

# Relaciones y operaciones transaccionales

Sesión Conceptual 1





# Inicio

{desafío}  
latam\_



15 minutos

*/\**Realizar consultas usando el comando JOIN para el consumo de datos entre tablas relacionadas.*\*/*

*/\**Realizar subconsultas para la anidación de datos entre tablas.*\*/*

## Objetivo



# Desarrollo

{desafío}  
latam\_

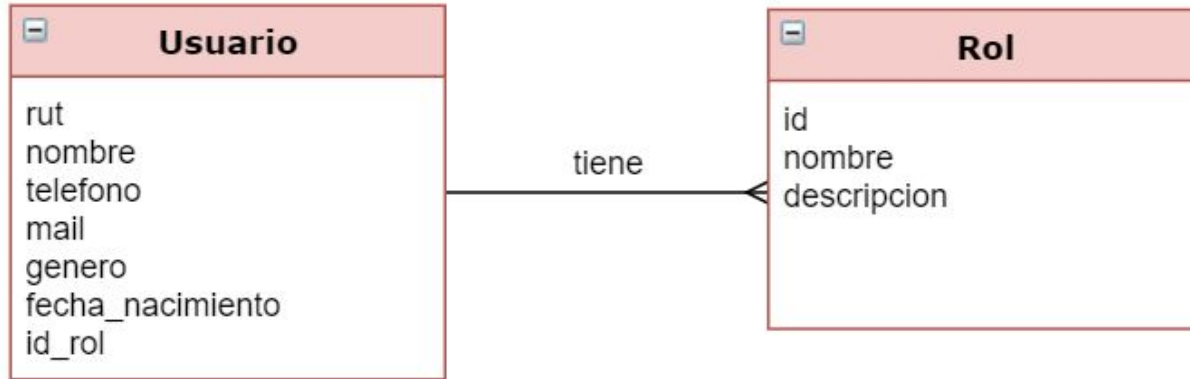


150 minutos

# Unión de tablas

- Se pueden relacionar dos tablas por medio de las claves primarias y foráneas.
- Veamos el siguiente caso:
  - “Un sistema con diferentes secciones restringidas por niveles de usuarios, desea almacenar los datos de su personal e identificarlos según su rol en el proyecto, teniendo los roles de administrador, marketing, editor y subscriptor, cada uno con diferentes funciones.”

# Unión de tablas



Clave primaria: Tabla "Rol" por medio del campo "id"

Clave foránea: Tabla "Usuario" por medio del campo "is\_rol"

# Unión de tablas

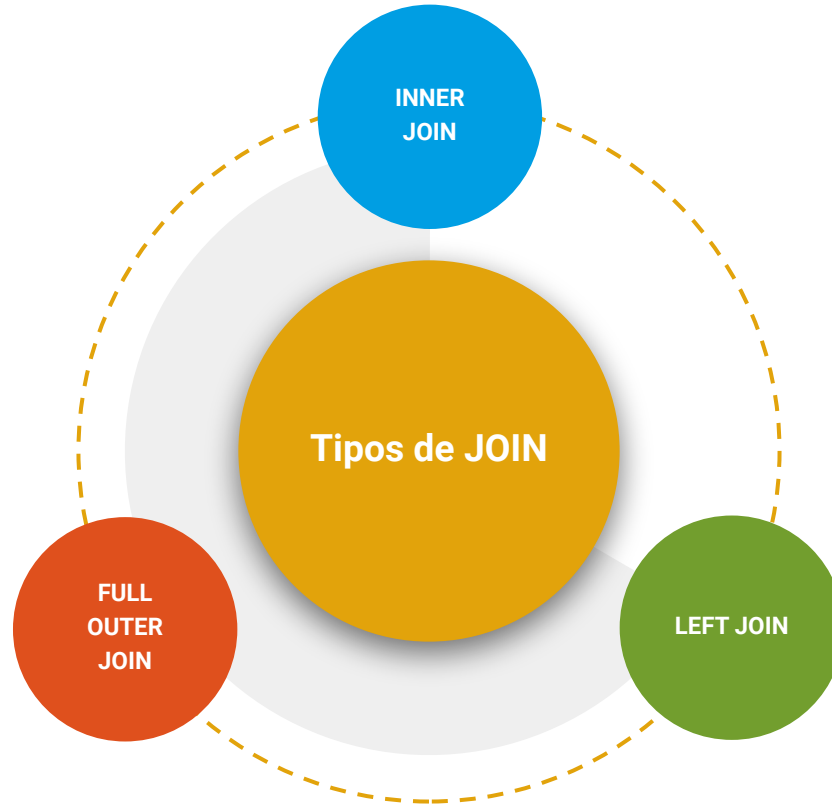
rut	nombre	teléfono	mail	género	fecha_nacimiento	id_rol
1-9	Juan Soto	99999999	juan.soto@gmail.com	M	24-06-2000	1
2-7	Jorge Perez	88888888	jorge.perez@hotmail.com	M	12-03-1998	4
3-5	Sara Morales	7777777	sara.morales@gmail.com	F	11-02-1990	

