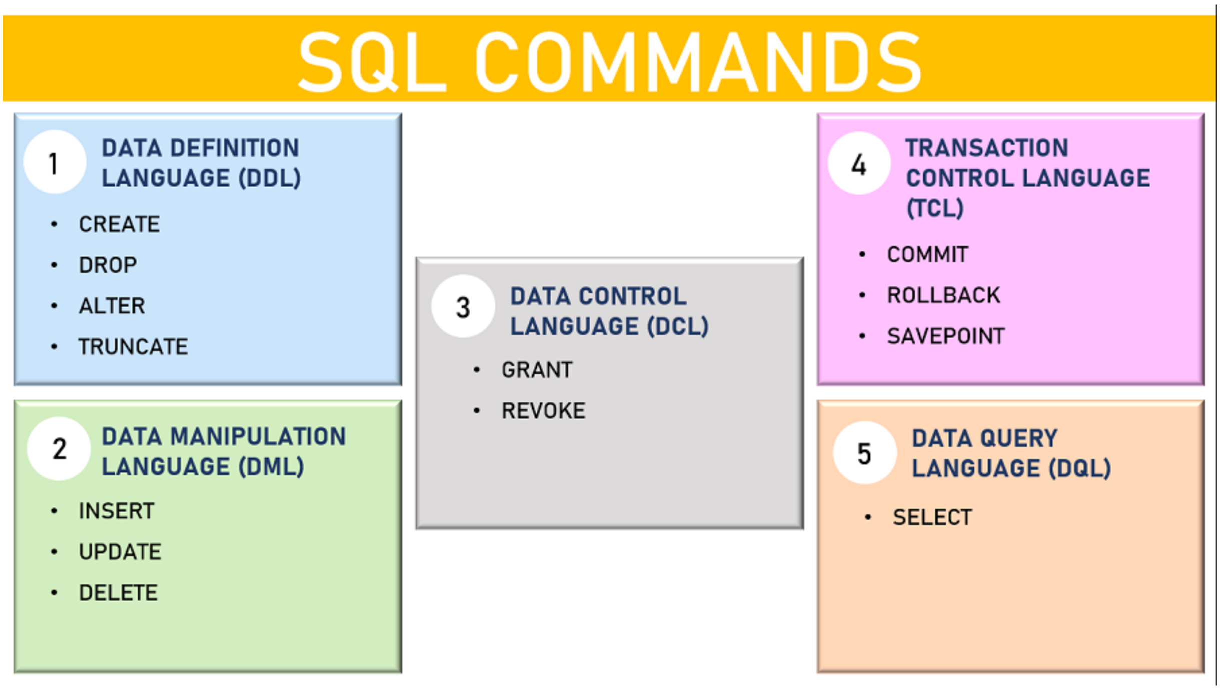
SQL Scenario-Based Interview Questions & Answers

List of some of the most popular SQL interview questions along with sample answers that are commonly asked for data-related positions like Data Analyst, Business Analyst, or Data Engineer roles:

Basic to Intermediate Questions

1. **What is SQL, and what are its main functions?**
   - **Answer**: SQL (Structured Query Language) is a standard language for managing and manipulating relational databases. Its main functions include:
     - **DDL (Data Definition Language)**: Defines database structure (e.g., `CREATE`, `ALTER`, `DROP`).
     - **DML (Data Manipulation Language)**: Manages data in tables (e.g., `SELECT`, `INSERT`, `UPDATE`, `DELETE`).
     - **DCL (Data Control Language)**: Controls data access (e.g., `GRANT`, `REVOKE`).
     - **TCL (Transaction Control Language)**: Manages transactions (e.g., `COMMIT`, `ROLLBACK`, `SAVEPOINT`).



SQL COMMANDS

1 DATA DEFINITION LANGUAGE (DDL)
- CREATE
- DROP
- ALTER
- TRUNCATE

2 DATA MANIPULATION LANGUAGE (DML)
- INSERT
- UPDATE
- DELETE

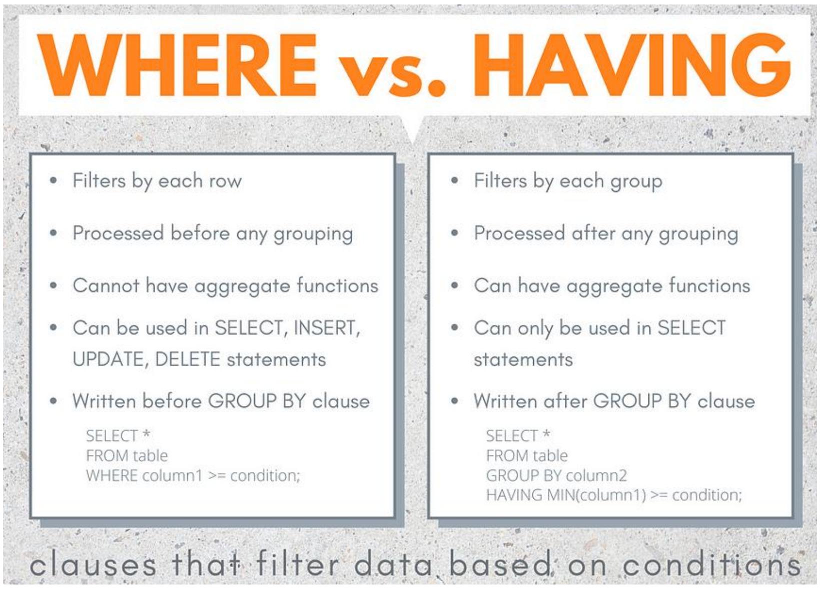
3 DATA CONTROL LANGUAGE (DCL)
- GRANT
- REVOKE

4 TRANSACTION CONTROL LANGUAGE (TCL)
- COMMIT
- ROLLBACK
- SAVEPOINT

5 DATA QUERY LANGUAGE (DQL)
- SELECT

2. What is the difference between `WHERE` and `HAVING` clauses?

- **Answer**: `WHERE` filters rows before aggregation, whereas `HAVING` filters aggregated data. Use `WHERE` with individual rows and `HAVING` with groups created by `GROUP BY`.



