

Guía de estudio - Consultas en Múltiples Tablas



¡Hola! Te damos la bienvenida a esta nueva guía de estudio.

¿En qué consiste esta guía?

La siguiente guía de estudio tiene como objetivo practicar y ejercitar los contenidos que hemos visto en clase, además de profundizar temas adicionales que complementan lo aprendido. Particularmente trabajaremos entre 2 o más tablas y combinando este contenido con consultas agrupadas, con las siguientes cláusulas:

- Select
- Inner Join
- Left Join o Right Join
- Full Join

¡Vamos con todo!



Tabla de contenidos

Repaso de conceptos básicos	3
Condiciones necesarias para juntar datos entre 2 tablas	3
Tipos de join	3
Actividad 1	5
Setup	5
Revisando las tablas	7
Actividad: Trabajando con 2 tablas, parte I	7
Tips	7
Countries	7
States	7
¡Manos a la obra! - Cruzando 2 tablas	8
Actividad: Trabajando con 3 tablas, parte II	9
Tips	9
¡Manos a la obra! - Cruzando 3 tablas	9



¡Comencemos!

Repaso de conceptos básicos

Con frecuencia, los datos que necesitaremos en nuestro trabajo estarán repartidos en múltiples tablas. Por lo cual, parte importante de trabajar con SQL es saber cruzar estos datos y obtener la información que necesitemos, sin importar en cuántas tablas estén repartidos.

Condiciones necesarias para juntar datos entre 2 tablas

A menos que queramos obtener una combinación de todos los registros de la tabla 1 con cada uno de los registros de la tabla 2, para unir dos tablas necesitaremos un elemento en común. Por ejemplo, si queremos unir los registros de posteos con los registros de comentarios, necesitamos saber a qué posteo pertenece un comentario. Lo mismo si queremos saber qué tweets ha publicado cada usuario; necesitamos saber a qué usuario pertenece un tweet.

En otras palabras, tiene que haber una (o más) columnas que nos permitan juntar la información de la tabla 1 con la tabla 2.

Tipos de join

Hay 4 formas de cruzar datos en múltiples tablas que responden a distintas necesidades que podríamos tener.

