



SQL Server & Database

Compiled By: Prof. Dr. Sher Khan Baloch

Table of Contents

Introduction	2
Data Management	3
Database Entity	5
Database Management System (DBMS):.....	6
Database Models	7
Relational Database Management System (RDBMS).....	9
Create Database and Tables	10
Insert, Update, Delete, Truncate	11
SQL Constraints	12
Select Command	13

Introduction

Database:

A database is a collection of information that is organized so that it can be easily accessed, managed, and updated.

OR

A structured set of data held in a computer, especially one that is accessible in various ways.

Table:

Database table is a tabular format and it is used to store the data of our applications, software, organization, etc.

The table is made of Rows and Columns.

A row represents a record.

A column represents a field.

What Is SQL?

SQL is a standard language for storing, manipulating, and retrieving data in databases.

SQL stands for **Structured Query Language**.

SQL lets you access and manipulate databases.

SQL is an ANSI (American National Standards Institute) Standard.

What Can SQL Do?

SQL can execute queries against a database.

SQL can retrieve, insert, update, and delete data from a database.

SQL can create new databases, new tables, stored procedures, and views in a database.

SQL can set permissions on tables, procedures, and views.

RDBMS:

RDBMS stands for relational database management system.

RDBMS is the basis for SQL and all modern database systems such as MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.

Every table is broken up into smaller entities called fields. The fields in the Customers table consist of customer ID, customer Name, Contact Name, Address, City, postal code, and country.

A field is a column in a table that is designed to maintain specific information about every record in the table.

A record, also called a row, is each entry that exists in a table. For example, there are 91 records in the above customer's table. A record is a horizontal entity in a table.

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

Data Management

Organization often maintain large amount of data, which are generated because of day-to-day operations.

- A Database:
 - Is an organization form of such data.
 - May consist of one or more related data items called records.
 - Is a data collection to which different questions can be asked.
- For Example:
 - What are the phone numbers and addresses of the five nearest post offices?
 - Do we have any books in our library that deal with health food?

Data:

Data is raw, unorganized facts that need to be processed. Data can be something simple and seemingly random and useless until it is organized.

Example: 71000 – No meaning, unorganized.

Information:

When this data is gathered, analyzed, and processed, it yields information.

Example: 71000 postal code of Hyderabad Sindh.

Database:

A database is an organized collection of data such that its contents can be easily accessed, managed, and updated.

Real Life Example:

A phone book is a database consisting of names, addresses, and telephone numbers.

Data Management:

Data management deals with managing large amounts of information, which involves:

- The storage of information
- The provision of mechanisms for the manipulation of information
- Providing safety of information stored under various circumstances

The two different approaches for managing data are as follows:

1. File Based System
2. Database Systems

1. File Based System

In a file-based system, data is stored in discrete files and a collection of such files is stored on a computer.

Rows in the table were called records and columns were called fields.

- Data redundancy and inconsistency
- Data isolation
- Concurrent access anomalies
- Security problems
- Integrity problems

2. Database Systems

Database systems evolved in the late 1960s to address common issues in applications handling large volumes of data, which are also data intensive.

At any point of time, data can be retrieved from the database, added and searched based on some criteria in these databases.

Databases are used to store data in an efficient and organized manner. A database allows quick and easy management of data.

- The amount of redundancy in the stored data can be reduced.
- No more inconsistencies in data
- The stored data can be shared.
- Standards can be set and followed
- Data integrity can be maintained. (No repetition and null values)
- Security of data can be implemented.

Database Entity

- An entity is a person, place, thing, object, event or even a concept, which can be distinctly identified.
- Each entity has certain characteristics known as attributes.
- For example, the student entity might include attributes like student, number, name and grade. Each attribute should be named appropriately.
- For example, the entities in a university are students, faculty members, and courses.
- A grouping of related entities becomes an entity set. Each entity set is given a name. The name of the entity set reflects the contents.

Database Management System (DBMS):

- A DBMS is a collection of related records and a set of program that access, manipulate these records, and enables the user to enter, store, and manage data.
- A database is a collection of interrelated data, and a DBMS is a set of programs used to add or modify the data.
- DBMS supports one of the four database models.

Example:

- Computerized library systems
- Flight reservations systems
- Automated teller machines
- Computerized parts inventory systems

Database Models

Databases can be differentiated based on functions and model of the data.

The analysis and design of data models has been the basis of the evolution of databases.

A data model describes a container for storing data, and the process of storing and retrieving data from that container.

Each model has evolved from the previous one. The commonly used database models are as follows.

1. Flat-File Data Model
2. Hierarchical Data Model
3. Network Data Model
4. Relational Data Model

1. Flat-File Data Model:

In this model, the database consists of only one table or file.

This model is used for simple databases, For example, to store the roll numbers, names, subjects, and marks of a group of students.

This model cannot handle very complex data. It can cause redundancy when data is repeated more than once.

2. Hierarchical Data Model:

In this model, different records are inter-related through hierarchical or tree like structures.

A parent record can have several children, but a child can have only one parent.

To find data stored in this model, the user needs to know the structure of the tree.

3. Network Data Model:

This model was developed to overcome the problems of hierarchical model.

This model is similar to the hierarchical Data Model. It is actually a subset of the network model.

The set theory of the network model does not use a single-parent tree hierarchical. It allows a child to have more than one parents. Thus, the records are physically linked.

Advantages:

Relationships are easier to implement in the network database model than in the hierarchical model.

This model enforces database integrity.

