**`Basic MySQL Questions**

1. **What is MySQL?**
   **Answer:**
   MySQL is an open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) to manage data. It is commonly used for web applications and supports multiple storage engines like InnoDB and MyISAM.

---

2. **What are the different types of SQL statements?**
   **Answer:**
   - **DDL (Data Definition Language):** CREATE, ALTER, DROP, TRUNCATE.
   - **DML (Data Manipulation Language):** INSERT, UPDATE, DELETE.
   - **DQL (Data Query Language):** SELECT.
   - **TCL (Transaction Control Language):** COMMIT, ROLLBACK, SAVEPOINT.
   - **DCL (Data Control Language):** GRANT, REVOKE.

---

3. **What is the difference between CHAR and VARCHAR?**
   **Answer:**
   - CHAR: Fixed-length storage (e.g., CHAR(10) always uses 10 bytes, even for shorter strings).
   - VARCHAR: Variable-length storage (e.g., VARCHAR(10) uses only the required space plus 1 or 2 bytes for length).

---

4. **What are MySQL storage engines? Which ones are commonly used?**
   **Answer:**
   Storage engines are the underlying technologies used to store, handle, and retrieve data in MySQL.
   - Commonly used engines:
     - **InnoDB**: Supports transactions, foreign keys, and row-level locking.
     - **MyISAM**: Fast but does not support transactions or foreign keys.

---

5. **How do you create a table in MySQL?**
   **Answer:**

sql

Copy code

CREATE TABLE employees (

```
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    salary DECIMAL(10, 2),
    hire_date DATE
);
```

---

**Intermediate MySQL Questions**

6. **What is the difference between WHERE and HAVING?**
   **Answer:**
   - **WHERE:** Filters rows before grouping (applied to raw data).
   - **HAVING:** Filters grouped data (applied after aggregation).

sql

Copy code

```
SELECT department, COUNT(*)
FROM employees
GROUP BY department
HAVING COUNT(*) > 5; -- Filters grouped departments
```

---

7. **What is the difference between INNER JOIN, LEFT JOIN, and RIGHT JOIN?**
   **Answer:**
   - **INNER JOIN:** Returns matching rows from both tables.
   - **LEFT JOIN:** Returns all rows from the left table, and matching rows from the right table (fills unmatched with NULL).
   - **RIGHT JOIN:** Returns all rows from the right table, and matching rows from the left table.
     Example:

sql

Copy code

```
SELECT employees.name, departments.name
FROM employees
LEFT JOIN departments ON employees.dept_id = departments.id;
```

---

8. **What are indexes in MySQL, and why are they used?**
   **Answer:**
   Indexes improve query performance by allowing faster lookups. Types of indexes in MySQL:
   - **Primary Key:** Ensures uniqueness and fast access.
   - **Unique Index:** Ensures all values in a column are distinct.
   - **Full-Text Index:** Used for full-text searches.
   - **Composite Index:** Combines multiple columns for faster searching.

---

9. **Explain the ACID properties of MySQL transactions.**
   **Answer:**
   - **Atomicity:** Ensures transactions are all-or-nothing.
   - **Consistency:** Maintains database integrity before and after the transaction.
   - **Isolation:** Ensures transactions do not interfere with each other.
   - **Durability:** Ensures committed transactions are saved permanently.

---

10. **How do you optimize a MySQL query?**
    **Answer:**
    - Use indexes effectively.
    - Avoid SELECT *; only fetch required columns.
    - Use EXPLAIN to analyze query performance.
    - Optimize joins by filtering unnecessary rows.
    - Use caching for frequent queries.
    - Avoid subqueries; prefer joins when possible.

---

**Advanced MySQL Questions**

11. **What are triggers in MySQL?**
    **Answer:**
    Triggers are automatic operations that execute in response to events like INSERT, UPDATE, or DELETE on a table.
    Example:

sql

Copy code

CREATE TRIGGER before_insert_employee

BEFORE INSERT ON employees

```
FOR EACH ROW
SET NEW.hire_date = NOW();
```

---

12. **What is the difference between UNION and UNION ALL?**
    **Answer:**
    - **UNION:** Combines results from multiple queries and removes duplicates.
    - **UNION ALL:** Combines results from multiple queries, including duplicates.

