**SQL WORKSHOP**

**What is Data?**

Data is a collection of a distinct small unit of information. It can be used in a variety of forms like text, numbers, media, bytes, etc. it can be stored in pieces of paper or electronic memory, etc.

Word 'Data' is originated from the word 'datum' that means 'single piece of information.' It is plural of the word datum.

In computing, Data is information that can be translated into a form for efficient movement and processing. Data is interchangeable.

**What is Database?**

A **database** is an organized collection of data, so that it can be easily accessed and managed.

You can organize data into tables, rows, columns, it to make it easier to find relevant information.

The **main purpose** of the database is to operate a large amount of information by storing, retrieving, and managing data.

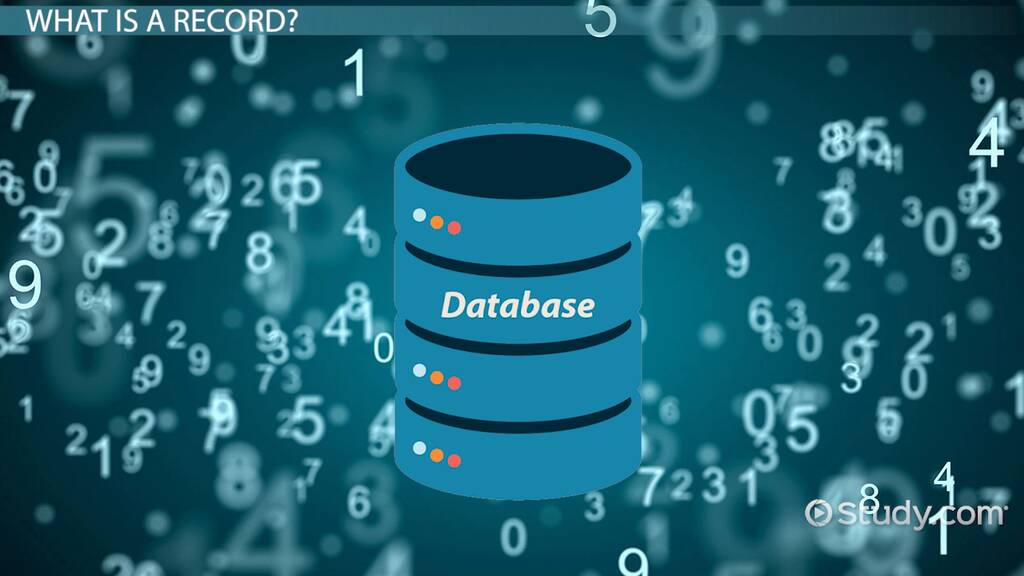
There are many **dynamic websites** on the World Wide Web nowadays which are handled through databases. For example, a model that checks the availability of rooms in a hotel. It is an example of a dynamic website that uses a database.

There are many **databases available** like SQL Server, MySQL, Sybase, Oracle, MongoDB, Informix, PostgreSQL, etc.

Modern databases are managed by the database management system (DBMS).

**SQL** (Structured Query Language) is used to operate on the data stored in a database. SQL depends on relational algebra and tuple relational calculus.

A cylindrical structure is used to display the image of a database.



**RDBMS (Relational Database Management System)**

The word RDBMS is termed as 'Relational Database Management System.' It is represented as a table that contains rows and column.

RDBMS is based on the Relational model; it was introduced by E. F. Codd.

* Table
* Record/ Tuple
* Field/Column name /Attribute

**Introduction to SQL:**

* SQL stands for Structured Query Language
* SQL lets you access and manipulate databases
* SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987.

**Create Database Query**

Create Database db\_name;

Use db\_name;

**First of Fall Create table and insert some data in database**

# SQL INSERT INTO Statement

The **INSERT INTO** statement is used to insert new records in a table.

It is possible to write the INSERT INTO statement in two ways:

**First**

INSERT INTO *table\_name* (*column1*,*column2*,*column3*, ...)  
VALUES (*value1*,*value2*,*value3*, ...);

**Second**

INSERT INTO *table\_name* VALUES (*value1*,*value2*,*value3*, ...);

**SQL Select Statement**

**Example:** SELECT *column1*,*column2, ...*FROM *table\_name*;

All Columns Select in table

SELECT \* FROM *table\_name*;

Here \* (asterisk) denoted all columns

## The SQL SELECT DISTINCT Statement

The SELECT DISTINCT statement is used to return only distinct (different) values.

Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values

### **SELECT DISTINCT Syntax**

SELECT DISTINCT *column1*,*column2, ...*FROM *table\_name*;

**Where Clauses**

The WHERE clause is used to filter records.

**Syntax:**

SELECT *column1*,*column2,*FROM *table\_name* WHERE *condition*;

Example:

SELECT \* FROM Employee WHERE gender=’male’;

Here = is the conditional operator. Here below we have a more operator

|  |  |
| --- | --- |
| = | Equal |
| > | Greater than |
| < | Less than |
| >= | Greater than or equal |
| <= | Less than or equal |
| != | Not equal |
| BETWEEN | Between a certain range with AND |
| LIKE | Search for a pattern |
| IN | To specify multiple possible values for a column it is shorthand of or operator |

**Between Operator:**

The BETWEEN operator selects values within a given range. The values can be numbers, text, or dates.

