Must learn SQL questions

1. What is SQL, and why is it used?

Answer: SQL (Structured Query Language) is a standard language used to interact with databases. It is used for querying, updating, inserting, and deleting data, as well as managing database structures.

---

2. Write a query to fetch the second-highest salary from the Employee table.

sql
CopyEdit

```
SELECT DISTINCT salary
FROM Employee
ORDER BY salary DESC
LIMIT 1 OFFSET 1;
```

or

sql
CopyEdit

```
SELECT MAX(salary)
FROM Employee
WHERE salary < (SELECT MAX(salary) FROM Employee);
```

---

3. What are the different types of SQL commands?

Answer:

- **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`

Must learn SQL questions

4. Write a query to find duplicate records in a table.

sql
CopyEdit

```
SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name
HAVING COUNT(*) > 1;
```

---

5. What is the difference between DELETE and TRUNCATE?

Answer:

- `DELETE` removes specific rows and allows rollback.
- `TRUNCATE` removes all rows and cannot be rolled back.

---

6. Write a query to get the department with the highest number of employees.

sql
CopyEdit

```
SELECT department, COUNT(*) AS emp_count
FROM Employee
GROUP BY department
ORDER BY emp_count DESC
LIMIT 1;
```

---

7. What are joins in SQL? Name the types of joins.

Answer: Joins combine records from two or more tables based on a related column. Types:

- `INNER JOIN`
- `LEFT JOIN`
- `RIGHT JOIN`
- `FULL OUTER JOIN`
- `SELF JOIN`
- `CROSS JOIN`

---

8. Write a query to fetch records where name starts with ‘A’.

Must learn SQL questions

sql
CopyEdit
```
SELECT * FROM Employee WHERE name LIKE 'A%';
```

---

9. What is a primary key, and how is it different from a unique key?

Answer: A **Primary Key** uniquely identifies records and cannot have NULL values. A **Unique Key** also ensures uniqueness but can have one NULL value.

---

10. Write a query to fetch employees who earn more than the average salary.

sql
CopyEdit
```
SELECT * FROM Employee
WHERE salary > (SELECT AVG(salary) FROM Employee);
```

---

11. What is a foreign key, and why is it important?

Answer: A **Foreign Key** links two tables by referencing the Primary Key in another table, maintaining referential integrity.

---

12. Write a query to get the top 3 highest salaries in the Employee table.

sql
CopyEdit
```
SELECT DISTINCT salary
FROM Employee
ORDER BY salary DESC
LIMIT 3;
```

---

13. What is the difference between WHERE and HAVING clauses?

Answer:

- `WHERE` filters rows before grouping.
- `HAVING` filters groups after aggregation.

Must learn SQL questions

---

14. Write a query to fetch common records from two tables.

sql
CopyEdit

```
SELECT * FROM TableA
INTERSECT
SELECT * FROM TableB;
```

---

15. What is normalization? Explain its types.

Answer: Normalization reduces redundancy and improves consistency. Types:

1. **1NF** – No duplicate columns
2. **2NF** – 1NF + No partial dependency
3. **3NF** – 2NF + No transitive dependency
4. **BCNF** – Stricter 3NF

---

16. Write a query to create a table with constraints.

sql
CopyEdit

```
CREATE TABLE Employee (
    emp_id INT PRIMARY KEY,
    name VARCHAR(50) UNIQUE,
    dept_id INT,
    FOREIGN KEY (dept_id) REFERENCES Department(dept_id)
);
```

---

17. What are indexes in SQL, and what are their types?

Answer: Indexes improve query performance. Types:

- **Clustered Index** – Sorts and stores data physically
- **Non-clustered Index** – Points to data without sorting it

---

18. Write a query to count the number of employees in each department.

sql
CopyEdit

Must learn SQL questions

```
SELECT department, COUNT(*) AS emp_count
FROM Employee
GROUP BY department;
```

---

19. What is the difference between clustered and non-clustered indexes?

Answer:

- **Clustered Index** determines the physical storage order.
- **Non-clustered Index** stores pointers to actual rows.

---

20. Write a query to find employees who have not been assigned a department.

sql
CopyEdit

```
SELECT * FROM Employee WHERE department IS NULL;
```

---

21. What are aggregate functions in SQL? Give examples.

Answer: Aggregate functions perform calculations on multiple rows. Examples:

- `SUM()`
- `AVG()`
- `COUNT()`
- `MAX()`
- `MIN()`

---

22. Write a query to combine the results of two tables using UNION.

sql
CopyEdit

```
SELECT name FROM TableA
UNION
SELECT name FROM TableB;
```

---

23. What is the difference between UNION and UNION ALL?

Must learn SQL questions

Answer:

- `UNION` removes duplicates.
- `UNION ALL` includes duplicates.

---

24. Write a query to fetch the nth highest salary in a table.

sql
CopyEdit

```
SELECT DISTINCT salary
FROM Employee
ORDER BY salary DESC
LIMIT 1 OFFSET n-1;
```

---

25. What is a self-join, and when would you use it?

Answer: A self-join joins a table to itself, useful for hierarchical data like employee-manager relationships.

---

26. Write a query to get the total salary paid to employees in each department.

sql
CopyEdit

```
SELECT department, SUM(salary) AS total_salary
FROM Employee
GROUP BY department;
```

---

27. What is the difference between RANK(), DENSE_RANK(), and ROW_NUMBER()?

Answer:

- `RANK()` – Skips rankings for ties.
- `DENSE_RANK()` – No gaps in rankings.
- `ROW_NUMBER()` – Assigns unique numbers without ties.

Must learn SQL questions

28. Write a query to update the salary of employees by 10%.

sql
CopyEdit

```
UPDATE Employee
SET salary = salary * 1.1;
```

---

29. What are ACID properties in a database?

Answer:

- **Atomicity** – Transactions are all or nothing.
- **Consistency** – Data remains valid.
- **Isolation** – Transactions do not interfere.
- **Durability** – Changes persist.

---

30. Write a query to delete duplicate records from a table.

sql
CopyEdit

```
DELETE FROM Employee
WHERE id NOT IN (
    SELECT MIN(id)
    FROM Employee
    GROUP BY name, salary, department
);
```