



Relaciones y operaciones transaccionales _

Sesión Conceptual 01



- Objetivo de la sesión

- Conceptualización
- Ejercicios
- Quiz

- Cierre



Inicio



20 minutos

{desafío}
latam_

- Realizar consultas usando el comando JOIN para el consumo de datos entre tablas relacionadas.
- Realizar subconsultas para la anidación de datos entre tablas.



Objetivos



Desarrollo



110 minutos

/* Operaciones de unión entre Tablas */

Unión de tablas

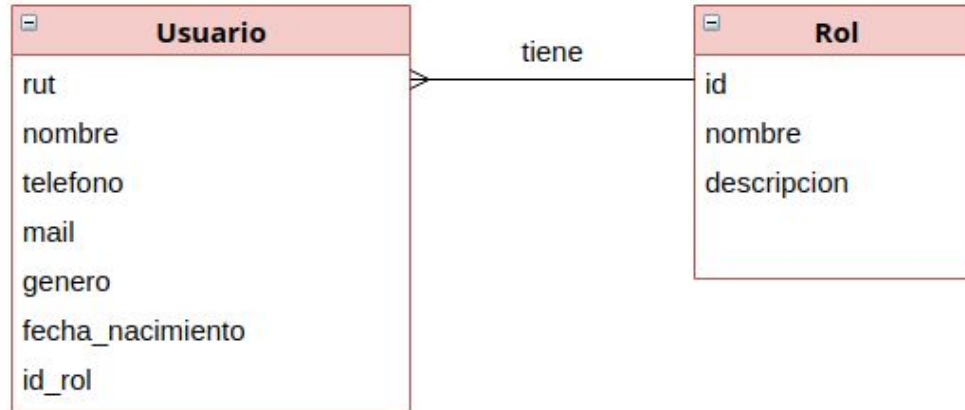
- Se pueden relacionar dos tablas por medio de las claves primarias y foráneas.
- Veamos el siguiente caso:
 - “Un sistema con diferentes secciones restringidas por niveles de usuarios, desea almacenar los datos de su personal e identificarlos según su rol en el proyecto, teniendo los roles de administrador,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



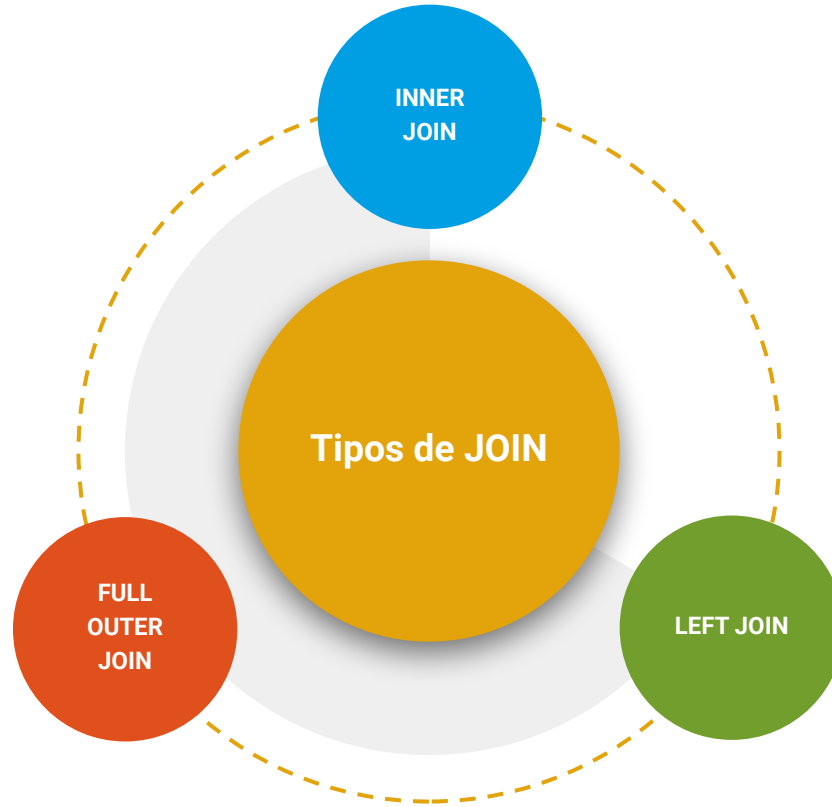
Unión de tablas

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	

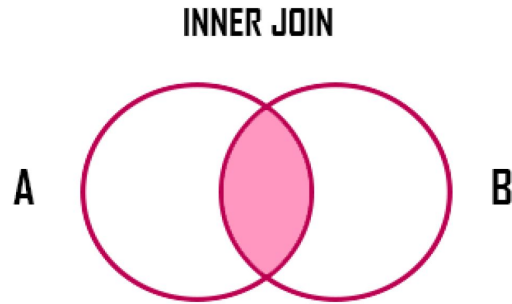
Estructura y registros de la tabla Usuario.

