MySQL

Theory question of Mysql

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

=> SQL "LIKE" statement, the "%" symbol represents zero or more characters, while the "_" (underscore) represents a single character, allowing you to search for patterns within data using wildcard matching; essentially, you can use them to find rows where a column value partially matches a specified pattern.

"%":

Can match any number of characters (including none) at any position within a string.

"_":

Only matches a single character at the position where it is placed.

9) Explain normalization in the context of databases ?

Normalization is the process of organizing the data in the database. Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies. Normalization divides the larger table into smaller and links them using relationships.

Data normalization ensures that your data remains clean, consistent, and error-free by breaking it into smaller tables and linking them through relationships. This process reduces redundancy, improves data integrity, and optimizes database performance.

Advantages of Normalization

- Elimination of Data Redundancy: Reduces the repetition of data, improving efficiency and consistency.

- Ensuring Data Consistency: Prevents inconsistencies and contradictions by eliminating redundancy.
- Simplification of Data Management: Makes it easier to manage, update, and retrieve data by breaking down complex structures into simpler tables.
- Improved Database Design: Organizes data in a structured way, making the database more flexible and adaptable.
- Avoiding Update Anomalies: Ensures that each table contains only one type of data, avoiding anomalies when updating records.
- Standardization: Ensures that data is stored in a consistent and uniform manner.

10) What does a join in MySQL mean?

In a DBMS, a join is an operation that combines rows from two or more tables based on a related column between them.

Joins are used to retrieve data from multiple tables by linking them together using a common key or column.

Types of Joins:

1. Inner Join
2. left join
3. Right Join

* Inner Join

An inner join combines data from two or more tables based on a specified condition, known as the join condition.

The result of an inner join includes only the rows where the join condition is met in all participating tables.

It essentially filters out non-matching rows and returns only the rows that have matching values in both tables.

Syntax:

```
SELECT columns
FROM table1
INNER JOIN table2
ON table1.column = table2.column
```

. Left Join (Left Join):

A left outer join returns all the rows from the left table and the matching rows from the right

table.

If there is no match in the right table, the result will still include the left table's row with NULL values in the right table's columns.

Example:

```
SELECT Customers.CustomerName, Orders.Product

FROM Customers

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

Right Join (Right Join):

A right outer join is similar to a left outer join, but it returns all rows from the right table and

the matching rows from the left table.

If there is no match in the left table, the result will still include the right table's row with NULL

values in the left table's columns.

Example: Using the same Customers and Orders tables.

```
SELECT Customers.CustomerName, Orders.Product

FROM Customers

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