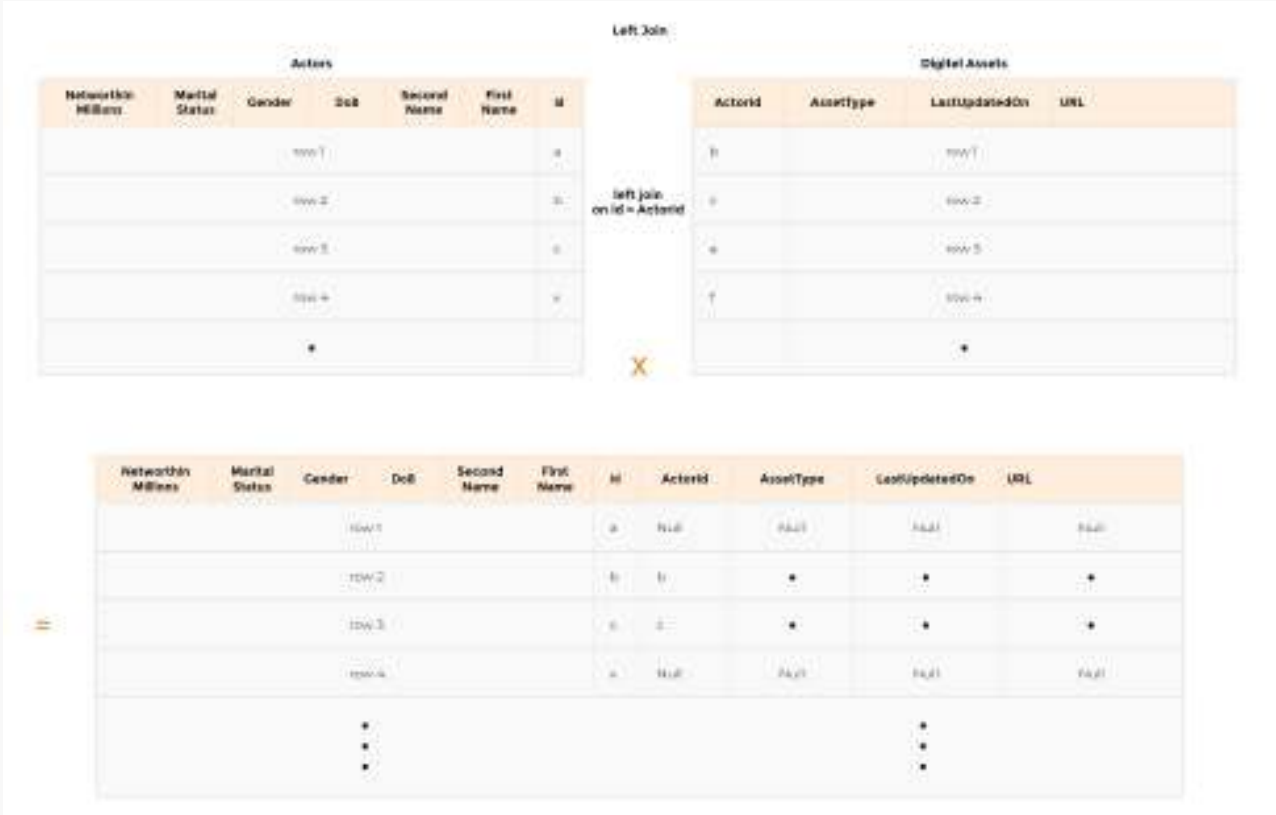
Left and Right Joins

This lesson discusses left and right joins.