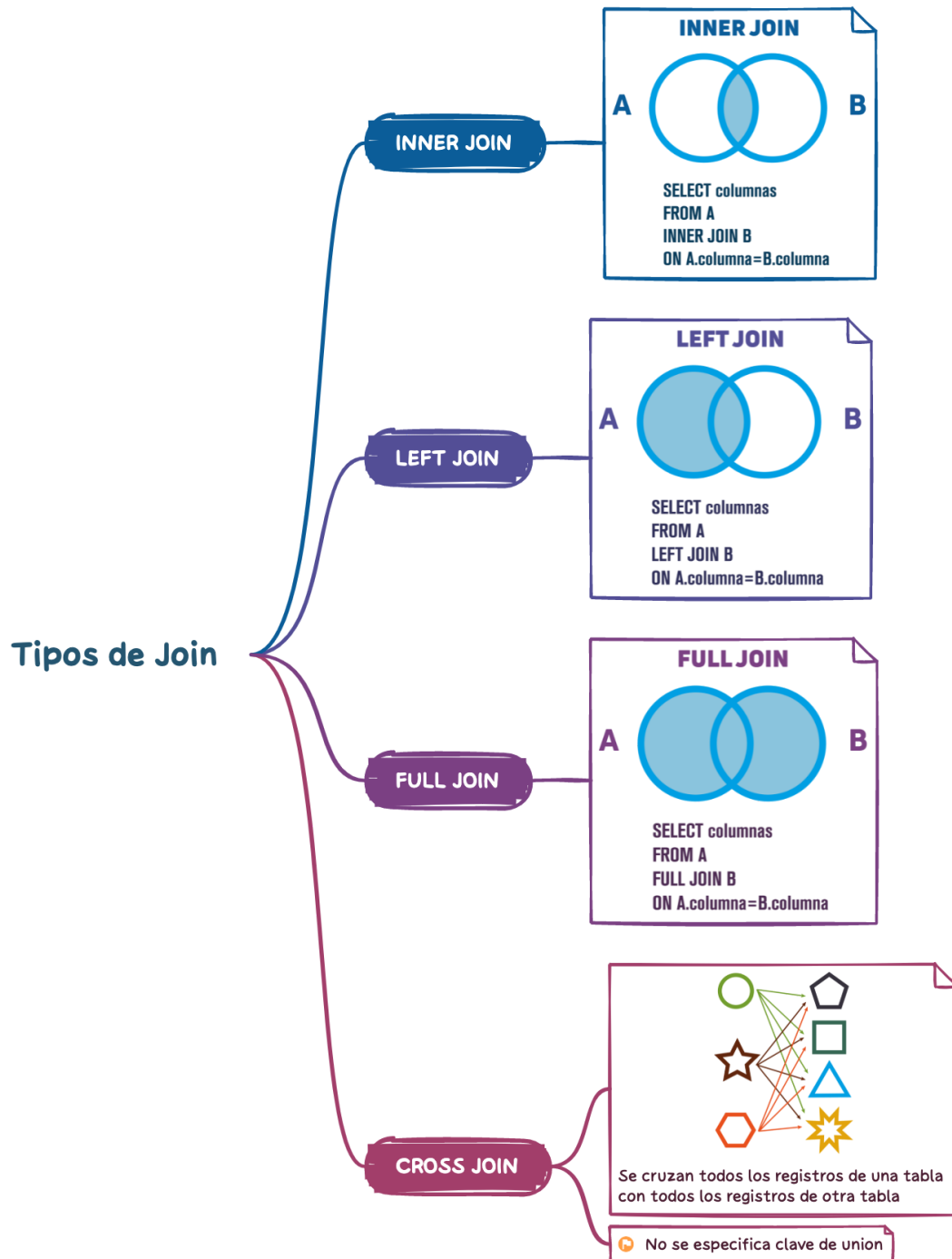


Imagen 1. Tipos de joins
Fuente: Desafío Latam.



Actividad 1

A continuación, trabajaremos con una base de datos compuesta por 3 tablas, que muestra información de todos los países, estados y ciudades del mundo. Para esto será necesario cargar los datos previamente llevando a cabo los siguientes pasos.

Setup

1. Descargar los archivos countries.csv, states.csv y cities.csv de la plataforma.
2. Crear la base de datos países.
3. Crear la tabla countries.
 - a. Utilizaremos los mismos nombres y columnas del archivo CSV

```
CREATE TABLE countries(  
  id INTEGER,  
  name VARCHAR,  
  iso3 CHAR(3),  
  capital VARCHAR,  
  currency CHAR(3),  
  currency_name VARCHAR,  
  tid CHAR(3),  
  region VARCHAR,  
  subregion VARCHAR,  
  latitude DECIMAL(10,8),  
  longitude DECIMAL(11,8)  
);
```

4. Copiar los datos \copy countries from '/direccion-al-archivo/countries.csv' DELIMITER ',' CSV HEADER

Ejemplo de código de inserción:

```
\copy countries from '/mnt/c/Users/Public/EJercicio 1/countries.csv'  
delimiter ',' csv header;
```

Al insertar los datos exitosamente deberías obtener: COPY 250 haciendo alusión a que se insertaron correctamente. Si tienes problemas copiando los datos puedes intentar importándolos con pgadmin4.

5. Crear la tabla `states`, que contiene todos los estados de cada país.

```
CREATE TABLE states(  
  id INTEGER,  
  name VARCHAR,  
  country_id INTEGER,  
  state_code CHAR(5),  
  state_type VARCHAR,  
  latitude DECIMAL(10,8),  
  longitude DECIMAL(11,8)  
);
```

6. Luego, copia los datos con `\copy states from '/direccion-al-archivo/states.csv' DELIMITER ';' CSV HEADER '`. Deberíamos obtener como respuesta COPY 4979. Al igual que en el caso anterior, si tienes problemas con la carga puedes intentarlo con pgAdmin.

Ejemplo de código de inserción:

```
\copy states from '/mnt/c/Users/Public/EJercicio 1/states.csv' delimiter  
';' csv header;
```

7. Crear la tabla `cities`, que contiene todas las ciudades de cada estado.

```
CREATE TABLE cities(  
  id INTEGER,  
  name VARCHAR,  
  state_id INTEGER,  
  latitude DECIMAL(10,8),  
  longitude DECIMAL(11,8)  
);
```

8. Carga los datos con `\copy cities from '/direccion-al-archivo/cities.csv' DELIMITER ';' CSV HEADER '`. Deberíamos obtener como respuesta COPY 148250 y al igual que en el caso anterior, si tienes problemas con la carga puedes intentarlo con pgAdmin.

Ejemplo de código de inserción:

```
\copy cities from '/mnt/c/Users/Public/EJercicio 1/cities.csv' delimiter  
';' csv header;
```



La información oficial de este dataset proviene de [Data World](https://dataworld.org/). Los archivos han sido modificados respecto al dataset original para intencionar ciertos

aprendizajes. El Link al dataset oficial es <https://data.world/dr5hn/country-state-city/>

Cómo son muchos datos, te recomendamos evitar seleccionarlos todos simultáneamente. Para evitarlo, agrega `limits` y `order by` a las consultas acorde a lo pedido.

