

# 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](#)
- [Relational Integrity Constraints](#)
- [Operations in Relational Model](#)
- [Best Practices for creating a Relational Model](#)
- [Advantages of using Relational Model](#)
- [Disadvantages of using Relational Model](#)

## Relational Model Concepts

1. **Attribute:** Each column in a Table. Attributes are the properties which define a relation. e.g., Student\_Rollno, NAME, etc.
2. **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.
3. **Tuple** – It is nothing but a single row of a table, which contains a single record.
4. **Relation Schema:** A relation schema represents the name of the relation with its attributes.
5. **Degree:** The total number of attributes which in the relation is called the degree of the relation.
6. **Cardinality:** Total number of rows present in the Table.
7. **Column:** The column represents the set of values for a specific attribute.
8. **Relation instance** – Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.
9. **Relation key** - Every row has one, two or multiple attributes, which is called relation key.
10. **Attribute domain** – Every attribute has some pre-defined value and scope which is known as attribute domain

### Table also called Relation

© guru99.com

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

**Primary Key** (points to CustomerID)

**Domain** (points to CustomerName, Ex: NOT NULL)

**Tuple OR Row** (points to the rows of the table)

**Total # of rows is Cardinality**

**Column OR Attributes** (points to the columns of the table)

**Total # of column is Degree**

(./images/1/091318\_0803\_RelationalD1.png)

## Relational Integrity Constraints

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

There are many types of Integrity Constraints in DBMS. 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 in DBMS are based 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:**

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

Customer

InvoiceNo	CustomerID	Amount
1	1	\$100
2	1	\$200
3	2	\$150

Billing

(./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.

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



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

(./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.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active




CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Active
4	Alibaba	Active

(./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.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Active
4	Alibaba	Active



CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
4	Alibaba	Active

(./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

CustomerID	CustomerName	Status		CustomerID	CustomerName	Status
1	Google	Active		2	Amazon	Active
2	Amazon	Active				
4	Alibaba	Active				

(./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 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 (<https://www.guru99.com/sql.html>) 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.

## Summary

- The Relational database modelling 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 approach in DBMS
- 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 the operations performed in Relational Model constraints
- 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 in DBMS are simplicity, structural independence, ease of use, query capability, data independence, scalability, etc.
- Few relational databases have limits on field lengths which can't be exceeded.

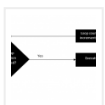
◀ [Prev \(/dbms-schemas.html\)](/dbms-schemas.html)

[Report a Bug](#)

Next ▶ [\(/er-diagram-tutorial-dbms.html\)](/er-diagram-tutorial-dbms.html)


## YOU MIGHT LIKE:

### PL-SQL




[\(/loops-pl-sql.html\)](/loops-pl-sql.html)  
[\(/loops-pl-sql.html\)](/loops-pl-sql.html)

### SQL

[\(/auto-increment.html\)](/auto-increment.html)  
 [\(/auto-increment.html\)](/auto-increment.html)

### SQL


[\(/sql-cheat-sheet.html\)](/sql-cheat-sheet.html)  
 [\(/sql-cheat-sheet.html\)](/sql-cheat-sheet.html)

## Oracle PL/SQL LOOP with Example

(/loops-pl-sql.html)

### SQL

---

(/top-20-sql-management-  
 tools.html) (/top-20-sql-management-  
tools.html)

**20 Best SQL (Database) Management Tools in 2021 [Free/Paid]**


(/top-20-sql-management-  
tools.html)

## MySQL AUTO\_INCREMENT with Examples

(/auto-increment.html)

### SQL

---

(/regular-expressions.html)  
 (/regular-expressions.html)

**MYSQL Regular Expressions (REGEXP) with Syntax & Examples**


(/regular-expressions.html)

## SQL Cheat Sheet [2021 Update]

(/sql-cheat-sheet.html)

### PL-SQL

---

(/oracle-plsql-for-  
 loop.html) (/oracle-plsql-for-loop.html)

**Oracle PL/SQL FOR LOOP with Example**

(/oracle-plsql-for-loop.html)

© Copyright - Guru99 2021

[Privacy Policy \(/privacy-policy.html\)](/privacy-policy.html). | [Affiliate Disclaimer \(/affiliate-earning-disclaimer.html\)](/affiliate-earning-disclaimer.html). | [ToS \(/terms-of-service.html\)](/terms-of-service.html).