---

13. **What are stored procedures and functions in MySQL?**
    **Answer:**
    - **Stored Procedure:** A set of SQL statements executed as a program.

```
sql
Copy code
DELIMITER $$
CREATE PROCEDURE GetEmployeeCount()
BEGIN
  SELECT COUNT(*) FROM employees;
END $$
```

    - **Function:** Returns a single value.

```
sql
Copy code
DELIMITER $$
CREATE FUNCTION CalculateTax(salary DECIMAL(10, 2))
RETURNS DECIMAL(10, 2)
BEGIN
  RETURN salary * 0.1;
END $$
```

---

14. **What is replication in MySQL?**
    **Answer:**
    Replication is the process of synchronizing data across multiple servers. Types include:
    - **Master-Slave Replication:** One master server writes; slaves replicate data.

- **Master-Master Replication:** Both servers can write and replicate data bidirectionally.

---

15. **What is a deadlock in MySQL, and how do you resolve it?**
    **Answer:**
    A deadlock occurs when two transactions block each other while waiting for resources.
    - **Resolution:**
      - Analyze deadlock logs.
      - Use shorter transactions.
      - Use proper indexing.
      - Avoid cyclic dependencies between tables.

---

16. **How do you implement pagination in MySQL?**
    **Answer:**
    Use LIMIT and OFFSET.

sql

Copy code

```
SELECT *
FROM employees
LIMIT 10 OFFSET 20; -- Fetches rows 21-30
```

---

17. **What are MySQL partitions?**
    **Answer:**
    Partitioning splits a large table into smaller, more manageable pieces. Types:
    - **Range Partitioning:** Based on a range of values.
    - **List Partitioning:** Based on a list of values.
    - **Hash Partitioning:** Based on a hash function.
      Example:

sql

Copy code

```
CREATE TABLE sales (
  id INT,
  sale_date DATE
) PARTITION BY RANGE (YEAR(sale_date)) (
```

```
    PARTITION p1 VALUES LESS THAN (2010),
    PARTITION p2 VALUES LESS THAN (2020)
);
```

**18. What is the difference between DELETE, TRUNCATE, and DROP?**

**Answer:**

- **DELETE:** Removes rows based on a condition. Can be rolled back if wrapped in a transaction.

sql

Copy code

```
DELETE FROM employees WHERE department = 'HR';
```

- **TRUNCATE:** Removes all rows in a table without logging individual row deletions. Cannot be rolled back.

sql

Copy code

```
TRUNCATE TABLE employees;
```

- **DROP:** Deletes the table structure and data permanently.

sql

Copy code

```
DROP TABLE employees;
```

---

**19. Explain the difference between clustered and non-clustered indexes.**

**Answer:**

- **Clustered Index:**
    - Determines the physical order of data in the table.
    - Only one clustered index is allowed per table (e.g., primary key).
- **Non-Clustered Index:**
    - Does not affect the physical order of the table.
    - Multiple non-clustered indexes can exist on a table.

---

**20. How do you handle NULL values in MySQL?**

**Answer:**

- Use IS NULL or IS NOT NULL to filter rows.

- Replace NULL values using COALESCE.

sql

Copy code

SELECT name, COALESCE(salary, 0) AS adjusted_salary FROM employees;

---

**21. What are subqueries in MySQL?**

**Answer:**
A subquery is a query nested inside another query. Types include:

- **Scalar Subquery:** Returns a single value.

sql

Copy code

SELECT name FROM employees WHERE salary = (SELECT MAX(salary) FROM employees);

- **Correlated Subquery:** Refers to the outer query.

sql

Copy code

SELECT name FROM employees e1 WHERE salary > (SELECT AVG(salary) FROM employees e2 WHERE e1.department = e2.department);

---

**22. How do you perform a full outer join in MySQL?**

**Answer:**
MySQL does not support FULL OUTER JOIN natively but can be simulated using UNION.

sql

Copy code

SELECT *

FROM employees e

LEFT JOIN departments d ON e.dept_id = d.id

UNION

SELECT *

FROM employees e

RIGHT JOIN departments d ON e.dept_id = d.id;

---

**23. Explain the concept of normalization.**

**Answer:**
Normalization is the process of organizing data to reduce redundancy and improve integrity. Normal forms include:

1. **1NF:** Ensure atomicity and uniqueness of rows.
2. **2NF:** Remove partial dependencies.
3. **3NF:** Remove transitive dependencies.
4. **BCNF:** Further decomposition for stricter dependencies.

