1. ¿A qué se denomina **JOIN** en una base de datos?

Es el proceso de tomar datos de varias tablas y colocarlos en un filtro general de manera visual. Podría relacionarse a un SELECT que combina columnas entre varias tablas.

1. Nombre y explique 2 tipos de **JOIN**.

**Inner Join**: Toma datos no nulos en común entre dos tablas y los une. Es decir quiere decir que solamente traiga los datos que esten en común en ambas tablas

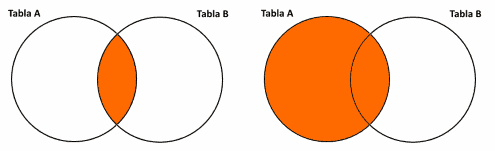
**Left join**: Toma datos de la primera tabla e intenta relacionarlos con la segunda tabla, mostrando la totalidad de la primera tabla

1. ¿Para qué se utiliza el **GROUP BY**?

Este identifica una columna seleccionada para utilizarla para agrupar resultados iguales. Divide datos en grupos por los valores de la columna especificada y devuelve una fila de resultados para cada grupo.

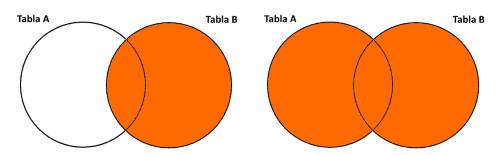
1. ¿Para qué se utiliza el **HAVING**?

El having se utiliza para filtrar los contenidos de la consulta *luego* de haber realizado un group by

1. Dado lo siguientes diagramas indique a qué tipo de **JOIN** corresponde cada uno:

Inner join Left Join

1. Escriba una consulta genérica por cada uno de los diagramas a continuación:



Right join Full Join

**SELECT**

**one.name,**

**two.name**

**FROM table1 AS one**

**RIGHT JOIN table2 AS two**

**ON one.table1id = two.table2id;**

**Select \* tablaA tA**

**FULL JOIN tablaB tB ON tB.id == tA.id**

**SELECT**

**m.title as 'Movie Title',**

**g.name as 'Genre Name'**

**FROM movies AS m**

**FULL JOIN genres AS g**

**ON m.genre\_id = g.id;**

**SEGUNDA PARTE**

1. Mostrar el título y el nombre del género de todas las series.

SELECT s.title, g.name FROM series s INNER JOIN genres g;

SELECT SER.TITLE, GEN.NAME FROM MOVIES\_DB.SERIES AS SER JOIN MOVIES\_DB.GENRES AS GEN ON SER.GENRE\_ID = GEN.ID;

1. Mostrar el título de los episodios, el nombre y apellido de los actores que trabajan en cada uno de ellos.

SELECT title, first\_name, last\_name FROM episodes AS ep

INNER JOIN actor\_episode AS ac\_ep

INNER JOIN actors AS ac

ON ep.id = ac\_ep.episode\_id AND ac.id = ac\_ep.actor\_id

1. Mostrar el título de todas las series y el total de temporadas que tiene cada una de ellas.

**SELECT SER.TITLE , COUNT(SER.ID) AS TEMPORADAS FROM MOVIES\_DB.SEASONS AS SEA**

**JOIN MOVIES\_DB.SERIES AS SER ON SEA.SERIE\_ID = SER.ID**

**GROUP BY SER.ID;**

**Otra opción:** SELECT ser.title, (SELECT count(sea.number) FROM seasons AS sea WHERE ser.id = sea.serie\_id) AS "total de temporadas"

FROM series AS ser INNER JOIN seasons AS sea ON ser.id = sea.serie\_id

GROUP BY ser.id;

1. Mostrar el nombre de todos los géneros y la cantidad total de películas por cada uno, siempre que sea mayor o igual a 3.

***SELECT name, (SELECT COUNT(id) FROM movies WHERE genre\_id = gen.id) cantidad FROM genres AS gen***

***HAVING cantidad > 2;***

***OTRA OPCION:  
SELECT GEN.NAME , COUNT(GEN.ID) AS PELICULAS FROM MOVIES\_DB.GENRES AS GEN***

***JOIN MOVIES\_DB.MOVIES AS MOV ON MOV.GENRE\_ID = GEN.ID***

***GROUP BY GEN.ID***

***HAVING PELICULAS > 2;***

1. Mostrar sólo el nombre y apellido de los actores que trabajan en todas las películas de la guerra de las galaxias y que estos no se repitan.

**SELECT DISTINCT ACT.FIRST\_NAME AS NOMBRE, ACT.LAST\_NAME AS APELLIDO FROM MOVIES\_DB.ACTOR\_MOVIE AS ACT\_MOV**

**JOIN MOVIES\_DB.MOVIES AS MOV ON MOV.ID = ACT\_MOV.MOVIE\_ID AND MOV.TITLE LIKE 'La Guerra de las galaxias%'**

**JOIN MOVIES\_DB.ACTORS AS ACT ON ACT.ID = ACT\_MOV.ACTOR\_ID;**

**Otra opción:** SELECT DISTINCT a.first\_name, a.last\_name

FROM movies AS m

INNER JOIN actors AS a

INNER JOIN actor\_movie AS am

ON am.actor\_id = a.id AND am.movie\_id = m.id

WHERE m.title LIKE "La Guerra de las galaxias%";

Otra opcion: SELECT ACT.FIRST\_NAME AS NOMBRE, ACT.LAST\_NAME AS APELLIDO FROM MOVIES\_DB.ACTOR\_MOVIE AS ACT\_MOV

JOIN MOVIES\_DB.MOVIES AS MOV ON MOV.ID = ACT\_MOV.MOVIE\_ID

JOIN MOVIES\_DB.ACTORS AS ACT ON ACT.ID = ACT\_MOV.ACTOR\_ID

WHERE MOV.TITLE LIKE 'La Guerra de las galaxias%'

GROUP BY ACT.FIRST\_NAME, ACT.LAST\_NAME;