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 15.21.2, “Forcing InnoDB Recovery”.

13.1.33 DROP TABLESPACE Statement

```
DROP [UNDO] TABLESPACE tablespace_name
    [ENGINE [=] engine_name]
```

This statement drops a tablespace that was previously created using CREATE TABLESPACE. It is supported by the NDB and InnoDB storage engines.

The UNDO keyword, introduced in MySQL 8.0.14, must be specified to drop an undo tablespace. Only undo tablespaces created using CREATE UNDO TABLESPACE syntax can be dropped. An undo tablespace must be in an empty state before it can be dropped. For more information, see Section 15.6.3.4, “Undo Tablespaces”.

ENGINE sets the storage engine that uses the tablespace, where *engine_name* is the name of the storage engine. Currently, the values InnoDB and NDB are supported. If not set, the value of default_storage_engine is used. If it is not the same as the storage engine used to create the tablespace, the DROP TABLESPACE statement fails.

tablespace_name is a case-sensitive identifier in MySQL.

For an InnoDB general tablespace, all tables must be dropped from the tablespace prior to a DROP TABLESPACE operation. If the tablespace is not empty, DROP TABLESPACE returns an error.

An NDB tablespace to be dropped must not contain any data files; in other words, before you can drop an NDB tablespace, you must first drop each of its data files using ALTER TABLESPACE ... DROP DATAFILE.

Notes

- A general InnoDB tablespace is not deleted automatically when the last table in the tablespace is dropped. The tablespace must be dropped explicitly using DROP TABLESPACE *tablespace_name*.
- A DROP DATABASE operation can drop tables that belong to a general tablespace but it cannot drop the tablespace, even if the operation drops all tables that belong to the tablespace. The tablespace must be dropped explicitly using DROP TABLESPACE *tablespace_name*.
- Similar to the system tablespace, truncating or dropping tables stored in a general tablespace creates free space internally in the general tablespace .ibd data file which can only be used for new InnoDB data. Space is not released back to the operating system as it is for file-per-table tablespaces.

InnoDB Examples

This example demonstrates how to drop an InnoDB general tablespace. The general tablespace ts1 is created with a single table. Before dropping the tablespace, the table must be dropped.

```
mysql> CREATE TABLESPACE `ts1` ADD DATAFILE 'ts1.ibd' Engine=InnoDB;

mysql> CREATE TABLE t1 (c1 INT PRIMARY KEY) TABLESPACE ts1 Engine=InnoDB;

mysql> DROP TABLE t1;
```