(8) What is the significance of “%” and “_” operators in the LIKE statement?

In MySQL, LIKE operator, the "%" and "_" symbols act as wildcards, allowing for pattern matching in strings. The % represents zero or more characters, while the _ represents a single character.

Explanation:

- % (Percent):

This wildcard signifies any sequence of characters, including an empty sequence (zero characters). It's used to represent any number of characters at a specific position within the search string. For example, 'A%' would match strings like "A", "Apple", "Apple Pie", and "A_".

- _ (Underscore):

This wildcard stands for a single character. It's used to represent exactly one character at a particular location. For instance, 'A_' would match strings like "Ab", "Az", and "Ah" but not "App" or "A".

(9) Explain normalization in the context of databases.

In MySQL and other database management systems, database normalization is a process of organizing data to minimize redundancy and improve data integrity. It involves breaking down large, complex tables into smaller, more manageable tables and establishing relationships between them. This process helps prevent data anomalies (problems like insertion, update, and deletion anomalies) and ensures data consistency.

Normalization is typically achieved through a series of stages called normal forms. The most common ones are:

1NF (First Normal Form):

Each column in a table should contain only atomic (indivisible) values. This means no repeating groups of columns or data.

2NF (Second Normal Form):

A table is in 2NF if it is in 1NF and all non-key attributes are fully dependent on the entire primary key.

3NF (Third Normal Form):

A table is in 3NF if it is in 2NF and all non-key attributes are not transitively dependent on the primary key.

Boyce-Codd Normal Form (BCNF): A stricter version of 3NF, where every determinant is a

candidate key.

10) What does a join in MySQL mean?

A JOIN in MySQL is used to combine rows from two or more tables based on a related column between them — usually a primary key in one table and a foreign key in another.

It allows you to pull information that is spread across multiple tables into a single result set.

There are different types of joins, such as:

- INNER JOIN: Returns only the rows that have matching values in both tables.
- LEFT JOIN (LEFT OUTER JOIN): Returns all rows from the leŌ table, and matching rows

  from the right table. If no match exists, NULL values are returned for the right table's

  columns.

- RIGHT JOIN (RIGHT OUTER JOIN): Similar to LEFT JOIN but returns all rows from the right

  table, with NULLs for unmatched rows in the leŌ table.

- FULL OUTER JOIN: Returns all rows when there is a match in either the leŌ or right table.

11) What do you understand about DDL, DCL, and DML in MySQL?

In MySQL (and all SQL-based systems), SQL commands are categorized based on what they do.

The three major categories are:

(1) DDL (Data Definition Language) : DDL commands deal with the structure of the database:

(creating, altering, dropping tables, schemas, indexes, etc.)

Common DDL Commands:

CREATE - Creates a new table, database, index, or view.

ALTER - Modifies an existing database object (like a table).

DROP - Deletes a table, database, index, or view permanently.

TRUNCATE - Removes all records from a table quickly, but keeps the table itself.

RENAME - Renames a database object like a table.

(2) DCL (Data Control Language) : DML commands deal with data inside tables — inserting, updating, deleting, and querying.

Common DML Commands:

SELECT - Retrieves data from one or more tables.

INSERT - Adds new records into a table.

UPDATE - Modifies existing data in a table.

DELETE - Removes records from a table.

(3) DML (Data Manipulation Language) : DCL commands are used to control permissions and access to the database and its objects.

Common DCL Commands:

GRANT - Gives users privileges to perform certain tasks (SELECT, INSERT, UPDATE, etc.).

REVOKE - Removes privileges that were previously given.

12) What is the role of the MySQL JOIN clause in a query, and what are some common types of joins?

A JOIN clause in MySQL is used to combine rows from two or more tables based on a related column between them (like a primary key and a foreign key).

It helps you fetch data together from multiple tables as if it were one table.

The JOIN condition matches rows based on specified relationships between columns.

In simple words:

JOIN connects different tables to pull related information in a single query.

INNER JOIN :

INNER JOINs are used to fetch only common matching records. The INNER JOIN clause allows retrieving only those records from Table A and Table B that meet the join condition. It is the most widely used type of JOIN.

```
SELECT Customers.Name, Orders.Amount

FROM Customers

INNER JOIN Orders

ON Customers.CustomerID = Orders.CustomerID;
```

LEFT JOIN (or LEFT OUTER JOIN) :

LEFT JOINs allow retrieving all records from Table A along with those records from Table B for which the join condition is met. For the records from Table A that do not match the condition, the NULL values are displayed.

```
SELECT Customers.Name, Orders.Amount

FROM Customers

LEFT JOIN Orders

ON Customers.CustomerID = Orders.CustomerID;
```

RIGHT JOIN (or RIGHT OUTER JOIN) :

A RIGHT JOIN is similar to a LEFT JOIN, but it returns all records from Table B along with those records from Table A for which the join condition is met. For the records from Table B that do not match the condition, the NULL values are displayed.

```
SELECT Customers.Name, Orders.Amount

FROM Customers

RIGHT JOIN Orders

ON Customers.CustomerID = Orders.CustomerID;
```

CROSS JOIN :

MySQL CROSS JOIN, also known as a cartesian join, retrieves all combinations of rows from each table. In this type of JOIN, the result set is returned by multiplying each row of table A with all rows in table B if no additional condition is introduced.

```
SELECT Customers.Name, Orders.Amount

FROM Customers

CROSS JOIN Orders;
```