SQL Guide: 50 Commonly Asked Questions with Advanced Concepts

Contents

1 Basic SQL Questions and Answers **3**
- 1.1 What is SQL? 3
- 1.2 What is a database? 3
- 1.3 What are the different types of DBMS? 3
- 1.4 What is a primary key? 3
- 1.5 What is a foreign key? 3
- 1.6 What is the SELECT statement used for? 3
- 1.7 How do you insert data into a table? 3
- 1.8 How do you update data in a table? 4
- 1.9 How do you delete data from a table? 4
- 1.10 What is the WHERE clause used for? 4
- 1.11 How do you sort query results? 4
- 1.12 What does GROUP BY do? 4
- 1.13 What is the HAVING clause? 4
- 1.14 What are the different types of JOINs? 4
- 1.15 What is an INNER JOIN? 5
- 1.16 What is a LEFT JOIN? 5
- 1.17 What is a RIGHT JOIN? 5
- 1.18 What is a FULL JOIN? 5
- 1.19 How do you remove duplicates from query results? 5
- 1.20 How do you limit the number of rows returned? 5
- 1.21 What is NULL in SQL? 5
- 1.22 How do you check for NULL values? 5
- 1.23 How do you ensure a column cannot have NULL values? 6
- 1.24 What does the COUNT function do? 6
- 1.25 What does the SUM function do? 6

2 Intermediate SQL Questions and Answers **6**
- 2.1 What does the AVG function do? 6
- 2.2 What does the MAX function do? 6
- 2.3 What does the MIN function do? 6
- 2.4 What is a subquery? 6
- 2.5 What is a correlated subquery? 7
- 2.6 What is the CASE statement? 7
- 2.7 What does UNION do? 7

2.8 What is the difference between UNION and UNION ALL? 7
2.9 What does INTERSECT do? 7
2.10 What does EXCEPT do? 7
2.11 What is a view? 7
2.12 What is an index? 8
2.13 How do you delete a table? 8
2.14 How do you modify a table structure? 8
2.15 What does TRUNCATE TABLE do? 8

3 Advanced SQL Concepts **8**
3.1 What is a transaction? 8
3.2 What does COMMIT do? 8
3.3 What does ROLLBACK do? 8
3.4 What is a trigger? 9
3.5 What is a stored procedure? 9
3.6 What is a user-defined function? 9
3.7 What are window functions? 9
3.8 What is a Common Table Expression (CTE)? 9
3.9 What is a recursive CTE? 10
3.10 How do you create a pivot table in SQL? 10

1 Basic SQL Questions and Answers

1.1 What is SQL?

SQL (Structured Query Language) is a standardized language used to manage and manipulate relational databases. It allows users to create, read, update, and delete data, as well as manage database structures.

1.2 What is a database?

A database is an organized collection of data, typically stored and accessed electronically from a computer system, managed by a Database Management System (DBMS).

1.3 What are the different types of DBMS?

DBMS types include:

- Relational DBMS (e.g., MySQL, PostgreSQL)
- NoSQL DBMS (e.g., MongoDB, Cassandra)
- Hierarchical DBMS (e.g., IBM IMS)
- Network DBMS
- Object-oriented DBMS

1.4 What is a primary key?

A primary key is a unique identifier for each record in a table. It must contain unique values and cannot contain NULL values.

1.5 What is a foreign key?

A foreign key is a column or set of columns in one table that uniquely identifies a row of another table, establishing a link between them.

1.6 What is the SELECT statement used for?

The SELECT statement retrieves data from a database. Example:

```
SELECT * FROM employees;
```

1.7 How do you insert data into a table?

Use the INSERT INTO statement:

```
INSERT INTO employees (id
id, name, age) VALUES (1, 'John Doe', 30);
```

1.8 How do you update data in a table?

Use the UPDATE statement:

```
UPDATE employees SET age = 31 WHERE id = 1;
```

1.9 How do you delete data from a table?

Use the DELETE statement:

```
DELETE FROM employees WHERE id = 1;
```

1.10 What is the WHERE clause used for?

The WHERE clause filters records based on specified conditions:

```
SELECT * FROM employees WHERE age > 30;
```

1.11 How do you sort query results?

Use ORDER BY:

```
SELECT * FROM employees ORDER BY age DESC;
```

1.12 What does GROUP BY do?

GROUP BY groups rows with the same values into summary rows:

```
SELECT department, COUNT(*) FROM employees GROUP BY department;
```

1.13 What is the HAVING clause?

HAVING filters grouped records:

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

1.14 What are the different types of JOINs?

Types include:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

1.15 What is an INNER JOIN?

INNER JOIN returns rows with matching values in both tables:

```
SELECT employees.name, departments.department_name
FROM employees INNER JOIN departments
ON employees.department_id = departments.id;
```

1.16 What is a LEFT JOIN?

LEFT JOIN returns all rows from the left table and matching rows from the right table:

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

1.17 What is a RIGHT JOIN?

RIGHT JOIN returns all rows from the right table and matching rows from the left table.

1.18 What is a FULL JOIN?

FULL JOIN returns all rows when there is a match in either table.

1.19 How do you remove duplicates from query results?

Use DISTINCT:

```
SELECT DISTINCT department FROM employees;
```

1.20 How do you limit the number of rows returned?

Use LIMIT:

```
SELECT * FROM employees LIMIT 10;
```

1.21 What is NULL in SQL?

NULL represents a missing or unknown value, not equivalent to zero or an empty string.

