Combining Conditions

This lesson discusses how to combine multiple conditions in a MySQL query.

Combining Conditions

In this lesson we'll learn how to combine multiple conditions in the **WHERE** clause.

Example Syntax

> SELECT **col1, col2, ... coln**
>
> FROM **table**
>
> WHERE **col3 LIKE "%some-string%"**
>
> AND
>
> **col4** = 55;

> 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/10lesson.sh** and wait for the MySQL prompt to start-up.

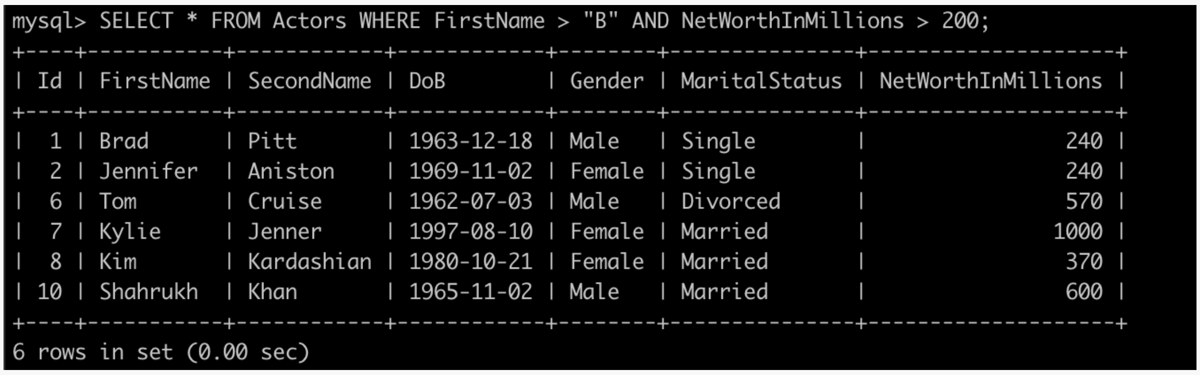
```
-- The lesson queries are reproduced below for convenient copy/paste into the terminal.

-- Query 1
SELECT * FROM Actors WHERE FirstName > "B" AND NetWorthInMillions > 200;

-- Query 2
SELECT * FROM Actors WHERE FirstName > "B" OR NetWorthInMillions > 200;
```

```
-- Query 3
SELECT * FROM Actors WHERE (FirstName > 'B' AND FirstName < 'J') OR (SecondName >'I' AND Sec

-- Query 4
SELECT * FROM Actors WHERE NOT(FirstName > "B" OR NetWorthInMillions > 200);

-- Query 5
SELECT * FROM Actors WHERE NOT NetWorthInMillions = 200;

-- Query 6
SELECT * FROM Actors WHERE (NOT NetWorthInMillions) = 200;

-- Query 7
SELECT * FROM Actors WHERE FirstName > "B" XOR NetWorthInMillions > 200;
```

Terminal

1. We can use the **AND** operator to query for actors whose first name starts with the letter 'B' or any character thereafter and whose net worth is greater than 200 million dollars. Any row that satisfies both these conditions is displayed.

