DBMS Lab Assignment 1

Objective:

To practice and write SQL queries using Data Definition Language (DDL) and Data Manipulation Language (DML) commands.

Theory and Concepts:

What is SQL?

- SQL stands for Structured Query Language. It is used for storing and managing data in the Relational Database Management System (RDBMS).
- It is a standard language for Relational Database Systems. It enables a user to create, read, update and delete relational databases and tables.
- All the RDBMS like MySQL, Informix, Oracle, MS Access and SQL Server use SQL as their standard database language.
- SQL allows users to query the database in a number of ways, using English-like statements.

SQL follows the following rules:

- SQL is not case-sensitive, but it is common practice to write keywords in uppercase for better readability.
- SQL statements are independent of text lines; a single SQL statement can span one or multiple lines.
- With SQL statements, you can perform various operations in a database, such as creating, reading, updating, and deleting data.
- SQL is based on concepts from tuple relational calculus and relational algebra, providing a foundation for manipulating relational data.

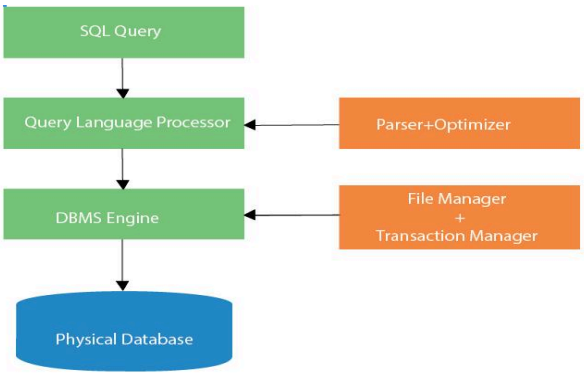
What is SQL Process?

When an SQL command is executed in any Relational Database Management System (RDBMS), the system processes the request to find the most efficient way to perform the task. The **SQL engine** plays a key role in determining how to interpret and execute the query.

Components Involved in the SQL Process:

- Query Dispatcher: Directs the query to the appropriate engine.
- Optimization Engine: Determines the most efficient execution plan for the query.
- Query Engine: Responsible for the actual execution of the SQL query.
- Classic Query Engine: Handles non-SQL queries; does not handle logical file operations.

- SQL Query Engine: Dedicated to processing SQL queries and manages tasks such as parsing, optimizing, and executing SQL statements.

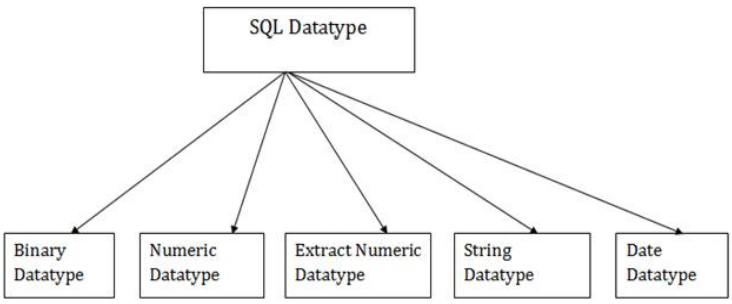


Advantages of SQL:

- High Speed: SQL queries are optimized for performance, allowing for fast data retrieval and manipulation.
- No Coding Needed: SQL uses declarative language, meaning users specify what they want without having to write detailed procedural code.
- Well-Defined Standards: SQL follows standardized rules and syntax, ensuring consistency across different RDBMS systems.
- Portability: SQL commands are largely portable across various database systems, making it easier to switch or integrate different systems.
- Interactive Language: SQL allows for interactive data querying and manipulation, making it suitable for real-time data access and analysis.
- Multiple Data Views: SQL supports creating various views and abstractions of data, allowing users to see and interact with data in different ways.

What is SQL Data Type?

- SQL Datatype is used to define the values that a column can contain.
- Every column is required to have a name and data type in the database table.



Commonly used Data Types

- INT: An integer data type used to store whole numbers.
- VARCHAR(n): A variable-length string data type that can store up to `n` characters.
- CHAR(n): A fixed-length string data type that always stores exactly `n` characters, padding with spaces if necessary.
- TEXT: A variable-length string data type used to store large amounts of text.
- DATE: A data type used to store calendar dates (year, month, and day).
- TIME: A data type used to store time of day (hours, minutes, seconds).
- DATETIME: A data type used to store both date and time information.
- TIMESTAMP: A data type used to store a combination of date and time, typically with timezone information.
- FLOAT: A floating-point number data type used to store approximate numeric values with fractional components.
- DECIMAL(p, s): A fixed-point number data type with precision `p` (total number of digits) and scale `s` (number of digits to the right of the decimal point).
- BOOLEAN: A data type used to store true or false values.

