Here are common interview questions and answers for SQL, with a focus on MySQL. It's important to be able to write the queries as well as explain the concepts.

**SQL / MySQL Fundamentals**

**1. What is the difference between SQL and MySQL?**

* **Answer:** This is a classic and fundamental question.
  + **SQL (Structured Query Language):** This is a standardized language used to interact with and manipulate relational databases. It's the "what" you do. You use SQL to create databases and tables, insert, update, and delete data, and retrieve information.
  + **MySQL:** This is a Relational Database Management System (RDBMS). It's the "how" and the "where." MySQL is a software product that provides the environment to store, manage, and retrieve data. It is one of many database systems (like SQL Server, PostgreSQL, Oracle) that use SQL as their standard language.

**2. What are the different types of SQL commands?**

* **Answer:** SQL commands are broadly categorized into four main types:
  + **DDL (Data Definition Language):** Used for creating and modifying the structure of database objects.
    - CREATE (e.g., CREATE TABLE, CREATE DATABASE)
    - ALTER (e.g., ALTER TABLE)
    - DROP (e.g., DROP TABLE)
  + **DML (Data Manipulation Language):** Used for managing data within tables.
    - SELECT
    - INSERT
    - UPDATE
    - DELETE
  + **DCL (Data Control Language):** Used for controlling access to data.
    - GRANT
    - REVOKE
  + **TCL (Transaction Control Language):** Used for managing transactions.
    - COMMIT
    - ROLLBACK

**3. What is a Primary Key? What is a Foreign Key?**

* **Answer:** These are the cornerstones of relational database design.
  + **Primary Key:** A column or a set of columns that uniquely identifies each row in a table. A primary key:
    - Must contain a unique value for each row.
    - Cannot contain NULL values.
    - Can only have one per table.
  + **Foreign Key:** A column or a set of columns in one table that refers to the primary key in another table. It establishes a link between two tables, enforcing referential integrity.

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

* **Answer:** All three commands remove data, but they do so in different ways.
  + **DELETE:** Removes rows one at a time. It is a DML command and is much slower for large datasets because it logs each deletion. It can be rolled back. You can use a WHERE clause to delete specific rows.
  + **TRUNCATE:** Removes all rows from a table by deallocating the data pages. It is a DDL command and is much faster than DELETE. It cannot be rolled back. It resets the identity/auto-increment value.
  + **DROP:** Deletes the entire table, including its structure, data, indexes, and constraints. It is a DDL command and cannot be rolled back.

**5. What is an SQL Join, and what are the different types?**

* **Answer:** A JOIN is used to combine rows from two or more tables based on a related column between them.
  + **INNER JOIN:** Returns only the rows where there is a match in both tables.
  + **LEFT JOIN (or LEFT OUTER JOIN):** Returns all rows from the left table and the matched rows from the right table. Rows in the left table with no match in the right will have NULL values.
  + **RIGHT JOIN (or RIGHT OUTER JOIN):** Returns all rows from the right table and the matched rows from the left table.
  + **FULL JOIN (or FULL OUTER JOIN):** Returns all rows when there is a match in one of the tables. It's a combination of LEFT and RIGHT joins.
  + **CROSS JOIN:** Returns the Cartesian product of the two tables (every row from the first table is joined with every row from the second table).

**Intermediate to Advanced Questions**

**6. What is the difference between the WHERE and HAVING clauses?**

* **Answer:** Both are used for filtering, but they operate at different stages of a query.
  + **WHERE:** Filters rows *before* any grouping or aggregation is performed. You cannot use aggregate functions (like SUM, AVG, COUNT) in a WHERE clause.
  + **HAVING:** Filters groups of rows *after* they have been grouped by GROUP BY. You can and should use aggregate functions here to filter on the results of the aggregation.

**7. How do you find the second highest salary from an Employees table?**

* **Answer:** This is a very common technical question. There are multiple ways to solve it.
  + **Using LIMIT and OFFSET (MySQL/PostgreSQL):**

SQL

SELECT Salary

FROM Employees

ORDER BY Salary DESC

LIMIT 1 OFFSET 1;

* + **Using a Subquery and MAX:**

SQL

SELECT MAX(Salary)

FROM Employees

WHERE Salary < (SELECT MAX(Salary) FROM Employees);

* + **Using DENSE\_RANK() (More flexible and modern):**

SQL

SELECT Salary

FROM (

SELECT Salary, DENSE\_RANK() OVER (ORDER BY Salary DESC) as rank\_num

FROM Employees

) AS ranked\_salaries

WHERE rank\_num = 2;

**8. What is a "subquery"? What are its types?**

* **Answer:** A subquery (or nested query) is a query within another SQL query.
  + **Types:**
    - **Single-row Subquery:** Returns zero or one row. Used with single-value comparison operators (=, >, <).
    - **Multi-row Subquery:** Returns one or more rows. Used with multi-value operators (IN, ANY, ALL).
    - **Correlated Subquery:** The inner query's result depends on the outer query's row. It's executed once for each row processed by the outer query, which can be inefficient.

**9. What is an Index, and why is it important?**

* **Answer:** An index is a database object that improves the speed of data retrieval operations. It's similar to an index in a book; instead of reading the entire book to find a topic, you look up the page number in the index.
  + **Benefits:**
    - Speeds up SELECT queries.
    - Helps with sorting and filtering operations.
  + **Drawbacks:**
    - Slows down INSERT, UPDATE, and DELETE operations because the index must also be updated.
    - Requires additional storage space.

**10. Explain the concept of Normalization.**

* **Answer:** Normalization is the process of organizing data in a database to reduce redundancy and improve data integrity. It's a systematic approach to breaking down a large table into smaller, more manageable tables and defining relationships between them.
  + **Normal Forms:** The process is guided by "normal forms" (1NF, 2NF, 3NF, BCNF, etc.), with 3NF being a common goal for most business applications.
  + **1NF (First Normal Form):** Each column contains a single, atomic value, and there are no repeating groups of columns.
  + **2NF (Second Normal Form):** Is in 1NF and all non-key columns are fully dependent on the primary key.
  + **3NF (Third Normal Form):** Is in 2NF and all non-key columns are not transitively dependent on the primary key (i.e., they don't depend on other non-key columns).

Practical SQL Query Questions:

Be prepared to write queries for common scenarios like:

* Finding duplicates in a table.
* Calculating sums, averages, or counts (GROUP BY, HAVING).
* Joining three or more tables.
* Using CASE statements to categorize data.