The BETWEEN operator is inclusive: begin and end values are included.

**Syntax:**

SELECT *column\_name(s)*FROM *table\_name* WHERE *column\_name*BETWEEN *value1* AND *value2;*

**Example:**

SELECT \* FROM Products WHERE id BETWEEN 5 AND 10;

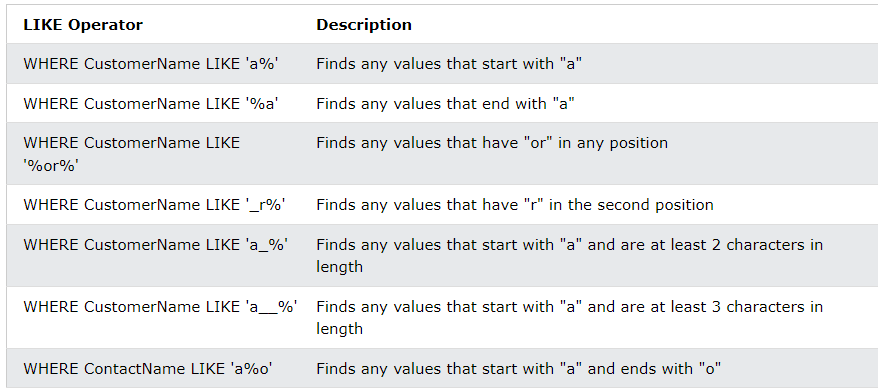
AND

OR

**Like Operator**

**Syntax:**

SELECT *column1,column2,...*FROM *table\_name* WHERE *columnN* LIKE *pattern*;

****

**In Operator**

The IN operator allows you to specify multiple values in a WHERE clause.

The IN operator is a shorthand for multiple OR conditions.

**Syntax:**

SELECT *column\_name(s)*FROM *table\_name* WHERE *column\_name* IN (*value1*,*value2*, ...);

**Example**

SELECT \* FROM Customers WHERE Country IN ('pakistan', 'india', 'UK');

## The SQL AND, OR and NOT Operators

The WHERE clause can be combined with AND, OR, and NOT operators.

The **AND** and **OR** operators are used to filter records based on more than one condition:

* The AND operator displays a record if all the conditions separated by AND are TRUE.
* The OR operator displays a record if any of the conditions separated by OR is TRUE.

The NOT operator displays a record if the condition(s) is NOT TRUE.

## AND Example

SELECT \* FROM Customers WHERE Country='Germany' AND City='Berlin';

## OR Example

SELECT \* FROM Customers WHERE City='Berlin' OR City='München';

## NOT Example

SELECT \* FROM Customers WHERE NOT Country='Germany';

## Combining AND, OR and NOT

**Example:**

SELECT \* FROM Customers  
WHERE Country='Germany' AND (City='Berlin' OR City='München');

**Another Example:**

SELECT \* FROM Customers  
WHERE NOT Country='Germany' AND NOT Country='USA';

# SQL ORDER BY Keyword

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

**Syntax**

SELECT *column1*,*column2, ...* FROM *table\_name* ORDER BY *column1, column2, ...*ASC|DESC;

**Example:**

SELECT \* FROM Customers ORDER BY Country;

**Another Example**

SELECT \* FROM Customers ORDER BY Country ASC;

**Order by DESC (Descending)**

**Example:**

SELECT \* FROM Customers ORDER BY Country DESC;

# Aggregate Functions:

# SQL MIN(), MAX(), Count , avg,sum Functions

**Min()**

SELECT MIN(*column\_name*) FROM *table\_name*;

**Example:**

Select min(price) from product;

**Max()**

SELECT MAX(Price) FROM product;

### **COUNT()**

SELECT COUNT(P\_ID) FROM products;

**AVG()**

SELECT AVG(Price) FROM Products;

**SUM()**

SELECT SUM(price) FROM products;

**SQL Update Statement:**

The UPDATE statement is used to modify the existing records in a table.

**Syntax**

UPDATE *table\_name* SET *column1*=*value1*,*column2*=*value2*, ... WHERE *condition*;

**Example:**

UPDATE Customers  
SET ContactName = 'Arif Alvi', City= 'Frankfurt'  
WHERE CustomerID = 1;

Here where clause is mandatory if you will not write where condition so all data will have updated.

**SQL DELETE STATEMENT**

The DELETE statement is used to delete existing records in a table.

**Syntax:**

DELETE FROM *table\_name*WHERE *condition*;

Here where clause is mandatory if you will not write where condition so all data will have deleted.

**Example**

DELETE FROM Customers WHERE CustomerName='ali';

**Delete All Records:**

**Example:**

DELETE FROM Customers;

## LIMIT Clauses

**Syntax:**

SELECT \* FROM Customers LIMIT 3;

Here 3 denoted three first rows.

And other syntax with two parameter

**Syntax:**

SELECT \* FROM Customers LIMIT 2,6;

Here 2 denoted where you start. It is also called offset.

Here 6 is denoted my limit how many rows you want