Constraints

- Specify rules for data in a table.
- Ensures the accuracy and reliability of data in the table.
- Constraints can be specified when the table is created with the CREATE TABLE statement, or after the table is created with the ALTER TABLE statement

Commonly Used Constraints

- **PRIMARY KEY:** Ensures that each value in a column or a group of columns is unique and not null.
- **FOREIGN KEY:** Enforces a link between the data in two tables to maintain referential integrity.
- **UNIQUE:** Ensures all values in a column or a group of columns are distinct across the table.
- **NOT NULL:** Ensures that a column cannot have a null value.
- **CHECK:** Ensures that all the values are limited to the range that we have placed in a column..
- **DEFAULT:** Sets a default value for a column when no value is specified during insertion.

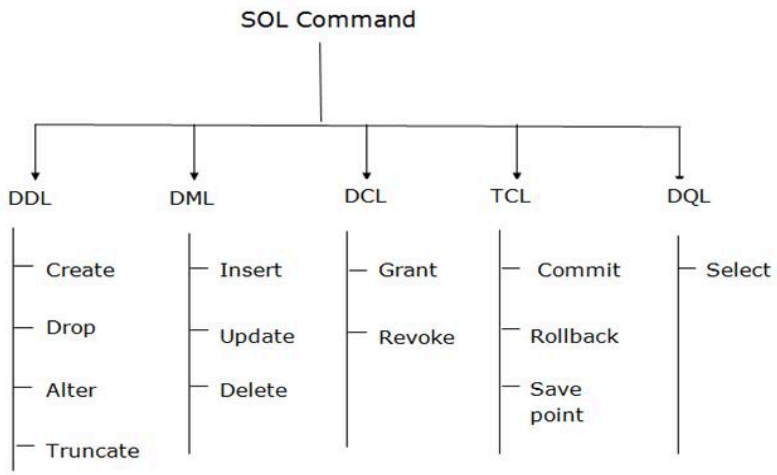
SQL Commands

- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.

- SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

TYPES OF SQL COMMANDS

There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.



Data Definition Language (DDL)

1. DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
2. All the commands of DDL are auto-committed, which means it permanently saves all the changes in the database.
3. Here are some commands that come under DDL:
    i. CREATE
    ii. ALTER
    iii. DROP
    iv. TRUNCATE

Data Definition Language (DDL)-

CREATE: It is used to create a new table in the database.

Syntax:

```
CREATE TABLE table_name (
   column1 datatype constraint,
   column2 datatype constraint,
   column3 datatype constraint,
```

);

Example:

```
CREATE TABLE classroom (

Rollno  INT PRIMARY KEY,

Name VARCHAR(50) NOT NULL,

House CHAR(12) NOT NULL,

Grade CHAR(1) );
```

Data Definition Language (DDL)-

DROP: It is used to delete both the structure and record stored in the table.

Syntax:

```
DROP TABLE table_name;
```

Example:

```
DROP TABLE classroom;
```

Data Definition Language (DDL)-

TRUNCATE: It is used to delete all the rows from the table and free the space containing the table.

Syntax:

```
TRUNCATE TABLE table_name;
```

Example:

```
TRUNCATE TABLE EMPLOYEE;
```

Data Manipulation Language(DML)

DML commands are used to manage and manipulate the data stored in a database. Unlike Data Definition Language (DDL) commands, which affect the structure of the database, DML commands are focused on modifying the data within the tables.

Characteristics:

- Non-Auto-committed: Changes made by DML commands are not automatically saved. They can be rolled back if needed.
- Transactional Control: You can use transactions to group multiple DML operations into a single unit of work. This allows you to commit or roll back changes as needed.

Data Manipulation Language(DML) -

INSERT: The INSERT statement is a SQL query. It is used to insert data into the row of a table.

Syntax 1:

INSERT INTO <Table_Name>(attr1, attr2, attr3) values(att_vl1, attr_val2, att_vl3);

INSERT INTO table_name
(column1, column2, ...)
VALUES (value1, value2, ...);

Dept_id	Dept_Name	Location
1	CSE	Hyderabad
2	ECE	KGP
3	CIVIL	Jaipur

Example:

INSERT INTO Departments (Dept_id, Dept_Name, Location)
VALUES (1, 'CSE', 'Hyderabad'), (2, 'ECE', 'KGP'), (3, 'CIVIL', 'Jaipur');

Syntax 2:

INSERT INTO <Table_Name> values(att_vl1, attr_val2…);

INSERT INTO table_name
VALUES (value1, value2, ...);

Data Query Language

DQL is used to retrieve or fetch data from the database. It consists of the SELECT statement, which is used to query and extract data based on certain conditions.

