MySQL Interview Questions (2+ Years Experience)

Q: What is the difference between INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN?

A: INNER JOIN: Returns only matching rows from both tables. LEFT JOIN: Returns all rows from the left table + matching rows from the right. RIGHT JOIN: Returns all rows from the right table + matching rows from the left. FULL JOIN: Returns all rows when there is a match in one of the tables (MySQL emulates using UNION).

Q: What are indexes in MySQL? Why are they used?

A: Indexes speed up SELECT queries by allowing faster searching. Types: PRIMARY KEY, UNIQUE, FULLTEXT, INDEX. Downside: Slow down write operations (INSERT/UPDATE/DELETE).

Q: What is the difference between WHERE and HAVING clause?

A: WHERE: Filters rows before grouping. HAVING: Filters rows after grouping (used with GROUP BY).

Q: Explain normalization and its types.

A: Normalization is organizing data to reduce redundancy & improve integrity. 1NF: Atomic values. 2NF: 1NF + no partial dependency. 3NF: 2NF + no transitive dependency.

Q: What's the difference between DELETE, TRUNCATE, and DROP?

A: DELETE: Removes rows, can use WHERE, rollback possible. TRUNCATE: Removes all rows, faster, resets auto-increment, usually no rollback. DROP: Deletes table schema + data permanently.

Q: What is a stored procedure? Advantages?

A: Stored procedure: Precompiled SQL set stored in DB. Advantages: Better performance, reusable, enhances security.

Q: How do transactions work in MySQL?

A: Transaction = sequence of queries executed as a unit. Follows ACID: Atomicity, Consistency, Isolation, Durability.

Q: What is the difference between MyISAM and InnoDB?

A: MyISAM: No transactions, table-level locking, faster reads. InnoDB: Supports transactions, row-level locking, foreign keys, better concurrency.

Q: What are triggers in MySQL?

A: Triggers: SQL code executed automatically in response to an event (INSERT, UPDATE, DELETE).

Q: How to optimize a slow SQL query?

A: Use EXPLAIN to analyze query. Add indexes. Avoid SELECT *. Use LIMIT. Use caching (e.g., Redis, query cache).