Rollno, NAME,etc.

**Student (**Student\_Rollno, NAME, CGPA….)

1. **Tables** – In the Relational model the, relations are saved in the table format. It is stored along with its entities. A table has two properties rows and columns. Rows represent records and columns represent attributes.

**Student (**Student\_Rollno int, NAME varchar, CGPA double….)

1. **Tuple** – It is nothing but a single row of a table, which contains a single record.
2. **Relation Schema:** A relation schema represents the name of the relation with its attributes.

Student (St\_ID int, St\_Name Varchar, CGPA double)

1. **Degree:** The total number of attributes in a relation is called the degree of the relation.
2. **Cardinality:**Total number of rows present in the Table.

**A = {a, b, c} |A| = 3 A = {b, a, c}**

**Student (St\_ID, St\_Name, CGPA)**

**Student = { <1, A, 2>, <3, C, 4>, <2, B, 3> } |Student| = 3**

**Student = { <1, A, 2>, <2, B, 3>, <3, C, 4>, <4, C, 4> }**

|  |  |  |  |
| --- | --- | --- | --- |
| St\_ID | St\_Name | St\_Age | CGPA |
| 1 | A |  | 2 |
| 2 | B |  | 3 |
| 3 | C |  | 4 |
| 4 | C |  | 4 |

1. **Column:** The column represents the set of values for a specific attribute.
2. **Relation instance** – Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.

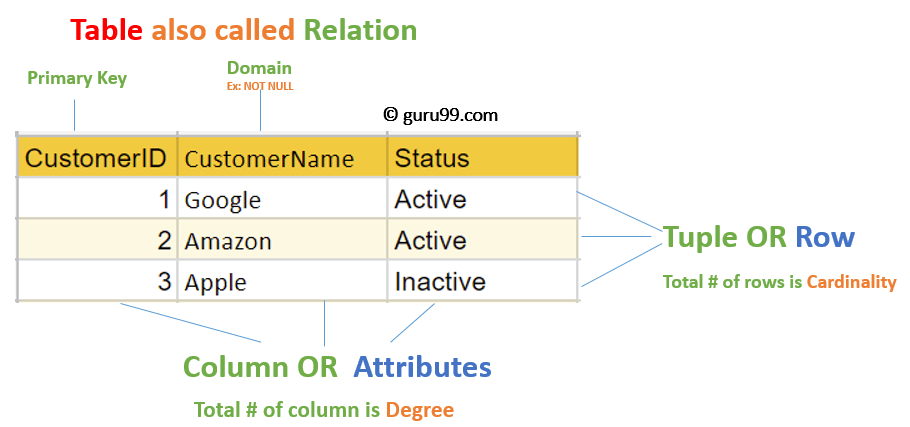
**B = {a, b, c, a, b} is B a set** ? Not a set because repeated values

**C = {1, 2, 3, a, b} is C a set**? Not a set because not well defined

1. **Relation key** - Every row has one, two or multiple attributes, which is called relation key.
2. **Attribute domain** – Every attribute has some pre-defined value and scope which is known as attribute domain

**Y = f(x) x is a domain Y is a range**

**Y = x2+ x + 1**

[](https://www.guru99.com/images/1/091318_0803_RelationalD1.png)

**Relational Integrity constraints**

Relational Integrity constraints is referred to conditions which must be present for a valid relation. These integrity constraints are derived from the rules in the mini-world that the database represents.

There are many types of integrity constraints. Constraints on the Relational database management system is mostly divided into three main categories are:

1. Domain constraints
2. Key constraints
3. Referential integrity constraints

**Domain Constraints**

Domain constraints can be violated if an attribute value is not appearing in the corresponding domain or it is not of the appropriate data type.

Domain constraints specify that within each tuple, and the value of each attribute must be unique. This is specified as data types which include standard data types integers, real numbers, characters, Booleans, variable length strings, etc.

**Example:**

Create DOMAIN CustomerName

CHECK (value not NULL)

The example shown demonstrates creating a domain constraint such that CustomerName is not NULL

**Key constraints**

An attribute that can uniquely identify a tuple in a relation is called the key of the table. The value of the attribute for different tuples in the relation has to be unique.

**Example:**

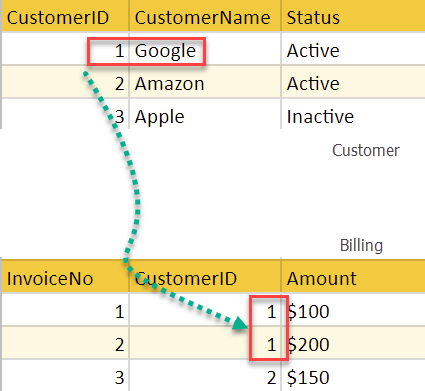
In the given table, CustomerID is a key attribute of Customer Table. It is most likely to have a single key for one customer, CustomerID =1 is only for the CustomerName =" Google".

|  |  |  |
| --- | --- | --- |
| CustomerID | CustomerName | Status |
| 1 | Google | Active |
| 2 | Amazon | Active |
| 3 | Apple | Inactive |
|  |  |  |

**Referential integrity constraints**

Referential integrity constraints is base on the concept of Foreign Keys. A foreign key is an important attribute of a relation which should be referred to in other relationships. Referential integrity constraint state happens where relation refers to a key attribute of a different or same relation. However, that key element must exist in the table.