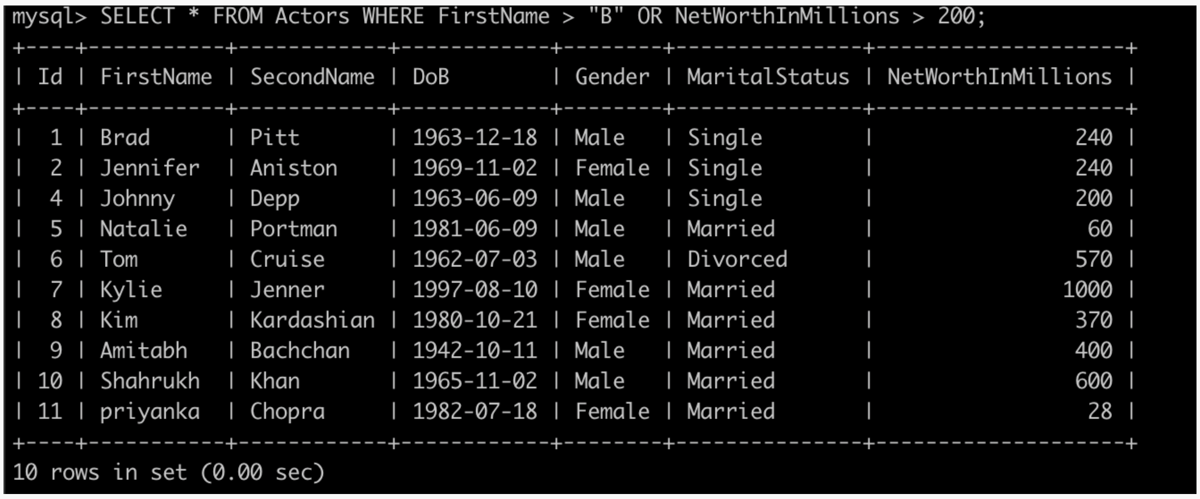
```
SELECT * FROM Actors WHERE FirstName > "B" AND NetWorthInMillion
s > 200;
```



```
mysql> SELECT * FROM Actors WHERE FirstName > "B" AND NetWorthInMillions > 200;
+----+-----------+------------+------------+--------+---------------+--------------------+
| 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 |
|  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 |
| 10 | Shahrukh  | Khan       | 1965-11-02 | Male   | Married       |                600 |
+----+-----------+------------+------------+--------+---------------+--------------------+
6 rows in set (0.00 sec)
```

2. We can use the **OR** operator to match rows that satisfy at least one of several conditions specified in the **WHERE** clause. We can now query for actors whose first name starts with the letter B or any character thereafter or has a net worth of 200 million dollars or greater. The query now returns four more rows than the previous **AND** query.

```
SELECT * FROM Actors WHERE FirstName > "B" OR NetWorthInMillions
> 200;
```

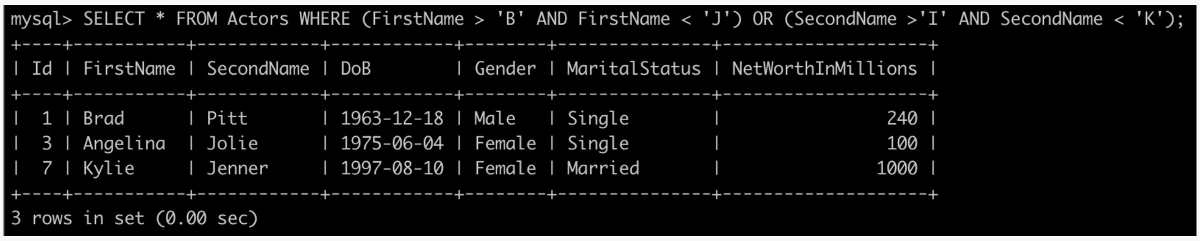


```
mysql> SELECT * FROM Actors WHERE FirstName > "B" OR NetWorthInMillions > 200;
+----+-----------+------------+------------+--------+---------------+--------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+--------------------+
|  1 | Brad      | Pitt       | 1963-12-18 | Male   | Single        |                240 |
|  2 | Jennifer  | Aniston    | 1969-11-02 | Female | Single        |                240 |
|  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 |
+----+-----------+------------+------------+--------+---------------+--------------------+
10 rows in set (0.00 sec)
```

3. We can also combine the **AND** and the **OR** operators. Consider the following query:

```
SELECT * FROM Actors WHERE (FirstName > 'B' AND FirstName < 'J')
OR (SecondName >'I' AND SecondName < 'K');
```

