SQL

Introduction to SQL

1. **What is SQL?**
    - SQL (Structured Query Language) is used to interact with databases to create, read, update, and delete (CRUD) data.
    - SQL is crucial for backend testing, database validation, and ensuring data integrity.
2. **Why is SQL important for SDETs?**
    - Validate data stored in databases.
    - Write efficient queries to fetch test data for automated tests.
    - Troubleshoot backend-related issues.

SQL Basics

1. Data Types

- **Numeric:** INT, FLOAT, DECIMAL
- **String:** VARCHAR, CHAR, TEXT
- **Date and Time:** DATE, TIME, TIMESTAMP

2. Database Concepts

- **Table:** Organized collection of data in rows and columns.
- **Primary Key:** Unique identifier for each row.
- **Foreign Key:** Maintains relationships between tables.
- **Index:** Speeds up data retrieval.

3. SQL Commands

- **DDL (Data Definition Language):** CREATE, ALTER, DROP, TRUNCATE
- **DML (Data Manipulation Language):** SELECT, INSERT, UPDATE, DELETE
- **DCL (Data Control Language):** GRANT, REVOKE
- **TCL (Transaction Control Language):** COMMIT, ROLLBACK, SAVEPOINT

Writing Queries

1. SELECT Statement

Syntax:

sql
Copy code
SELECT column1, column2 FROM table_name WHERE condition;

-
- Examples:

Fetch all records:

sql
Copy code
SELECT * FROM employees;

-

Fetch specific columns:

sql
Copy code
SELECT first_name, salary FROM employees;

-

2. Filtering Data with WHERE

Syntax:

sql
Copy code
SELECT * FROM table_name WHERE column_name = value;

-
- Operators: =, !=, >, <, >=, <=, LIKE, IN, BETWEEN, IS NULL.

Example:

sql
Copy code
```
SELECT * FROM employees WHERE salary > 50000;
```

-

3. Sorting Data with ORDER BY

Syntax:

sql
Copy code
```
SELECT * FROM table_name ORDER BY column_name ASC/DESC;
```

-

Example:

sql
Copy code
```
SELECT * FROM employees ORDER BY salary DESC;
```

-

4. Limiting Results

Syntax:

sql
Copy code
```
SELECT * FROM table_name LIMIT number;
```

-

Example:

sql
Copy code
```
SELECT * FROM employees LIMIT 10;
```

-

Advanced SQL Concepts

1. Aggregate Functions

- Functions: COUNT(), SUM(), AVG(), MAX(), MIN()

Example:

sql
Copy code
SELECT COUNT(*) AS total_employees FROM employees;

-

2. Grouping Data with GROUP BY

Syntax:

sql
Copy code
SELECT column, aggregate_function(column) FROM table GROUP BY column;

-

Example:

sql
Copy code
SELECT department_id, COUNT(*) FROM employees GROUP BY department_id;

-

3. Filtering Groups with HAVING

Example:

sql
Copy code

SELECT department_id, AVG(salary) FROM employees GROUP BY department_id HAVING AVG(salary) > 50000;

-

4. Joining Tables

- Types: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN

Example:

sql
Copy code
SELECT employees.name, departments.name
FROM employees
INNER JOIN departments
ON employees.department_id = departments.id;

-

Commonly Used Queries for Testing

Find Duplicate Records

sql
Copy code
SELECT name, COUNT(*)
FROM employees
GROUP BY name
HAVING COUNT(*) > 1;

1.

Identify Missing Foreign Key Relationships

sql
Copy code
SELECT orders.id
FROM orders

```
LEFT JOIN customers
ON orders.customer_id = customers.id
WHERE customers.id IS NULL;
```

2.

Validate Data Consistency

```
sql
Copy code
SELECT * FROM orders WHERE total_amount != (price * quantity);
```

3.

Best Practices for Writing SQL Queries

1. Use appropriate indexing for faster query execution.
2. Avoid using SELECT *; specify the columns needed.
3. Use JOINs instead of subqueries where possible for better performance.
4. Use LIMIT or OFFSET for large datasets to avoid performance bottlenecks.
5. Ensure proper error handling in automation scripts interacting with SQL.

Common Interview Questions

Write a query to find the nth highest salary in a table.

```
sql
Copy code
SELECT DISTINCT salary
FROM employees
ORDER BY salary DESC
LIMIT 1 OFFSET n-1;
```

1.

Find all employees with the same salary in a table.

```
sql
```

Copy code
```
SELECT salary, COUNT(*)
FROM employees
GROUP BY salary
HAVING COUNT(*) > 1;
```

2.

Write a query to retrieve employees who have joined in the last 30 days.

sql
Copy code
```
SELECT * FROM employees
WHERE join_date >= DATE_SUB(CURRENT_DATE, INTERVAL 30 DAY);
```

3.

- Important interview Question

Basic SQL Questions

1. **What is SQL, and why is it used in testing?**
   - SQL (Structured Query Language) is used to interact with relational databases. It is crucial in testing to validate backend data integrity, consistency, and correctness during software testing processes.

---

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

---

1. **What is the difference between `DELETE` and `TRUNCATE`?**
   - **`DELETE`**: Removes specific rows, can include `WHERE` conditions, and logs individual row deletions (slower, rollback possible).
   - **`TRUNCATE`**: Removes all rows from a table, faster, no rollback for individual deletions, and does not trigger delete triggers.

