

15.1.32 DROP TABLE Statement

```
DROP [TEMPORARY] TABLE [IF EXISTS]
    tbl_name [, tbl_name] ...
    [RESTRICT | CASCADE]
```

DROP TABLE removes one or more tables. You must have the DROP privilege for each table.

Be careful with this statement! For each table, it removes the table definition and all table data. If the table is partitioned, the statement removes the table definition, all its partitions, all data stored in those partitions, and all partition definitions associated with the dropped table.

Dropping a table also drops any triggers for the table.

DROP TABLE causes an implicit commit, except when used with the `TEMPORARY` keyword. See Section 15.3.3, “Statements That Cause an Implicit Commit”.

Important

When a table is dropped, privileges granted specifically for the table are *not* automatically dropped. They must be dropped manually. See Section 15.7.1.6, “GRANT Statement”.

If any tables named in the argument list do not exist, DROP TABLE behavior depends on whether the `IF EXISTS` clause is given:

- Without `IF EXISTS`, the statement fails with an error indicating which nonexisting tables it was unable to drop, and no changes are made.
- With `IF EXISTS`, no error occurs for nonexisting tables. The statement drops all named tables that do exist, and generates a `NOTE` diagnostic for each nonexistent table. These notes can be displayed with SHOW WARNINGS. See Section 15.7.7.42, “SHOW WARNINGS Statement”.

`IF EXISTS` can also be useful for dropping tables in unusual circumstances under which there is an entry in the data dictionary but no table managed by the storage engine. (For example, if an abnormal server exit occurs after removal of the table from the storage engine but before removal of the data dictionary entry.)

The `TEMPORARY` keyword has the following effects:

- The statement drops only `TEMPORARY` tables.
- The statement does not cause an implicit commit.
- No access rights are checked. A `TEMPORARY` table is visible only with the session that created it, so no check is necessary.

Including the `TEMPORARY` keyword is a good way to prevent accidentally dropping non-`TEMPORARY` tables.

The `RESTRICT` and `CASCADE` keywords do nothing. They are permitted to make porting easier from other database systems.

`DROP TABLE` is not supported with all `innodb_force_recovery` settings. See Section 17.21.3, “Forcing InnoDB Recovery”.

© 2024 Oracle
