

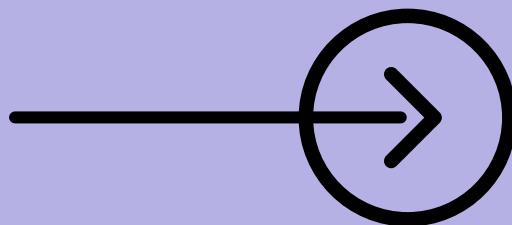
Part - 1

Advanced SQL Interview

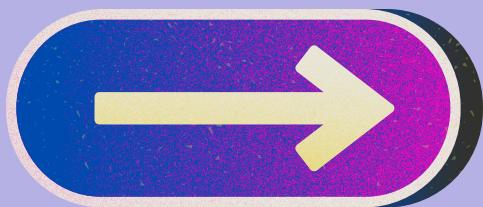
Q & A



Krishan kumar
@Krishan kumar



**How do you find duplicates in a
given table?**



Finding Duplicates:

→ To identify duplicates in a SQL table, you can use the **GROUP BY** and **HAVING** clauses.

This method groups the data based on columns you want to check for duplicates and filters groups with more than one occurrence.



I want example To understand these concept

Example Scenario:

- Suppose you have a table named Employees with columns EmployeeID, FirstName, LastName, and Email.

To find duplicate records based on the Email column, the query would look like this:

```
SELECT Email, COUNT(*)  
FROM Employees  
GROUP BY Email  
HAVING COUNT(*) > 1;
```

- Output:

The result will list duplicate email addresses along with the count of how often each appears in the table.

Getting Detailed Duplicate Records

Detailed Duplicate Records:

→ To view full details for each duplicate row, you can use a **CTE** (**Common Table Expression**) or a **subquery**:

```
WITH DuplicateEmails AS (
    SELECT Email
    FROM Employees
    GROUP BY Email
    HAVING COUNT(*) > 1)
SELECT e.*
FROM Employees e
JOIN DuplicateEmails d ON e.Email = d.Email;
```

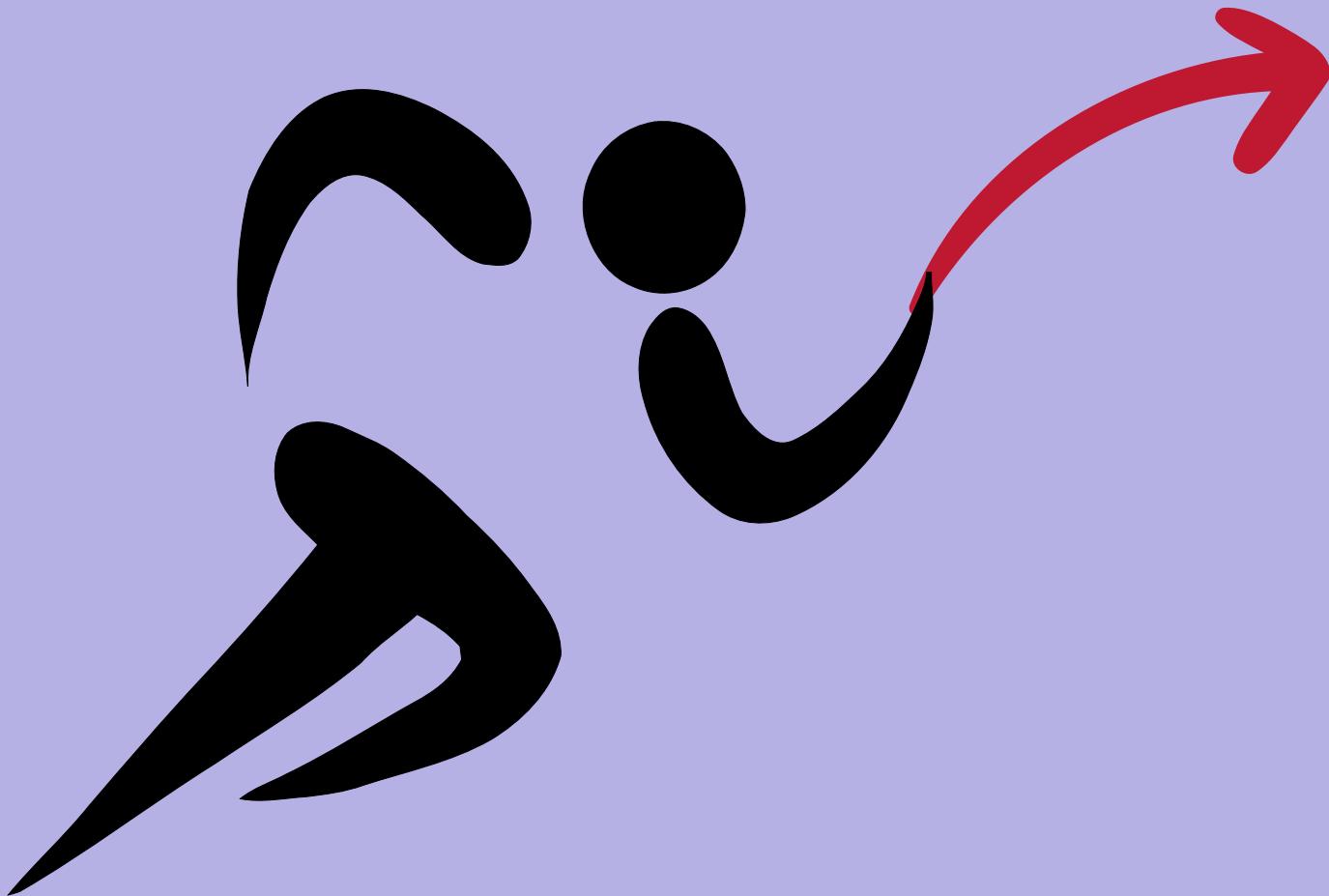


I create some additional Notes

Additional Notes:

→ Notes:

- **Flexible Columns:** To find duplicates based on multiple columns, replace Email with those columns in both the **SELECT** and **GROUP BY** clauses.
- **Performance:** For large tables, ensure your columns used for filtering are indexed to speed up the duplicate detection process.



Swipe right for the jackpot of the day! 😊





wanna see some
Counter Questions



1. Can you find duplicates based on more than one column?

- Yes, just add the additional columns to both **SELECT** and **GROUP BY**.

For example, GROUP BY FirstName, LastName.

Next Question



2. What is the difference between GROUP BY and PARTITION BY when finding duplicates?

→ **GROUP BY** groups the entire result set and is paired with **HAVING** for filtering duplicates.

PARTITION BY is used within window functions and can't directly find duplicates.



Next Question

3. How can you handle duplicate entries in SQL after finding them?

- You can delete duplicates using a **DELETE** with a **ROW_NUMBER()** function in a CTE to isolate duplicates.



Next Question

4. Are there other methods to find duplicates without GROUP BY?

- Yes, using the **ROW_NUMBER()** window function with **PARTITION BY** and filtering rows where **ROW_NUMBER > 1**.



5. Does this approach work with NULL values?

- Yes, GROUP BY treats NULLs as equal, so all NULLs in a column will be grouped together.



you completed one interview question
with me,

can you do me a favour



Find This Useful



**Time to hit that like button
and give it some love! 😍**

Visit my LinkedIn for such amazing Content 😊

[in krishan kumar](#)