WHERE vs. HAVING

- Filters by each row
- Processed before any grouping
- Cannot have aggregate functions
- Can be used in SELECT, INSERT, UPDATE, DELETE statements
- Written before GROUP BY clause

```
SELECT *
FROM table
WHERE column1 >= condition;
```

- Filters by each group
- Processed after any grouping
- Can have aggregate functions
- Can only be used in SELECT statements
- Written after GROUP BY clause

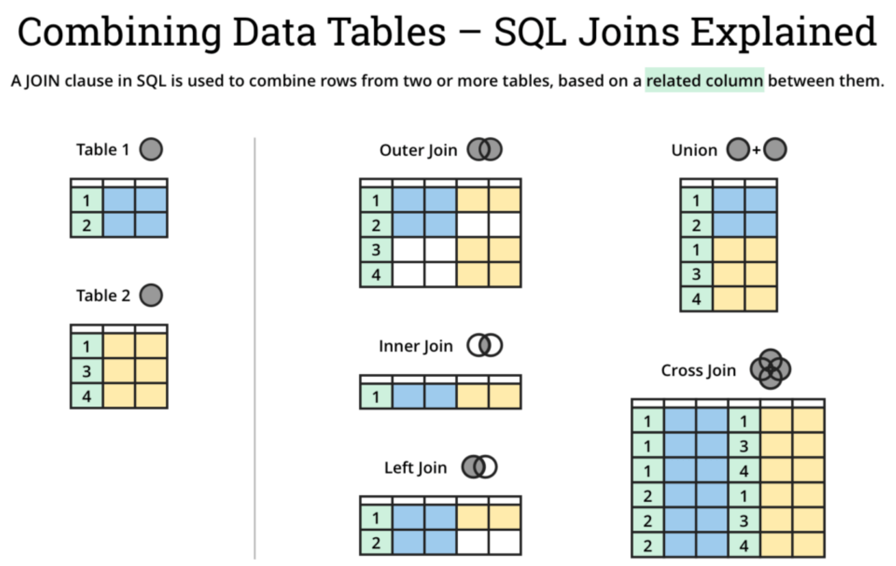
```
SELECT *
FROM table
GROUP BY column2
HAVING MIN(column1) >= condition;
```

clauses that filter data based on conditions

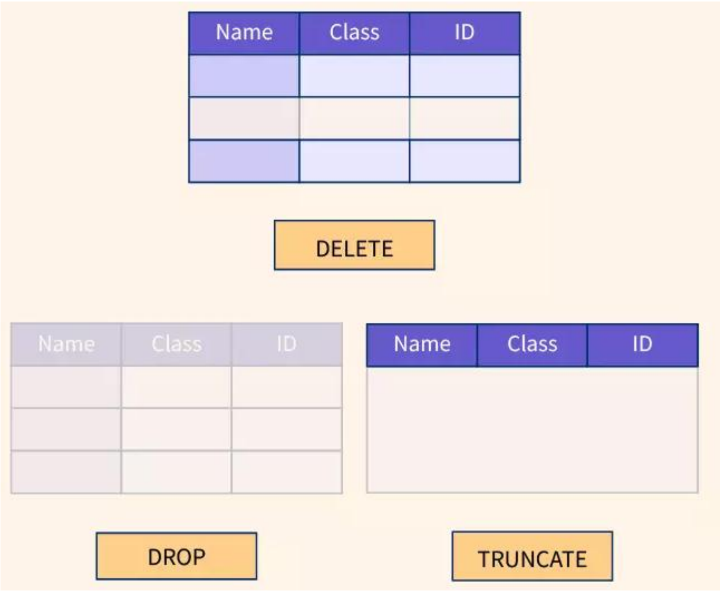
3. How do you use the `JOIN` clause? Explain different types of joins.

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

- **CROSS JOIN**: Produces a Cartesian product of the two tables.