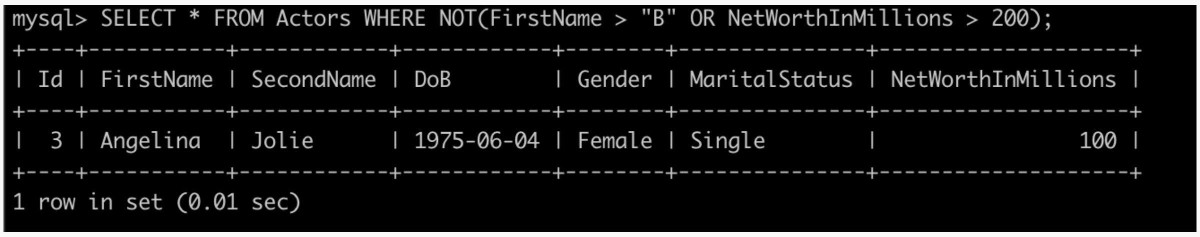
Each clause within the parentheses in the above query selects an actor meeting the criteria and the OR-ing of the two clauses prints three actors.



```
mysql> SELECT * FROM Actors WHERE (FirstName > 'B' AND FirstName < 'J') OR (SecondName >'I' AND SecondName < 'K');
+----+-----------+------------+------------+--------+---------------+--------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+--------------------+
|  1 | Brad      | Pitt       | 1963-12-18 | Male   | Single        |                240 |
|  3 | Angelina  | Jolie      | 1975-06-04 | Female | Single        |                100 |
|  7 | Kylie     | Jenner     | 1997-08-10 | Female | Married       |               1000 |
+----+-----------+------------+------------+--------+---------------+--------------------+
3 rows in set (0.00 sec)
```

4. **NOT** is a unary operator that negates a boolean statement. For example, the following query gives us the complement of the result set that includes actors with a net worth greater than two hundred million dollars and whose first name starts with alphabet 'B' or higher. The complement includes only one row.

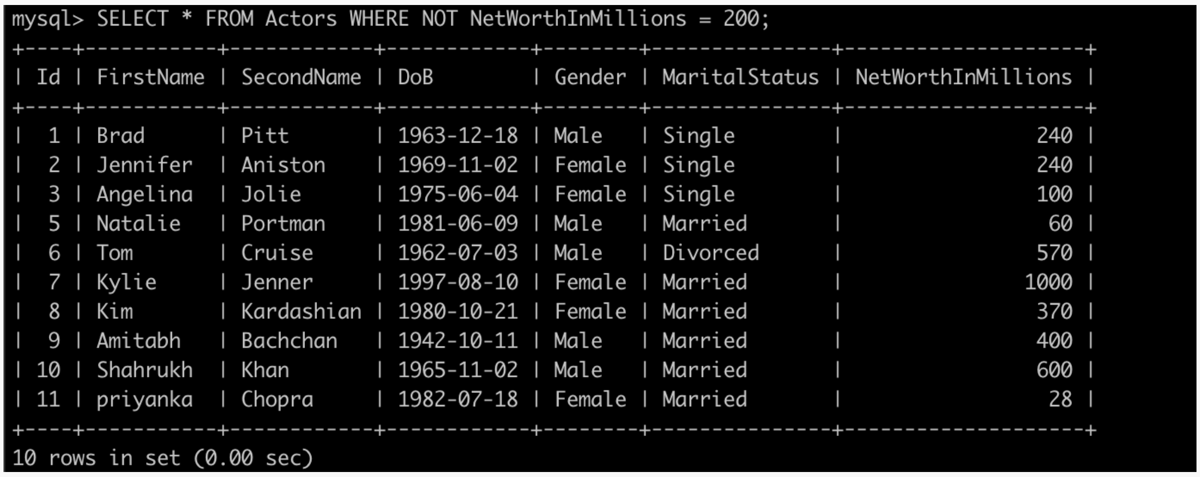
```
SELECT * FROM Actors WHERE NOT(FirstName > "B" OR NetWorthInMilli
ons > 200);
```



```
mysql> SELECT * FROM Actors WHERE NOT(FirstName > "B" OR NetWorthInMillions > 200);
+----+-----------+------------+------------+--------+---------------+--------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+--------------------+
|  3 | Angelina  | Jolie      | 1975-06-04 | Female | Single        |                100 |
+----+-----------+------------+------------+--------+---------------+--------------------+
1 row in set (0.01 sec)
```

The rows matching the two conditions in the parentheses are excluded by the **NOT** and everything else is included. The **NOT** operator's precedence can be tricky. Consider the following query:

