Inserting Data

This lesson teaches the various ways of inserting data in MySQL.

Insert Data

In the previous lessons we created our example table, **Actors**. But a table without any data is not very useful. In this lesson we'll learn how to add data into a table using the **INSERT** statement. We'll retrieve the added rows using the **SELECT** keyword. We'll learn more about using **SELECT** in the next lesson, but for now, it suffices to know that it is used for retrieving rows from a table.

Example Syntax

> INSERT INTO **table (col1, col2 … coln)**
>
> VALUES (**val1, val2, … valn)**;

> 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/7lesson.sh** and wait for the MySQL prompt to start-up.

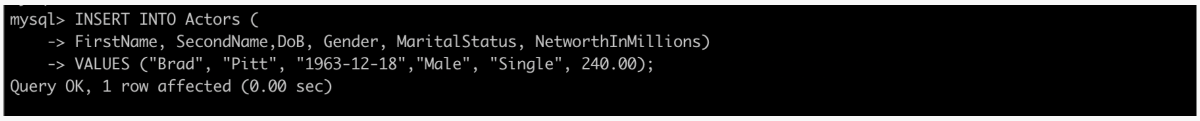
```
-- The lesson queries are reproduced below for convenient copy/paste into the terminal.

-- Query 1
INSERT INTO Actors (
FirstName, SecondName, DoB, Gender, MaritalStatus, NetworthInMillions)
VALUES ("Brad", "Pitt", "1963-12-18", "Male", "Single", 240.00);

-- Query 2
```

```
INSERT INTO Actors (
FirstName, SecondName, DoB, Gender, MaritalStatus, NetworthInMillions)
VALUES

("Jennifer", "Aniston", "1969-11-02", "Female", "Single", 240.00),
("Angelina", "Jolie", "1975-06-04", "Female", "Single", 100.00),
("Johnny", "Depp", "1963-06-09", "Male", "Single", 200.00);

-- Query 3
INSERT INTO Actors
VALUES (DEFAULT, "Dream", "Actress", "9999-01-01", "Female", "Single", 000.00);

-- Query 4
INSERT INTO Actors VALUES (NULL, "Reclusive", "Actor", "1980-01-01", "Male", "Single", DEFAUL

-- Query 5
INSERT INTO Actors () VALUES ();

-- Query 6
INSERT INTO Actors SET DoB="1950-12-12", FirstName="Rajnikanth", SecondName="",  Gender="Male
```

Terminal

1. Now we'll add a row to our, so far empty, **Actors** table using the **INSERT** command. Copy and paste the following query:

   ```
   INSERT INTO Actors (
   FirstName, SecondName, DoB, Gender, MaritalStatus, NetworthInMill
   ions)
   VALUES ("Brad", "Pitt", "1963-12-18", "Male", "Single", 240.00);
   ```

   ```
   mysql> INSERT INTO Actors (
       -> FirstName, SecondName,DoB, Gender, MaritalStatus, NetworthInMillions)
       -> VALUES ("Brad", "Pitt", "1963-12-18","Male", "Single", 240.00);
   Query OK, 1 row affected (0.00 sec)
   ```

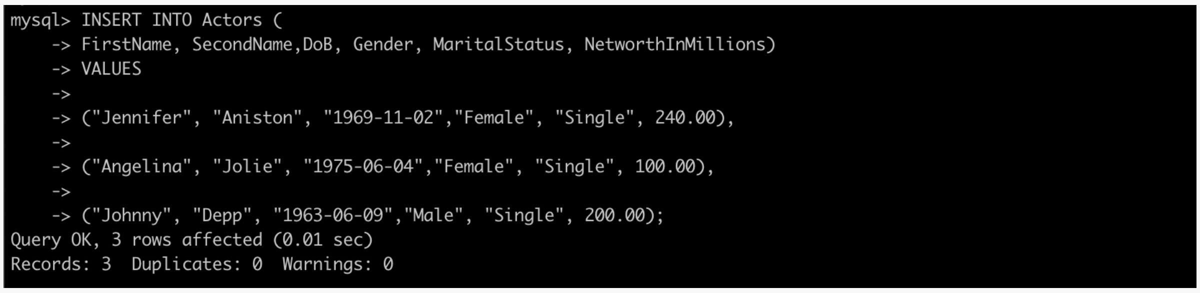
   The prompt will display the message, "Query OK, 1 row affected", if the row is inserted successfully.

   Note that the order of the column names is the same as in the table, but this isn't necessary. We can list the column names in any order as long as the values match the same order.

2. We can also add multiple records using the **INSERT** statement. The syntax requires us to separate the records using a comma. Execute the following command to insert multiple records:

   ```
   INSERT INTO Actors (
   ```

```
FirstName, SecondName, DoB, Gender, MaritalStatus, NetworthInMill
ions)


VALUES

("Jennifer", "Aniston", "1969-11-02", "Female", "Single", 240.00
),

("Angelina", "Jolie", "1975-06-04", "Female", "Single", 100.00),

("Johnny", "Depp", "1963-06-09", "Male", "Single", 200.00);
```

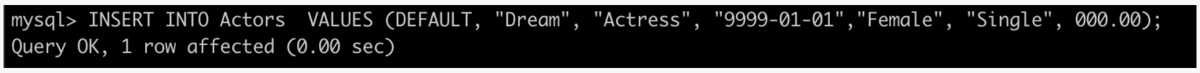


```
mysql> INSERT INTO Actors (
    -> FirstName, SecondName,DoB, Gender, MaritalStatus, NetworthInMillions)
    -> VALUES
    ->
    -> ("Jennifer", "Aniston", "1969-11-02","Female", "Single", 240.00),
    ->
    -> ("Angelina", "Jolie", "1975-06-04","Female", "Single", 100.00),
    ->
    -> ("Johnny", "Depp", "1963-06-09","Male", "Single", 200.00);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

This style of inserting rows is much faster than inserting one row at a time. Adding multiple rows gives MySQL the opportunity to optimize inserts.

3. We can also use an alternative syntax to insert data into a table that doesn't require listing out the column names. For example:

```
INSERT INTO Actors
VALUES (DEFAULT, "Dream", "Actress", "9999-01-01", "Female", "Sin
gle", 000.00);
```



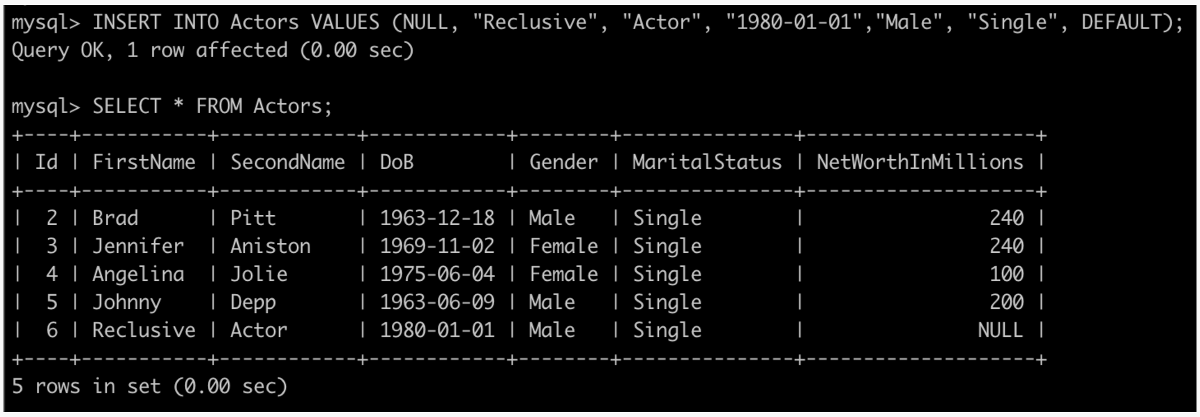
```
mysql> INSERT INTO Actors  VALUES (DEFAULT, "Dream", "Actress", "9999-01-01","Female", "Single", 000.00);
Query OK, 1 row affected (0.00 sec)
```

Since we skipped the column names when using the alternative syntax to insert rows, the order of the values should be the same as the order of the columns in the table or that listed by the describe table query. Note that we used the **DEFAULT** keyword for the **ID** column. We could have also used **NULL** or **0** for MySQL to automatically assign the next higher integer in the sequence to the ID column of the new row.

4. When inserting a row into a table we can skip a column and instruct MySQL to populate it with the default value using the **DEFAULT** keyword. Copy and paste the SQL query below in the terminal and

observe the results:

```
INSERT INTO Actors VALUES (NULL, "Reclusive", "Actor", "1980-01-0
1", "Male", "Single", DEFAULT);
```



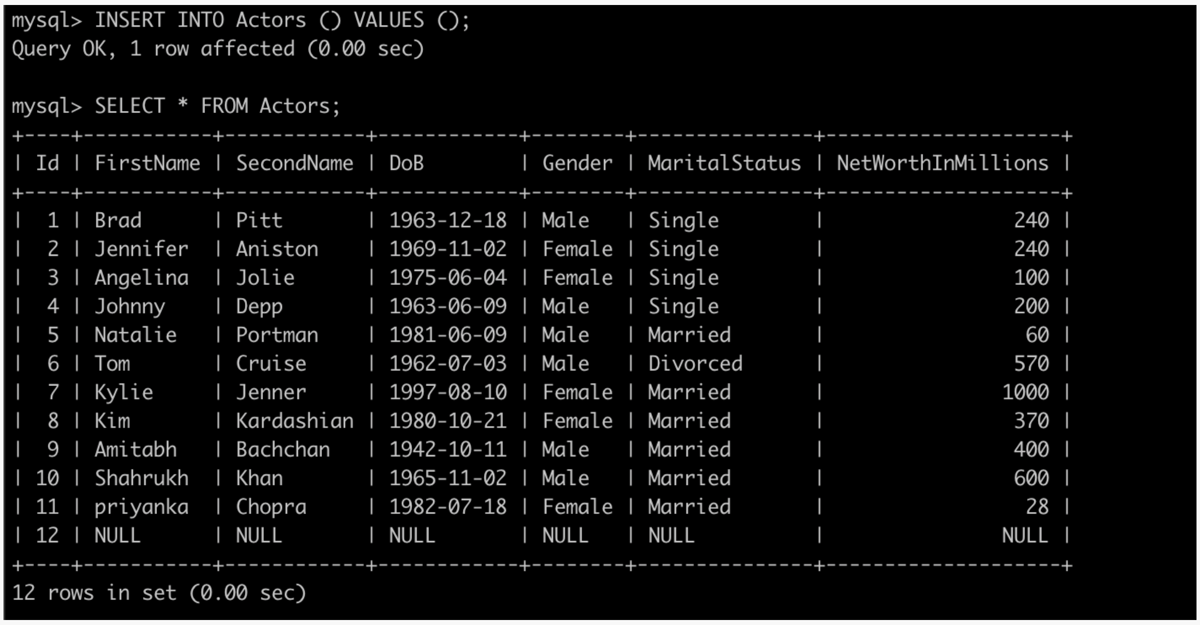
```
mysql> INSERT INTO Actors VALUES (NULL, "Reclusive", "Actor", "1980-01-01","Male", "Single", DEFAULT);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Actors;
+----+-----------+------------+------------+--------+---------------+-------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+-------------------+
|  2 | Brad      | Pitt       | 1963-12-18 | Male   | Single        |               240 |
|  3 | Jennifer  | Aniston    | 1969-11-02 | Female | Single        |               240 |
|  4 | Angelina  | Jolie      | 1975-06-04 | Female | Single        |               100 |
|  5 | Johnny    | Depp       | 1963-06-09 | Male   | Single        |               200 |
|  6 | Reclusive | Actor      | 1980-01-01 | Male   | Single        |              NULL |
+----+-----------+------------+------------+--------+---------------+-------------------+
5 rows in set (0.00 sec)
```

You can see the column **NetWorthInMillions** for the inserted row takes on the default value of "**NULL**". We could specify a default numeric value for the column when creating the table but since we didn't, the default value is set to NULL.

5. Another interesting aspect is we can insert a row with all default values. If a column doesn't have a default value defined, it is assigned **NULL** as default. Consider the query below:

```
INSERT INTO Actors () VALUES ();
```

The query adds a row with all **NULL** values as shown below:

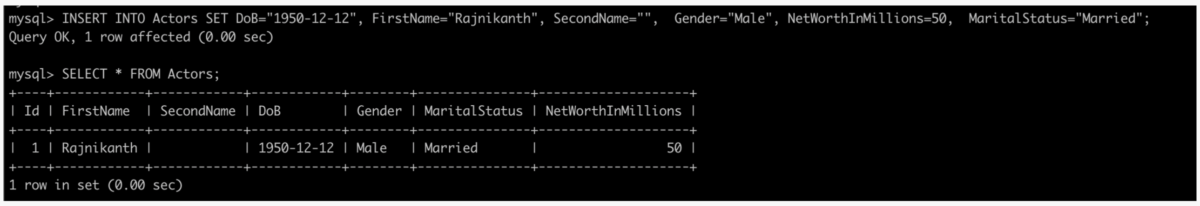


```
mysql> INSERT INTO Actors () VALUES ();
Query OK, 1 row affected (0.00 sec)

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        |               100 |
|  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 | Amitabh   | Bachchan   | 1942-10-11 | Male   | Married       |               400 |
| 10 | Shahrukh  | Khan       | 1965-11-02 | Male   | Married       |               600 |
| 11 | priyanka  | Chopra     | 1982-07-18 | Female | Married       |                28 |
| 12 | NULL      | NULL       | NULL       | NULL   | NULL          |              NULL |
+----+-----------+------------+------------+--------+---------------+-------------------+
12 rows in set (0.00 sec)
```

The above query will fail if any one of the table columns is specified as not-null. **DEFAULT** keyword also comes in handy when working with the **TIMESTAMP** column. The default value for a **TIMESTAMP** column is the current timestamp, which may be what we want when inserting a new row.

6. Yet another way to add rows to a table is to use the column name and the value together. This alternative syntax makes use of the **SET** keyword:

```sql
INSERT INTO Actors SET DoB="1950-12-12", FirstName="Rajnikanth",
 SecondName="",  Gender="Male", NetWorthInMillions=50,  MaritalSt
atus="Married";
```



```
mysql> INSERT INTO Actors SET DoB="1950-12-12", FirstName="Rajnikanth", SecondName="",  Gender="Male", NetWorthInMillions=50,  MaritalStatus="Married";
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM Actors;
+----+------------+------------+------------+--------+---------------+--------------------+
| Id | FirstName  | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+------------+------------+------------+--------+---------------+--------------------+
|  1 | Rajnikanth |            | 1950-12-12 | Male   | Married       |                 50 |
+----+------------+------------+------------+--------+---------------+--------------------+
1 row in set (0.00 sec)
```

We can arrange the column and value pairs as we desire. No ambiguity is created since we are explicitly calling out the value for each column.