Revisando las tablas

En este problema hay 3 tablas. En la primera parte de la guía trabajarás únicamente con dos tablas. En la 2º parte trabajarás cruzando los datos a lo largo de las 3 tablas.

Actividad: Trabajando con 2 tablas, parte I



Tips

Se tiene una tabla de países y una de estados, y necesitamos ubicar la información de cada estado junto a su país.

Es preciso recordar que, para juntar información de ambas tablas, necesitamos algún elemento en común. En este caso, si abrimos los archivos csv con un software de hoja de cálculo como excel, google spreadsheet u otro, o incluso si lo abrimos con el editor de código, encontraremos en las cabeceras de la tabla lo siguiente:

Countries

id	name	iso3	capital	currency	currency_name	tld	region	subregion	latitude	longitude
1	Afghanistan	AFG	Kabul	AFN	Afghan afghani	.af	Asia	Southern Asia	33.00000000	65.00000000
2	Aland Islands	ALA	Mariehamn	EUR	Euro	.ax	Europe	Northern Europe	60.11666700	19.90000000
3	Albania	ALB	Tirana	ALL	Albanian lek	.al	Europe	Southern Europe	41.00000000	20.00000000

States

id	name	country_id	state_code	type	latitude	longitude
3901	Badakhshan	1	BDS		36.73477250	70.81199530
3871	Badghis	1	BDG		35.16713390	63.76953840
3875	Baghlan	1	BGL		36.17890260	68.74530640

Veremos que los `country_id` de un estado coinciden con los id del país; a partir de esto podemos cruzar la información sobre las columnas. Para lograrlo podemos utilizar:

```
SELECT * FROM tabla1  
INNER JOIN tabla2 ON tabla1.id = tabla2.country_id;
```

O

```
SELECT * FROM tabla1, tabla2  
WHERE tabla1.id = tabla2.country_id;
```

De preferencia utilizaremos `inner join`.

Si queremos ordenar y limitar los resultados, solo tenemos que agregar estas cláusulas al final de la instrucción:

```
SELECT *  
FROM tabla1  
INNER JOIN tabla2 on tabla1.id = tabla2.country_id  
ORDER BY tabla1.latitude  
LIMIT(10)
```

Si necesitamos ocupar una columna de una tabla podemos utilizar `tabla1.columna` o `tabla2.columna`, o simplemente `columna` en caso de que no haya posibilidad de confusión.



¡Manos a la obra! - Cruzando 2 tablas

Realiza las siguientes actividades

1. Selecciona el cruce de ambas tablas ordenadas alfabéticamente por el nombre del país y limitando a 10 resultados.
2. Selecciona los primeros 5 estados de Austria.



Pista: Tienes que cruzar los datos de países y estados, y filtrar por el nombre del país.

3. Cuenta la cantidad de estados que tiene cada país, limitando los resultados a 10.
4. ¿Cuáles son los 3 países con más estados? Muestra el país junto con la cantidad de estados.

5. ¿Cuál es el país con el estado que está más al norte? Muestra el país y el estado con la latitud correspondiente.
 - (La latitud debe ser del estado, no la del país).
6. ¿Tenemos la misma cantidad de resultados con inner join que con left join?
 - Cuenta ambos casos y compara.
7. ¿Cuáles son los 5 estados con más ciudades? Muéstranos.

Actividad: Trabajando con 3 tablas, parte II



Tips

Se pueden cruzar datos en múltiples tablas utilizando múltiples joins. Por ejemplo:

```
SELECT *  
FROM tabla1  
INNER JOIN tabla2 ON tabla1.id = tabla2.country_id  
INNER JOIN tabla3 ON tabla2.id = tabla3.state_id  
WHERE (si aplica)  
ORDER BY (si aplica)  
LIMIT (si aplica)
```

Similar al caso anterior de cruzar dos tablas, el resultado será una única tabla con las columnas especificadas de las 3 tablas cruzadas.

Es posible cruzar múltiples tablas de esta forma, e incluso, combinar inner joins con left joins u outer join si la situación lo requiere.

Tenemos que ser cuidadosos y especificar correctamente las claves de unión de ambas tablas.



¡Manos a la obra! - Cruzando 3 tablas

- Selecciona todos los países con sus estados y ciudades.
 - Limita los resultados a 50.
- Muestra cada país junto con la cantidad de ciudades que tiene.
- Muestra cada país junto con la ciudad más al norte (mayor latitud).

- Considera este ejercicio en particular. ¿Existe alguna diferencia entre ocupar `inner join` en ambos cruces, ocupar `left join` en ambos o uno de cada uno de los tipos de joins? Cuenta los resultados y responde.



¡Continúa aprendiendo y practicando!