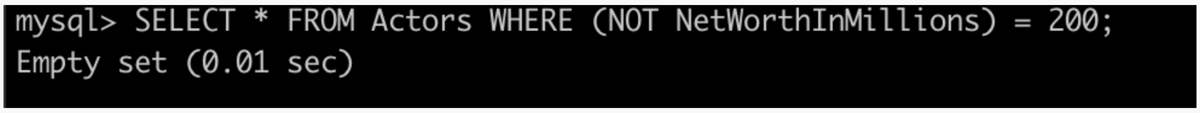
```
SELECT * FROM Actors WHERE NOT NetWorthInMillions = 200;
```



```
mysql> SELECT * FROM Actors WHERE NOT NetWorthInMillions = 200;
+----+-----------+------------+------------+--------+---------------+--------------------+
| 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 |
|  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 |
+----+-----------+------------+------------+--------+---------------+--------------------+
10 rows in set (0.00 sec)
```

The above query returns all the actors with a net worth not equal to 200 million dollars. However, if we put parentheses as follows around the **NOT** operator, the result is an empty set:

```
SELECT * FROM Actors WHERE (NOT NetWorthInMillions) = 200;
```



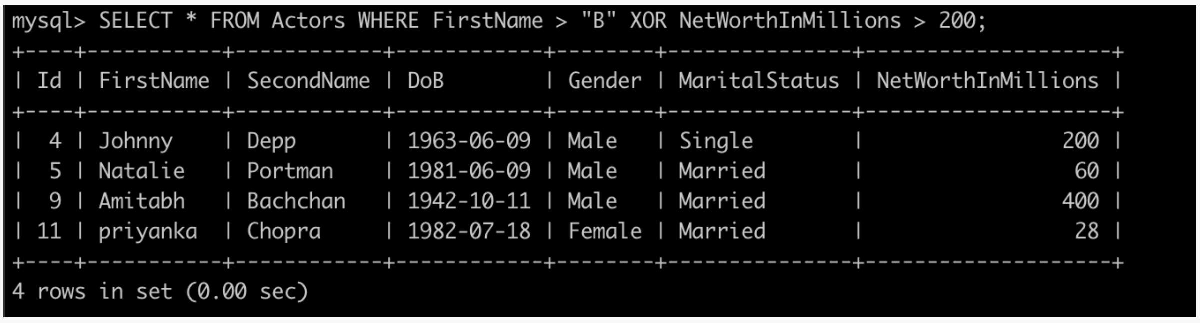
```
mysql> SELECT * FROM Actors WHERE (NOT NetWorthInMillions) = 200;
Empty set (0.01 sec)
```

The **NOT** operator is applied to the column NetWorthInMillions which has all non-zero values for all the rows in the table. Applying **NOT** on a non-zero column value makes it a zero, and since zero isn't equal to 200, no rows are displayed.

Also note that if the table had a row with a zero value for the column NetWorthInMillions, it will still not display anything because **NOT** of zero is non-zero, which isn't equal to 200.

5. MySQL supports exclusive OR through the **XOR** operator. Exclusive OR returns true when one of the two conditions is true. If both conditions are true, or both are false, the result of the XOR operations is false. If we XOR the conditions from the previous query, we are returned four rows. The rows satisfy either of the conditions but not both. All the other rows in the table either fail or satisfy both conditions and aren't included in the result set.

```sql
SELECT * FROM Actors WHERE FirstName > "B" XOR NetWorthInMillion
s > 200;
```



```
mysql> SELECT * FROM Actors WHERE FirstName > "B" XOR NetWorthInMillions > 200;
+----+-----------+------------+------------+--------+---------------+--------------------+
| Id | FirstName | SecondName | DoB        | Gender | MaritalStatus | NetWorthInMillions |
+----+-----------+------------+------------+--------+---------------+--------------------+
|  4 | Johnny    | Depp       | 1963-06-09 | Male   | Single        |                200 |
|  5 | Natalie   | Portman    | 1981-06-09 | Male   | Married       |                 60 |
|  9 | Amitabh   | Bachchan   | 1942-10-11 | Male   | Married       |                400 |
| 11 | priyanka  | Chopra     | 1982-07-18 | Female | Married       |                 28 |
+----+-----------+------------+------------+--------+---------------+--------------------+
4 rows in set (0.00 sec)
```