Command: SELECT

- The SELECT statement is similar to the projection operation in relational algebra. It allows you to choose specific columns (attributes) from a table and filter rows based on conditions.

The SELECT clause is used to retrieve data from a database:

1. **Column**: SELECT column1, column2 FROM table; retrieves specific columns.
2. **All Columns:** SELECT * FROM table; retrieves all columns from a table.
3. **Distinct**: SELECT DISTINCT column FROM table; retrieves unique values from the specified column.

Clause

The WHERE clause is essential for filtering data based on specified conditions and returning it in the result set. It is commonly used in SELECT, INSERT, UPDATE, and DELETE statements to work on specific data. This clause follows the FROM clause in a SELECT statement and precedes any ORDER BY or GROUP BY clauses.

Within the WHERE clause, you can specify one or more conditions that the data must meet to be included in the result. Conditions can involve comparisons (e.g., equal to, not equal to, greater than, less than), logical operators (AND, OR), and other expressions.

WHERE Clause
The WHERE clause is used to specify a condition while fetching data from a single table or by joining multiple tables. Only the records that meet the specified condition are retrieved.

Syntax:

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

Parameters:

- SELECT: Specifies the columns to be retrieved.
- column1, column2, ...: List of specific columns to retrieve. Use * to select all columns.
- table_name: Name of the table from which the data is selected.
- WHERE: Filters records based on specific conditions.

Data Manipulation Language(DML) -

Update: This command is used to update or modify the value of a column in the table.
Syntax 1:

```
UPDATE <Table_name> SET <attribute> = <value>
WHERE <Condition>

UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

Example:

```
UPDATE Dept SET dept_name = 'IT'
WHERE location = 'Jaipur' ;
```

Data Manipulation Language(DML) -

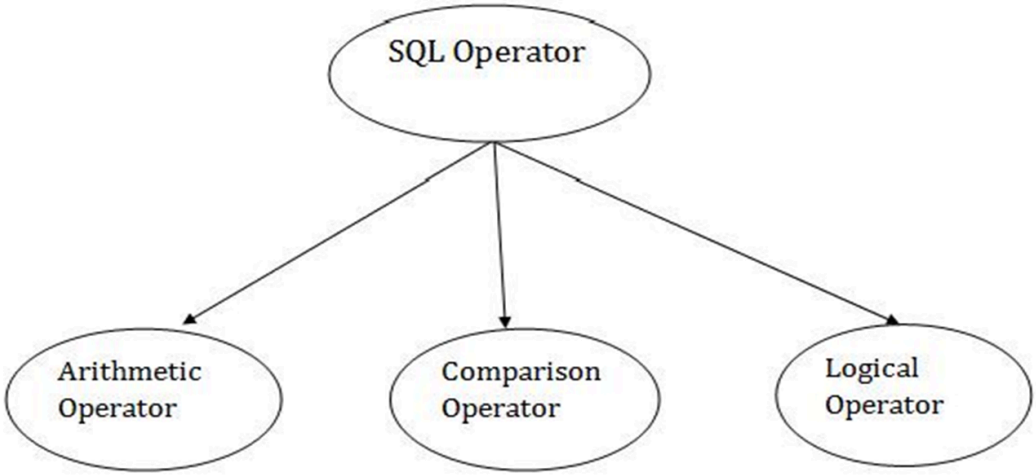
Delete: This command is used to remove rows from a table.

Syntax 1:

```
DELETE FROM table_name
WHERE condition;
```

SQL Operators

SQL operators are used to specify conditions in SQL statements and to perform operations on data in tables.



1. Arithmetic Operators

- Used for mathematical calculations.

Operator	Description	Example
+	Addition	SELECT 5 + 3;
-	Subtraction	SELECT 5 - 3;
*	Multiplication	SELECT 5 * 3;
/	Division	SELECT 6 / 3;

%	Modulus (Remainder)	SELECT 5 % 3;

---

2. Comparison Operators

- Used to compare values.

Operator	**Description**	**Example**
=	Equal to	SELECT * FROM table WHERE age = 30;
!= or <>	Not equal to	SELECT * FROM table WHERE age != 30;
>	Greater than	SELECT * FROM table WHERE age > 30;
<	Less than	SELECT * FROM table WHERE age < 30;
>=	Greater than or equal to	SELECT * FROM table WHERE age >= 30;
<=	Less than or equal to	SELECT * FROM table WHERE age <= 30;
BETWEEN	Between an inclusive range	SELECT * FROM table WHERE age BETWEEN 20 AND 30;
IN	Match any value in a set	SELECT * FROM table WHERE age IN (20, 30, 40);

---

3. Logical Operators

- Used to combine multiple conditions.

Operator	Description	Example
AND	Returns true if all conditions are true	SELECT * FROM table WHERE age > 20 AND salary > 50000;
OR	Returns true if at least one condition is true	SELECT * FROM table WHERE age > 30 OR salary > 50000;
NOT	Returns true if the condition is false	SELECT * FROM table WHERE NOT age > 30;

SQL Comments explain sections of SQL statements or prevent SQL statements from being executed.
There are 3 **types of comments** in SQL**:**

1. **Single-line comments**
2. **Multi-line comments**
3. **In-line comments**

SQL Single Line Comments
SQL Single Line Comments contain a single line comment. They start and end in a single line.

Single Line comments in SQL can be inserted using '--' before the line.

SQL Multi-Line Comments
SQL Multi-line comments contain multiple lines in a single comment. They start from one line and end in a different line.

A multi-line comment starts with '/*' and is terminated when '*/' is encountered.

SQL In-Line Comments
SQL In-line comments are an extension of multi-line comments, these comments are used in between of a SQL statement.

An In-Line comment starts with '/*' and end with '*/'.

Assignment 1 Questions

Objective:

a. Create the World table and insert the provided data.
b. Execute all the queries and verify the output.
c. Drop the table after completing the queries

1. Write the SQL statement to create the World table with the following schema:
    i. name (VARCHAR, primary key)
    ii. continent (VARCHAR)
    iii. area (INT)
    iv. population (INT)
    v. gdp (BIGINT)
2. Insert the following data into the World table:
    i. ('Afghanistan', 'Asia', 652230, 25500100, 20343000000)
    ii. ('Albania', 'Europe', 28748, 2831741, 12960000000)
    iii. ('Algeria', 'Africa', 2381741, 37100000, 188681000000)
    iv. ('Andorra', 'Europe', 468, 78115, 3712000000)
    v. ('Angola', 'Africa', 1246700, 20609294, 100990000000)
3. Write a query to retrieve all the data from the World table.
4. Write a query to retrieve only the name, population, and area columns from the World table.
5. Write a query to retrieve the distinct continents present in the World table.
6. Write a query to find the name, population, and area of all countries that are either:
    i. Bigger than or equal to 3,000,000 km² or
    ii. Have a population greater than or equal to 25,000,000.
7. Write a query to retrieve all countries in the `World` table that are located in the continent 'Africa'.
8. Write a query to retrieve the `name` and `population` of countries that have either:
    a. a population greater than 10,000,000 **OR**

b. a GDP greater than 100,000,000,000.

9. Write a query to retrieve the `name`, `area`, and `population` of countries that are in the continent 'Europe' **AND** have an area greater than 10,000 km².
10. Write a query to retrieve the `name` and `population` of countries that are **NOT** located in 'Europe'.
11. Write a query to retrieve the `name`, `area`, and `gdp` of countries where the `area` is less than 1,000,000 km².
12. Write a query to calculate the GDP per capita (GDP/population) for each country and display the `name` and the calculated GDP per capita.
13. Write a query to retrieve all countries where the `gdp` per capita (GDP/population) is greater than 5,000.
14. Write a query to retrieve all countries that are in 'Asia' **AND** have a population greater than 20,000,000 **OR** have an area greater than 500,000 km².
15. Write a query to update the population of 'Angola' to 21,000,000.
16. Write a query to delete the row for 'Andorra' from the World table.
17. Write a query to drop the World table.

Company Related Questions and Platform

Question 1 : LINK

Query a list of CITY names from STATION for cities that have an even ID number. Print the results in any order, but exclude duplicates from the answer.

Question 2 : LINK

Find the difference between the total number of CITY entries in the table and the number of distinct CITY entries in the table.

Question 3 : LINK

Write a solution to find the ids of products that are both low fat and recyclable. Return the result table in any order.

Question 4 : LINK

Find the names of the customer that are not referred by the customer with id = 2. Return the result table in any order.