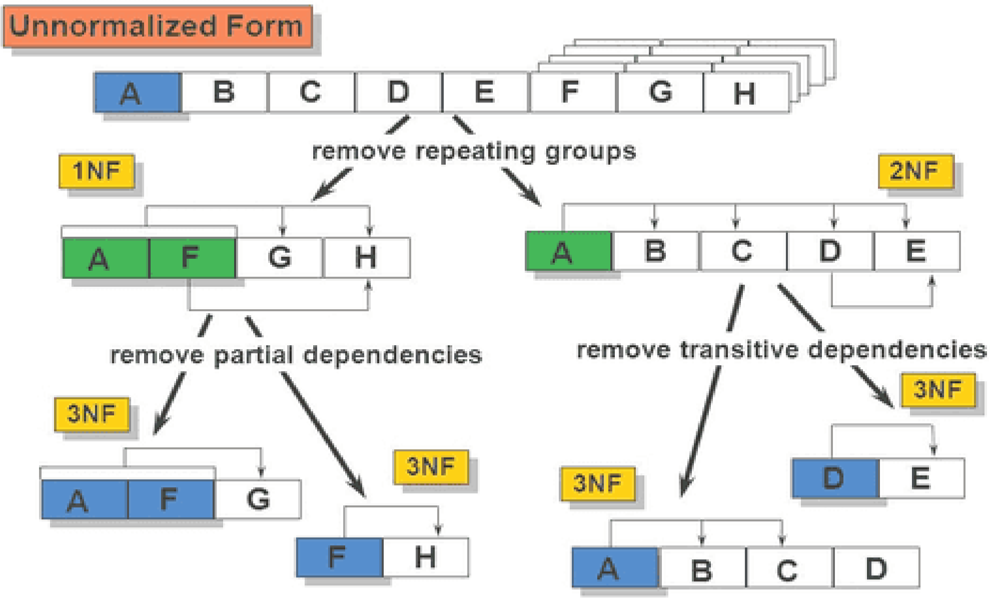

Combining Data Tables – SQL Joins Explained
A JOIN clause in SQL is used to combine rows from two or more tables, based on a related column between them.
Table 1
1
2
Table 2
1
3
4
Outer Join
1
2
3
4
Inner Join
1
Left Join
1
2
Union
1
2
1
3
4
Cross Join
1 1
1 3
1 4
2 1
2 3
2 4

4. <span>**Explain the difference between `DELETE`, `TRUNCATE`, and `DROP` statements.**</span>
    - **Answer**:
        - `DELETE`: Removes rows from a table based on a condition. It can be rolled back.
        - `TRUNCATE`: Removes all rows from a table without logging individual row deletions. Cannot be rolled back.
        - `DROP`: Deletes the entire table, including its structure. It cannot be rolled back.



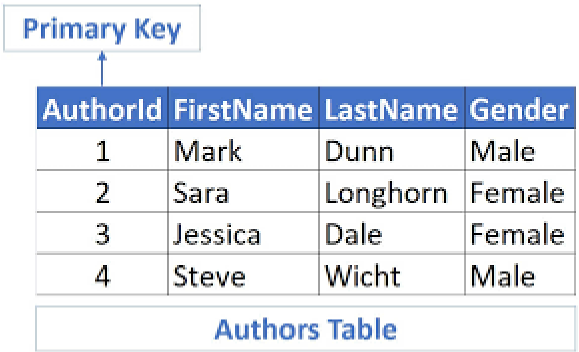
5. What is normalization? Explain its different forms.

- **Answer**: Normalization is the process of organizing data to reduce redundancy. Forms include:
  - **1NF**: Ensures columns hold atomic values and each record is unique.
  - **2NF**: Removes partial dependencies on the primary key.
  - **3NF**: Removes transitive dependencies on non-primary attributes.
  - **BCNF**: Stronger version of 3NF, requiring every determinant to be a candidate key.



6. What is a primary key? Can it have null values?

- **Answer**: A primary key uniquely identifies each row in a table and cannot have null values. It ensures data integrity by enforcing



Primary Key

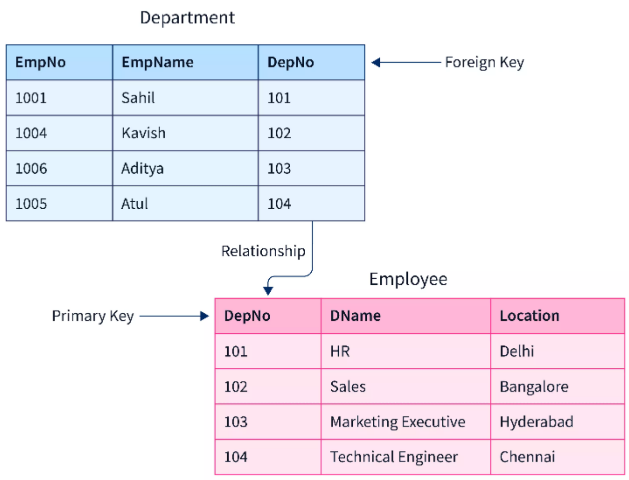
AuthorId	FirstName	LastName	Gender
1	Mark	Dunn	Male
2	Sara	Longhorn	Female
3	Jessica	Dale	Female
4	Steve	Wicht	Male

Authors Table

uniqueness.

7. What is a foreign key?

- **Answer**: A foreign key is a column or set of columns in a table that establishes a link between data in two tables. It enforces referential integrity by linking the foreign key in one table to the primary key in another.



Department

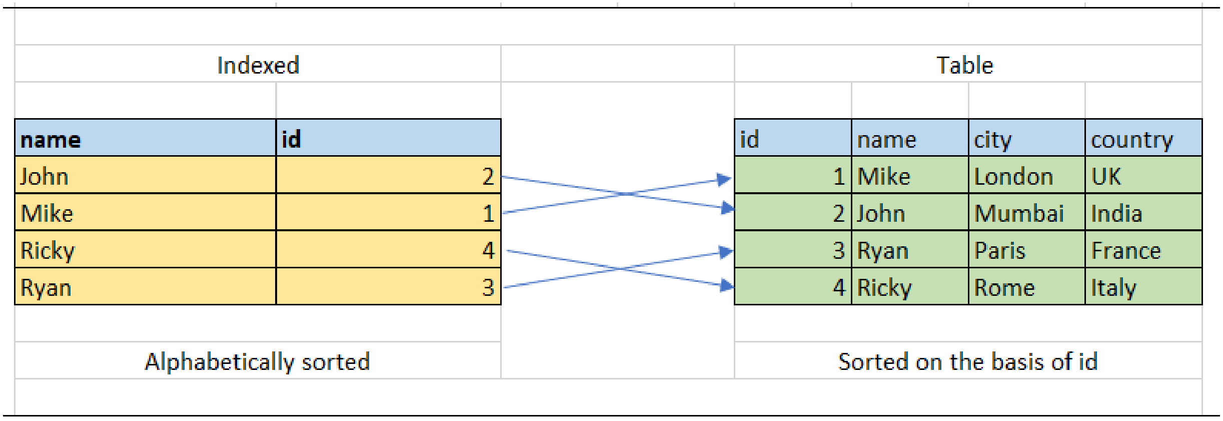
EmpNo	EmpName	DepNo
1001	Sahil	101
1004	Kavish	102
1006	Aditya	103
1005	Atul	104

