MySQL ON DELETE CASCADE: Deleting Data from Related Tables Automatically

Founder

5-6 minutes

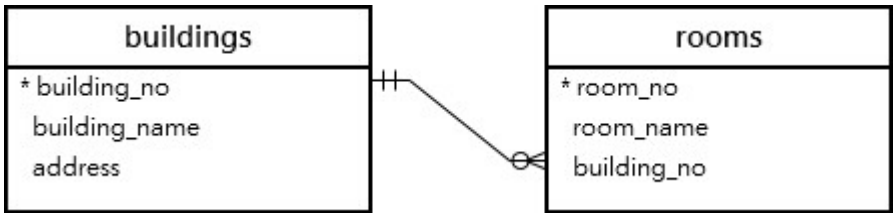
Summary: in this tutorial, you will learn how to use MySQL `ON DELETE CASCADE` referential action for a foreign key to delete data from multiple related tables.

In the previous tutorial, you learned how to delete data from multiple related tables using a single `DELETE` statement. However, MySQL provides a more effective way called `ON DELETE CASCADE` referential action for a foreign key that allows you to delete data from child tables automatically when you delete the data from the parent table.

Let's take a look at an example of using MySQL `ON DELETE CASCADE` .

Suppose that we have two tables:`buildings` and `rooms` . In this database model, each building has one or many rooms. However, each room belongs to one only one building. A room would not exist without a building.

The relationship between the `buildings` and `rooms` tables is one-to-many (1:N) as illustrated in the following database diagram:



When you delete a row from the `buildings` table, you also want to

delete all rows in the `rooms` table that references to the row in the `buildings` table. For example, when you delete a row with building no. 2 in the `buildings` table as the following query:

```
DELETE FROM buildings
WHERE building_no = 2;
```

Code language: SQL (Structured Query Language) (sql)

You also want the rows in the `rooms` table that refers to building number 2 will be also removed.

The following are steps that demonstrate how the `ON DELETE CASCADE` referential action works.

Step 1. Create the `buildings` table:

```
CREATE TABLE buildings (
    building_no INT PRIMARY KEY AUTO_INCREMENT,
    building_name VARCHAR(255) NOT NULL,
    address VARCHAR(255) NOT NULL
);
```

Code language: SQL (Structured Query Language) (sql)

Step 2. Create the `rooms` table:

```
CREATE TABLE rooms (
    room_no INT PRIMARY KEY AUTO_INCREMENT,
    room_name VARCHAR(255) NOT NULL,
    building_no INT NOT NULL,
    FOREIGN KEY (building_no)
        REFERENCES buildings (building_no)
        ON DELETE CASCADE
);
```

Code language: SQL (Structured Query Language) (sql)

Notice that the `ON DELETE CASCADE` clause at the end of the foreign key constraint definition.

Step 3. Insert rows into the `buildings` table:

```
INSERT INTO buildings(building_name,address)
VALUES('ACME Headquaters','3950 North 1st Street CA
95134'),
      ('ACME Sales','5000 North 1st Street CA 95134');
```

Code language: SQL (Structured Query Language) (sql)

Step 4. Query data from the `buildings` table:

```
SELECT * FROM buildings;
```

Code language: SQL (Structured Query Language) (sql)

	building_no	building_name	address
▶	1	ACME Headquaters	3950 North 1st Street CA 95134
	2	ACME Sales	5000 North 1st Street CA 95134

We have two rows in the `buildings` table.

Step 5. Insert rows into the `rooms` table:

```
INSERT INTO rooms(room_name,building_no)
VALUES('Amazon',1),
      ('War Room',1),
      ('Office of CEO',1),
      ('Marketing',2),
      ('Showroom',2);
```

Code language: SQL (Structured Query Language) (sql)

Step 6. Query data from the `rooms` table:

```
SELECT * FROM rooms;
```

Code language: SQL (Structured Query Language) (sql)

	room_no	room_name	building_no
▶	1	Amazon	1
	2	War Room	1
	3	Office of CEO	1
	4	Marketing	2
	5	Showroom	2

We have three rooms that belong to building no 1 and two rooms that belong to the building no 2.

Step 7. Delete the building with building no. 2:

```
DELETE FROM buildings
```

```
WHERE building_no = 2;
```

Code language: SQL (Structured Query Language) (sql)

Step 8. Query data from `rooms` table:

```
SELECT * FROM rooms;
```

Code language: SQL (Structured Query Language) (sql)

	room_no	room_name	building_no
▶	1	Amazon	1
	2	War Room	1
	3	Office of CEO	1

As you can see, all the rows that reference to `building_no` 2 were automatically deleted.

Notice that `ON DELETE CASCADE` works only with tables with the storage engines that support foreign keys e.g., InnoDB.

Some table types do not support foreign keys such as MyISAM so you should choose appropriate storage engines for the tables that you plan to use the MySQL `ON DELETE CASCADE` referential action.

Tips to find tables affected by MySQL `ON DELETE CASCADE` action

Sometimes, it is useful to know which table is affected by the `ON DELETE CASCADE` referential action when you delete data from a table. You can query this data from the `referential_constraints` in the `information_schema` database as follows:

```
USE information_schema;

SELECT
    table_name
FROM
    referential_constraints
WHERE
    constraint_schema = 'database_name'
```

```
        AND referenced_table_name = 'parent_table'
        AND delete_rule = 'CASCADE'
```

Code language: SQL (Structured Query Language) (sql)

For example, to find tables that associated with the `buildings` table with the `CASCADE` deletion rule in the `classicmodels` database, you use the following query:

```
USE information_schema;

SELECT
    table_name
FROM
    referential_constraints
WHERE
    constraint_schema = 'classicmodels'
        AND referenced_table_name = 'buildings'
        AND delete_rule = 'CASCADE'
```

Code language: SQL (Structured Query Language) (sql)

	table_name
▶	rooms

In this tutorial, you have learned how to use the MySQL `ON DELETE CASCADE` referential action for a foreign key to delete data automatically from the child tables when you delete data from the parent table.

Was this tutorial helpful?