Select command in SQL

- The SELECT command is used in SQL to retrieve data from one or more tables in a database.

Syntax of select command

- The basic syntax for the SELECT command is as follows:

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

- The columns that you want to retrieve are listed after the SELECT keyword and separated by commas. The table or tables from which you want to retrieve the data are specified after the FROM keyword. You can also include a WHERE clause to filter the results based on specific conditions.

Examples of select command

- Here's an example of a SELECT command that retrieves all columns from a table named "customers":

  ```
  SELECT *
  FROM customers;
  ```

- And here's an example of a SELECT command that retrieves only the customer name and email address columns from the same "customers" table, where the country is "USA":

  ```
  SELECT customer_name, email_address
  FROM customers
  WHERE country = 'USA';
  ```

- Note that the values for string columns should be enclosed in single quotes in the WHERE clause.

Additional Options in select command

- Here are some additional details and options you can use with the SELECT command in SQL:

1. **FROM Keyword :** To specify which table to retrieve data from, you can use the table name after the **FROM** keyword. If you want to retrieve data from multiple tables, you can separate the table names with commas. For example, SELECT * FROM table1, table2;

2. **WHERE Clause :** To filter the results based on specific conditions, you can use the WHERE clause. The WHERE clause is followed by a Boolean expression that evaluates to true or false for each

row in the table. Only rows that evaluate to true will be returned. For example, SELECT * FROM orders WHERE order_date > '2022-01-01';

3. **Comparison Operators :** You can use comparison operators such as =, <, >, <=, and >= in the WHERE clause to compare column values. You can also use logical operators such as AND, OR, and NOT to combine multiple conditions. For example, SELECT * FROM customers WHERE country = 'USA' AND city = 'New York';

4. **DISTINCT Keyword :** If you want to retrieve only distinct values from a column, you can use the DISTINCT keyword after the SELECT keyword. For example, SELECT DISTINCT country FROM customers;

5. **ORDER BY Clause :** You can use the ORDER BY clause to sort the results based on one or more columns. The ORDER BY clause is followed by the column name(s) and the sorting direction (ASC for ascending, DESC for descending). For example, SELECT * FROM products ORDER BY price DESC;

6. **Aggregate Functions :** You can use aggregate functions such as COUNT, SUM, AVG, MIN, and MAX to perform calculations on columns. These functions can be used in combination with the SELECT statement to retrieve summarized data. For example, SELECT COUNT(*) FROM orders WHERE status = 'completed';

7. **LIKE Keyword :** The LIKE keyword is used in SQL to search for a specified pattern in a column. The pattern can include wildcards, which represent any character or a group of characters.

   For example, suppose you have a table named "customers" with a column named "customer_name", and you want to retrieve all the customers whose names start with "J". You can use the following query:
   SELECT customer_name
   FROM customers
   WHERE customer_name LIKE 'J%';

   The **%** wildcard represents any sequence of zero or more characters, so the query will retrieve all the customer names that start with "J".

   Similarly, suppose you want to retrieve all the customers whose names have exactly five characters and end with "son". You can use the following query:
   SELECT customer_name
   FROM customers

WHERE customer_name LIKE '____son';

The _ wildcard represents any single character, so the query will retrieve all the customer names that have four characters and end with "son".

Note that the LIKE keyword is case-sensitive by default.