

Chapter 29: DELETE

The DELETE statement is used to delete records from a table.

Section 29.1: DELETE all rows

Omitting a [WHERE](#) clause will delete all rows from a table.

```
DELETE FROM Employees
```

See TRUNCATE documentation for details on how TRUNCATE performance can be better because it ignores triggers and indexes and logs to just delete the data.

Section 29.2: DELETE certain rows with WHERE

This will delete all rows that match the [WHERE](#) criteria.

```
DELETE FROM Employees  
WHERE FName = 'John'
```

Section 29.3: TRUNCATE clause

Use this to reset the table to the condition at which it was created. This deletes all rows and resets values such as auto-increment. It also doesn't log each individual row deletion.

```
TRUNCATE TABLE Employees
```

Section 29.4: DELETE certain rows based upon comparisons with other tables

It is possible to [DELETE](#) data from a table if it matches (or mismatches) certain data in other tables.

Let's assume we want to [DELETE](#) data from Source once its loaded into Target.

```
DELETE FROM Source  
WHERE EXISTS ( SELECT 1 -- specific value in SELECT doesn't matter  
              FROM Target  
              Where Source.ID = Target.ID )
```

Most common RDBMS implementations (e.g. MySQL, Oracle, PostgreSQL, Teradata) allow tables to be joined during [DELETE](#) allowing more complex comparison in a compact syntax.

Adding complexity to original scenario, let's assume Aggregate is built from Target once a day and does not contain the same ID but contains the same date. Let us also assume that we want to delete data from Source *only* after the aggregate is populated for the day.

On MySQL, Oracle and Teradata this can be done using:

```
DELETE FROM Source  
WHERE Source.ID = TargetSchema.Target.ID  
AND TargetSchema.Target.Date = AggregateSchema.Aggregate.Date
```

In PostgreSQL use:

```
DELETE FROM Source
USING TargetSchema.Target, AggregateSchema.Aggregate
WHERE Source.ID = TargetSchema.Target.ID
AND TargetSchema.Target.DataDate = AggregateSchema.Aggregate.AggDate
```

This essentially results in INNER JOINS between Source, Target and Aggregate. The deletion is performed on Source when the same IDs exist in Target AND date present in Target for those IDs also exists in Aggregate.

Same query may also be written (on MySQL, Oracle, Teradata) as:

```
DELETE Source
FROM Source, TargetSchema.Target, AggregateSchema.Aggregate
WHERE Source.ID = TargetSchema.Target.ID
AND TargetSchema.Target.DataDate = AggregateSchema.Aggregate.AggDate
```

Explicit joins may be mentioned in [Delete](#) statements on some RDBMS implementations (e.g. Oracle, MySQL) but not supported on all platforms (e.g. Teradata does not support them)

Comparisons can be designed to check mismatch scenarios instead of matching ones with all syntax styles (observe `NOT EXISTS` below)

```
DELETE FROM Source
WHERE NOT EXISTS ( SELECT 1 -- specific value in SELECT doesn't matter
                  FROM Target
                  Where Source.ID = Target.ID )
```