Employee

DepNo	DName	Location
101	HR	Delhi
102	Sales	Bangalore
103	Marketing Executive	Hyderabad
104	Technical Engineer	Chennai

8. What is an index in SQL, and why is it used?

- **Answer**: An index is a database object that improves the speed of data retrieval operations on a table. It is created on columns that are frequently searched or filtered. However, indexes slow down `INSERT`, `UPDATE`, and `DELETE` operations.



Indexed

name	id
John	2
Mike	1
Ricky	4
Ryan	3

Alphabetically sorted

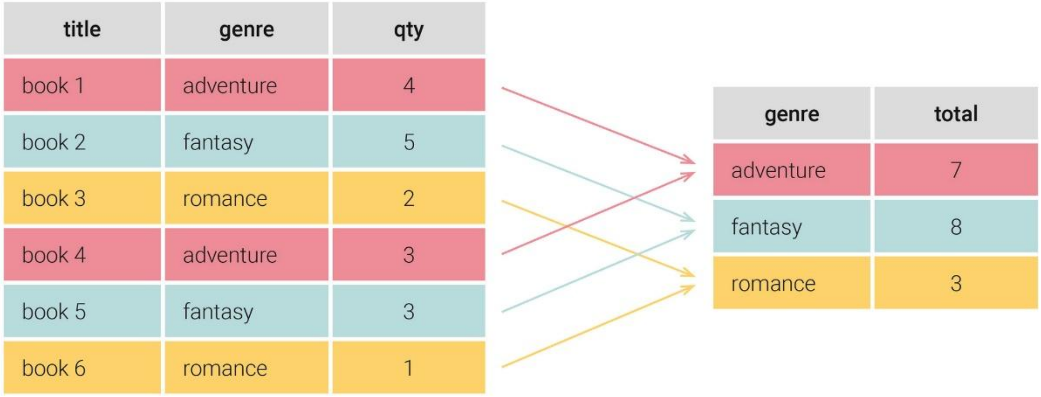
Table

id	name	city	country
1	Mike	London	UK
2	John	Mumbai	India
3	Ryan	Paris	France
4	Ricky	Rome	Italy

Sorted on the basis of id

9. Explain the `GROUP BY` clause and its use with aggregate functions.

- **Answer**: `GROUP BY` groups rows sharing a property so aggregate functions (like `SUM`, `COUNT`, `AVG`, `MIN`, `MAX`) can be applied to each group. For example, to find the total sales per product, you'd use `GROUP BY` on the product column.

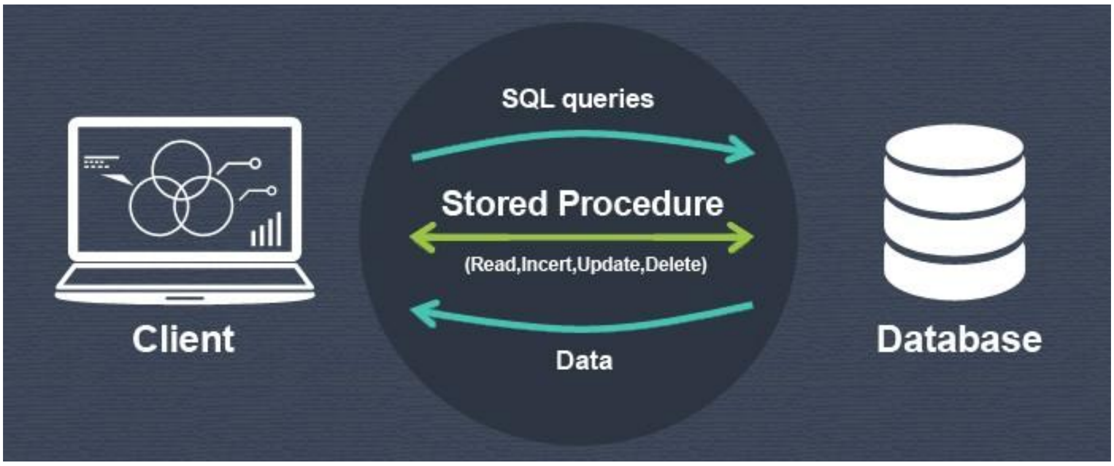


title	genre	qty
book 1	adventure	4
book 2	fantasy	5
book 3	romance	2
book 4	adventure	3
book 5	fantasy	3
book 6	romance	1

