

Deleting Data

This lesson discusses how to delete data in MySQL.

Deleting Data

We can delete rows from a table using the **DELETE** statement. A delete statement deletes an entire row and not individual columns. If changing a particular column value for a row is desired, use the **UPDATE** statement, which we will cover next. Also realize that deleting all the rows of a table doesn't delete the table itself.

Example Syntax

```
DELETE FROM table  
  
WHERE col3 > 5  
  
ORDER BY col1  
  
LIMIT 5;
```

Connect to the terminal below by clicking in the widget. Once connected, the command line prompt will show up. Enter or copy and paste the command `./DataJek/Lessons/13lesson.sh` and wait for the MySQL prompt to start-up.

```
-- The lesson queries are reproduced below for convenient copy/paste into the terminal.
```



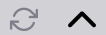
```
-- Query 1  
DELETE FROM Actors WHERE FirstName="priyanka";
```

```
-- Query 2
DELETE FROM Actors WHERE Gender="Male";

-- Query 3
DELETE FROM Actors ORDER BY NetWorthInMillions DESC LIMIT 3;

-- Query 4
DELETE FROM Actors;
```

● Terminal



1. We'll start with deleting just one row. In order to target a single row, we'll need to use the **WHERE** clause similar to how we used it with a select statement. Let's delete the row for the actress priyanka.

```
DELETE FROM Actors WHERE FirstName="priyanka";
```

```
mysql> SELECT * FROM Actors;
+----+-----+-----+-----+-----+-----+-----+
| Id | FirstName | SecondName | DoB      | Gender | MaritalStatus | NetWorthInMillions |
+----+-----+-----+-----+-----+-----+-----+
| 1 | Brad      | Pitt       | 1963-12-18 | Male   | Single        | 240 |
| 2 | Jennifer  | Aniston    | 1969-11-02 | Female | Single        | 240 |
| 3 | Angelina  | Jolie      | 1975-06-04 | Female | Single        | 180 |
| 4 | Johnny    | Depp       | 1963-06-09 | Male   | Single        | 280 |
| 5 | Natalie   | Portman    | 1981-06-09 | Male   | Married       | 60 |
| 6 | Tom       | Cruise     | 1962-07-03 | Male   | Divorced      | 570 |
| 7 | Kylie     | Jenner     | 1997-08-10 | Female | Married       | 1880 |
| 8 | Kim       | Kardashian | 1980-10-21 | Female | Married       | 370 |
| 9 | Anilabh   | Bachchan   | 1942-10-11 | Male   | Married       | 480 |
| 10 | Shahrukh  | Khan       | 1965-11-02 | Male   | Married       | 680 |
| 11 | priyanka  | Chopra     | 1982-07-18 | Female | Married       | 28 |
+----+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql> DELETE FROM Actors WHERE FirstName="priyanka";
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Actors;
+----+-----+-----+-----+-----+-----+-----+
| Id | FirstName | SecondName | DoB      | Gender | MaritalStatus | NetWorthInMillions |
+----+-----+-----+-----+-----+-----+-----+
| 1 | Brad      | Pitt       | 1963-12-18 | Male   | Single        | 240 |
| 2 | Jennifer  | Aniston    | 1969-11-02 | Female | Single        | 240 |
| 3 | Angelina  | Jolie      | 1975-06-04 | Female | Single        | 180 |
| 4 | Johnny    | Depp       | 1963-06-09 | Male   | Single        | 280 |
| 5 | Natalie   | Portman    | 1981-06-09 | Male   | Married       | 60 |
| 6 | Tom       | Cruise     | 1962-07-03 | Male   | Divorced      | 570 |
| 7 | Kylie     | Jenner     | 1997-08-10 | Female | Married       | 1880 |
| 8 | Kim       | Kardashian | 1980-10-21 | Female | Married       | 370 |
| 9 | Anilabh   | Bachchan   | 1942-10-11 | Male   | Married       | 480 |
| 10 | Shahrukh  | Khan       | 1965-11-02 | Male   | Married       | 680 |
+----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

Note that the string comparison for the FirstName performed by MySQL doesn't take case into account and if we specified the **WHERE** clause with an uppercase as FirstName="PRIYANKA" the row would still be deleted.

2. The delete statement will delete all the matching rows, which in the previous example is only one. As an example, we can write a query to delete all male actors as follows:

```
DELETE FROM Actors WHERE Gender="Male";
```

```
mysql> SELECT * FROM Actors;
+----+-----+-----+-----+-----+-----+-----+
| Id | FirstName | SecondName | DoB      | Gender | MaritalStatus | NetWorthInMillions |
+----+-----+-----+-----+-----+-----+-----+
| 1  | Brad      | Pitt       | 1963-12-18 | Male   | Single        | 240 |
| 2  | Jennifer  | Aniston    | 1969-11-02 | Female | Single        | 240 |
| 3  | Angelina  | Jolie      | 1975-06-04 | Female | Single        | 180 |
| 4  | Johnny    | Depp       | 1963-06-09 | Male   | Single        | 200 |
| 5  | Natalie   | Portman    | 1981-06-09 | Male   | Married       | 60  |
| 6  | Tom       | Cruise     | 1962-07-03 | Male   | Divorced      | 570 |
| 7  | Kylie     | Jenner     | 1997-08-10 | Female | Married       | 1000 |
| 8  | Kim       | Kardashian | 1980-10-21 | Female | Married       | 370 |
| 9  | Aritabh   | Bachchan   | 1942-10-11 | Male   | Married       | 400 |
| 10 | Shahrukh  | Khan       | 1965-11-02 | Male   | Married       | 600 |
+----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> DELETE FROM Actors WHERE Gender="Male";
Query OK, 6 rows affected (0.01 sec)

mysql> SELECT * FROM Actors;
+----+-----+-----+-----+-----+-----+-----+
| Id | FirstName | SecondName | DoB      | Gender | MaritalStatus | NetWorthInMillions |
+----+-----+-----+-----+-----+-----+-----+
| 2  | Jennifer  | Aniston    | 1969-11-02 | Female | Single        | 240 |
| 3  | Angelina  | Jolie      | 1975-06-04 | Female | Single        | 180 |
| 7  | Kylie     | Jenner     | 1997-08-10 | Female | Married       | 1000 |
| 8  | Kim       | Kardashian | 1980-10-21 | Female | Married       | 370 |
+----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

3. Next, suppose that out of the remaining female actors in our database, we want to delete the top three actresses by net worth. We can accomplish that by using the **ORDER BY** and **LIMIT** clauses.

```
DELETE FROM Actors ORDER BY NetWorthInMillions DESC LIMIT 3;
```

```
mysql> SELECT * FROM Actors;
+----+-----+-----+-----+-----+-----+-----+
| Id | FirstName | SecondName | DoB      | Gender | MaritalStatus | NetWorthInMillions |
+----+-----+-----+-----+-----+-----+-----+
| 2  | Jennifer  | Aniston    | 1969-11-02 | Female | Single        | 240 |
| 3  | Angelina  | Jolie      | 1975-06-04 | Female | Single        | 180 |
| 7  | Kylie     | Jenner     | 1997-08-10 | Female | Married       | 1000 |
| 8  | Kim       | Kardashian | 1980-10-21 | Female | Married       | 370 |
+----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> DELETE FROM Actors ORDER BY NetWorthInMillions DESC LIMIT 3;
Query OK, 3 rows affected (0.01 sec)

mysql> SELECT * FROM Actors;
+----+-----+-----+-----+-----+-----+-----+
| Id | FirstName | SecondName | DoB      | Gender | MaritalStatus | NetWorthInMillions |
+----+-----+-----+-----+-----+-----+-----+
| 3  | Angelina  | Jolie      | 1975-06-04 | Female | Single        | 180 |
+----+-----+-----+-----+-----+-----+-----+
1 rows in set (0.00 sec)
```

```
1 row in set (0.00 sec)
```

The above query removes the top three actresses by net worth.

4. We can also remove all the rows from a table using the following query:

```
DELETE FROM Actors;
```

```
mysql> SELECT * FROM Actors;
+----+-----+-----+-----+-----+-----+-----+
| Id | FirstName | SecondName | DoB      | Gender | MaritalStatus | NetWorthInMillions |
+----+-----+-----+-----+-----+-----+-----+
| 3  | Angelina  | Jolie      | 1975-06-04 | Female | Single        | 180                 |
+----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> DELETE FROM Actors;
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Actors;
Empty set (0.00 sec)
```

Notice that the table can still be queried even after we have removed all the rows from it.