---

1. **What is a Primary Key?**

- A primary key is a column (or combination of columns) that uniquely identifies each row in a table. It cannot have `NULL` values and must be unique.

---

1. **What is a Foreign Key?**
   - A foreign key in one table is a reference to a primary key in another table. It ensures referential integrity between the tables.

Intermediate SQL Questions

1. **What are JOINS in SQL? Explain types of joins.**

   - Joins combine rows from two or more tables based on a related column.
     - **INNER JOIN**: Returns matching rows between tables.
     - **LEFT JOIN (OUTER JOIN)**: Returns all rows from the left table and matching rows from the right table.
     - **RIGHT JOIN (OUTER JOIN)**: Returns all rows from the right table and matching rows from the left table.
     - **FULL JOIN**: Returns rows when there is a match in either table.
     - **CROSS JOIN**: Returns the Cartesian product of the tables.

sql
Copy code
```
SELECT A.id, B.name
FROM TableA A
INNER JOIN TableB B ON A.id = B.id;
```

2.

---

1. **What is the difference between `WHERE` and `HAVING`?**

   - **`WHERE`**: Filters rows before grouping, works on individual rows.
   - **`HAVING`**: Filters groups of rows, applied after `GROUP BY`.

sql
Copy code
```
SELECT department, COUNT(*)
FROM employees
GROUP BY department
HAVING COUNT(*) > 5;
```

2.

- ---

1. **What is a Subquery?**

    - A subquery is a query nested within another query. It can be used in `SELECT`, `INSERT`, `UPDATE`, or `DELETE` statements.
    - Example:

sql
Copy code
```
SELECT name
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

2.

- ---

1. **What is the difference between `UNION` and `UNION ALL`?**
    - `UNION`: Combines results of two queries and removes duplicates.
    - `UNION ALL`: Combines results of two queries, including duplicates.

- ---

1. **What is the purpose of indexes?**
    - Indexes improve the speed of data retrieval but may slow down `INSERT`/`UPDATE`/`DELETE` operations.
    - Types:
        - **Clustered Index**: Sorts and stores data rows in the table.
        - **Non-Clustered Index**: Creates a separate structure for storing the index.

- ---

Advanced SQL Questions

1. **What are Triggers in SQL?**

    - A trigger is a database object that is automatically executed in response to certain events on a table, like `INSERT`, `UPDATE`, or `DELETE`.

sql
Copy code
```
CREATE TRIGGER after_insert_employee
AFTER INSERT ON employees
FOR EACH ROW
BEGIN
   INSERT INTO logs (action) VALUES ('New Employee Added');
END;
```

2.

- ---

1. **What are Stored Procedures, and why are they used?**

    - A stored procedure is a prepared SQL code that can be reused and executed multiple times.

sql
Copy code
```
CREATE PROCEDURE GetEmployees()
BEGIN
   SELECT * FROM employees;
END;
```

2.

- Benefits:
    - Improves performance.
    - Reduces redundancy.
    - Provides security through abstraction.

- ---

1. **What is the difference between `RANK()`, `DENSE_RANK()`, and `ROW_NUMBER()`?**

    - `RANK()`: Assigns ranks with gaps for duplicate values.
    - `DENSE_RANK()`: Assigns ranks without gaps.
    - `ROW_NUMBER()`: Assigns a unique number to each row.

sql
Copy code
```
SELECT name, salary,
    RANK() OVER (ORDER BY salary DESC) AS rank,
    DENSE_RANK() OVER (ORDER BY salary DESC) AS dense_rank,
    ROW_NUMBER() OVER (ORDER BY salary DESC) AS row_number
FROM employees;
```

2.

- ---

1. **How do you optimize SQL queries?**
    - Use indexes.
    - Avoid `SELECT *`.

- Use `EXISTS` instead of `IN` for subqueries.
- Avoid correlated subqueries.
- Use proper data types and minimize joins where possible.

---

1. **What are Common Table Expressions (CTEs)?**

   - A CTE is a temporary result set that can be referred to within a `SELECT`, `INSERT`, `UPDATE`, or `DELETE`.

sql
Copy code
```
WITH SalesCTE AS (
   SELECT product, SUM(quantity) AS total_sales
   FROM sales
   GROUP BY product
)
SELECT * FROM SalesCTE WHERE total_sales > 100;
```

2.

Scenario-Based Questions

How do you find duplicate rows in a table?

sql
Copy code
```
SELECT name, COUNT(*)
FROM employees
GROUP BY name
HAVING COUNT(*) > 1;
```

1.

How do you fetch the second highest salary?

sql
Copy code
```
SELECT MAX(salary)
FROM employees
WHERE salary < (SELECT MAX(salary) FROM employees);
```

1.

- ---

How do you find all employees who joined in the last 30 days?

sql
Copy code
```
SELECT * FROM employees
WHERE join_date >= DATEADD(DAY, -30, GETDATE());
```

1.

- ---

1. **How do you identify unused indexes in a database?**
    - Use database performance analysis tools or query metadata views (e.g., `sys.dm_db_index_usage_stats` in SQL Server).

- ---

1. **How do you test data integrity in SQL?**
    - Validate foreign key constraints.
    - Compare source and target data after migration.
    - Check for orphan records.