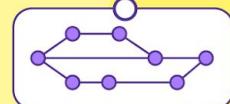


DEV.F:



MASTER en Data Science / AI





JOINS y subconsultas

🐱💻 Objetivos de aprendizaje 🐱💻

Objetivos de aprendizaje



- Empezaremos **estudiando** 🐱💻 la presentación de **JOINS** 🤖❤️ donde haremos una exploración sobre las formas de combinar tablas dentro de una Base de Datos 📊❤️.
- Después, haremos un **challenge** 🐱💻, para reforzar los conocimientos de Bases de Datos Relacionales en referencia a **las formas comunes de unir dos tablas**.
- Por último, daremos un breve espacio para **preguntas** y **conclusiones** 🐱💻.

Agenda

6:30 - 6:35 PM **Saludos y anuncios.**

6:35 - 6:40 PM **Objetivos de aprendizaje.**

6:40 - 7:10 PM Presentación: **Joins**

7:10 - 8:10 PM Notebook

8:10 - 8:25 PM BREAK.

8:25 - 9:25 PM Challenge

9:25 - 9:30 PM **Conclusiones.**



Temas

- 1. Llaves Primarias y Foráneas**
- 2. JOINS**
- 3. Subconsultas**



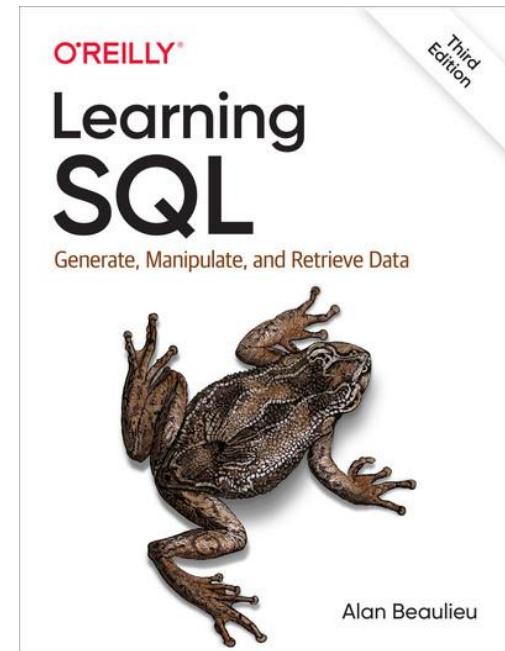
Recursos y meta-recursos de aprendizaje.



Chapter 5. Querying Multiple Tables, Chapter 9. SubQuerys

https://learning.oreilly.com/library/view/learning-sql-3rd/9781492057604/ch05.html#cartesian_product

<https://learning.oreilly.com/library/view/learning-sql-3rd/9781492057604/ch09.html>



Joining Data, JOINS and UNIONs

[Joining Data | Kaggle](#)

[JOINS and UNIONs | Kaggle](#)



