Inner Join

This lesson demonstrates how to perform an inner join.

Inner Join

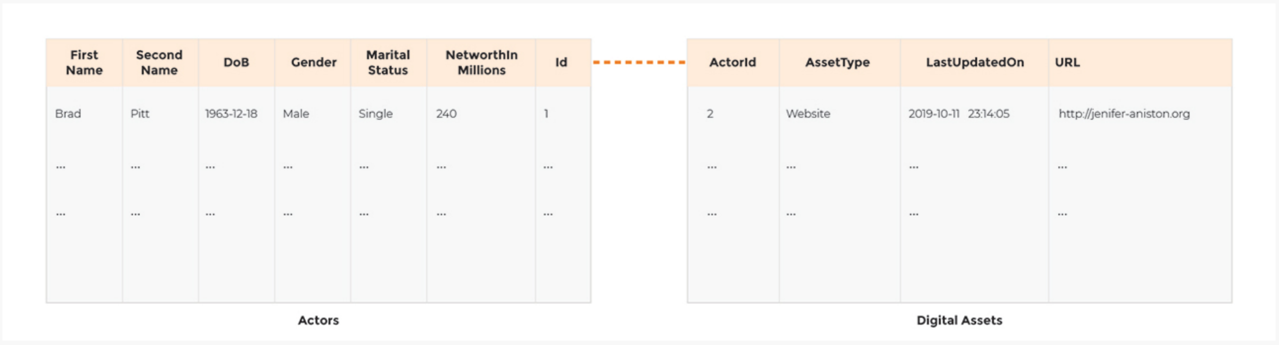
In the previous lesson, we saw how to join a table with itself. In this lesson we'll join two different tables. We'll introduce another table called **DigitalAssets** that'll contain the online public properties such as Twitter, Facebook, and Pinterest belonging to a celebrity. The table structure is shown below:

Column Name	Column Type
URL	VARCHAR(200)
AssetType	Enum('Facebook','Twitter', 'Instagram','Pinterest','Website')
LastUpdatedOn	TIMESTAMP
ActorId	INT

A few rows from the table are shown below:

```
+-------------------------------------+-----------+---------------------+---------+
| URL                                 | AssetType | LastUpdatedOn       | ActorId |
+-------------------------------------+-----------+---------------------+---------+
| http://jennifer-aniston.org         | Website   | 2019-10-11 23:14:05 |       2 |
| http://www.angelina-jolie.com       | Website   | 2019-05-01 12:54:02 |       3 |
| http://www.tomcruise.com            | Website   | 2019-10-23 09:56:33 |       6 |
| https://twitter.com/iamsrk          | Twitter   | 2019-08-18 18:39:08 |      10 |
| https://twitter.com/jenniferannistn | Twitter   | 2019-02-13 03:04:25 |       2 |
+-------------------------------------+-----------+---------------------+---------+
```

Note that the primary key of the table is the **URL** column as every URL is guaranteed to be unique. The **DigitalAssets** table is linked with the **Actors** table with the common column of ID for the actor as shown below. However, note that the column names in the two tables are different.



First Name	Second Name	DoB	Gender	Marital Status	NetworthIn Millions	Id
Brad	Pitt	1963-12-18	Male	Single	240	1
...	...	...	...	...	...	...
...	...	...	...	...	...	...

Actors

ActorId	AssetType	LastUpdatedOn	URL
2	Website	2019-10-11 23:14:05	http://jenifer-aniston.org
...	...	...	...
...	...	...	...