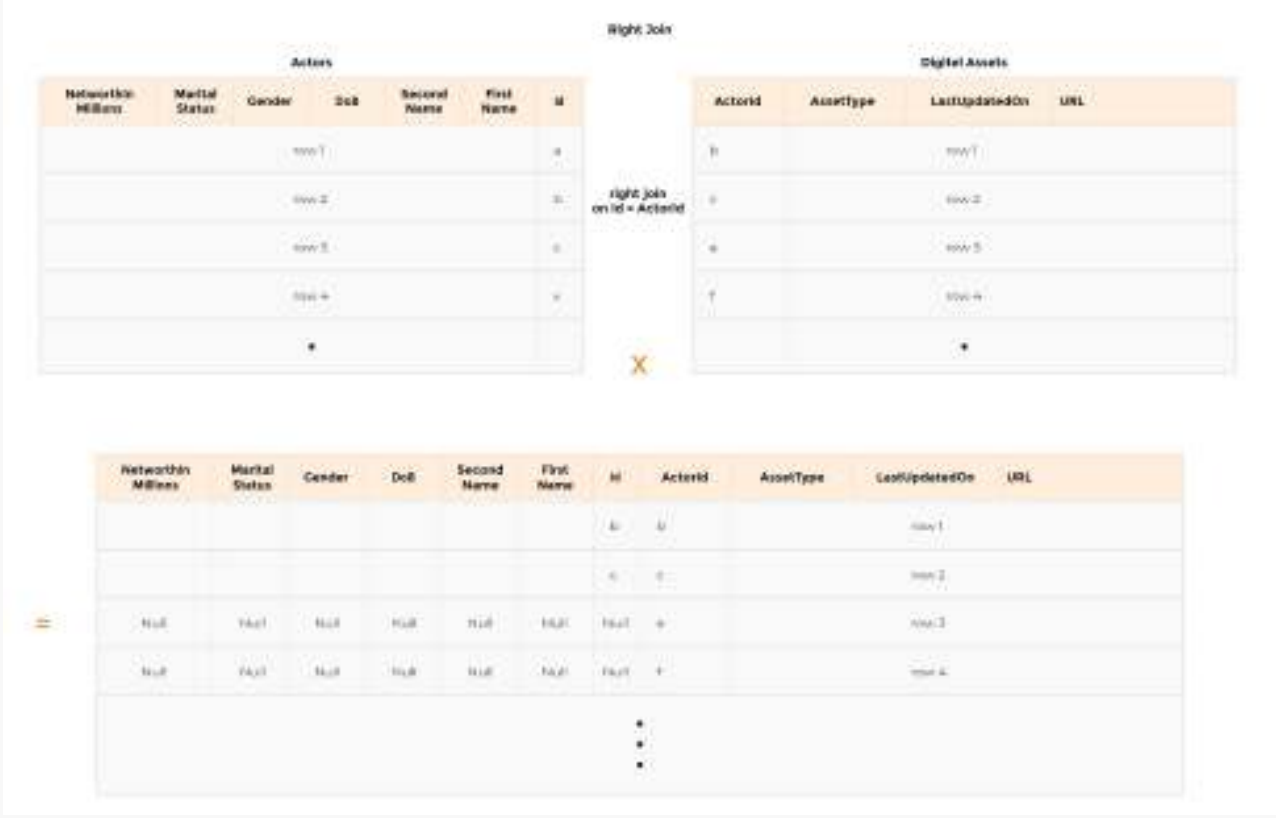
Left & Right Joins

In this lesson we'll look at left and right joins. The two joins add additional rows to the result set for one of the tables participating in the join. We can best exemplify the two joins pictorially as follows:

Left Join



Right Join



Syntax for Left Join

```
SELECT *

FROM table1

LEFT [OUTER] JOIN table2

ON <join condition>
```

Syntax for Right Join

```
SELECT *

FROM table1

RIGHT [OUTER] JOIN table2

ON <join condition>
```

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/28lesson.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, AssetType, URL
FROM Actors
LEFT JOIN DigitalAssets
ON Actors.Id = DigitalAssets.ActorID;

-- Query 2
SELECT FirstName, SecondName, AssetType, URL
FROM DigitalAssets
LEFT JOIN Actors
ON Actors.Id = DigitalAssets.ActorID;

-- Query 3
SELECT FirstName, SecondName, AssetType, URL
FROM Actors
RIGHT JOIN DigitalAssets
ON Actors.Id = DigitalAssets.ActorID;
```

1. We'll start with the query from the inner join lesson that output all the actors with digital assets. If you remember, the inner join query only outputs celebrities who have a digital presence. If we use the **LEFT JOIN** instead, we'll get a list of all the actors with or without digital presence. The query is shown below:

```
SELECT FirstName, SecondName, AssetType, URL

FROM Actors

LEFT JOIN DigitalAssets

ON Actors.Id = DigitalAssets.ActorID;
```

```
mysql> SELECT FirstName, SecondName, AssetType, URL
    ->
    -> FROM Actors
    ->
    -> LEFT JOIN DigitalAssets
    ->
    -> ON Actors.Id = DigitalAssets.ActorID;
+-----------+------------+-----------+-----------------------------------------------------+
| FirstName | SecondName | AssetType | URL                                                 |
+-----------+------------+-----------+-----------------------------------------------------+
| Jennifer  | Aniston    | Website   | http://jennifer-aniston.org                         |
| Angelina  | Jolie      | Website   | http://www.angelina-jolie.com                       |
| Tom       | Cruise     | Website   | http://www.tomcruise.com                            |
| Shahrukh  | Khan       | Twitter   | https://twitter.com/iamsrk                          |
| Jennifer  | Aniston    | Twitter   | https://twitter.com/jenniferannistn                 |
| Angelina  | Jolie      | Twitter   | https://twitter.com/joliestweet                     |
| Kim       | Kardashian | Twitter   | https://twitter.com/KimKardashian                   |
| Natalie   | Portman    | Twitter   | https://twitter.com/natpdotcom                      |
| Tom       | Cruise     | Twitter   | https://twitter.com/TomCruise                       |
| Brad      | Pitt       | Website   | https://www.bradpittweb.com                         |
| Shahrukh  | Khan       | Facebook  | https://www.facebook.com/IamSRK                     |
| Jennifer  | Aniston    | Facebook  | https://www.facebook.com/JenniferAniston            |
| Johnny    | Depp       | Website   | https://www.facebook.com/JohnChristopherOfficial    |
| Kim       | Kardashian | Facebook  | https://www.facebook.com/KimKardashian              |
| Natalie   | Portman    | Facebook  | https://www.facebook.com/natalieportmandotcom       |
| Tom       | Cruise     | Facebook  | https://www.facebook.com/officialtomcruise          |
| Brad      | Pitt       | Instagram | https://www.instagram.com/bradpittoficial           |
| Kim       | Kardashian | Website   | https://www.kkwbeauty.com                           |
| Natalie   | Portman    | Website   | https://www.natalieportman.com                      |
| Angelina  | Jolie      | Pinterest | https://www.pinterest.com/angelinajolie5601         |
| Natalie   | Portman    | Pinterest | https://www.pinterest.com/natalieportmandotcom      |
| Kylie     | Jenner     | NULL      | NULL                                                |
| Amitabh   | Bachchan   | NULL      | NULL                                                |
| priyanka  | Chopra     | NULL      | NULL                                                |
+-----------+------------+-----------+-----------------------------------------------------+
24 rows in set (0.00 sec)
```

Note that the output now includes those actors who don't have a digital presence. The **LEFT JOIN** includes those rows from the table on its left that don't match with rows in the table to its right.

2. Interestingly, if we flip the order of the two tables in the query we get a different result:

```
SELECT FirstName, SecondName, AssetType, URL

FROM DigitalAssets

LEFT JOIN Actors

ON Actors.Id = DigitalAssets.ActorID;
```

```
mysql> SELECT FirstName, SecondName, AssetType, URL
    ->
    -> FROM DigitalAssets
    ->
    -> LEFT JOIN Actors
    ->
    -> ON Actors.Id = DigitalAssets.ActorID;
+-----------+------------+-----------+-----------------------------------------------------+
| FirstName | SecondName | AssetType | URL                                                 |
+-----------+------------+-----------+-----------------------------------------------------+
| Jennifer  | Aniston    | Website   | http://jennifer-aniston.org                         |
| Angelina  | Jolie      | Website   | http://www.angelina-jolie.com                       |
| Tom       | Cruise     | Website   | http://www.tomcruise.com                            |
| Shahrukh  | Khan       | Twitter   | https://twitter.com/iamsrk                          |
| Jennifer  | Aniston    | Twitter   | https://twitter.com/jenniferannistn                 |
| Angelina  | Jolie      | Twitter   | https://twitter.com/joliestweet                     |
| Kim       | Kardashian | Twitter   | https://twitter.com/KimKardashian                   |
| Natalie   | Portman    | Twitter   | https://twitter.com/natpdotcom                      |
| Tom       | Cruise     | Twitter   | https://twitter.com/TomCruise                       |
| Brad      | Pitt       | Website   | https://www.bradpittweb.com                         |
| Shahrukh  | Khan       | Facebook  | https://www.facebook.com/IamSRK                     |
| Jennifer  | Aniston    | Facebook  | https://www.facebook.com/JenniferAniston            |
| Johnny    | Depp       | Website   | https://www.facebook.com/JohnChristopherOfficial    |
| Kim       | Kardashian | Facebook  | https://www.facebook.com/KimKardashian              |
| Natalie   | Portman    | Facebook  | https://www.facebook.com/natalieportmandotcom       |
| Tom       | Cruise     | Facebook  | https://www.facebook.com/officialtomcruise          |
| Brad      | Pitt       | Instagram | https://www.instagram.com/bradpittoficial           |
| Kim       | Kardashian | Website   | https://www.kkwbeauty.com                           |
| Natalie   | Portman    | Website   | https://www.natalieportman.com                      |
| Angelina  | Jolie      | Pinterest | https://www.pinterest.com/angelinajolie5601         |
| Natalie   | Portman    | Pinterest | https://www.pinterest.com/natalieportmandotcom      |
+-----------+------------+-----------+-----------------------------------------------------+
21 rows in set (0.00 sec)
```

The outcome makes sense, because the **DigitalAssets** table doesn't have any rows that don't have an owner in the **Actors** table, so all the rows in the **DigitalAssets** table match with a row in the **Actors** table and become part of the output. Note that actors without digital presence are left out.

3. The **RIGHT JOIN** is very similar to the **LEFT JOIN**. The only difference is that in the case of the left join, the unmatched rows come from the table specified on the left of the **LEFT JOIN** clause whereas, in the case of right join, the unmatched rows come from the table specified on the right of the **RIGHT JOIN** clause. If we use right join in the first query of the lesson, we would not need to flip the tables as we did above.

