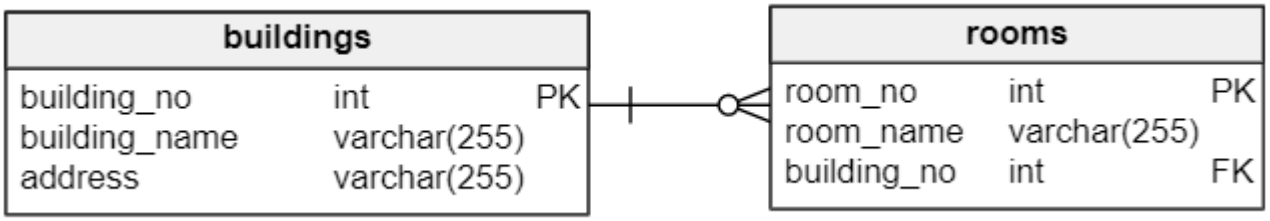
MySQL ON DELETE CASCADE example

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

Suppose we have two tables:`buildings` and `rooms` .In this database model, each building has one or more 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 we delete a row from the `buildings` table, we also want to delete the rows in the `rooms` table that references to the rows in the buildings table. For example, when we delete a row with building no. 2 in the `buildings` table as the following query:

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

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

The following are steps that demonstrate how MySQL `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
);
```

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
);
```

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

Step 3. Insert data 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');
```

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

```
SELECT * FROM buildings;
```

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 data 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);
```

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

```
SELECT * FROM rooms;
```

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 3 rooms that belong to building 1 and 2 rooms that belong to the building 2.

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

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

Step 8. Query data from `rooms` table:

SELECT * FROM rooms;

	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 deleted.