genre	total
adventure	7
fantasy	8
romance	3

10. What is a stored procedure?

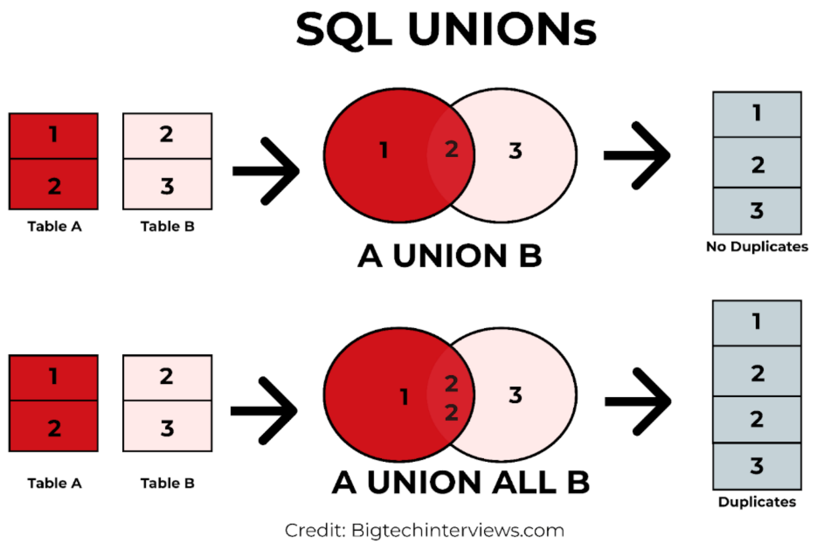
- **Answer**: A stored procedure is a set of SQL statements saved in the database. It can be reused and is typically used for operations like batch processing, data validation, and business logic encapsulation.



Advanced Questions

11. What is the difference between `UNION` and `UNION ALL`?

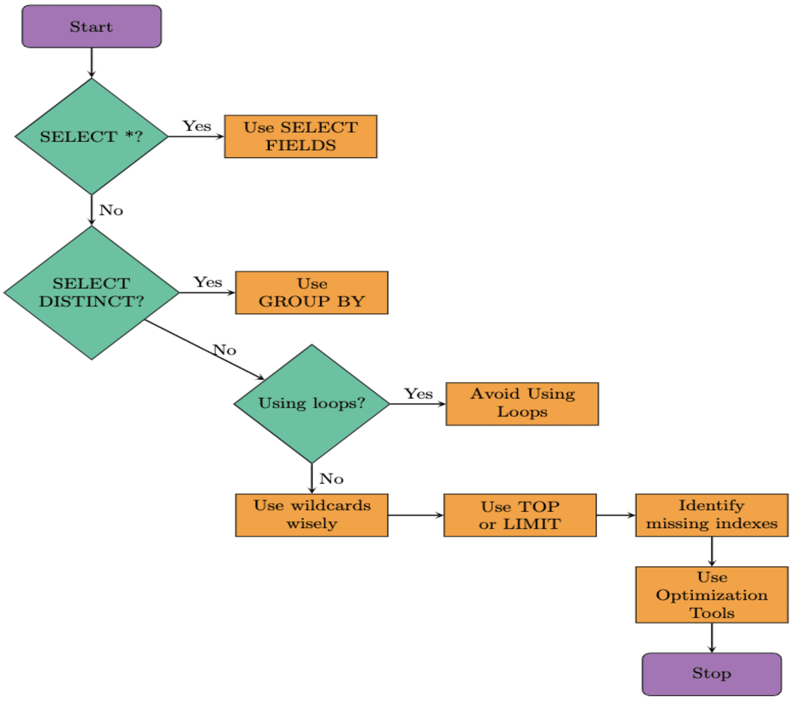
- **Answer**: `UNION` combines results of two queries and removes duplicates, whereas `UNION ALL` includes duplicates.



Credit: Bigtechinterviews.com

12. How do you optimize a SQL query?

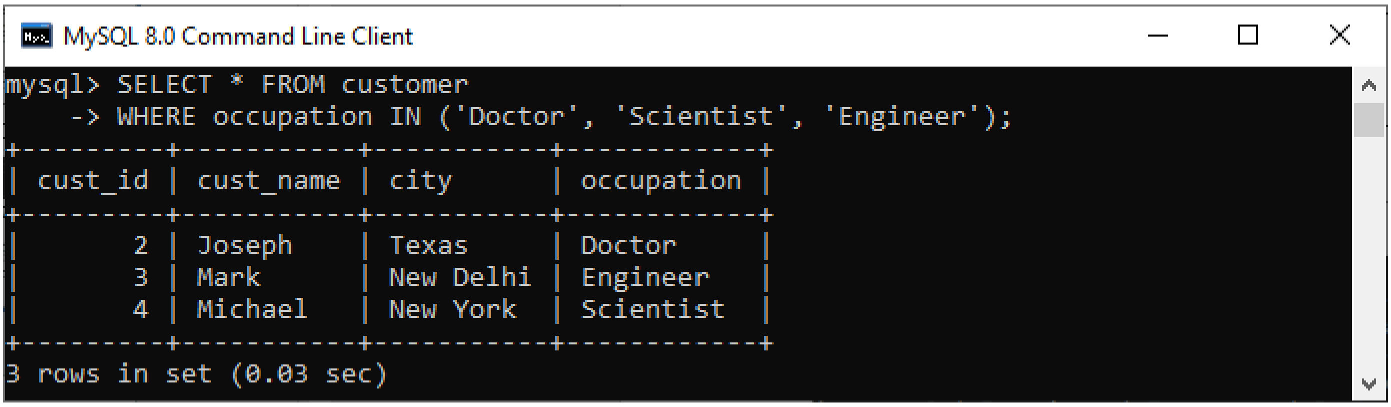
- **Answer**: Common optimization techniques include:
  - Creating indexes on frequently used columns in `JOIN`, `WHERE`, and `ORDER BY` clauses.
  - Using `EXISTS` instead of `IN` for subqueries.
  - Avoiding `SELECT *`; specifying only required columns.



- Analyzing the query execution plan to identify bottlenecks.