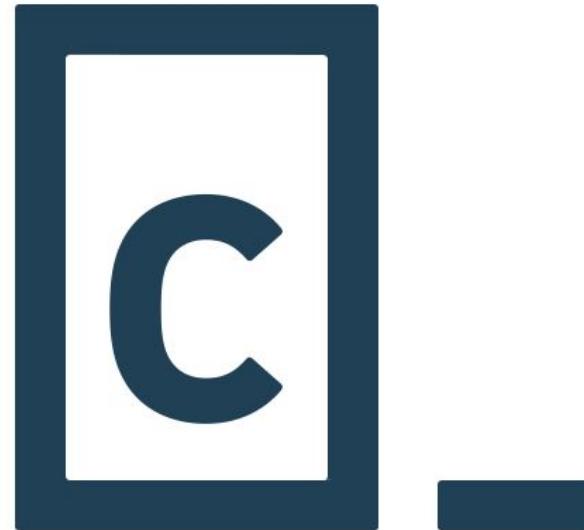
SQL JOIN Keyword

SQL JOIN Keyword



Multiple Tables

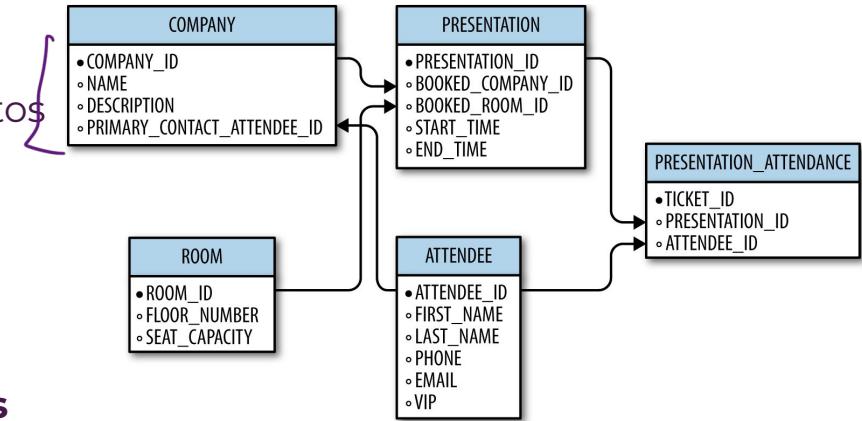
[Review - Learn SQL | Codecademy](#)



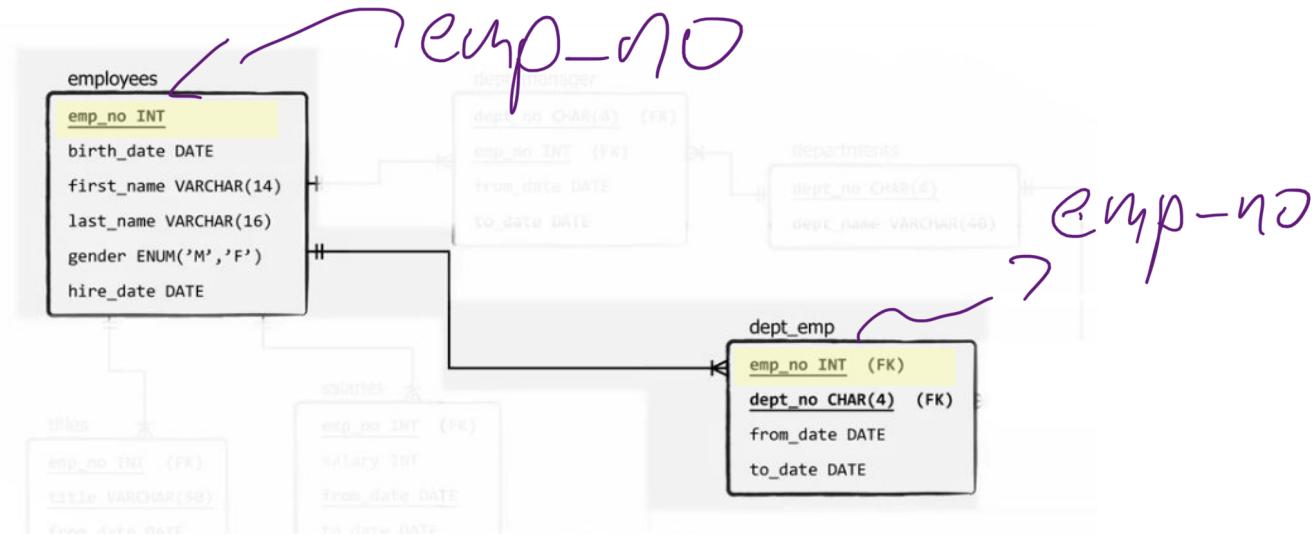
Cuando aprendemos SQL por primera vez, **es común trabajar** con datos **en una sola tabla.**

Aunque en el mundo laboral, las bases de datos **generalmente tienen** datos en **más de una tabla.**

Así que si queremos poder trabajar con esos datos, **aprenderemos cómo combinar varias tablas dentro de una consulta.**



Utilizando SQL se puede **combinar dos o más conjuntos** de datos **a través de JOINS** ocupando una **columna relacionada** entre ellas.





Llaves Primarias y Foráneas

(es a nivel tabla,
no a nivel bd)

