SQL - RDBMS Concepts

Table of content

What is RDBMS?

RDBMS stands for **R**elational **D**atabase **M**anagement **S**ystem. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as introduced by E. F. Codd in 1970.

What is a Table?

The data in an RDBMS is stored in database objects known as **tables**. This table is basically a collection of related data entries and it consists of numerous columns and rows.

Remember, a table is the most common and simplest form of data storage in a relational database. Following is an example of a CUSTOMERS table which stores customer's ID, Name, Age, Salary, City and Country –

ID	Name	Age	Salary	City	Country
1	Ramesh	32	2000.00	Hyderabad	India
2	Mukesh	40	5000.00	New York	USA
3	Sumit	45	4500.00	Muscat	Oman
4	Kaushik	25	2500.00	Kolkata	India
5	Hardik	29	3500.00	Bhopal	India
6	Komal	38	3500.00	Saharanpur	India
7	Ayush	25	3500.00	Delhi	India
8	Javed	29	3700.00	Delhi	India

Learn **SQL** in-depth with real-world projects through our **SQL certification course**. Enroll and become a certified expert to boost your career.

What is a Field?

Every table is broken up into smaller entities called fields. A field is a column in a table that is designed to maintain specific information about every record in the table.

For example, our CUSTOMERS table consists of different fields like ID, Name, Age, Salary, City and Country.

What is a Record or a Row?

A record is also called as a row of data is each individual entry that exists in a table. For example, there are 7 records in the above

CUSTOMERS table. Following is a single row of data or record in the CUSTOMERS table –

ID	Name	Age	Salary	City	Country
1	Ramesh	32	2000.00	Hyderabad	India

A record is a horizontal entity in a table.

What is a Column?

A column is a vertical entity in a table that contains all information associated with a specific field in a table.

For example, our CUSTOMERS table have different columns to represent ID, Name, Age, Salary, City and Country.

What is a NULL Value?

A NULL value in a table is a value in a field that appears to be blank, which means a field with a NULL value is a field with no value.

It is very important to understand that a NULL value is different than a zero value or a field that contains spaces. A field with a NULL value is the one that has been left blank during a record creation. Following table has three records where first record has NULL value for the salary and second record has a zero value for the salary.

ID	Name	Age	Salary	City	Country
1	Ramesh	32		Hyderabad	India
2	Mukesh	40	00.00	New York	USA
3	Sumit	45	4500.00	Muscat	Oman

SQL Constraints

Constraints are the rules enforced on data columns on a table. These are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the database.

Constraints can either be column level or table level. Column level constraints are applied only to one column whereas, table level constraints are applied to the entire table.

Following are some of the most commonly used constraints available in SQL –

S.No.	Constraints
1	**NOT NULL Constraint** Ensures that a column cannot have a NULL value.
2	**DEFAULT Constraint** Provides a default value for a column when none is specified.
3	**UNIQUE Key** Ensures that all the values in a column are different.
4	**PRIMARY Key** Uniquely identifies each row/record in a database table.
5	**FOREIGN Key** Uniquely identifies a row/record in any another database table.
6	**CHECK Constraint** Ensures that all values in a column satisfy certain conditions.
7	**INDEX Constraint** Used to create and retrieve data from the database very quickly.

Data Integrity

The following categories of data integrity exist with each RDBMS –

- **Entity Integrity** – This ensures that there are no duplicate rows in a table.

- **Domain Integrity** – Enforces valid entries for a given column by restricting the type, the format, or the range of values.

- **Referential integrity** – Rows cannot be deleted, which are used by other records.

- **User-Defined Integrity** – Enforces some specific business rules that do not fall into entity, domain or referential integrity.

Database Normalization

Database normalization is the process of efficiently organizing data in a database. There are two reasons of this normalization process –

- Eliminating redundant data, for example, storing the same data in more than one table.

- Ensuring data dependencies make sense.

Both these reasons are worthy goals as they reduce the amount of space a database consumes and ensures that data is logically stored. Normalization consists of a series of guidelines that help guide you in creating a good database structure.

Normalization guidelines are divided into normal forms; think of a form as the format or the way a database structure is laid out. The

aim of normal forms is to organize the database structure, so that it complies with the rules of first normal form, then second normal form and finally the third normal form.

It is your choice to take it further and go to the Fourth Normal Form, Fifth Normal Form and so on, but in general, the Third Normal Form is more than enough for a normal Database Application.

- [First Normal Form (1NF)](#)

- [Second Normal Form (2NF)](#)

- [Third Normal Form (3NF)](#)

Print Page

Advertisements