13. Explain EXISTS vs. IN.

- **Answer**: IN checks if a value exists within a specified set, while EXISTS checks for the existence of rows in a subquery. EXISTS is generally faster for subqueries.



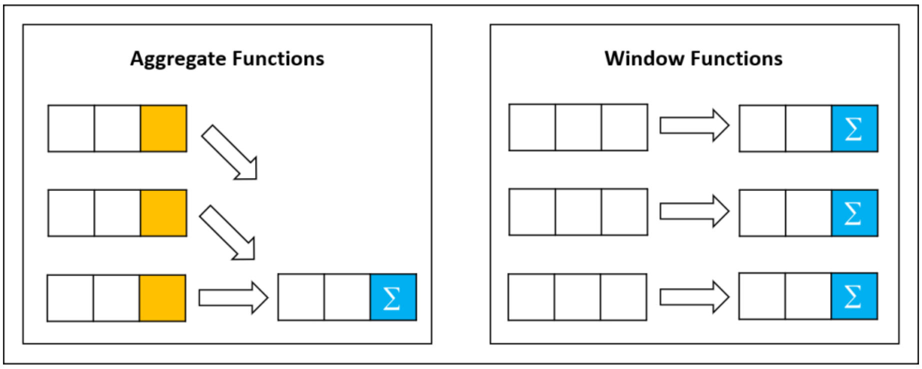
```
mysql> SELECT * FROM customer
    -> WHERE occupation IN ('Doctor', 'Scientist', 'Engineer');
+---------+-----------+-----------+------------+
| cust_id | cust_name | city      | occupation |
+---------+-----------+-----------+------------+
|       2 | Joseph    | Texas     | Doctor     |
|       3 | Mark      | New Delhi | Engineer   |
|       4 | Michael   | New York  | Scientist  |
+---------+-----------+-----------+------------+
3 rows in set (0.03 sec)
```

14. What are window functions, and give an example?

- **Answer**: Window functions perform calculations across a set of rows related to the current row. Examples include ROW_NUMBER, RANK, LEAD, and LAG.

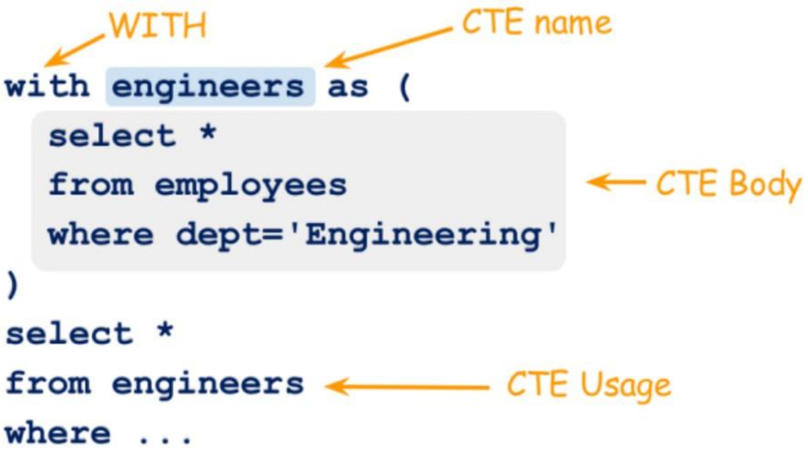
```
SELECT employee_id, department, salary,
       RANK() OVER (PARTITION BY department
ORDER BY salary DESC) as rank
FROM employees;
```

This ranks employees’ salaries within each department.



15. What is a CTE (Common Table Expression)?

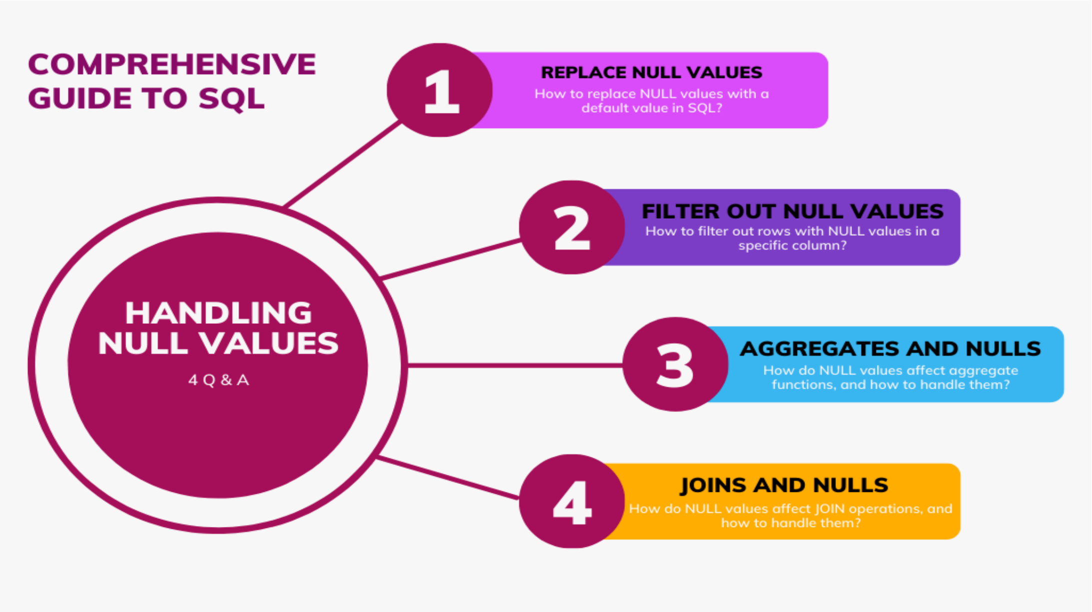
- **Answer**: A CTE is a temporary result set defined in a `WITH` clause, used to simplify complex queries by breaking them into modular parts.



```
with engineers as (
  select *
  from employees
  where dept='Engineering'
)
select *
from engineers
where ...
```

16. How do you handle NULL values in SQL?

- **Answer**: Use `IS NULL` or `IS NOT NULL` to filter rows with null values. Functions like `COALESCE` and `IFNULL` help replace nulls with specified values.



COMPREHENSIVE GUIDE TO SQL

HANDLING NULL VALUES
4 Q & A

1 REPLACE NULL VALUES
How to replace NULL values with a default value in SQL?

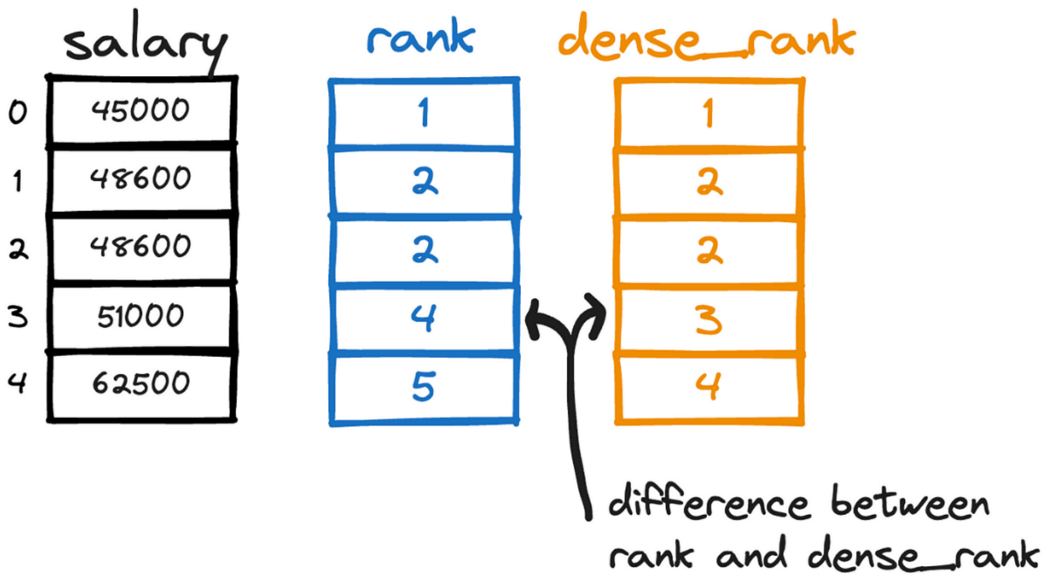
2 FILTER OUT NULL VALUES
How to filter out rows with NULL values in a specific column?

3 AGGREGATES AND NULLS
How do NULL values affect aggregate functions, and how to handle them?

4 JOINS AND NULLS
How do NULL values affect JOIN operations, and how to handle them?

17. Explain the difference between `RANK` and `DENSE_RANK` functions.

- **Answer**: Both are ranking functions. `RANK` assigns ranks with gaps for ties, while `DENSE_RANK` assigns consecutive ranks without gaps.



	salary	rank	dense_rank
0	45000	1	1
1	48600	2	2
2	48600	2	2
3	51000	4	3
4	62500	5	4

difference between
rank and dense_rank

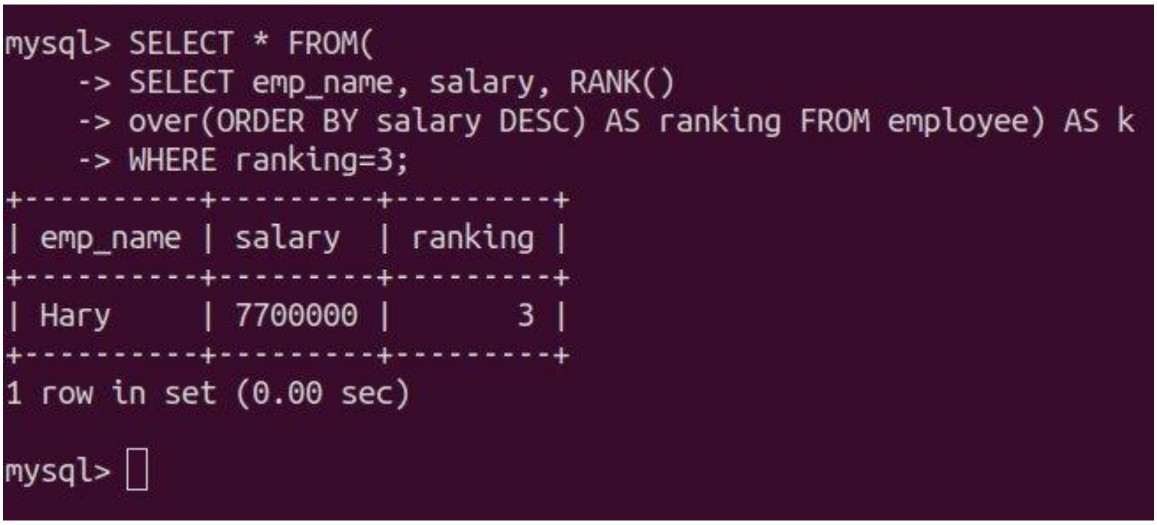
18. **How do you calculate the nth highest salary in a table?**

- **Answer**: You can use the `LIMIT` with offset or subquery.

```
SELECT salary
FROM employees
ORDER BY salary DESC
LIMIT 1 OFFSET (n - 1);
```

Or with a subquery:

```
SELECT salary FROM employees e1
WHERE (n - 1) = (
    SELECT COUNT(DISTINCT salary) FROM
employees e2 WHERE e2.salary > e1.salary
);
```

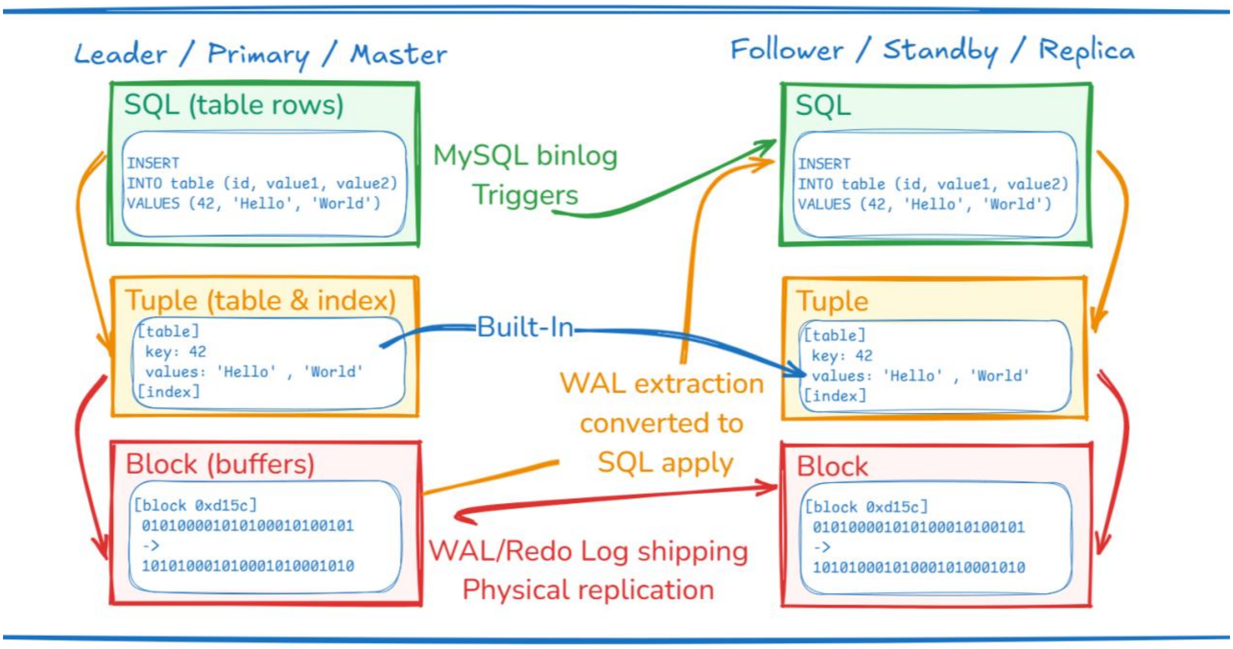


```
mysql> SELECT * FROM(
    -> SELECT emp_name, salary, RANK()
    -> over(ORDER BY salary DESC) AS ranking FROM employee) AS k
    -> WHERE ranking=3;
+----------+---------+---------+
| emp_name | salary  | ranking |
+----------+---------+---------+
| Hary     | 7700000 |       3 |
+----------+---------+---------+
1 row in set (0.00 sec)

mysql> 
```

19. What are triggers in SQL?

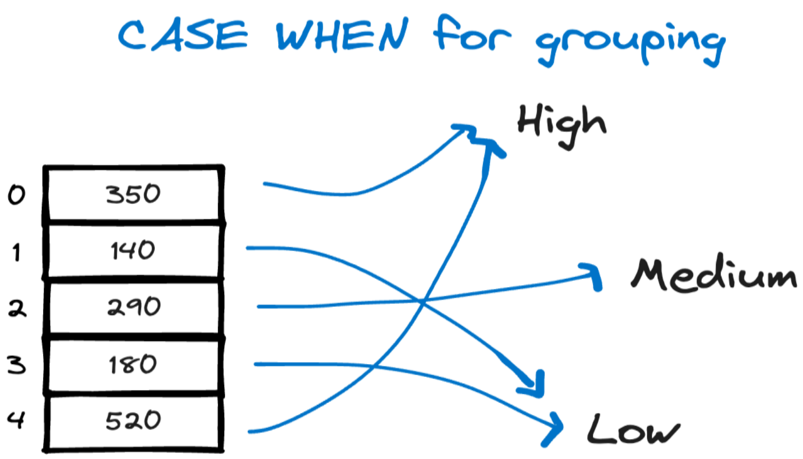
- **Answer**: Triggers are automated actions executed before or after events (`INSERT`, `UPDATE`, `DELETE`) on a table. They are commonly used for auditing, data validation, and enforcing business rules.



20. Explain the use of `CASE` statements.

- **Answer**: `CASE` statements allow conditional logic within SQL.

```
SELECT employee_id,
       salary,
       CASE
           WHEN salary > 50000 THEN 'High'
           WHEN salary BETWEEN 30000 AND
50000 THEN 'Medium'
           ELSE 'Low'
       END AS salary_range
FROM employees;
```



These questions cover core SQL concepts, advanced querying, optimization techniques, and specific SQL functions that are frequently discussed in interviews. Familiarizing yourself with these can help build strong foundations for SQL-based roles.

If you like the content, please consider supporting us by sharing it with others who may benefit. Your support helps us continue creating valuable resources!

Scan any QR using PhonePe App

Aditya chandak