Llaves Primarias y Foráneas

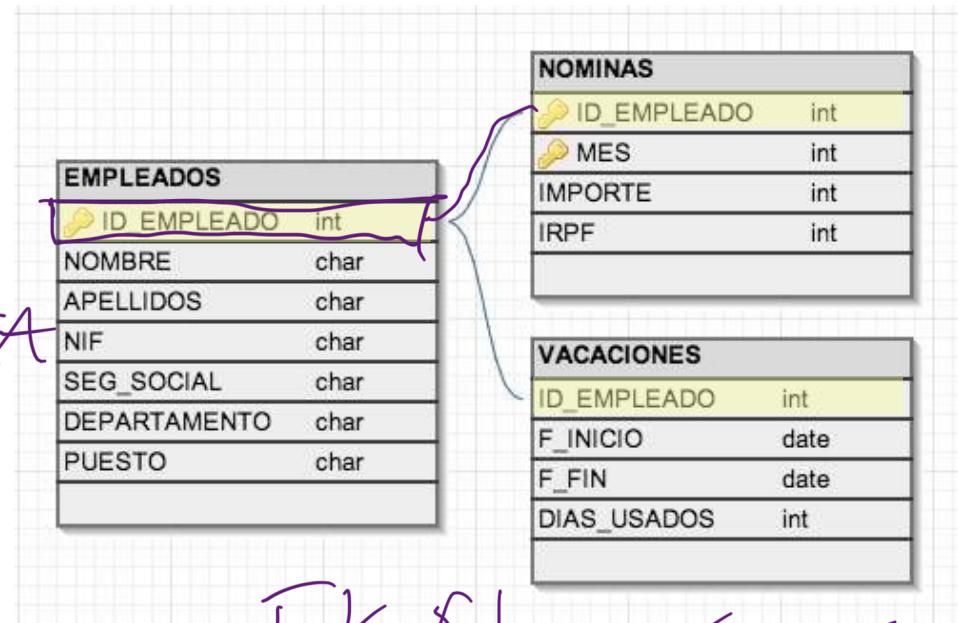
a.k.a. clave principal, clave externa

a.k.a. PK, FK

Normalmente en una BD relacional, se tiene en cada tabla una columna que es la **llave primaria** , donde cada entrada **representa de forma única una sola fila** en esa tabla.

PK
UNICA

Ahora, para establecer una relación entre dos tablas nos apoyamos de **asociar la llave principal de una tabla a la otra** y a esta columna se le denomina **llave foránea**.



Llaves Primarias y Foráneas

ejemplo

EMPLEADOS

id_empleado	nombre	apellido	NIF	seg_social	departamento	puesto
1123	liliana	davalos	M0200096H	72795608040	3	2
3356	marlo	rogel	M0201073G	91806083439	6	5
1189	mia	prior	M0200854B	93884843272	9	8

NOMINAS

id_empleado	mes	importe	IRFP
1123	202112	21000	1995
3356	202112	37000	3515
1189	202112	5600	531
1123	202111	21000	1995
3356	202111	37000	3515

VACACIONES

id_empleado	F_inicio	F_fin	dias_usados
1123	03/01/2018	05/01/2018	3
3356	21/04/2019	26/04/2019	4
3356	26/12/2020	30/12/2020	5

Select *
FROM empleados
Left Join vacaciones ON empleados.id_empleado = vacaciones.id_empleado
i) - cond 1 | F_inicio

Llaves Primarias y Foráneas

Ejemplo 2

1189 | NULL

Select Order (OrderID, OrderItemID, OrderAmount)

Customers	
CustID	Name
1	Customer1
2	Customer2
3	Customer3

Orders		
OrderID	CustID	OrderTotal
1	1	100
2	1	150
3	2	200
4	2	50
5	3	25

OrderItems		
OrderItemID	OrderID	OrderAmount
1	1	20
2	1	80
3	2	120
4	2	30
5	3	100
6	3	25
7	3	75
8	4	50
9	5	25

cursor of



Foreign key

Primary key



JOINS

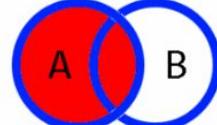
JOIN

Diferentes formas de combinar las tablas:

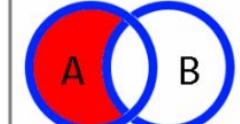
- Inner Join *id en ambas tablas*
- Left Join *todos los q. estén en la tabla izquierda*
- Right Join
- Full Join* *trae a todos*
- Self Join*
- Cartesian Join (Cross)

* Es necesario validar en su aplicación dependiendo del RDBMS

LEFT OUTER JOIN

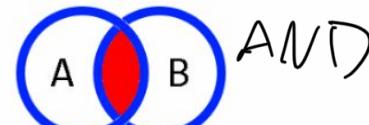


```
SELECT *  
FROM TableA a  
LEFT JOIN TableB b  
ON a.KEY = b.KEY
```



```
SELECT *  
FROM TableA a  
LEFT JOIN TableB b  
ON a.KEY = b.KEY  
WHERE b.KEY IS NULL
```

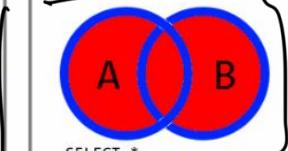
INNER JOIN



AND

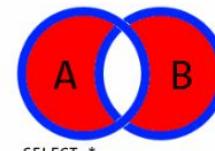
```
SELECT *  
FROM TableA a  
INNER JOIN TableB b  
ON a.KEY = b.KEY
```

