Aggregate Methods

This lesson discusses MySQL aggregate functions.

Aggregate Methods

In this lesson, we'll demonstrate working with a few of the important aggregate functions.

Example Syntax

> SELECT **AggregateFunction(col1)**
>
> FROM **table**;

> 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/22lesson.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 COUNT(*) FROM Actors;

-- Query 2
SELECT SUM(NetworthInMillions) FROM Actors;

-- Query 3
SELECT AVG(NetWorthInMillions) FROM Actors;

-- Query 4
SELECT MIN(NetWorthInMillions) FROM Actors;

-- Query 5
```

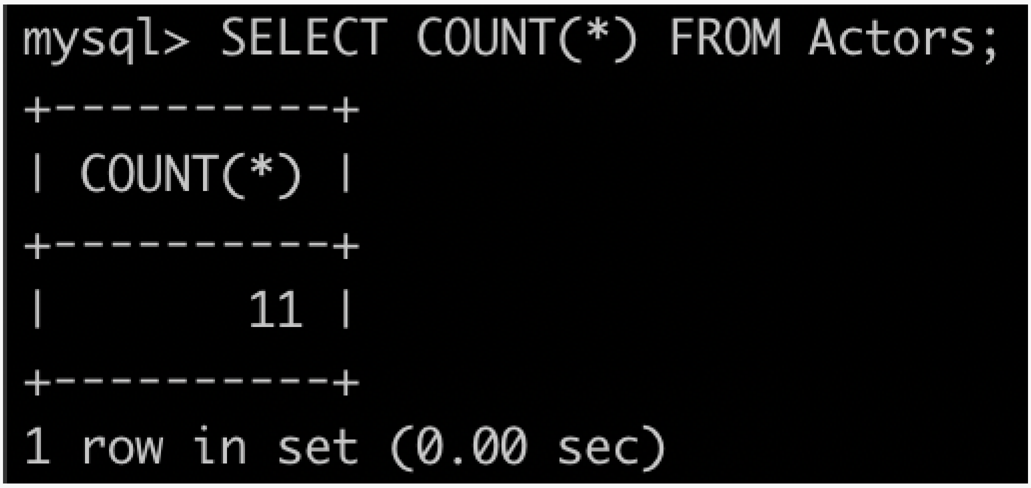
```
SELECT MAX(NetWorthInMillions) FROM Actors;


-- Query 6
SELECT STDDEV(NetWorthInMillions) FROM Actors;
```

Terminal

1. We can count the number of rows in a table using the **COUNT** function.

```
SELECT COUNT(*) FROM Actors;
```

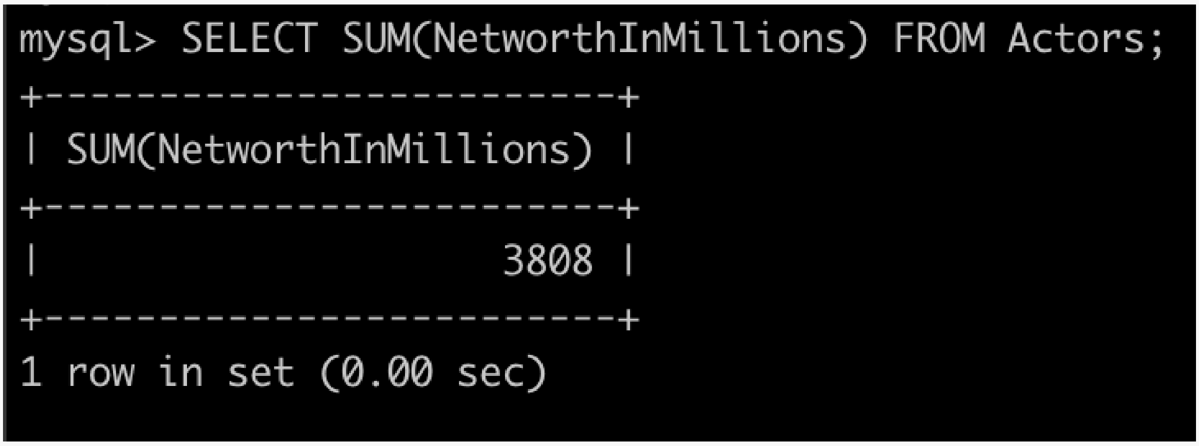


```
mysql> SELECT COUNT(*) FROM Actors;
+----------+
| COUNT(*) |
+----------+
|       11 |
+----------+
1 row in set (0.00 sec)
```

Note the output of the query is a single value rather than rows.

2. Using the **SUM** function, we can add up the numeric values of a column. For instance, the following query will sum the net worth of all the actors in our example setup to report the cumulative worth of all the actors.

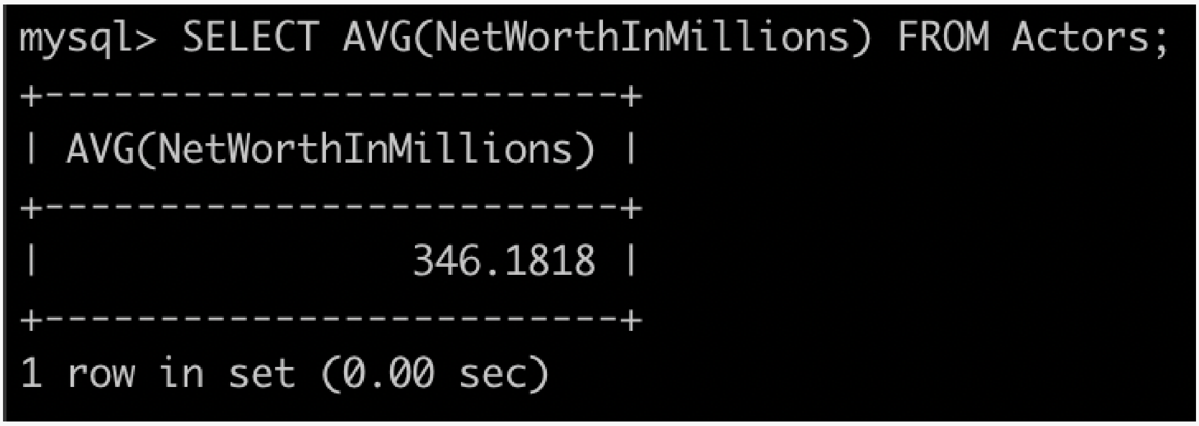
```
SELECT SUM(NetworthInMillions) FROM Actors;
```



```
mysql> SELECT SUM(NetworthInMillions) FROM Actors;
+-------------------------+
| SUM(NetworthInMillions) |
+-------------------------+
|                    3808 |
+-------------------------+
1 row in set (0.00 sec)
```

3. We can use the **AVG** function to calculate the average net worth of actors as follows:

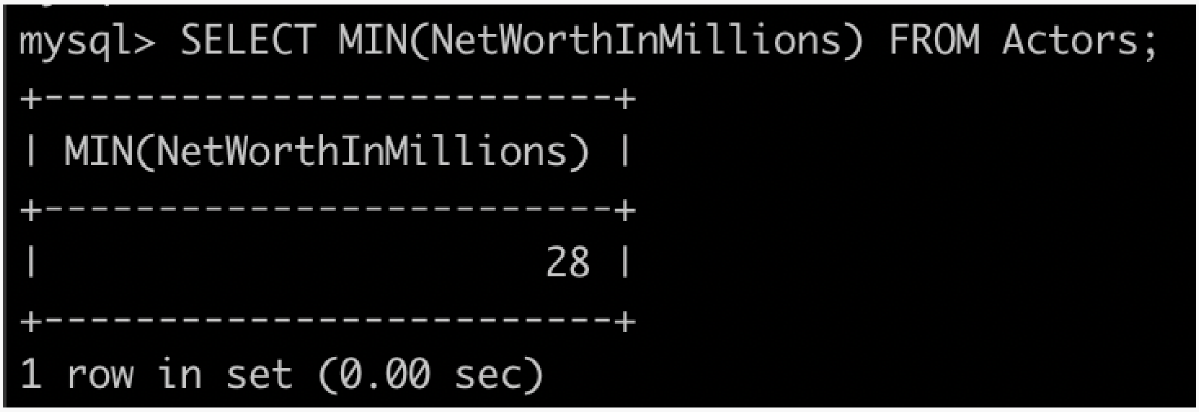
```
SELECT AVG(NetWorthInMillions) FROM Actors;
```