Digital Assets

Syntax

SELECT *

FROM **table1**

INNER 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/26lesson.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
-- Query 1
SELECT FirstName, SecondName, AssetType, URL
FROM Actors
INNER JOIN DigitalAssets
ON Actors.Id = DigitalAssets.ActorID;

-- Query 2
SELECT FirstName, SecondName, AssetType, URL
FROM Actors
INNER JOIN DigitalAssets
USING(Id);

-- Query 3
SELECT FirstName, SecondName, AssetType, URL
FROM Actors, DigitalAssets
WHERE ActorId=Id;

-- Query 4
SELECT FirstName, SecondName, AssetType, URL
FROM Actors, DigitalAssets;

-- Query 5
SELECT FirstName, SecondName, AssetType, URL
FROM Actors
INNER JOIN DigitalAssets;

-- Query 6
-- Makes no sense to join tables on FirstName and URL columns as they aren't related.
SELECT *
FROM Actors
INNER JOIN DigitalAssets ON URL = FirstName;

-- Query 7
-- Again no sense in combining net worth and actor id. Additionally, one is an int and the ot
SELECT *
FROM Actors
INNER JOIN DigitalAssets
ON NetWorthInMillions = ActorId;
```

Terminal

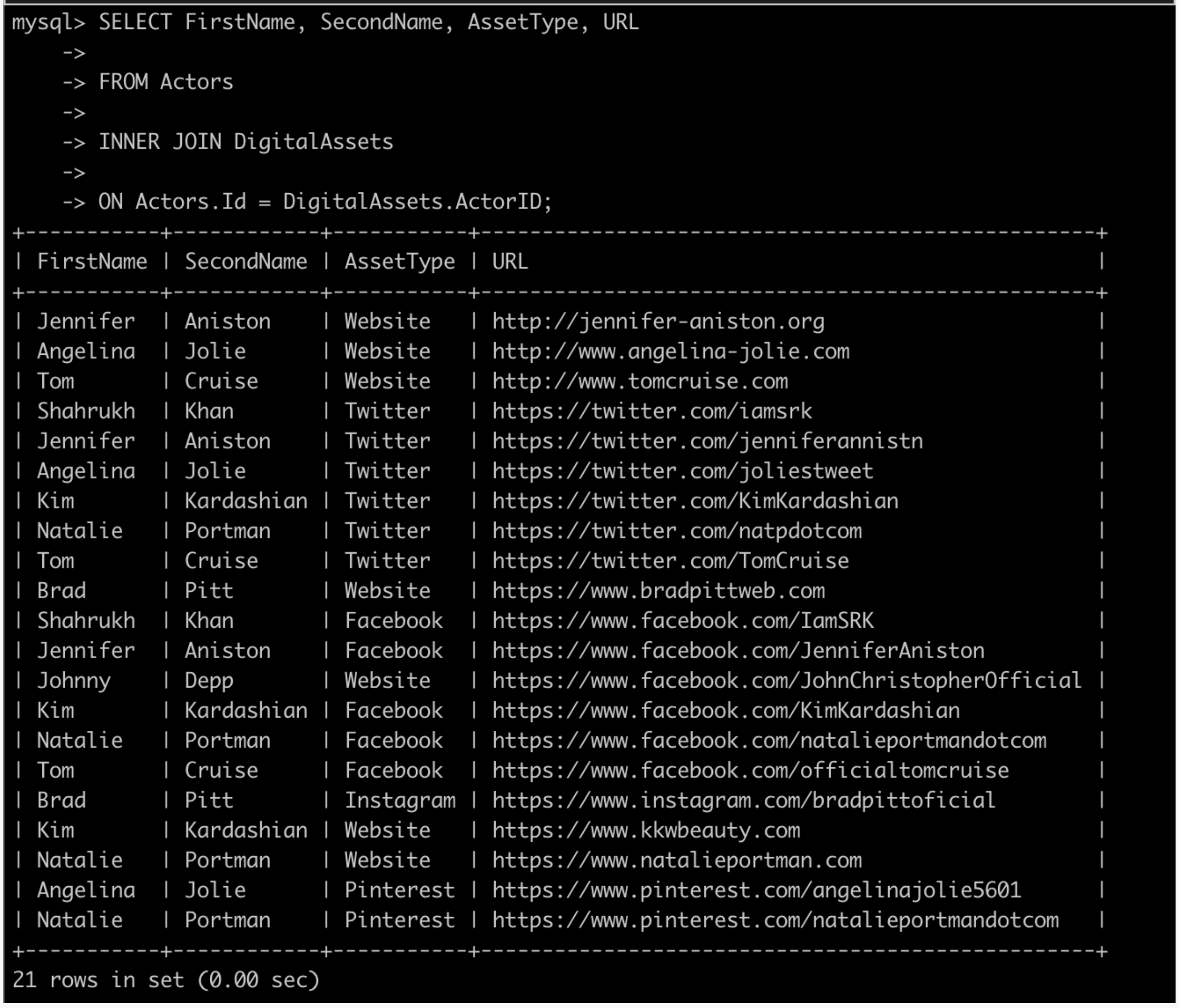
1. Using the **INNER JOIN**, we are now able to answer queries such as listing the Facebook pages for each celebrity. Note that each table in isolation can't answer this query as the **Actors** table doesn't hold the digital assets information for each actor and the **DigitalAssets** table doesn't hold the names for each actor.

