/\*

We will start with a simple example, and will start with an INNER JOIN.  
  
As you can see below, there are 3 tables:  
**character**: Each character is a row and is represented by a unique identifier (id), e.g. 1 is Doogie Howser  
**character\_tv\_show:** For each character, which show is he/she in?  
**character\_actor**: For each character, who is the actor?  
  
See that in **character\_tv\_show**, instead of storing both the character and TV show names (e.g. Willow Rosenberg and Buffy the Vampire Slayer), it stores the character\_id as a substitute for the character name. This character\_id refers to the matching id row from the **character** table.  
  
This is done so data is not duplicated. For example, if the name of a character were to change, you would only have to change the name of the character in one row.  
  
This allows us to "join" the tables together "on" that reference/common column.

***character***

|  |  |
| --- | --- |
| **id** | **name** |
| 1 | Doogie Howser |
| 2 | Barney Stinson |
| 3 | Lily Aldrin |
| 4 | Willow Rosenberg |

***character\_tv\_show***

|  |  |  |
| --- | --- | --- |
| **id** | **character\_id** | **tv\_show\_name** |
| 1 | 4 | Buffy the Vampire Slayer |
| 2 | 3 | How I Met Your Mother |
| 3 | 2 | How I Met Your Mother |
| 4 | 1 | Doogie Howser, M.D. |

***character\_actor***

|  |  |  |
| --- | --- | --- |
| **id** | **character\_id** | **actor\_name** |
| 1 | 4 | Alyson Hannigan |
| 2 | 3 | Alyson Hannigan |
| 3 | 2 | Neil Patrick Harris |
| 4 | 1 | Neil Patrick Harris |

Consider the above tables

\*/

--Can you use an inner join to pair each character name with the actor who plays them?

SELECT character.name , character\_actor.actor\_name

FROM character

INNER JOIN character\_actor

ON character.id = character\_actor.character\_id

/\*

In order to not duplicate any names, we need to have more tables, and use multiple joins.  
  
We have tables for characters, TV shows, and actors. Those tables represent things (also known as entities).  
  
In addition those tables, we have the relationship tables **character\_tv\_show** and **character\_actor**, which capture the relationship between two entities.  
  
This is a flexible way of capturing the relationship between different entities, as some TV show characters might be in multiple shows, and some actors are known for playing multiple characters.

***character***

|  |  |
| --- | --- |
| **id** | **name** |
| 1 | Doogie Howser |
| 2 | Barney Stinson |
| 3 | Lily Aldrin |
| 4 | Willow Rosenberg |

***character\_tv\_show***

|  |  |  |
| --- | --- | --- |
| **id** | **character\_id** | **tv\_show\_name** |
| 1 | 4 | Buffy the Vampire Slayer |
| 2 | 3 | How I Met Your Mother |
| 3 | 2 | How I Met Your Mother |
| 4 | 1 | Doogie Howser, M.D. |

***character\_actor***

|  |  |  |
| --- | --- | --- |
| **id** | **character\_id** | **actor\_name** |
| 1 | 4 | Alyson Hannigan |
| 2 | 3 | Alyson Hannigan |
| 3 | 2 | Neil Patrick Harris |
| 4 | 1 | Neil Patrick Harris |

***tv\_show***

|  |  |
| --- | --- |
| **id** | **name** |
| 1 | Buffy the Vampire Slayer |
| 2 | How I Met Your Mother |
| 3 | Doogie Howser, M.D. |

***actor***

|  |  |
| --- | --- |
| **id** | **name** |
| 1 | Alyson Hannigan |
| 2 | Neil Patrick Harris |

\*/

-- Can you use two joins to pair each character name with the actor who plays them?

SELECT character.name , actor.name

FROM character

INNER JOIN character\_actor

ON character.id = character\_actor.character\_id

INNER JOIN actor

ON character\_actor.actor\_id = actor.id;

--You can also use joins with the WHERE clause.

/\*

Can you return a list of characters and TV shows that are not named "Willow Rosenberg" and in the show "How I Met Your Mother"?

\*/

SELECT character.name , tv\_show.name

FROM character

INNER JOIN character\_tv\_show

ON character.id = character\_tv\_show.character\_id

INNER JOIN tv\_show

ON character\_tv\_show.tv\_show\_id = tv\_show.id

WHERE character.name != 'Willow Rosenberg' AND tv\_show.name != 'How I Met Your Mother';

/\*

A LEFT JOIN returns all of the data from the first (or "left") table, and if there isn't corresponding data for the second table, it returns NULL for those columns.

Can you use left joins to match character names with the actors that play them?

\*/

SELECT character.name , actor.name

FROM character

LEFT JOIN character\_actor

ON character.id = character\_actor.character\_id

LEFT JOIN actor

ON character\_actor.actor\_id = actor.id;

/\* If you want to use an alias for a table, you add AS \*alias\_name\* after the table name.

Can you use left joins to match character names with the actors that play them, and use aliases to make the query shorter?

\*/

SELECT c.name , a.name

FROM character AS c

LEFT JOIN character\_actor AS ca

ON c.id = ca.character\_id

LEFT JOIN actor AS a

ON ca.actor\_id = a.id;

/\*

Sometimes, it may make sense for you to do a self join. In that case, you need to use table aliases to determine which data is from the "first"/"left" table.

Can you run a query that returns the name of an employee and the name of their boss? Use column aliases to make the columns employee\_name and boss\_name.

\*/

SELECT e1.name AS employee\_name, e2.name AS boss\_name

FROM employees AS e1

INNER JOIN employees AS e2

ON e1.boss\_id=e2.id;