SL. NO.	DELETE	DROP	TRUNCATE
1	Used to Delete single or multiple records depending on specific condition.	Delete entire table/View and its data.	Removes complete data, table/view but preserves its structure or scheme.
2	DML	DLL	DLL
3	**WHERE** clause can be used.	**WHERE** clause cannot be used.	**WHERE** clause cannot be used.
4	Need DELETE **permission**		Need ALTER **permission.**
5	Eliminate records one by one		Deletes entire data page containing records.
6	The DELETE command acquires the **lock** on every deleting record; thus, it requires more locks and resources.		The TRUNCATE command requires fewer **locks** and resources before deleting the data page because it acquires the lock on the data page
7	**Slower** than TRUNCATE	**Slower** than TRUNCATE	**Faster** than DELETE
8	It records all the deleted data rows in the **transaction log**.		It records only the deleted data pages in the **transaction log**.
9	You can restore the data using the COMMIT or ROLLBACK command.	It *cannot be rolled back.*	You cannot restore the deleted data after executing this command.
10	The DELETE statement deletes the records and does not interfere with the **table's identity**. (ie. auto-increment primary key)		The TRUNCATE statement does not delete the table structure but **resets the identity** of the table
11	**Integrity constraints** do not get removed in the DELETE command.	Integrity constraints get removed in the DROP command.	Integrity constraint doesn't get removed in the Truncate command.
12	DELETE FROM table_name WHERE condition;	DROP TABLE Shippers;	TRUNCATE TABLE Categories;

Ques. why TRUNCATE operation in SQL is DDL ??

Ans.

1. TRUNCATE directly affects the structure of a table, even though it primarily deletes data. It deallocates all the data pages associated with the table, which is a structural operation rather than a row-by-row operation like DELETE.
2. Since TRUNCATE deallocates storage and resets metadata (like the auto-increment counter in some databases), it affects how the database engine handles the table's storage, which aligns with DDL's purpose of defining and managing schema objects.
3. TRUNCATE is minimally logged, meaning it does not log individual row deletions but rather logs the deallocation of entire pages of data. This behavior is typical of DDL operations, which often bypass or minimize detailed logging compared to DML operations like DELETE, which logs each row change.
4. TRUNCATE is not always transactional, meaning it can't be rolled back in some database systems once executed, unlike DML operations such as DELETE. This non-transactional nature further aligns it with DDL operations.
5. Since TRUNCATE works at the table level and deals with the entire dataset at once, it's faster than DELETE, which operates row-by-row. This bulk operation nature makes it more similar to DDL operations like DROP or CREATE.