```
SELECT FirstName, SecondName, AssetType, URL

FROM Actors

INNER JOIN DigitalAssets
```

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



```
mysql> SELECT FirstName, SecondName, AssetType, URL
    ->
    -> FROM Actors
    ->
    -> INNER 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 |
+-----------+------------+-----------+---------------------------------------------+
21 rows in set (0.00 sec)
```

2. If the two tables had the same column name for the actor's ID then we could have used the alternative syntax with **USING** clause to make the query slightly less verbose as shown below:

```
SELECT FirstName, SecondName, AssetType, URL

FROM Actors

INNER JOIN DigitalAssets

USING(Id);
```

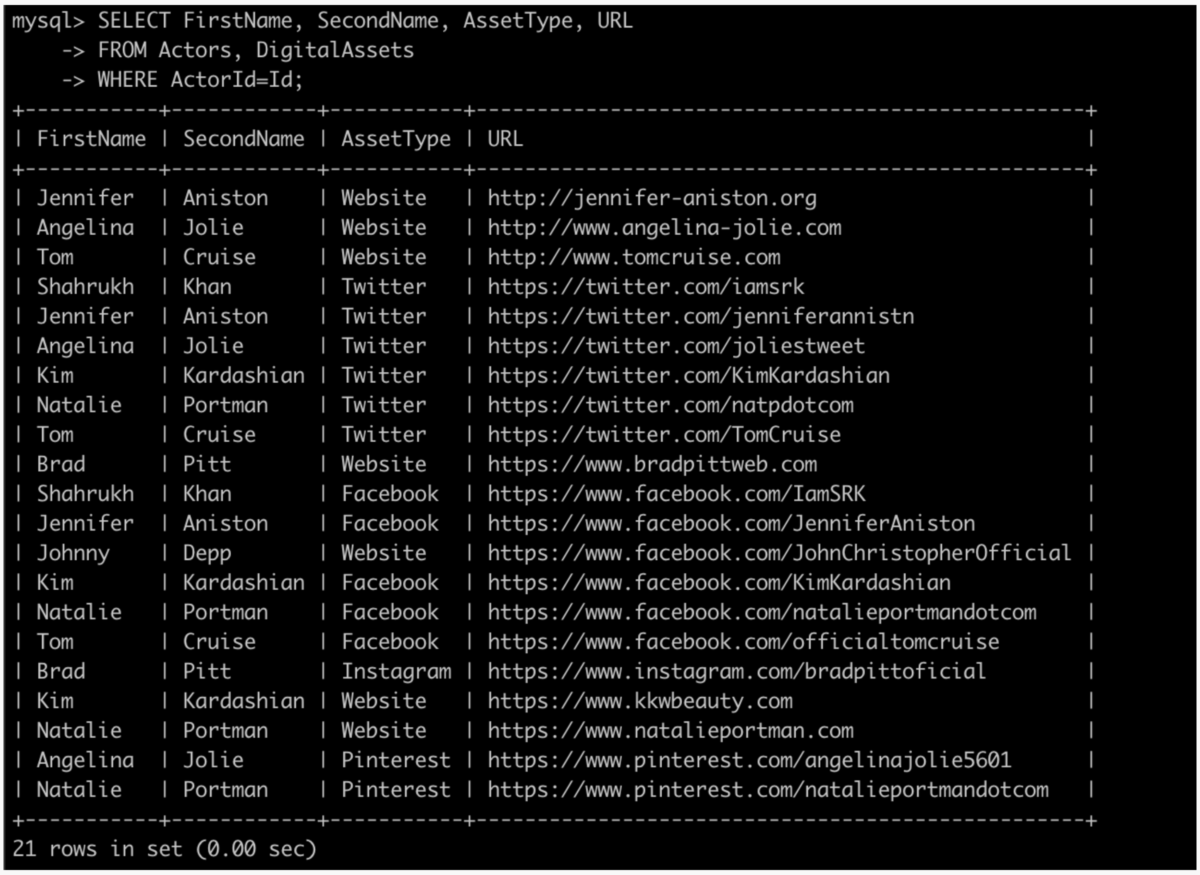
Note that the columns listed in the **SELECT** clause are unique across the two tables. However, if the two tables had columns with the same names then we would need to disambiguate the two by fully qualifying the column with the table name.

Also notice that celebrities with no digital assets, or assets with no corresponding celebrity entries, in the **Actors** table aren't captured with the results of the query. The server picks rows from both tables with the results of the query. The server picks rows from both tables that have the same value for the two columns. Or you can think of it

as an intersection of the two tables based on the IDs of the celebrities.

3. It's not necessary to use the **INNER JOIN** clause to get an inner join between two tables. We can also use the **WHERE** clause to achieve the same effect as shown below:

```
SELECT FirstName, SecondName, AssetType, URL
FROM Actors, DigitalAssets
WHERE ActorId=Id;
```



```
mysql> SELECT FirstName, SecondName, AssetType, URL
    -> FROM Actors, DigitalAssets
    -> WHERE ActorId=Id;
+-----------+------------+-----------+-----------------------------------------------------+
| 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)
```

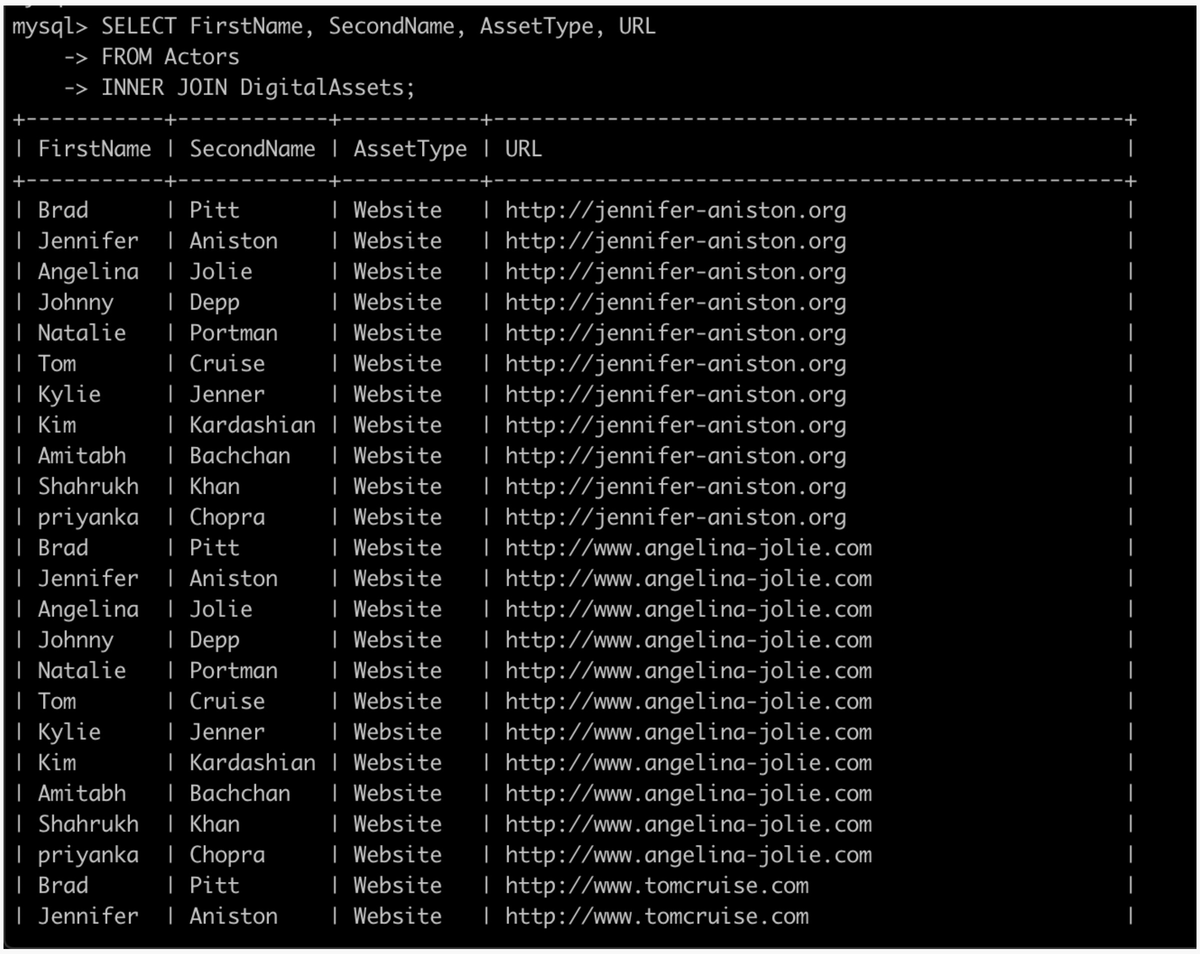
There's no difference in using the **WHERE** clause or the INNER JOIN clause in query performance, rather it is just a matter of taste.

4. We can also create a cartesian product between the two tables as we did in the self join section. We can use either the where or the inner join syntax. Both are shown below:

```
SELECT FirstName, SecondName, AssetType, URL
FROM Actors, DigitalAssets;
```

Or,

```sql
SELECT FirstName, SecondName, AssetType, URL
FROM Actors
INNER JOIN DigitalAssets;
```

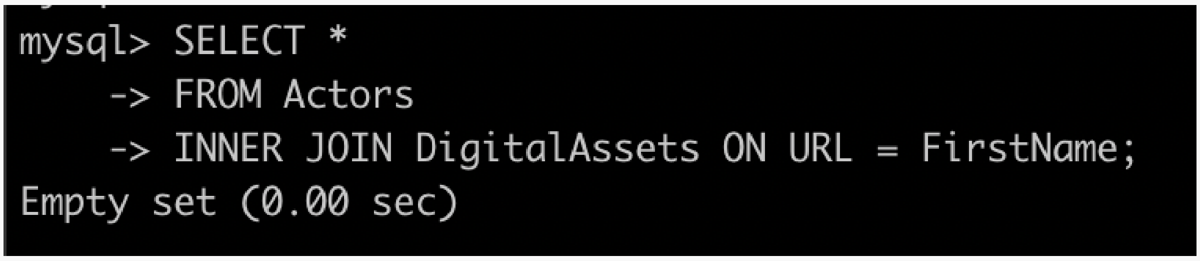