This model achieves sufficient data independence.

Disadvantages:

The databases in this model are difficult to design.

The programmer has to be familiar with the internal structure to access the database.

4. Relational Data Model:

Edgar Frank (EF) Codd (19 August 1923 – 18 April 2003) was an English computer scientist who, while working for IBM, invented the relational model for database management in 1969. In this relational data model, where all data is represented in terms of tuples (rows), grouped into relations (tables).

The term **Relation** is derived from the set theory of mathematics. In the relational model, unlike the hierarchical and network models, there are no physical links.

All data is maintained in the form of tables consisting of rows and columns. Data in two tables is related through common columns and not physical links.

This led to the development of what came to be called the relational model database.

Operators are provided for operating on rows in tables. This model represents the database as a collection of relations.

- A row is called a tuple or record.
- A column is called an attribute.
- The table is called a relation.
- Several attributes can belong to the same domain.

Relational Database Management System (RDBMS)

Relational Model is an attempt to simplify database structures.

A RDBMS is a software program that helps to create, maintain and manipulate a relational database.

A relational database is a database divided into logical units called tables, where tables are related to one another within the database.

Represents all data in the database as simple row-column tables of data values.

Tables are related in a relational database, allowing adequate data to be retrieved in a single query (although the desired data may exist in more than one table). Example: SQL Server, MySQL, Oracle is a RDBMS software.

Create Database and Tables

```
CREATE DATABASE SQLTutorials;
```

```
USE SQLTutorials;
```

```
CREATE TABLE Students_Bio_Data (Std_ID int, Std_Name varchar(50), Father_Name varchar(50), Roll_No int, Class varchar(50));
```

```
SELECT * FROM Students_Bio_Data
```

```
CREATE TABLE Employee (EmpID int, EmpName varchar(50), EmpDpt varchar(50), Salary int)
```

Insert, Update, Delete, Truncate

-- Insert

```
INSERT INTO Students_Bio_Data values (1, 'Sher Khan Baloch', 'Abdul Wahab', 101, 14);
```

```
INSERT INTO Students_Bio_Data values (2, 'Afaque Buledi', 'Abdul Wahab', 102, 13);
```

```
INSERT INTO Students_Bio_Data values (3, 'Fayaz Buledi', 'Abdul Wahab', 103, 10);
```

```
INSERT INTO Students_Bio_Data values (4, 'Faiza', 'Abdul Wahab', 104, 9);
```

```
INSERT INTO Students_Bio_Data (Std_ID, Std_Name) values (5, 'Fiza');
```

-- Update

```
UPDATE Employee set EmpName = 'Sher Khan' where EmpID = 1;
```

```
UPDATE Employee set EmpDpt = 'Faculty' where EmpID = 2;
```

```
UPDATE Employee set Salary = 50000 where EmpID = 3;
```

-- Delete

```
DELETE FROM Employee where EmpID = 1;
```

```
DELETE FROM Employee where EmpName = 'Afaque';
```

-- Truncate

```
TRUNCATE TABLE Employee;
```

```
SELECT * FROM Students_Bio_Data
```

```
SELECT * FROM Employee
```

SQL Constraints

SQL constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in a table. If there is any violation between the constraints and the data action, the action is aborted.

Constraints can be column level or table level. Column-level constraints apply to a column, and table-level constraints apply to the whole table.

The following constraints are commonly used in SQL.

NOT NULL: Ensures that a column cannot have a NULL value.

UNIQUE: Ensures that all values in a column are different.

PRIMARY KEY: A combination of NOT NULL and UNIQUE. Uniquely identifies each row in a table.

FOREIGN KEY: Uniquely identifies a Row/Record in another table.

CHECK: Ensures that all values in a column satisfy a specific condition.

DEFAULT: Sets a default value for a column when no value is specified.

INDEX: Used to create and retrieve data from the database very quickly.

However, you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

```
USE SQLTutorials;
```

```
-- Unique
```

```
CREATE TABLE Employee (EmpID int unique, EmpName varchar(50), EmpDpt varchar(50), Salary int)
```

```
-- Not Null
```

```
CREATE TABLE Employee (EmpID int unique not null, EmpName varchar(50) not null, EmpDpt varchar(50) not null, Salary int not null)
```

```
-- Primary Key
```

```
CREATE TABLE Voter_List (Voter_ID int primary key, Voter_Name varchar(50) not null, Voter_Age int not null);
```

```
-- Check
```

```
CREATE TABLE Voter_List (Voter_ID int primary key, Voter_Name varchar(50) not null, Voter_Age int not null check(Voter_Age >= 18));
```

```
-- Default
```

```
CREATE TABLE Voter_List (Voter_ID int primary key, Voter_Name varchar(50) not null, Voter_Age int not null default(18));
```

```
SELECT * FROM Voter_List;
```

```
INSERT INTO Voter_List VALUES (1, 'Afaq', 21);
```

Select Command

Use SQLTutorials;

-- Select

```
SELECT * FROM Voter_List;
```

```
SELECT * FROM Voter_List WHERE Voter_ID = 4 OR Voter_ID = 6;
```

```
SELECT * FROM Voter_List ORDER BY Voter_Name ASC;
```

```
SELECT * FROM Voter_List ORDER BY Voter_Name DESC;
```

```
SELECT Voter_Name, Voter_Age FROM Voter_List;
```

```
SELECT Voter_Name, Voter_Age FROM Voter_List WHERE Voter_ID = 4 OR Voter_ID = 6;
```