

# What's the difference between TRUNCATE and DELETE in SQL



What's the difference between `TRUNCATE` and `DELETE` in SQL?

240

If your answer is platform specific, please indicate that.



sql

database

truncate



106

edited Jun 3 at 16:39



[aioobe](#)

337k

78

714

763

asked Sep 26 '08 at 13:53



[David Aldridge](#)

44.3k

8

51

80

11 The FAQ says it's fine to answer your own question. – [Dave Webb](#) Sep 26 '08 at 13:54

Can you clarify in your question Oracle or SQL Server (only tagged Oracle so far) as this seems to have implementation specific differences. – [Guy](#) Sep 29 '08 at 11:54

1 I took off the Oracle tag. Let it apply to any RDBMS – [David Aldridge](#) Oct 1 '08 at 4:15

3 All the answers are platform-specific. There is no TRUNCATE command in standard SQL. It is therefore a proprietary feature and means a different thing to each DBMS vendor. – [nvogel](#) Jul 8 '11 at 12:54

The answer is very implementation specific, as it must be, since as sqlvogel pointed out, this is a non-standard command (TRUNCATE). Either leave this tagged 'oracle' or let's make it a community-wiki style answer, and put in the consequences for each major RDBMS (Oracle, MS-MSQL, PostgreSQL all implement TRUNCATE...) – [reedstrm](#) Oct 15 '12 at 17:11

## 31 Answers

1

2

next

Here's a list of differences. I've highlighted Oracle-specific features, and hopefully the community can add in other vendors' specific

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook





If you want to quickly delete all of the rows from a table, and you're really sure that you want to do it, and you do not have foreign keys against the tables, then a TRUNCATE is probably going to be faster than a DELETE.

Various system-specific issues have to be considered, as detailed below.

## Statement type

Delete is DML, Truncate is DDL ([What is DDL and DML?](#))

## Commit and Rollback

Variable by vendor

### SQL\*Server

Truncate can be rolled back.

### PostgreSQL

Truncate can be rolled back.

### Oracle

Because a TRUNCATE is DDL it involves two commits, one before and one after the statement execution. Truncate can therefore not be rolled back, and a failure in the truncate process will have issued a commit anyway.

However, see Flashback below.

## Space reclamation

Delete does not recover space, Truncate recovers space

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



## Row scope

Delete can be used to remove all rows or only a subset of rows. Truncate removes all rows.

### Oracle

When a table is partitioned, the individual partitions can be truncated in isolation, thus a partial removal of all the table's data is possible.

## Object types

Delete can be applied to tables and tables inside a cluster. Truncate applies only to tables or the entire cluster. (May be Oracle specific)

## Data Object Identity

### Oracle

Delete does not affect the data object id, but truncate assigns a new data object id *unless* there has never been an insert against the table since its creation. Even a single insert that is rolled back will cause a new data object id to be assigned upon truncation.

## Flashback (Oracle)

Flashback works across deletes, but a truncate prevents flashback to states prior to the operation.

However, from 11gR2 the FLASHBACK ARCHIVE feature allows this, except in Express Edition

[Use of FLASHBACK in Oracle](http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm#ADFNS638) [http://docs.oracle.com/cd/E11882\\_01/appdev.112/e41502/adfns\\_flashback.htm#ADFNS638](http://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_flashback.htm#ADFNS638)

## Privileges

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook



Delete can be granted on a table to another user or role, but truncate cannot be without using a DROP ANY TABLE grant.

## Redo/Undo

Delete generates a small amount of redo and a large amount of undo. Truncate generates a negligible amount of each.

## Indexes

### Oracle

A truncate operation renders unusable indexes usable again. Delete does not.

## Foreign Keys

A truncate cannot be applied when an enabled foreign key references the table. Treatment with delete depends on the configuration of the foreign keys.

## Table Locking

### Oracle

Truncate requires an exclusive table lock, delete requires a shared table lock. Hence disabling table locks is a way of preventing truncate operations on a table.

## Triggers

DML triggers do not fire on a truncate.

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook



## Remote Execution

### Oracle

Truncate cannot be issued over a database link.

## Identity Columns

### SQL\*Server

Truncate resets the sequence for IDENTITY column types, delete does not.

## Result set

In most implementations, a `DELETE` statement can return to the client the rows that were deleted.

e.g. in an Oracle PL/SQL subprogram you could:

```
DELETE FROM employees_temp
WHERE     employee_id = 299
RETURNING first_name,
          last_name
INTO      emp_first_name,
          emp_last_name;
```

edited Jun 3 at 16:41

community wiki  
14 revs, 4 users 99%  
David Aldridge

---

Don't understand your 4th statement: if I say `DELETE [ ] FROM Table; then *all` rows in that table will be deleted unless a FK stops it. By the way, I guess this is SQL Server-specific, you can't use TRUNCATE on tables with FKs. – [Joe Pineda](#) Mar 27 '09 at 13:32

---

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



- 3 DELETE returns number of rows deleted, but TRUNCATE does not. It is very silly point but worth mentioning it :) – [Deepak Kumar Jha](#) Jan 12 '15 at 14:08

The difference between truncate and delete is listed below:

159

| Truncate  | Delete   |
|---|--|
| We can't Rollback after performing Truncate.  | We can Rollback after delete.  |
| Example:<br>BEGIN TRAN<br>TRUNCATE TABLE tranTest<br>SELECT * FROM tranTest<br>ROLLBACK<br>SELECT * FROM tranTest | Example:<br>BEGIN TRAN<br>DELETE FROM tranTest<br>SELECT * FROM tranTest<br>ROLLBACK<br>SELECT * FROM tranTest |
| Truncate reset identity of table.   | Delete does not reset identity of table.   |
| It locks the entire table.  | It locks the table row.  |

Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up with email](#)[Sign up with Google](#)[Sign up with Facebook](#)

|   |                                      |  |
|---|--------------------------------------|--|
| + |                                      |  |
|   | We can't use WHERE clause with it.   | We can use WHERE to filter data to delete. |
|   |                                      |  |
| + |                                      |  |
| + |                                      |  |
|   | Trigger is not fired while truncate. | Trigger is fired.                          |
|   |                                      |  |
| + |                                      |  |
| + |                                      |  |
|   | Syntax :                             | Syntax :                                   |
|   |                                      |  |
|   | 1) TRUNCATE TABLE table_name         | 1) DELETE FROM table_name                  |
|   |                                      | 2) DELETE FROM table_name WHERE            |
|   |                                      | example_column_id IN (1,2,3)               |
|   |                                      |  |
| + |                                      |  |
| + |                                      |  |

edited Sep 1 '17 at 13:55



Naresh Ravlani

1,330 10 24

answered Oct 15 '12 at 17:07



Bhaumik Patel

11.3k 5 22 31

*Truncate reset identity of table* What does it mean ? But, **What about Delete ?** – [Ravi](#) Jul 11 '15 at 8:29

- 1 @jWeaver: It means when you set Identity Specification property to True for Primary key field, so when you insert data into that table, primary key column holds value like 1,2,3,4,5....(if Identity starts from 1 and seed is 1), and when you truncate table it will lost all identity value, so when you start inserting data into that table again it will starts from 1 instead of where it is left last. In DELETE, it's reverse, it preserves identity value even after you execute DELETE statement. Sorry for 2nd comparison point mistake in DELETE column in above image. – [Bhaumik Patel](#) Jul 13 '15 at 23:42 ✎

Seems like you are answering for **SQL SERVER** – [Ravi](#) Jul 14 '15 at 3:05

- 1 Both TRUNCATE and DELETE can be rolledBack in **SQL SERVER**. And in the 2nd row DELETE dose not reset identity. Now how can you edit this post? That's the bad thing about using pictures in StackOverflow. – [Mahmood Jenami](#) Aug 3 '16 at 8:05 ✎

Truncate is being rolled back! (SQL SERVER) – [Aimal Khan](#) Aug 24 '18 at 22:33 ✎

**Join Stack Overflow** to learn, share knowledge, and build your career.

[Sign up with email](#)

[Sign up with Google](#)
[Sign up with Facebook](#)


## TRUNCATE

TRUNCATE removes all rows from a table. The operation cannot be rolled back and no triggers will be fired. As such, TRUNCATE is faster and doesn't use as much undo space as a DELETE.

## DELETE

The DELETE command is used to remove rows from a table. A WHERE clause can be used to only remove some rows. If no WHERE condition is specified, all rows will be removed. After performing a DELETE operation you need to COMMIT or ROLLBACK the transaction to make the change permanent or to undo it. Note that this operation will cause all DELETE triggers on the table to fire.

From: [http://www.orafaq.com/faq/difference\\_between\\_truncate\\_delete\\_and\\_drop\\_commands](http://www.orafaq.com/faq/difference_between_truncate_delete_and_drop_commands)

edited Apr 11 '17 at 0:33



Pang

7,154 16 68 108

answered Apr 15 '15 at 8:02



Mohit Singh

3,926 1 15 21

3 Nice explanation for a beginner like me – [Vikas Kukreti](#) May 3 '15 at 17:25

All good answers, to which I must add:

23

Since TRUNCATE TABLE is a DDL ([Data Definition Language](#)), not a DML ([Data Manipulation Language](#)) command, the Delete Triggers do not run.

edited Jan 22 '16 at 22:16



Dawid Ferenczy  
Rogożan

12.2k 8 50 55

answered Sep 26 '08 at 14:28



polara

2,622 5 18 19

Ah, triggers ... that's a good point. I'll add that to the list I made and credit you Polara, if that's OK. – [David Aldridge](#) Sep 26 '08 at 14:30

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook





## Summary of Delete Vs Truncate in SQL server

For Complete Article follow this link : <http://codaffection.com/sql-server-article/delete-vs-truncate-in-sql-server/>

19

|                                 | Delete               | Truncate                                |
|---------------------------------|----------------------|---|
| Command Type                    | DML                  | DDL                                     |
| Where Condition                 | support              | does not support                        |
| Reset Identity Column           | no                   | yes                                     |
| Acquired lock                   | row lock             | table and page lock                     |
| Transaction log                 | for each deleted row | one log indicating deallocation of page |
| Performance(Speed of execution) | slow                 | much faster than Delete                 |

Taken from dotnet mob article : [Delete Vs Truncate in SQL Server](#)

edited Aug 12 '18 at 3:32

answered May 14 '16 at 16:07



**Shamseer K**

**3,121** 1 19 27

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook





Dawid Ferenczy  
Rogożan

12.2k 8 50 55



27.6k 3 53 88

Is this SQLSERVER? – [David Aldridge](#) Sep 26 '08 at 14:11

To clarify, this is for SQL Server if the table has a column defined as IDENTITY. Delete would maintain the last auto-assigned ID, while Truncate resets the counter. – [Codewerks](#) Sep 27 '08 at 14:10

As the question is tagged ORACLE, then this answer is WRONG, therefore downvoted. – [Guy](#) Sep 29 '08 at 11:53

oops, didn't see the oracle tag :) – [mathieu](#) Sep 29 '08 at 15:00

+1 true, and it resets it to **0**. If you want it to be **1** instead: DBCC CHECKIDENT (table\_name, RESEED, 1) – [JohnB](#) Aug 6 '10 at 16:57



"Truncate doesn't log anything" is correct. I'd go further:

12

Truncate is not executed in the context of a transaction.



The speed advantage of truncate over delete should be obvious. That advantage ranges from trivial to enormous, depending on your situation.

However, I've seen truncate unintentionally break referential integrity, and violate other constraints. The power that you gain by modifying data outside a transaction has to be balanced against the responsibility that you inherit when you walk the tightrope without a net.

answered Sep 26 '08 at 14:12



[Walter Mitty](#)

15k 2 21 52



Yes, DELETE is slower, TRUNCATE is faster. Why?

DELETE must read the records, check constraints, update the block, update indexes, and generate redo/undo. All of that takes time.

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook



answered Sep 27 '08 at 13:46



DCookie

36k 11 65 82

PostgreSQL is similar to this as well. – [Gavin M. Roy](#) Sep 27 '08 at 14:00

6

TRUNCATE is the DDL statement whereas DELETE is a DML statement. Below are the differences between the two:

1. As TRUNCATE is a DDL ([Data definition language](#)) statement it does not require a commit to make the changes permanent. And this is the reason why rows deleted by truncate could not be rolled back. On the other hand DELETE is a DML ([Data manipulation language](#)) statement hence requires explicit commit to make its effect permanent.
2. TRUNCATE always removes all the rows from a table, leaving the table empty and the table structure intact whereas DELETE may remove conditionally if the where clause is used.
3. The rows deleted by TRUNCATE TABLE statement cannot be restored and you can not specify the where clause in the TRUNCATE statement.
4. TRUNCATE statements does not fire triggers as opposed of on delete trigger on DELETE statement

[Here](#) is the very good link relevant to the topic.

edited Jan 22 '16 at 22:27

Dawid Ferenczy  
Rogożan

12.2k 8 50 55

answered Dec 9 '09 at 11:31



Sachin Chourasiya

7,887 27 75 93

5

If accidentally you removed all the data from table using Delete/Truncate. You can rollback committed transaction. Restore the last backup and run transaction log till the time when Delete/Truncate is about to happen.

The related information below is from [a blog post](#):

Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up with email](#)[Sign up with Google](#)[Sign up with Facebook](#)

- Delete is a DML command.
- Delete statement is executed using a row lock, each row in the table is locked for deletion.
- We can specify filters in where clause.
- It deletes specified data if where condition exists.
- Delete activities a trigger because the operation are logged individually.
- Slower than Truncate because it Keeps logs

### Truncate

- Truncate is a DDL command.
- Truncate table always lock the table and page but not each row. As it removes all the data.
- Cannot use Where condition.
- It Removes all the data.
- Truncate table cannot activate a trigger because the operation does not log individual row deletions.
- Faster in performance wise, because it doesn't keep any logs.

Note: Delete and Truncate both can be rolled back when used with Transaction. If Transaction is done, means committed then we can not rollback Truncate command, but we can still rollback Delete command from Log files, as delete write records them in Log file in case it is needed to rollback in future from log files.

If you have a Foreign key constraint referring to the table you are trying to truncate, this won't work even if the referring table has no data in it. This is because the foreign key checking is done with DDL rather than DML. This can be got around by temporarily disabling the foreign key constraint(s) to the table.

Delete table is a logged operation. So the deletion of each row gets logged in the transaction log, which makes it slow. Truncate table also deletes all the rows in a table, but it won't log the deletion of each row instead it logs the deallocation of the data pages of the table, which makes it faster.

~ If accidentally you removed all the data from table using Delete/Truncate. You can rollback committed transaction. Restore the last backup and run transaction log till the time when Delete/Truncate is about to happen.

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook



1 Thank @Lucas: he inlined the content. I just put it in a blockquote to make it apparent that it was from a different source. – AstroCB Aug 21 '14 at 20:36

In SQL Server 2005 I believe that you **can** rollback a truncate

4

answered Mar 27 '09 at 12:26

Xander

1 yes i tested it – Ogrish Man Jun 9 '10 at 8:04

I'll add that in to the community answer – David Aldridge Jun 23 '15 at 9:36

TRUNCATE can be rolled back if wrapped in a transaction.

3

Please see the two references below and test yourself:-

<http://blog.sqlauthority.com/2007/12/26/sql-server-truncate-cant-be-rolled-back-using-log-files-after-transaction-session-is-closed/>

[http://sqlblog.com/blogs/kalen\\_delaney/archive/2010/10/12/tsql-tuesday-11-rolling-back-truncate-table.aspx](http://sqlblog.com/blogs/kalen_delaney/archive/2010/10/12/tsql-tuesday-11-rolling-back-truncate-table.aspx)

The TRUNCATE vs. DELETE is one of the infamous questions during SQL interviews. Just make sure you explain it properly to the Interviewer or it might cost you the job. The problem is that not many are aware so most likely they will consider the answer as wrong if you tell them that YES Truncate can be rolled back.

edited Aug 21 '14 at 17:21

answered Aug 21 '14 at 14:55



SQLnbe

637 8 11

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



The DELETE command is used to remove rows from a table. A WHERE clause can be used to only remove some rows. If no WHERE condition is specified, all rows will be removed. After performing a DELETE operation you need to COMMIT or ROLLBACK the transaction to make the change permanent or to undo it. Note that this operation will cause all DELETE triggers on the table to fire.

## TRUNCATE

TRUNCATE removes all rows from a table. The operation cannot be rolled back and no triggers will be fired. As such, TRUNCATE is faster and doesn't use as much undo space as a DELETE.

## DROP

The DROP command removes a table from the database. All the table's rows, indexes and privileges will also be removed. No DML triggers will be fired. The operation cannot be rolled back.

DROP and TRUNCATE are DDL commands, whereas DELETE is a DML command. Therefore DELETE operations can be rolled back (undone), while DROP and TRUNCATE operations cannot be rolled back.

From: [http://www.orafaq.com/faq/difference\\_between\\_truncate\\_delete\\_and\\_drop\\_commands](http://www.orafaq.com/faq/difference_between_truncate_delete_and_drop_commands)

edited Apr 11 '17 at 0:36



Pang

7,154 16 68 108

answered Apr 7 '15 at 9:33



MyUserQuestion

120 8

Does this add something to the current answers? – David Aldridge Apr 7 '15 at 11:06

A small correction to the original answer - delete also generates significant amounts of redo (as undo is itself protected by redo). This can be seen from autotrace output:

2

```
SQL> delete from t1;
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook



```

0      DELETE STATEMENT Optimizer=FIRST_ROWS (Cost=43 Card=1)
1      0      DELETE OF 'T1'
2      1      TABLE ACCESS (FULL) OF 'T1' (TABLE) (Cost=43 Card=1)

```

### Statistics

```

-----
30 recursive calls
12118 db block gets
213 consistent gets
142 physical reads
3975328 redo size
441 bytes sent via SQL*Net to client
537 bytes received via SQL*Net from client
4 SQL*Net roundtrips to/from client
2 sorts (memory)
0 sorts (disk)
10918 rows processed

```

answered Sep 27 '08 at 15:26



CaptainPicard

728 6 8

its old thread but as per my understanding truncate only generate good amount of redo log where as delete generates undo and redo both. –  
Saurabh Sinha Aug 3 '16 at 11:35

Here is my detailed answer on [the difference between DELETE and TRUNCATE in SQL Server](#)

2

- **Remove Data** : First thing first, both can be used to remove the rows from table. But a DELETE can be used to remove the rows not only from a Table but also from a VIEW or the result of an OPENROWSET or OPENQUERY subject to provider capabilities.
- **FROM Clause** : With DELETE you can also delete rows from one table/view/rowset\_function\_limited based on rows from another table by using another FROM clause. In that FROM clause you can also write normal JOIN conditions. Actually you can create a DELETE

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook



- **WHERE** : A TRUNCATE cannot have WHERE Conditions, but a DELETE can. That means with TRUNCATE you can't delete a specific row or specific group of rows. TRUNCATE TABLE is similar to the DELETE statement with no WHERE clause.
- **Performance** : TRUNCATE TABLE is faster and uses fewer system and transaction log resources. And one of the reason is locks used by either statements. The DELETE statement is executed using a row lock, each row in the table is locked for deletion. TRUNCATE TABLE always locks the table and page but not each row.
- **Transaction log** : DELETE statement removes rows one at a time and makes individual entries in the transaction log for each row. TRUNCATE TABLE removes the data by deallocating the data pages used to store the table data and records only the page deallocations in the transaction log.
- **Pages** : After a DELETE statement is executed, the table can still contain empty pages. TRUNCATE removes the data by deallocating the data pages used to store the table data.
- **Trigger** : TRUNCATE does not activate the delete triggers on the table. So you must be very careful while using TRUNCATE. One should never use a TRUNCATE if delete Trigger is defined on the table to do some automatic cleanup or logging action when rows are deleted.
- **Identity Column** : With TRUNCATE if the table contains an identity column, the counter for that column is reset to the seed value defined for the column. If no seed was defined, the default value 1 is used. DELETE doesn't reset the identity counter. So if you want to retain the identity counter, use DELETE instead.
- **Replication** : DELETE can be used against table used in transactional replication or merge replication. While TRUNCATE cannot be used against the tables involved in transactional replication or merge replication.
- **Rollback** : DELETE statement can be rolled back. TRUNCATE can also be rolled back provided it is enclosed in a TRANSACTION block and session is not closed. Once session is closed you won't be able to Rollback TRUNCATE.
- **Restrictions** : The DELETE statement may fail if it violates a trigger or tries to remove a row referenced by data in another table with a FOREIGN KEY constraint. If the DELETE removes multiple rows, and any one of the removed rows violates a trigger or constraint, the statement is canceled, an error is returned, and no rows are removed. And if DELETE is used against View, that View must be an Updatable view. TRUNCATE cannot be used against the table used in Indexed view. TRUNCATE cannot be used against the table referenced by a FOREIGN KEY constraint, unless a table that has a foreign key that references itself

---

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email

 Sign up with Google

Sign up with Facebook





Hi Mangal -- thanks for the answer, particularly for SQL\*Server specific issues. Do you think you could integrate those points in with the Community Wiki answer that was accepted? – [David Aldridge](#) Nov 5 '15 at 12:48

Yes sure, but how and where? Sorry I'm new here. – [Mangal Pardeshi](#) Nov 6 '15 at 3:40

"So if you want to retain the identity counter, use DELETE" could you use `DECLARE @ai as bigint SET @ai =IDENT_CURRENT('tablename')`  
`TRUNCATE TABLE tablename DBCC checkident('tablename', RESEED, @ai)` – [mpag](#) Nov 4 '16 at 23:43



The biggest difference is that truncate is non logged operation while delete is.

1

Simply it means that in case of a database crash , you cannot recover the data operated upon by truncate but with delete you can.



More details [here](#)

answered Sep 26 '08 at 13:56



[Learning](#)

6,884 3 25 45



DELETE Statement: This command deletes only the rows from the table based on the condition given in the where clause or deletes all the rows from the table if no condition is specified. But it does not free the space containing the table.

1

The Syntax of a SQL DELETE statement is:



DELETE FROM table\_name [WHERE condition];

TRUNCATE statement: This command is used to delete all the rows from the table and free the space containing the table.

answered Jun 26 '13 at 22:15



[Bhushan Patil](#)

11 1

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



**Delete** = Only Delete- so it can be rolled back  
**In DELETE** you can write conditions **using WHERE** clause  
Syntax - **Delete from [Table] where [Condition]**

## TRUNCATE

**TRUNCATE is** a DDL command  
You can't **rollback** in **TRUNCATE**, **TRUNCATE** removes the record permanently  
**Truncate = Delete+Commit** -so we can't roll back  
You can't **use** conditions(**WHERE** clause) **in TRUNCATE**  
Syntax - **Truncate table [Table]**

For more details visit

<http://www.zilckh.com/what-is-the-difference-between-truncate-and-delete/>

edited Sep 7 '13 at 12:46



Luc M

12.3k 12 60 81

answered Sep 7 '13 at 12:18



user27332

2,919 1 7 2

One further difference of the two operations is that if the table contains an identity column, the counter for that column is reset 1 (or to the seed value defined for the column) under **TRUNCATE**. **DELETE** does not have this affect.

1

answered Oct 6 '14 at 9:44



westyside

59 3

In short, truncate doesn't log anything (so is much faster but can't be undone) whereas delete is logged (and can be part of a larger transaction, will rollback etc). If you have data that you don't want in a table in dev it is normally better to truncate as you don't run the risk of filling up the transaction log

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook





0



A big reason it is handy, is when you need to refresh the data in a multi-million row table, but don't want to rebuild it. "Delete \*" would take forever, whereas the performance impact of Truncate would be negligible.

answered Sep 26 '08 at 14:17

[Jordan Ogren](#)

71 1 2 9



0



Can't do DDL over a dblink.

answered Sep 26 '08 at 23:44

joel garry



0



I'd comment on matthieu's post, but I don't have the rep yet...

In MySQL, the auto increment counter gets reset with truncate, but not with delete.

answered Sep 27 '08 at 1:08

[nathan](#)

3,942 2 22 28



0



It is not that truncate does not log anything in SQL Server. truncate does not log any information but it log the deallocation of data page for the table on which you fired TRUNCATE.

and truncated record can be rollback if we define transaction at beginning and we can recover the truncated record after rollback it. But can not recover truncated records from the transaction log backup after committed truncated transaction.

answered Jan 25 '14 at 7:19

**Join Stack Overflow** to learn, share knowledge, and build your career.

[Sign up with email](#)[Sign up with Google](#)[Sign up with Facebook](#)

Truncate can also be Rolled back here the example

0

```
begin Tran
delete from Employee

select * from Employee
Rollback
select * from Employee
```

edited Jul 30 '14 at 8:38



Robert

4,108 13 55 111

answered Jul 30 '14 at 8:32



Vinay Pandit

1

0

Truncate and Delete in SQL are two commands which are used to remove or delete data from a table. Though quite basic in nature both SQL commands can create a lot of trouble until you are familiar with details before using it. An incorrect choice of command can result in either a very slow process or can even blow up a log segment, if too much data needs to be removed and the log segment is not enough. That's why it's critical to know when to use truncate and delete command in SQL but before using these you should be aware of the Differences between Truncate and Delete, and based upon them, we should be able to find out when DELETE is a better option for removing data or TRUNCATE should be used to purge tables.

Refer check [click here](#)

answered Aug 22 '14 at 18:20



wpzone4u

59 1 2

0

By issuing a TRUNCATE TABLE statement, you are instructing SQL Server to delete every record within a table, without any logging or transaction processing taking place.

answered Oct 6 '14 at 0:25



Gerald

Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up with email](#)[Sign up with Google](#)[Sign up with Facebook](#)

0

DELETE statement can have a WHERE clause to delete specific records whereas TRUNCATE statement does not require any and wipes the entire table. Importantly, the DELETE statement logs the deleted data whereas the TRUNCATE statement does not.

answered Jun 13 '16 at 4:18



These are already in the community Wiki answer – [David Aldridge](#) Jun 14 '16 at 11:39

0

One more difference specific to microsoft sql server is with `delete` you can use `output` statement to track what records have been deleted, e.g.:

```
delete from [SomeTable]
output deleted.Id, deleted.Name
```

You cannot do this with `truncate`.

answered Apr 19 '17 at 6:58



0

**Truncate** command is used to re-initialize the table, it is a DDL command which delete all the rows of table. Whereas **DELETE** is a DML command which is used to delete row or set of rows according to some condition, if condition is not specified then this command will delete all the rows from the table.

answered Apr 26 '17 at 7:51



Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up with email](#)[Sign up with Google](#)[Sign up with Facebook](#)

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus does not count](#)).

Would you like to answer one of these [unanswered questions](#) instead?

---

**Join Stack Overflow** to learn, share knowledge, and build your career.

[Sign up with email](#)

 [Sign up with Google](#)

[Sign up with Facebook](#)