1.22 How do you check for NULL values?

Use IS NULL:

```
SELECT * FROM employees WHERE age IS NULL;
```

1.23 How do you ensure a column cannot have NULL values?

Use NOT NULL in table creation:

```
CREATE TABLE employees (
    id INT NOT NULL,
    name VARCHAR(50)
);
```

1.24 What does the COUNT function do?

COUNT returns the number of rows:

```
SELECT COUNT(*) FROM employees;
```

1.25 What does the SUM function do?

SUM calculates the total of a numeric column:

```
SELECT SUM(salary) FROM employees;
```

2 Intermediate SQL Questions and Answers

2.1 What does the AVG function do?

AVG calculates the average of a numeric column:

```
SELECT AVG(salary) FROM employees;
```

2.2 What does the MAX function do?

MAX returns the maximum value in a column:

```
SELECT MAX(salary) FROM employees;
```

2.3 What does the MIN function do?

MIN returns the minimum value in a column:

```
SELECT MIN(salary) FROM employees;
```

2.4 What is a subquery?

A subquery is a query nested inside another query:

```
SELECT * FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees);
```

2.5 What is a correlated subquery?

A correlated subquery references columns from the outer query:

```
SELECT * FROM employees e
WHERE EXISTS (
    SELECT 1 FROM departments d
    WHERE d.id = e.department_id
);
```

2.6 What is the CASE statement?

CASE provides conditional logic:

```
SELECT name,
       CASE WHEN age < 30 THEN 'Young'
            ELSE 'Senior' END AS age_group
FROM employees;
```

2.7 What does UNION do?

UNION combines the results of two queries, removing duplicates:

```
SELECT name FROM employees
UNION
SELECT name FROM contractors;
```

2.8 What is the difference between UNION and UNION ALL?

UNION ALL includes duplicates, while UNION removes them.

2.9 What does INTERSECT do?

INTERSECT returns rows common to both queries:

```
SELECT name FROM employees
INTERSECT
SELECT name FROM contractors;
```

2.10 What does EXCEPT do?

EXCEPT returns rows in the first query not present in the second:

```
SELECT name FROM employees
EXCEPT
SELECT name FROM contractors;
```

2.11 What is a view?

A view is a virtual table based on a query:

```
CREATE VIEW high_earners AS
SELECT * FROM employees WHERE salary > 50000;
```

2.12 What is an index?

An index improves query performance by creating a lookup structure:

```
CREATE INDEX idx_employee_name ON employees(name);
```

2.13 How do you delete a table?

Use DROP TABLE:

```
DROP TABLE employees;
```

2.14 How do you modify a table structure?

Use ALTER TABLE:

```
ALTER TABLE employees ADD COLUMN email VARCHAR(100);
```

2.15 What does TRUNCATE TABLE do?

TRUNCATE TABLE removes all rows from a table without logging individual deletions:

```
TRUNCATE TABLE employees;
```

3 Advanced SQL Concepts

3.1 What is a transaction?

A transaction is a sequence of operations treated as a single unit:

```
BEGIN TRANSACTION;
UPDATE accounts SET balance = balance - 100 WHERE id = 1;
UPDATE accounts SET balance = balance + 100 WHERE id = 2;
COMMIT;
```

3.2 What does COMMIT do?

COMMIT saves all changes made during a transaction:

```
COMMIT;
```

3.3 What does ROLLBACK do?

ROLLBACK undoes all changes made during a transaction:

```
ROLLBACK;
```

3.4 What is a trigger?

A trigger is a stored procedure that runs automatically when specific events occur:

```
CREATE TRIGGER update_timestamp
AFTER UPDATE ON employees
FOR EACH ROW
UPDATE employees SET last_updated = CURRENT_TIMESTAMP
WHERE id = OLD.id;
```

3.5 What is a stored procedure?

A stored procedure is a set of SQL statements stored in the database:

```
CREATE PROCEDURE raise_salary(IN employee_id INT, IN increase
   DECIMAL)
BEGIN
    UPDATE employees SET salary = salary + increase
    WHERE id = employee_id;
END;
```

3.6 What is a user-defined function?

A function returns a value based on input parameters:

```
CREATE FUNCTION get_age(birth_date DATE)
RETURNS INT
RETURN TIMESTAMPDIFF(YEAR, birth_date, CURDATE());
```

3.7 What are window functions?

Window functions perform calculations across a set of rows related to the current row:

```
SELECT name, salary,
       RANK() OVER (PARTITION BY department ORDER BY salary
          DESC) AS rank
FROM employees;
```

3.8 What is a Common Table Expression (CTE)?

A CTE is a temporary result set defined within a query:

```
WITH dept_counts AS (
    SELECT department, COUNT(*) AS emp_count
    FROM employees
    GROUP BY department
)
SELECT * FROM dept_counts WHERE emp_count > 5;
```

3.9 What is a recursive CTE?

A recursive CTE refers to itself to process hierarchical data:

```sql
WITH RECURSIVE org_chart AS (
    SELECT id, name, manager_id, 1 AS level
    FROM employees WHERE manager_id IS NULL
    UNION ALL
    SELECT e.id, e.name, e.manager_id, o.level + 1
    FROM employees e INNER JOIN org_chart o
    ON e.manager_id = o.id
)
SELECT * FROM org_chart;
```

3.10 How do you create a pivot table in SQL?

A pivot table transforms rows into columns:

```sql
SELECT *
FROM (
    SELECT department, year, sales
    FROM sales_data
) AS source
PIVOT (
    SUM(sales)
    FOR year IN ('2023', '2024', '2025')
) AS pivot_table;
```