

Part - 8

# SQL Basics

## Interview

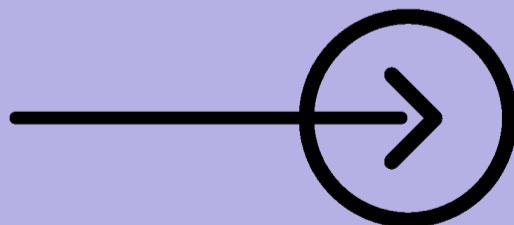
## Questions and Answers...!



Sharing with  
Counter questions ↑



*krishna kumar*  
@Krishan kumar



**What is the difference between  
DELETE and TRUNCATE  
statements?**



## **What is the primary purpose of the DELETE statement in SQL?**

- **The DELETE statement is used to remove specific rows from a table based on a given condition.**

**It allows for selective deletion and maintains logging and triggers,**

**Ensuring that each deletion is recorded and any associated actions are executed.**



**When would you use the TRUNCATE statement instead of DELETE?**

## Use of TRUNCATE statement instead of DELETE

→ You would use the TRUNCATE statement when you need to quickly remove all rows from a table without triggering logging and triggers.

**TRUNCATE is faster than DELETE for removing all records**

**because it doesn't process individual row deletions.**



How does DELETE differ from TRUNCATE in terms of functionality?

## **DELETE differ from TRUNCATE in terms of functionality**

→ **DELETE removes specific rows based on a condition, and it logs each deletion and triggers any associated actions.**

**TRUNCATE, on the other hand, removes all rows from a table without logging individual deletions or triggering actions,**

**and it resets any auto-increment counters.**



**What happens to the table structure after using DELETE and TRUNCATE?**

## **Table structure after using DELETE and TRUNCATE**

→ After using **DELETE**, the table structure remains intact, and the deleted rows are simply removed based on the specified condition.

Similarly, **TRUNCATE** also leaves the table structure intact but removes all rows and resets auto-increment values, if any.



**Can you roll back a **DELETE** operation? What about **TRUNCATE**?**

## **Impact on Roll back a DELETE operation and TRUNCATE?**

→ Yes, a **DELETE** operation can be rolled back if it is part of a transaction that has not been committed yet.

**However, TRUNCATE is generally not reversible and cannot be rolled back,**

**as it is a non-logged operation.**



wanna see some  
**Counter Questions**

## **1. How does TRUNCATE affect performance compared to DELETE??**

→ **TRUNCATE is generally faster than DELETE, especially for large tables, because it doesn't log individual row deletions or trigger associated actions.**

**It deallocates the data pages used by the table directly, leading to better performance for bulk deletions.**



**Next Question**

## **2. Can you use a WHERE clause with TRUNCATE? Why or why not?**

→ **No, you cannot use a WHERE clause with TRUNCATE because it is designed to remove all rows from a table without any condition.**

**If you need conditional deletion,**

**DELETE should be used instead.**



### **3. What are the implications of using TRUNCATE on a table with foreign key constraints?**

→ **TRUNCATE cannot be used on a table with foreign key constraints unless the foreign key constraints are temporarily disabled or removed.**

**This is because TRUNCATE does not allow partial deletions that could lead to referential integrity issues.**



*Next Question*

## **4. How does TRUNCATE handle auto-increment columns differently than DELETE?**

- **TRUNCATE resets the auto-increment counter to the starting value, typically 1, for the table, whereas DELETE does not reset the counter.**

**This means that after a TRUNCATE operation, any new row inserted will have the auto-increment column start from the beginning.**



## **5. In what scenarios is it more appropriate to use DELETE instead of TRUNCATE?**

- **DELETE** is more appropriate when you need to selectively remove rows based on a condition, when you need to trigger actions associated with row deletions, or when you require transaction logging for auditing purposes.

**TRUNCATE** is not suitable for these scenarios as it removes all rows unconditionally and without logging.



**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](#)