```
mysql> SELECT AVG(NetWorthInMillions) FROM Actors;
+-------------------------+
| AVG(NetWorthInMillions) |
+-------------------------+
|                346.1818 |
+-------------------------+
1 row in set (0.00 sec)
```

4. We can find the actor with the least net worth as follows:

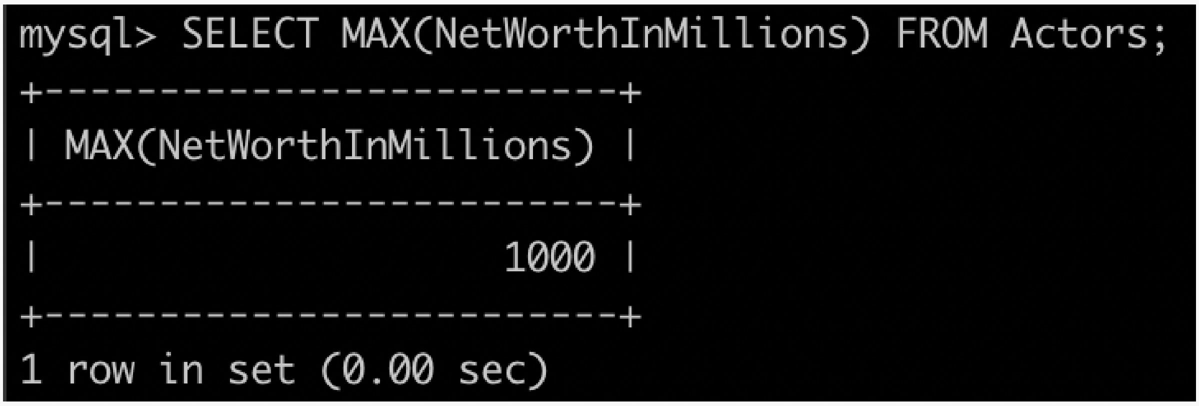
```
SELECT MIN(NetWorthInMillions) FROM Actors;
```



```
mysql> SELECT MIN(NetWorthInMillions) FROM Actors;
+-------------------------+
| MIN(NetWorthInMillions) |
+-------------------------+
|                      28 |
+-------------------------+
1 row in set (0.00 sec)
```

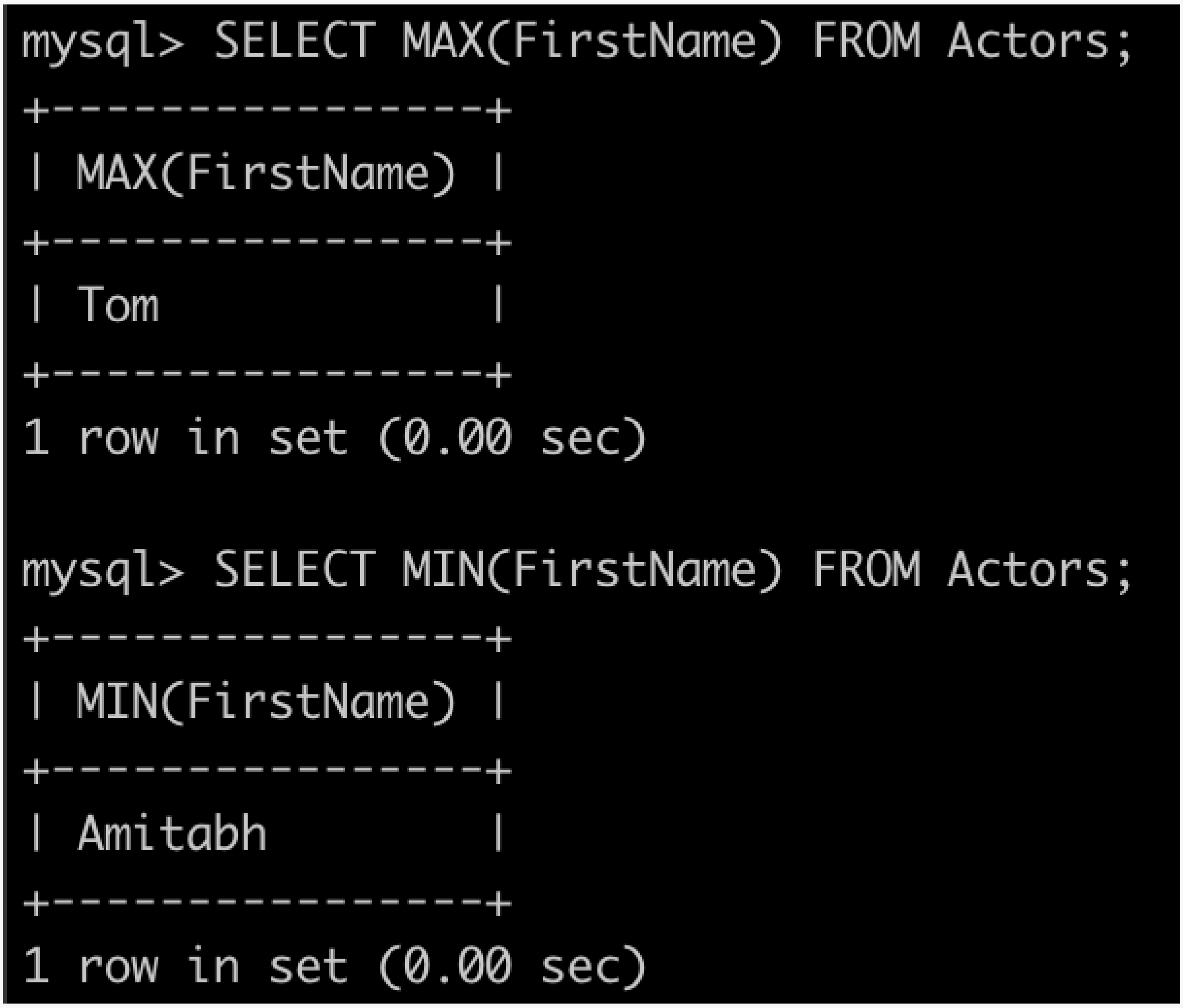
5. Similarly, we can find the actor with the most net worth as follows:

```
SELECT MAX(NetWorthInMillions) FROM Actors;
```



```
mysql> SELECT MAX(NetWorthInMillions) FROM Actors;
+-------------------------+
| MAX(NetWorthInMillions) |
+-------------------------+
|                    1000 |
+-------------------------+
1 row in set (0.00 sec)
```

Note that we can also apply the **MIN** and **MAX** functions to nonnumerical columns such as FirstName. MySQL would return the actor with the first name that occurs first or last when first names are sorted for **MIN** and **MAX** respectively. The queries are shown below:

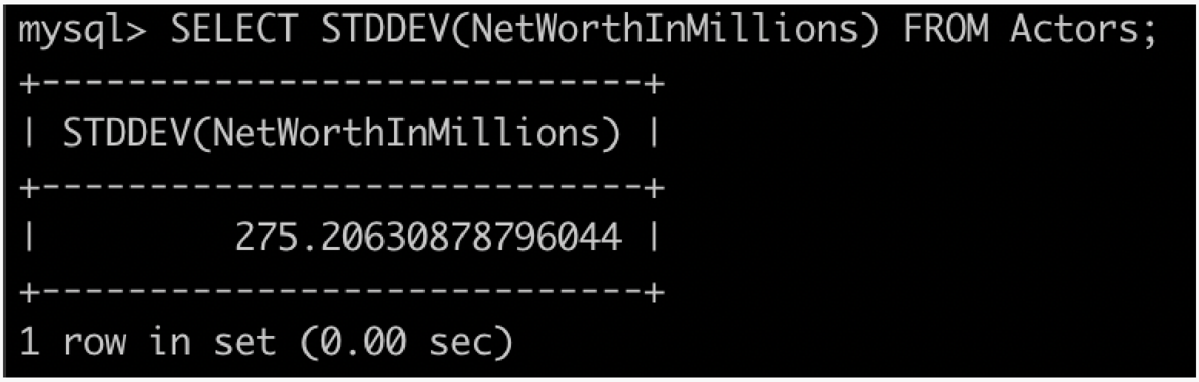


```
mysql> SELECT MAX(FirstName) FROM Actors;
+----------------+
| MAX(FirstName) |
+----------------+
| Tom            |
+----------------+
1 row in set (0.00 sec)

mysql> SELECT MIN(FirstName) FROM Actors;
+----------------+
| MIN(FirstName) |
+----------------+
| Amitabh        |
+----------------+
1 row in set (0.00 sec)
```

6. We can find the income disparity among actors using the standard deviation function **STD** or **STDDEV** as follows:

```
SELECT STDDEV(NetWorthInMillions) FROM Actors;
```



```
mysql> SELECT STDDEV(NetWorthInMillions) FROM Actors;
+----------------------------+
| STDDEV(NetWorthInMillions) |
+----------------------------+
|          275.2063087879604 |
+----------------------------+
1 row in set (0.00 sec)
```

You can find a comprehensive list of MySQL functions here