FULL OUTER JOIN



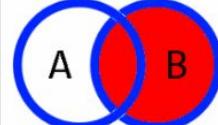
```
SELECT *  
FROM TableA a  
FULL OUTER JOIN TableB b  
ON a.KEY = b.KEY
```

OR

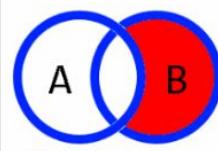


```
SELECT *  
FROM TableA a  
FULL OUTER JOIN TableB b  
ON a.KEY = b.KEY  
WHERE a.KEY IS NULL  
OR b.KEY IS NULL
```

RIGHT OUTER JOIN



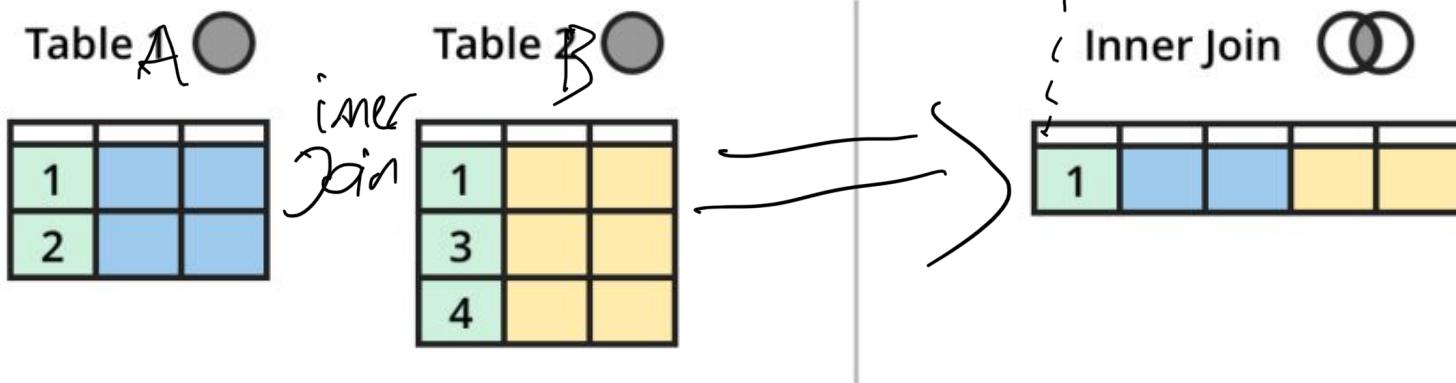
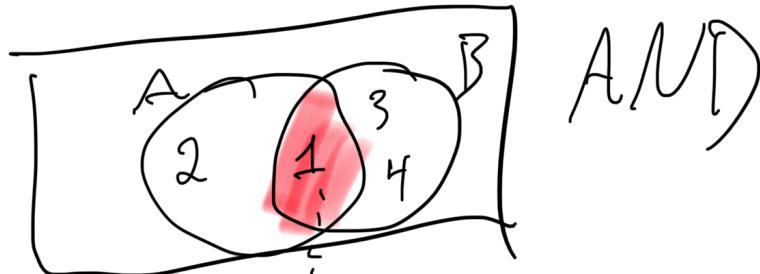
```
SELECT *  
FROM TableA a  
RIGHT JOIN TableB b  
ON a.KEY = b.KEY
```



```
SELECT *  
FROM TableA a  
RIGHT JOIN TableB b  
ON a.KEY = b.KEY  
WHERE a.KEY IS NULL
```