---

**24. What is a composite key?**

**Answer:**
A composite key is a primary key that consists of two or more columns to uniquely identify rows in a table.

sql

Copy code

```
CREATE TABLE orders (
   order_id INT,
   product_id INT,
   PRIMARY KEY (order_id, product_id)
);
```

---

**25. How do you measure query performance in MySQL?**

**Answer:**

- Use EXPLAIN to analyze query execution plans.
- Monitor the query cache and indexing usage.
- Use tools like MySQL Workbench or performance schema.
- Example:

sql

Copy code

```
EXPLAIN SELECT * FROM employees WHERE department = 'IT';
```

---

**26. What are MySQL views? How do you create and use them?**

**Answer:**
A view is a virtual table based on the result of a query.

- **Creation:**

sql

Copy code

CREATE VIEW EmployeeView AS

SELECT name, department, salary FROM employees WHERE salary > 50000;

- **Usage:**

sql

Copy code

SELECT * FROM EmployeeView;

---

**27. What are the advantages and disadvantages of using stored procedures?**

**Answer:**

- **Advantages:**
    - Improves performance as it reduces network traffic.
    - Centralizes business logic.
    - Enhances security by restricting direct access to tables.
- **Disadvantages:**
    - Harder to debug compared to application logic.
    - Less portable across different RDBMS.
    - Can become complex with large procedures.

---

**28. What is the purpose of the GROUP BY clause?**

**Answer:**
The GROUP BY clause groups rows that have the same values in specified columns and allows aggregate functions like COUNT, SUM, and AVG to be applied.

sql

Copy code

SELECT department, COUNT(*) AS employee_count

FROM employees

GROUP BY department;

---

**29. How do you handle duplicate records in MySQL?**

**Answer:**

- Remove duplicates using DISTINCT.

sql

Copy code

SELECT DISTINCT name FROM employees;

- Delete duplicate rows:

sql

Copy code

DELETE e1

FROM employees e1

JOIN employees e2

ON e1.id > e2.id AND e1.name = e2.name;

---

**30. How do transactions work in MySQL?**

**Answer:**
Transactions allow multiple operations to be executed as a single unit.

- Commands:
    - START TRANSACTION
    - COMMIT: Saves changes.
    - ROLLBACK: Reverts changes.
- Example:

sql

Copy code

START TRANSACTION;

UPDATE accounts SET balance = balance - 100 WHERE id = 1;

UPDATE accounts SET balance = balance + 100 WHERE id = 2;

COMMIT;

---

**31. How do you handle database backups in MySQL?**

**Answer:**

- Use the mysqldump utility for logical backups.

bash

Copy code

mysqldump -u root -p database_name > backup.sql

- Use binary logs for point-in-time recovery.
- Automate backups using cron jobs or MySQL Workbench.

---

**32. What are MySQL JSON functions?**

**Answer:**
MySQL provides JSON functions to manage JSON data.

- Example:

sql

Copy code

CREATE TABLE products (id INT, data JSON);

INSERT INTO products (id, data) VALUES (1, '{"price": 100, "stock": 50}');

SELECT JSON_EXTRACT(data, '$.price') AS price FROM products;

---

**33. What are user-defined variables in MySQL?**

**Answer:**
User-defined variables are session-specific and start with @.

sql

Copy code

SET @employee_count = (SELECT COUNT(*) FROM employees);

SELECT @employee_count;

---

**34. What is MySQL partition pruning?**

**Answer:**
Partition pruning is the process of optimizing queries by accessing only the required partitions of a table instead of scanning the entire table. This improves performance for large datasets.

---

**35. How do you implement recursive queries in MySQL?**

**Answer:**
Use **Common Table Expressions (CTEs)** introduced in MySQL 8.0.

sql

Copy code

```
WITH RECURSIVE employee_hierarchy AS (
   SELECT id, name, manager_id FROM employees WHERE manager_id IS NULL
   UNION ALL
   SELECT e.id, e.name, e.manager_id
   FROM employees e
   INNER JOIN employee_hierarchy eh ON e.manager_id = eh.id
)
SELECT * FROM employee_hierarchy;
```