Preparing for MySQL Interview

Key Concepts, Practical Questions, and Interview Tips

What is MySQL?

Definition: MySQL is an open-source relational database management system (RDBMS) based on SQL.

Uses: Widely used for web applications, data management, and analytics.

SQL vs. MySQL

- SQL is the language, while MySQL is the software that uses SQL.
- Difference in usage and capabilities.
- When to use MySQL vs. other database systems.

Key SQL Concepts

- **Databases and Tables:** Understand schema, rows, and columns.
- **Keys:** Primary Key, Foreign Key, and Unique Key.
- **Indexes:** What they are and why they are important.
- **Constraints:** NOT NULL, UNIQUE, CHECK, DEFAULT, etc.
- **Data Types:** VARCHAR, INT, DATE, etc.

SQL Query Examples

- **Question 1:** Write a query to fetch the first 10 records from a table.

  -
    ```
    SELECT * FROM table_name LIMIT 10;
    ```
- **Question 2:** Write a query to fetch the highest salary from the "employees" table.

  -
    ```
    SELECT MAX(salary) FROM employees;
    ```
- **Question 3:** How do you join two tables and fetch common data?

  -
    ```
    SELECT * FROM table1 INNER JOIN table2 ON table1.column = table2.column;
    ```

Intermediate MySQL Query Examples

- **Question 1:** Fetch employees with a salary greater than 50,000 and hired in the last 2 years.

  -
    ```
    SELECT * FROM employees WHERE salary > 50000 AND hire_date > DATE_SUB(CURDATE(), INTERVAL 2 YEAR);
    ```
- **Question 2:** Write a query to find duplicate records in a table.

  -
    ```
    SELECT column, COUNT(*) FROM table GROUP BY column HAVING COUNT(*) > 1;
    ```
- **Question 3:** Create a view to fetch active employees only.

  -
    ```
    CREATE VIEW active_employees AS SELECT * FROM employees WHERE status = 'active';
    ```

Advanced SQL Query Examples

- **Question 1:** How do you handle subqueries in MySQL? Example of fetching employees with the highest salary department-wise.

  -
    ```
    SELECT department, employee, salary FROM employees WHERE salary = (SELECT MAX(salary) FROM employees e WHERE e.department = employees.department);
    ```
- **Question 2:** How to optimize slow-running queries? Discuss indexing and query restructuring
  - **What is an Index?**
  - An index is a data structure that helps the database retrieve data faster by reducing the number of rows it needs to scan.
- **What is Query Restructuring?**
  It involves rewriting queries to improve their efficiency without changing the result.

Theoretical SQL Questions

- **Normalization:** Explain different normal forms (1NF, 2NF, 3NF).
- **Denormalization:** When to denormalize and its benefits.
- **ACID Properties:** Atomicity, Consistency, Isolation, Durability.
- **Transactions:** BEGIN, COMMIT, ROLLBACK, and SAVEPOINT.

SQL Optimization Techniques

- Use of **Indexes** to speed up queries.
- Benefits of **Partitioning** in large datasets.
- **Caching results** and **connection pooling** for performance.
- Monitoring tools like **MySQL Slow Query Log**.

Additional Theoretical Questions

- Difference between **DELETE**, **TRUNCATE**, and **DROP**.
- How do **joins** work in MySQL? (INNER, LEFT, RIGHT, FULL).
- Explain the **group by** and **having** clauses.
- Explain **AUTO_INCREMENT** in MySQL.

Real-World SQL Use Cases

- How to design a database for an e-commerce platform.
- Write a query to fetch all customers who placed an order in the last 30 days.
- Optimize a reporting query fetching data for large-scale analytics.

SQL Security & Best Practices

- Managing **User Privileges**.
- Avoiding **SQL Injections**: Best practices for securing queries.
- How to **encrypt sensitive data** in MySQL.

Final Tips for SQL Interviews

- Focus on both theoretical and practical SQL questions.
- Explain your approach to solving SQL queries.
- Always think of performance optimization when discussing queries.
- Practice regularly with real-world problems on platforms like LeetCode, HackerRank, etc.