Estructura y registros de la tabla Usuario.

id	nombre	descripción
1	administrador	control total
2	marketing	editar y crear anuncios
3	editor	editar artículos
4	subscriber	leer artículos

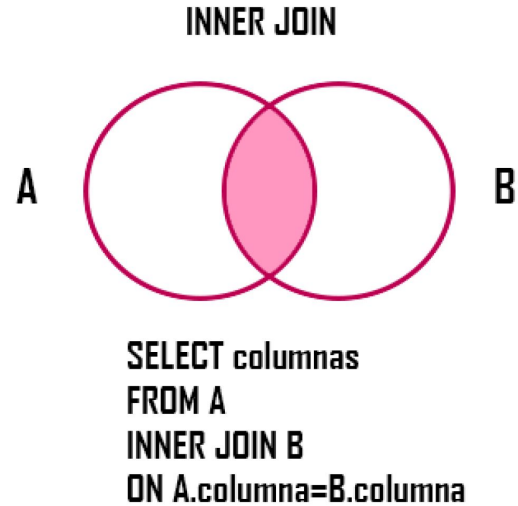
Estructura y registros de la tabla Rol.





# INNER JOIN

Este tipo de relación nos permite conocer aquellos elementos que tienen en conjunto una o más tablas.



## Resolvamos en conjunto

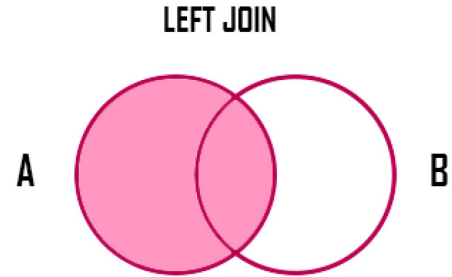
1. En nuestro ejemplo, si hiciéramos un JOIN entre las tablas usuario y rol obtendremos sólo dos registros. ¿Te imaginas por qué? comprobémoslo.
2. Si quisiéramos saber cuáles son los pokemones que ya poseemos y la información detallada, se podría consultar primero la tabla mis\_pokemones y luego consultar el detalle en la tabla pokemones.

Pokemones	
PK	pokedex
	nombre
	tipo1
	tipo2

mis_pokemones	
FK	pokedex
	fecha_captura
	lugar
	huevo
	peso
	estatura

# LEFT JOIN

Nos permite obtener toda la información de una tabla y el conjunto de la relación con otra. La diferencia en el código radica en agregarle la palabra “LEFT” antes de “JOIN”



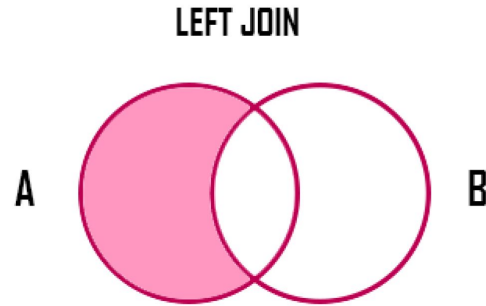
```
SELECT columnas  
FROM A  
LEFT JOIN B  
ON A.columna=B.columna
```

1. Aplicar un LEFT JOIN a la tabla Usuario y Rol.
2. ¿Qué ocurre si quiero obtener toda la información disponible de los pokemones, sin importar si lo he capturado o no?.

**Resolvamos  
en conjunto**

# LEFT JOIN Donde una columna en la tabla "B" es null

Este tipo de unión nos permite obtener la información de los registros que no se encuentran en la relación.



```
SELECT columnas  
FROM A  
LEFT JOIN B  
ON A.columna=B.columna  
WHERE B.columna IS NULL
```

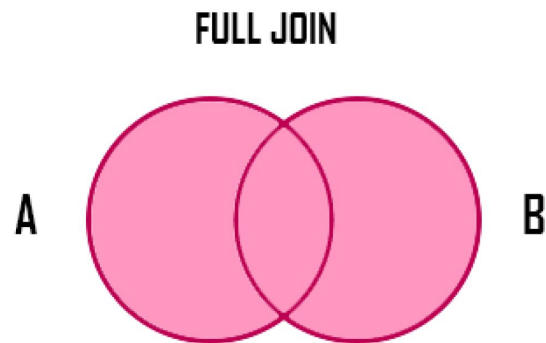
1. Nos gustaría saber aquellos roles que no han sido asignados a ningún usuario.
2. Nos gustaría saber ¿Qué pokemones aún no tenemos?

A vertical line separates the white left side from the orange right side. It features a series of white symbols: a closing curly brace '}', an '@' symbol, another closing curly brace '}', and a closing parenthesis ')'.

**Resolvamos  
en conjunto**

# FULL (OUTER) JOIN

Este tipo de unión nos permite obtener todos los registros disponibles en las tablas que estamos relacionando, sin filtrar la información.



```
SELECT columnas  
FROM A  
FULL OUTER JOIN B  
ON A.columna=B.columna
```



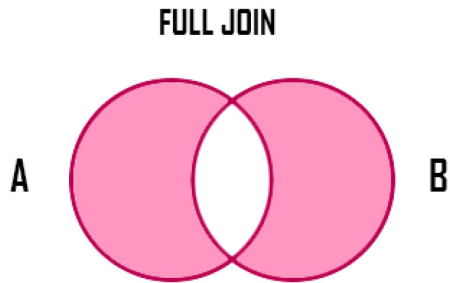
1. Aplicar FULL (OUTER) JOIN al ejemplo de Usuarios y Roles.
2. Para el caso de los pokemones, queremos saber toda la información disponible, sin importar si poseemos o no el pokemon.

**Resolvamos  
en conjunto**

## FULL OUTER JOIN

Donde una columna en la tabla "A" o "B" es null

FULL OUTER JOIN nos permite conocer aquellos registros que no tienen en común las tablas que estamos relacionando.



```
SELECT columnas  
FROM A  
FULL OUTER JOIN B  
ON A.columna=B.columna  
WHERE A.columna IS NULL  
OR B.columna IS NULL
```

1. Aplicar FULL OUTER JOIN al ejemplo de Usuarios y Roles.
2. Aplicar FULL OUTER JOIN al ejemplo de los Pokemones.

**Resolvamos  
en conjunto**

- Una subquery (o consulta interna) es una query implementada dentro de otra query principal de SQL, la cual debe enmarcarse dentro de la cláusula WHERE.
- Funcionan como una condición para restringir los datos.
- Las subconsultas son consultas o tablas temporales que sólo estarán para poder complementar el resultado de nuestra consulta principal.

## Subquery

# Operador WHERE

La sintaxis está dada de la siguiente manera:

```
SELECT columna1,columna2,...  
FROM nombre_tabla  
WHERE columna1 IN  
    (SELECT columna1  
     FROM nombre_tabla2  
     WHERE condicion);
```

Obtener como resultado los nombres de los pokemones que fueron obtenidos por huevos.

**Resolvamos  
en conjunto**

# Operador FROM

La subquery obtiene como resultado una subtabla temporal, sobre la cual se hará una nueva consulta. La sintaxis está dada de la siguiente manera.

```
SELECT x.columna1, xcolumna2, ...  
FROM (  
    SELECT columna1, columna2, ...  
    FROM nombre_tabla2  
    WHERE condicion  
    ) as x  
INNER JOIN nombre_tabla1 as y on x.columna1 = y.columna1;
```

Obtener los nombres de los pokemones capturados que tienen un peso mayor a 200 kilos.

**Resolvamos  
en conjunto**



# Reglas de las subqueries

Las consultas internas deben estar encapsuladas entre paréntesis.

Una subquery puede tener sólo una columna especificada en SELECT, con la excepción de múltiples columnas definidas en la consulta principal.

El comando ORDER BY no se puede utilizar en una consulta interna. La excepción es que esta instrucción si puede ser incluida en la consulta principal.

Para obtener un resultado similar a ORDER BY dentro de una consulta interna, se puede implementar el comando GROUP BY.

Aquellas consultas internas que retornen más de una fila sólo pueden ser utilizadas con operadores de múltiples valores como IN.



# Cierre

{desafío}  
latam\_



15 minutos

**¿Existe algún concepto que  
no hayas comprendido?**

**Volvamos a revisar los conceptos que más te  
hayan costado antes de seguir adelante**

**Reflexionemos**





*Academia de  
talentos digitales*

[www.desafiolatam.com](http://www.desafiolatam.com)



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam