

[Home](#) » [Resources](#) » [Knowledge Base](#) » [Library](#) » [MariaDB Documentation](#) » [SQL Statements & Structure](#) » [SQL Statements](#) » [Transactions](#) » [SQL statements That Cause an Implicit Commit](#)

SQL statements That Cause an Implicit Commit

Some SQL statements cause an implicit commit. As a rule of thumb, such statements are DDL statements. The same statements (except for SHUTDOWN) produce a 1400 error (SQLSTATE 'XAE09') if a XA transaction is in effect.

Here is the list:

```
ALTER DATABASE ... UPGRADE DATA DIRECTORY NAME
ALTER EVENT
ALTER FUNCTION
ALTER PROCEDURE
ALTER SERVER
ALTER TABLE
ALTER VIEW
ANALYZE TABLE
BEGIN
CACHE INDEX
CHANGE MASTER TO
CHECK TABLE
CREATE DATABASE
CREATE EVENT
CREATE FUNCTION
CREATE INDEX
CREATE PROCEDURE
CREATE ROLE
CREATE SERVER
CREATE TABLE
CREATE TRIGGER
CREATE USER
CREATE VIEW
DROP DATABASE
DROP EVENT
DROP FUNCTION
DROP INDEX
DROP PROCEDURE
DROP ROLE
DROP SERVER
DROP TABLE
DROP TRIGGER
DROP USER
DROP VIEW
FLUSH
GRANT
LOAD INDEX INTO CACHE
LOCK TABLES
OPTIMIZE TABLE
RENAME TABLE
RENAME USER
REPAIR TABLE
RESET
REVOKE
SET PASSWORD
SHUTDOWN
START SLAVE
START TRANSACTION
STOP SLAVE
TRUNCATE TABLE
```

`SET autocommit = 1` causes an implicit commit if the value was 0.

All these statements cause an implicit commit before execution. This means that, even if the statement fails with an error, the transaction is committed. Some of them, like `CREATE TABLE ... SELECT`, also cause a commit immediately after execution. Such statements couldn't be rolled back in any case.

If you are not sure whether a statement has implicitly committed the current transaction, you can query the `in_transaction` server system variable.

Note that when a transaction starts (not in autocommit mode), all locks acquired with `LOCK TABLES` are released. And acquiring such locks always commits the current transaction. To preserve the data integrity between transactional and non-transactional tables, the `GET_LOCK()` function can be used.

Exceptions

These statements do not cause an implicit commit in the following cases:

- `CREATE TABLE` and `DROP TABLE`, when the `TEMPORARY` keyword is used.
 - However, `TRUNCATE TABLE` causes an implicit commit even when used on a temporary table.
- `CREATE FUNCTION` and `DROP FUNCTION`, when used to create a UDF (instead of a stored function). However, `CREATE INDEX` and `DROP INDEX` cause commits even used with temporary tables.
- `UNLOCK TABLES` causes a commit only if a `LOCK TABLES` was used on non-transactional tables.
- `START SLAVE`, `STOP SLAVE`, `RESET SLAVE` and `CHANGE MASTER TO` only cause implicit commit since MariaDB 10.0.

Comments

No comments

Get Started

Download MariaDB and start working immediately! You can also get subscription details and learn more about the advantages of MariaDB's additional service offers.

[Download Now](#)

Products

- [MariaDB Server](#)
- [MariaDB MaxScale](#)
- [MariaDB ColumnStore](#)
- [Why MariaDB](#)
- [Get Started](#)
- [Pricing](#)
- [Product FAQs](#)
- [Enterprise Subscriptions Comparison](#)

Services

- [Remote DBA](#)
- [Technical Support Services](#)
- [Professional Services](#)
- [Training](#)
- [Technical Account Manager](#)
- [Migration Practice](#)

Resources

- [Customers](#)
- [Knowledge Base](#)
- [White Papers](#)
- [Datasheets & Guides](#)
- [Webinars](#)
- [Technical Presentations](#)
- [Blog](#)
- [Books](#)
- [Events](#)

About MariaDB

- [Investors](#)
- [Leadership](#)
- [Careers](#)
- [Newsroom](#)
- [Partners](#)
- [MariaDB.org](#)

Contact

Subscribe to our newsletter!

Add Me