

Part - 9

Data Retrieval

Interview

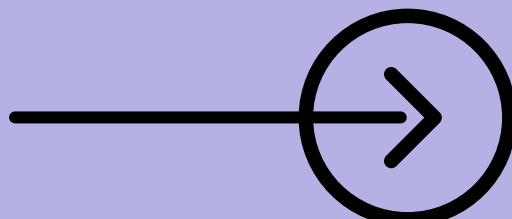
Questions and Answers...!



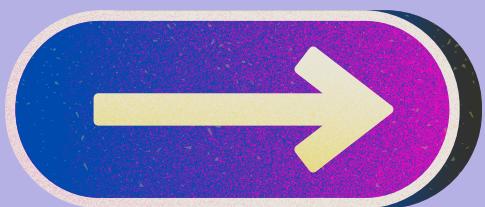
Sharing with
Counter questions ↑



krishna kumar
@Krishan kumar



**What is the difference and
similarity between UNION and
UNION ALL?**



The main difference between UNION and UNION ALL

- **UNION removes duplicate records from the combined result, showing only unique rows,**
while UNION ALL retains all records from both result sets, including duplicates.
- UNION is useful when you want to ensure no duplicate records,**
whereas UNION ALL is better if you want to keep all records, regardless of duplication.



Which one is faster between UNION and UNION ALL?

Who is faster

→ **UNION ALL** is generally faster than **UNION** because it doesn't need to perform the additional step of checking and removing duplicate records.

UNION, on the other hand, takes more time as it has to process and remove duplicates from the final result set.



When should you use **UNION** versus **UNION ALL**?

Where to use:

→ **Use UNION when you need to combine data sets but eliminate duplicate rows. This is often necessary when you want a clean, unique result set.**

Use UNION ALL when you need to combine data sets and keep all records, including duplicates, which can be useful when analyzing raw data or when duplicates are meaningful.



What is the similarity between UNION and UNION ALL?

Similarities:

→ Both **UNION** and **UNION ALL** are used to combine the result sets of two or more **SELECT** queries.

They require the same number of columns in each query, and the data types of these columns must be compatible across all **SELECT** statements for a successful query.



What is a simple example of using **UNION** and **UNION ALL**?

Examples:

→ Example of UNION:

```
SELECT FirstName, LastName FROM Employees
UNION
SELECT FirstName, LastName FROM Managers;
```

- This query combines the Employees and Managers tables but removes any duplicate entries, ensuring each record appears only once in the final result.

→ Example of UNION:

```
SELECT FirstName, LastName FROM Employees
UNION ALL
SELECT FirstName, LastName FROM Managers;
```

- This query combines the Employees and Managers tables and includes all duplicate entries, allowing repeated records to appear in the final result.



wanna see some
Counter Questions



1. What happens if the number of columns is different in the **SELECT statements of a **UNION** query?**

- The query will fail because both **SELECT** statements in a **UNION** or **UNION ALL** must have the same number of columns.

Each **SELECT must return the same number of columns in order to combine the result sets.**

Next Question



2. Can UNION and UNION ALL work with tables having different data types?

→ **No, the columns in the SELECT statements must have compatible data types.**

For example, if the first column is an integer in one SELECT query, it must also be an integer (or a data type that can be converted to integer) in the other.



Next Question

3. Can UNION or UNION ALL change the order of the result?

- **No, the UNION and UNION ALL operations do not sort the results.**

To control the order of the final output, you must use the ORDER BY clause at the end of the UNION or UNION ALL query.



Next Question

4. What is the effect of using UNION on performance compared to UNION ALL?

- **UNION** can slow down performance because it requires additional processing to remove duplicate rows.

In contrast, **UNION ALL** is faster as it doesn't remove duplicates, making it more efficient for large datasets where performance is critical.



5. Can you use aliases in the second SELECT statement in UNION queries?

→ **Column aliases should be defined in the first SELECT statement of a UNION query.**

These aliases are applied to the entire result set, so there's no need to redefine them in subsequent SELECT statements.



**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](#)