Join : Tabla X Tabla \Rightarrow Tablas

INNER JOIN



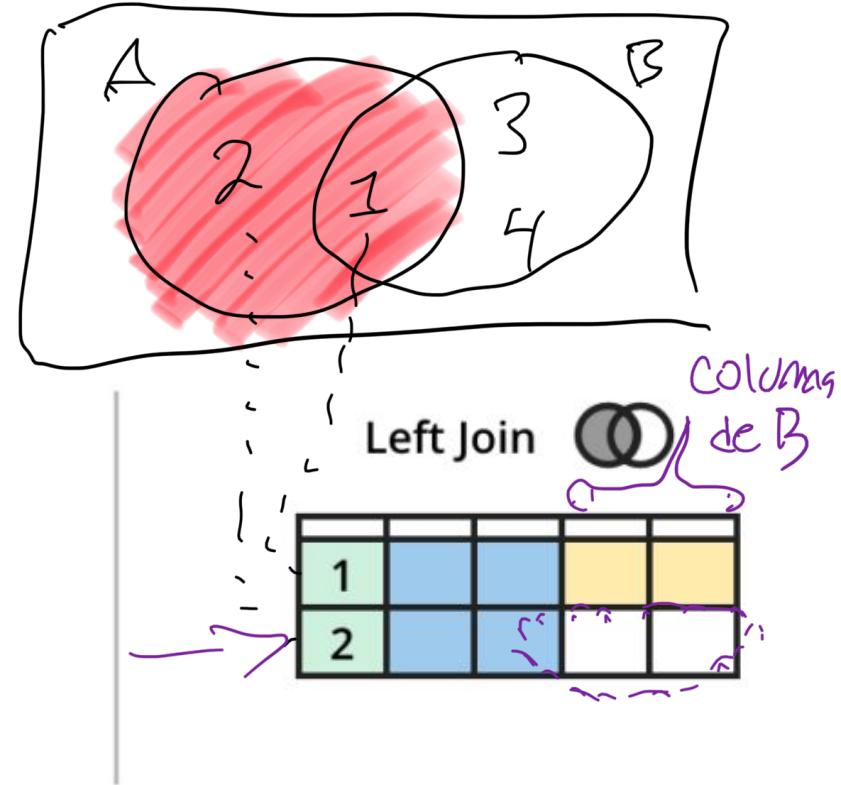
LEFT JOIN

Table A

1		
2		

Table B

1		
3		
4		



OUTER JOIN

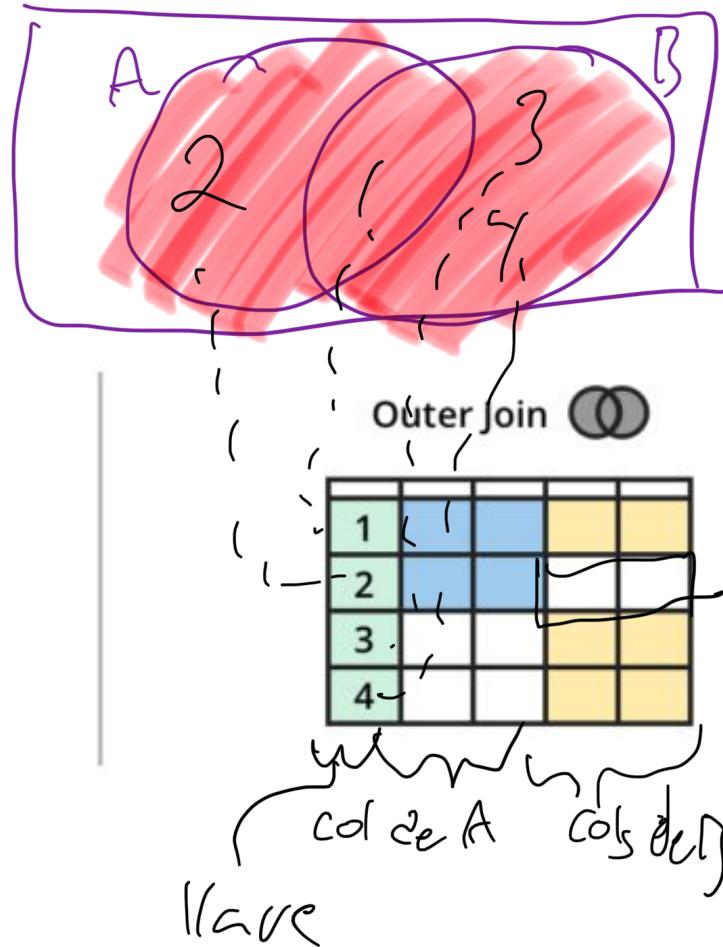
OB

Table A

1		
2		

Table B

1		
3		
4		

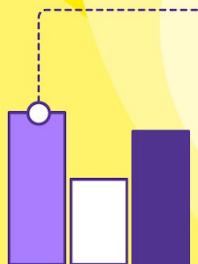
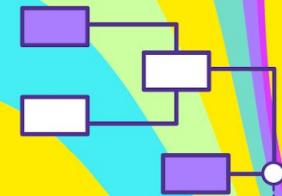


CheatSheet: SQL JOINs Cheat Sheet

Fuente: LearnSQL.com Team



Challenge: Joins y Subconsultas



MASTER en Data Science / AI

DEV.F:

