

LIMIT

This lesson discusses how to use the LIMIT clause.

LIMIT Clause

Usually tables in a production environment have thousands or millions of rows and a select query may return several hundred matched rows. This is problematic because outputting thousands of rows on the console or on a network connection can overwhelm the end-user in the former and is impractical in the latter scenario. The **LIMIT** clause allows us to restrict the number of rows returned from the result of a select query.

Example Syntax

```
SELECT col1, col2, ... coln

FROM table

WHERE col3 LIKE "%some-string%"

ORDER BY col3

LIMIT 10;
```

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/12lesson.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

```
SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMillions DESC LIMIT 3;
```

-- Query 2

```
SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMillions DESC LIMIT 4 OFFSET 3;
```

-- Query 3

```
SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMillions DESC LIMIT 3,4;
```

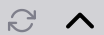
-- Query 4

```
SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMillions DESC LIMIT 1000 OFFSET 3;
```

-- Query 5

```
SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMillions DESC LIMIT 18446744073709551615;
```

● Terminal



1. Say we want to find the top three actors by net worth. We can execute the following query to get the desired result:

```
SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMillions DESC LIMIT 3;
```

```
mysql> SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMillions DESC LIMIT 3;
+-----+-----+
| FirstName | SecondName |
+-----+-----+
| Kylie     | Jenner     |
| Shahrukh  | Khan       |
| Tom       | Cruise     |
+-----+-----+
3 rows in set (0.00 sec)
```

2. Next, say we are required to retrieve the next 4 richest actors after the top three. We can do so by specifying the number of rows we want after the top three rows using the **OFFSET** keyword.

```
SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMillions DESC LIMIT 4 OFFSET 3;
```

```
mysql> SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMillions DESC LIMIT 4 OFFSET 3;
+-----+-----+
| FirstName | SecondName |
+-----+-----+
| Anitabh   | Bachchan   |
| Kim       | Kardashian |
| Brad      | Pitt       |
| Jennifer  | Aniston    |
+-----+-----+
4 rows in set (0.00 sec)
```

We can also use the alternative syntax as follows:

```
SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMilli
ons DESC LIMIT 3,4;
```

```
mysql> SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMillions DESC LIMIT 3,4;
+-----+-----+
| FirstName | SecondName |
+-----+-----+
| Arinash   | Bachchan   |
| Kim      | Kardashian |
| Brad     | Pitt       |
| Jennifer | Aniston    |
+-----+-----+
4 rows in set (0.00 sec)
```

The syntax is:

```
LIMIT <offset>, <number_of_row_to_print>;
```

- Note that we can specify as many rows as we would like to be retrieved, starting at the offset, we specify. For instance, we can ask for a thousand rows after the offset and we'll be returned all the rows after the top three.

```
SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMilli
ons DESC LIMIT 1000 OFFSET 3;
```

```
mysql> SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMillions DESC LIMIT 1000 OFFSET 3;
+-----+-----+
| FirstName | SecondName |
+-----+-----+
| Arinash   | Bachchan   |
| Kim      | Kardashian |
| Brad     | Pitt       |
| Jennifer | Aniston    |
| Johnny   | Depp       |
| Angelina | Jolie      |
| Natalie  | Portman    |
| Priyanka | Chopra     |
+-----+-----+
8 rows in set (0.00 sec)
```

The maximum number we can specify after the **LIMIT** keyword is 18446744073709551615, since that is the maximum value that can be stored in MySQL's **unsigned BIGINT** variable type. Any value higher than that and MySQL will complain.

```
SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMilli
ons DESC LIMIT 18446744073709551616;
```

```
mysql> SELECT FirstName, SecondName from Actors ORDER BY NetWorthInMillions DESC LIMIT 18446744073709551616;
+-----+-----+
| FirstName | SecondName |
+-----+-----+
| Apple     | Zener      |
| Sharmah  | Khan       |
| Tom      | Cruise     |
| Arinash   | Bachchan   |
| Kim      | Kardashian |
| Brad     | Pitt       |
| Jennifer | Aniston    |
| Johnny   | Depp       |
| Angelina | Jolie      |
+-----+-----+
9 rows in set (0.00 sec)
```

```

+-----+-----+
| Record | | Primary |
| pinyin | | Chapter |
+-----+-----+
| 18446744073709551616 | 18446744073709551616 |
+-----+-----+
mysql> SELECT * FROM table_name WHERE id=18446744073709551616 LIMIT 1;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '18446744073709551616' at line 1

```

As shown in the above snapshot MySQL issues an error when we use 18446744073709551616 for **LIMIT**.