```
SELECT FirstName, SecondName, AssetType, URL

FROM Actors

RIGHT JOIN DigitalAssets
```

```
ON Actors.Id = DigitalAssets.ActorID;
```

```
mysql> SELECT FirstName, SecondName, AssetType, URL
    ->
    -> FROM Actors
    ->
    -> RIGHT JOIN DigitalAssets
    ->
    -> ON Actors.Id = DigitalAssets.ActorID;
+-----------+------------+-----------+-----------------------------------------------+
| FirstName | SecondName | AssetType | URL                                           |
+-----------+------------+-----------+-----------------------------------------------+
| Jennifer  | Aniston    | Website   | http://jennifer-aniston.org                   |
| Angelina  | Jolie      | Website   | http://www.angelina-jolie.com                 |
| Tom       | Cruise     | Website   | http://www.tomcruise.com                      |
| Shahrukh  | Khan       | Twitter   | https://twitter.com/iamsrk                    |
| Jennifer  | Aniston    | Twitter   | https://twitter.com/jenniferannistn           |
| Angelina  | Jolie      | Twitter   | https://twitter.com/jolietweet                |
| Kim       | Kardashian | Twitter   | https://twitter.com/KimKardashian             |
| Natalie   | Portman    | Twitter   | https://twitter.com/natpdotcom                |
| Tom       | Cruise     | Twitter   | https://twitter.com/TomCruise                 |
| Brad      | Pitt       | Website   | https://www.bradpittweb.com                   |
| Shahrukh  | Khan       | Facebook  | https://www.facebook.com/IamSRK               |
| Jennifer  | Aniston    | Facebook  | https://www.facebook.com/JenniferAniston      |
| Johnny    | Depp       | Website   | https://www.facebook.com/JohnChristopherOfficial |
| Kim       | Kardashian | Facebook  | https://www.facebook.com/KimKardashian        |
| Natalie   | Portman    | Facebook  | https://www.facebook.com/nataliepartmandotcom |
| Tom       | Cruise     | Facebook  | https://www.facebook.com/officialtomcruise    |
| Brad      | Pitt       | Instagram | https://www.instagram.com/bradpittoficial     |
| Kim       | Kardashian | Website   | https://www.kkwbeauty.com                     |
| Natalie   | Portman    | Website   | https://www.natalieportman.com                |
| Angelina  | Jolie      | Pinterest | https://www.pinterest.com/angelinajolie5601   |
| Natalie   | Portman    | Pinterest | https://www.pinterest.com/nataliepartmandotcom |
+-----------+------------+-----------+-----------------------------------------------+
21 rows in set (0.00 sec)
```

4. Note that an alternative syntax for left and right joins is **LEFT OUTER JOIN** and **RIGHT OUTER JOIN** respectively, though there's no difference in functionality if you skip the **OUTER** keyword.