



MySQL Disable Foreign Key Checks

Summary: in this tutorial, you will learn how to disable foreign key constraint checks in MySQL.

Sometimes, it is very useful to disable [foreign key](https://www.mysqltutorial.org/mysql-foreign-key/) checks. For example, you can load data to the parent and child tables in any order with the foreign key constraint check disabled. If you don't disable foreign key checks, you have to load data into the parent tables first and then the child tables in sequence, which can be tedious.

Another scenario that you want to disable the foreign key check is when you want to [drop a table](https://www.mysqltutorial.org/mysql-drop-table). Unless you disable the foreign key checks, you cannot [drop a table](https://www.mysqltutorial.org/mysql-drop-table) referenced by a foreign key constraint.

To disable foreign key checks, you set the `foreign_key_checks` variable to zero as follows:

```
SET foreign_key_checks = 0;
```

To re-enable foreign key constraint check, you set the value of the `foreign_key_checks` to 1:

```
SET foreign_key_checks = 1;
```

Notice that setting `foreign_key_checks` to 1 does not trigger any validation of the existing table data. In other words, MySQL will not verify the consistency of the data that was added during the foreign key check disabled.

Disable foreign key check example

First, [create a new table](https://www.mysqltutorial.org/mysql-create-table/) named `countries` :

```
CREATE TABLE countries(  
    country_id INT AUTO_INCREMENT,  
    country_name VARCHAR(255) NOT NULL,
```

```
PRIMARY KEY(country_id)
) ENGINE=InnoDB;
```

Second, [create another table](https://www.mysqltutorial.org/mysql-create-table/) (<https://www.mysqltutorial.org/mysql-create-table/>) named `cities` :

```
CREATE TABLE cities(
  city_id INT AUTO_INCREMENT,
  city_name VARCHAR(255) NOT NULL,
  country_id INT NOT NULL,
  PRIMARY KEY(city_id),
  FOREIGN KEY(country_id)
    REFERENCES countries(country_id)
)ENGINE=InnoDB;
```

The table `cities` has a foreign key constraint that refers to the column `country_id` of the table `countries` .

Third, [insert a new row](https://www.mysqltutorial.org/mysql-insert-statement.aspx) (<https://www.mysqltutorial.org/mysql-insert-statement.aspx>) into the `cities` table:

```
INSERT INTO cities(city_name, country_id)
VALUES('New York',1);
```

MySQL issued the following error:

```
Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`classic
```

Fourth, disable foreign key checks:

```
SET foreign_key_checks = 0;
```

Fifth, insert a new row into the `cities` table:

```
INSERT INTO cities(city_name, country_id)
VALUES('New York',1);
```

This time the `INSERT` statement executed successfully due to the foreign key check disabled.

The following query (<https://www.mysqltutorial.org/mysql-select-statement-query-data.aspx>) returns the contents of the table `cities` :

```
SELECT * FROM cities;
```

Sixth, re-enable foreign key constraint check:

```
SET foreign_key_checks = 1;
```

When the foreign key checks re-enabled, MySQL did not re-validate data in the table. However, it won't allow you to [insert](https://www.mysqltutorial.org/mysql-insert-statement.aspx) or [update](https://www.mysqltutorial.org/mysql-update-data.aspx) data that violate the foreign key constraint.

Finally, insert a row into the `countries` table whose value in the column `country_id` is 1 to make the data consistent in both tables:

```
INSERT INTO countries(country_id, country_name)
VALUES(1, 'USA');
```

Drop tables that have foreign key constraints

Suppose that you want to drop the `countries` and `cities` tables.

First, drop the table `countries` :

```
DROP TABLE countries;
```

MySQL issued this error:

```
Error Code: 3730. Cannot drop table 'countries' referenced by a foreign key constraint 'citi
```

To fix this, you have two options:

- Drop the table `cities` first and then remove the table `countries` .
- Disable foreign key checks and drop tables in any sequence.

We'll demonstrate the second way which disables foreign key constraint check before dropping the tables.

Second, disable the foreign key check:

```
SET foreign_key_checks = 0;
```

Third, drop both tables `countries` and `cities` :

```
DROP TABLE countries;  
DROP TABLE cities;
```

Both statements executed successfully.

Finally, enable the foreign key check:

```
SET foreign_key_checks = 1;
```

In this tutorial, you have learned how to use the `SET foreign_key_checks = 0` to disable foreign key checks in MySQL.