**Example:**

[](https://www.guru99.com/images/1/091318_0803_RelationalD2.png)

In the above example, we have 2 relations, Customer and Billing.

Tuple for CustomerID =1 is referenced twice in the relation Billing. So we know CustomerName=Google has billing amount $300

**Operations in Relational Model**

Four basic update operations performed on relational database model are

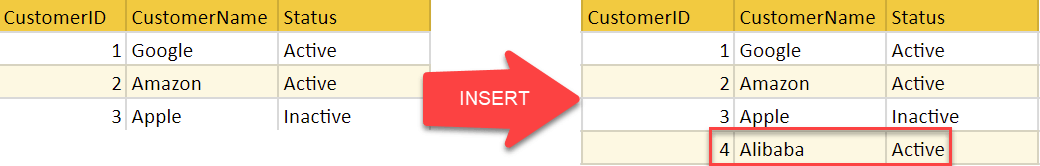
Insert, update, delete and select.

* Insert is used to insert data into the relation
* Delete is used to delete tuples from the table.
* Modify allows you to change the values of some attributes in existing tuples.
* Select allows you to choose a specific range of data.

Whenever one of these operations are applied, integrity constraints specified on the relational database schema must never be violated.

**Insert Operation**

The insert operation gives values of the attribute for a new tuple which should be inserted into a relation.

[](https://www.guru99.com/images/1/091318_0803_RelationalD3.png)

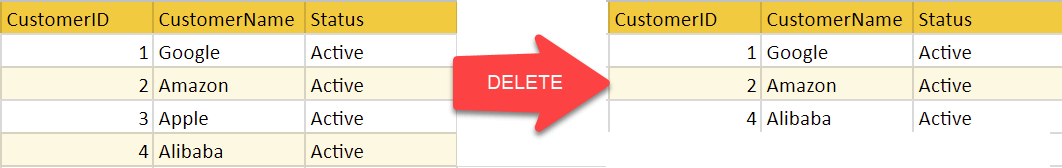
**Update Operation**

You can see that in the below-given relation table CustomerName= 'Apple' is updated from Inactive to Active.

[](https://www.guru99.com/images/1/091318_0803_RelationalD4.png)

**Delete Operation**

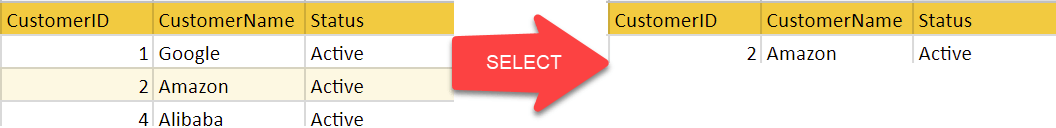
To specify deletion, a condition on the attributes of the relation selects the tuple to be deleted.

[](https://www.guru99.com/images/1/091318_0803_RelationalD5.png)

In the above-given example, CustomerName= "Apple" is deleted from the table.

The Delete operation could violate referential integrity if the tuple which is deleted is referenced by foreign keys from other tuples in the same database.

**Select Operation**

[](https://www.guru99.com/images/1/091318_0803_RelationalD6.png)

In the above-given example, CustomerName="Amazon" is selected

**Best Practices for creating a Relational Model**

* Data need to be represented as a collection of relations
* Each relation should be depicted clearly in the table
* Rows should contain data about instances of an entity
* Columns must contain data about attributes of the entity
* Cells of the table should hold a single value
* Each column should be given a unique name
* No two rows can be identical
* The values of an attribute should be from the same domain

**Advantages of using Relational model**

* **Simplicity**: A relational data model 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 is easy as tables consisting of rows and columns is 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 a 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.

**Summary**

* The Relational database model represents the database as a collection of relations (tables)
* Attribute, Tables, Tuple, Relation Schema, Degree, Cardinality, Column, Relation instance, are some important components of Relational Model
* Relational Integrity constraints are referred to conditions which must be present for a valid relation
* Domain constraints can be violated if an attribute value is not appearing in the corresponding domain or it is not of the appropriate data type
* Insert, Select, Modify and Delete are operations performed in Relational Model
* The relational database is only concerned with data and not with a structure which can improve the performance of the model
* Advantages of relational model is simplicity, structural independence, ease of use, query capability, data independence, scalability.
* Few relational databases have limits on field lengths which can't be exceeded.