```
mysql> SELECT FirstName, SecondName, AssetType, URL
    -> FROM Actors
    -> INNER JOIN DigitalAssets;
+-----------+------------+-----------+--------------------------------------------------+
| FirstName | SecondName | AssetType | URL                                              |
+-----------+------------+-----------+--------------------------------------------------+
| Brad      | Pitt       | Website   | http://jennifer-aniston.org                      |
| Jennifer  | Aniston    | Website   | http://jennifer-aniston.org                      |
| Angelina  | Jolie      | Website   | http://jennifer-aniston.org                      |
| Johnny    | Depp       | Website   | http://jennifer-aniston.org                      |
| Natalie   | Portman    | Website   | http://jennifer-aniston.org                      |
| Tom       | Cruise     | Website   | http://jennifer-aniston.org                      |
| Kylie     | Jenner     | Website   | http://jennifer-aniston.org                      |
| Kim       | Kardashian | Website   | http://jennifer-aniston.org                      |
| Amitabh   | Bachchan   | Website   | http://jennifer-aniston.org                      |
| Shahrukh  | Khan       | Website   | http://jennifer-aniston.org                      |
| priyanka  | Chopra     | Website   | http://jennifer-aniston.org                      |
| Brad      | Pitt       | Website   | http://www.angelina-jolie.com                    |
| Jennifer  | Aniston    | Website   | http://www.angelina-jolie.com                    |
| Angelina  | Jolie      | Website   | http://www.angelina-jolie.com                    |
| Johnny    | Depp       | Website   | http://www.angelina-jolie.com                    |
| Natalie   | Portman    | Website   | http://www.angelina-jolie.com                    |
| Tom       | Cruise     | Website   | http://www.angelina-jolie.com                    |
| Kylie     | Jenner     | Website   | http://www.angelina-jolie.com                    |
| Kim       | Kardashian | Website   | http://www.angelina-jolie.com                    |
| Amitabh   | Bachchan   | Website   | http://www.angelina-jolie.com                    |
| Shahrukh  | Khan       | Website   | http://www.angelina-jolie.com                    |
| priyanka  | Chopra     | Website   | http://www.angelina-jolie.com                    |
| Brad      | Pitt       | Website   | http://www.tomcruise.com                         |
| Jennifer  | Aniston    | Website   | http://www.tomcruise.com                         |
```

5. We can join any two columns from two tables that have the same type, or which can be converted to one another albeit with data loss. For instance, the following two queries are nonsensical, but the tables can still be joined on the columns that appear in the queries.

```sql
-- Makes no sense to join tables on FirstName and URL columns a
s they aren't related.

SELECT *
FROM Actors
INNER JOIN DigitalAssets ON URL = FirstName;
```

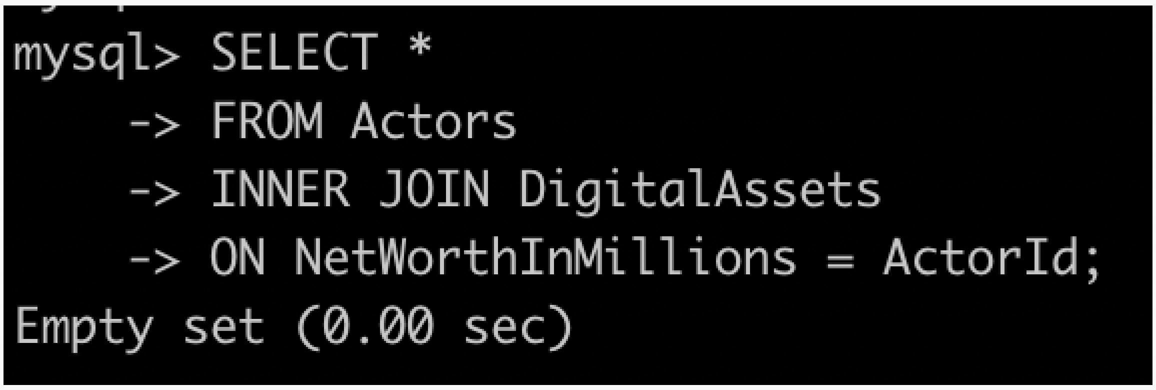


```
mysql> SELECT *
    -> FROM Actors
    -> INNER JOIN DigitalAssets ON URL = FirstName;
Empty set (0.00 sec)
```

Or,

```sql
-- Again no sense in combining net worth and actor id. Additional
ly, one is an int and the other a decimal but still comparable.

SELECT *
FROM Actors
INNER JOIN DigitalAssets
ON NetWorthInMillions = ActorId;
```



```
mysql> SELECT *
    -> FROM Actors
    -> INNER JOIN DigitalAssets
    -> ON NetWorthInMillions = ActorId;
Empty set (0.00 sec)
```

Both the queries result in empty sets.