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

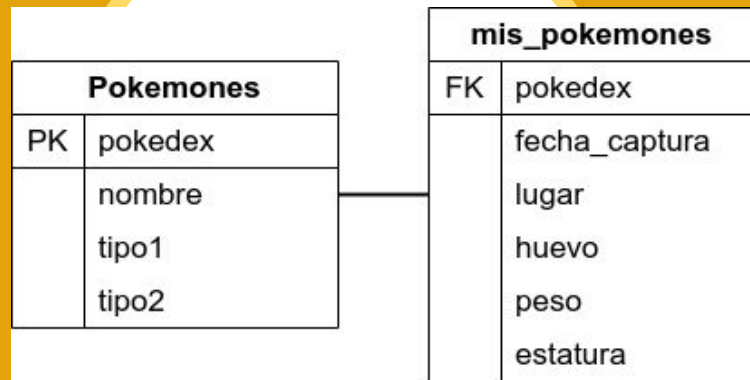


```
SELECT columnas  
FROM A  
INNER JOIN B  
ON A.columna=B.columna
```

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

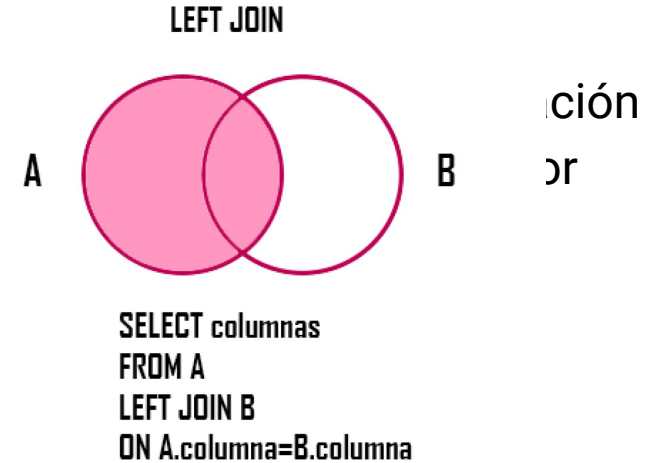
Ejercicio guiado

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.



LEFT JOIN

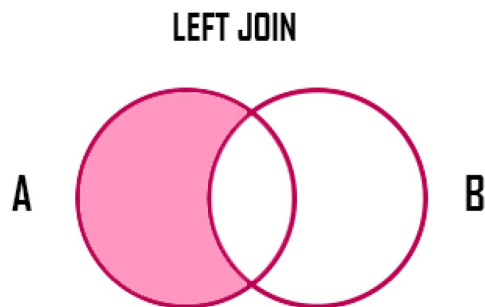
Nos permite obtener toda la información de un con otra. La diferencia en el código radica en a “LEFT” antes de “JOIN”



Para un desarrollo más completo de esta y las siguientes temáticas a tratar, se sugiere revisar los ejercicios contenidos en los **Materiales complementarios de la U2.**

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
```

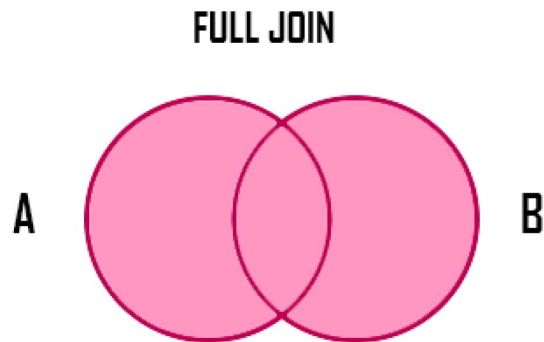
Ejercicio guiado

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



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
```

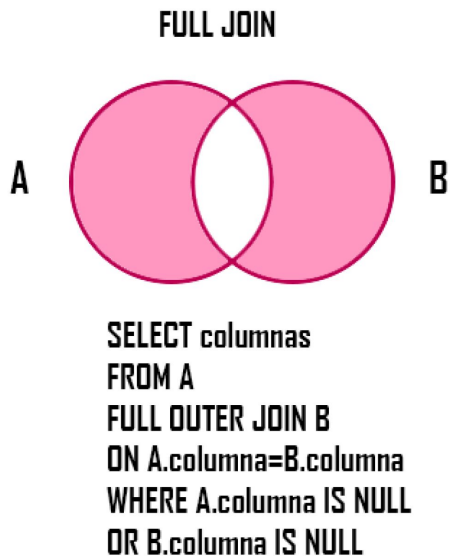

Ejercicio guiado

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.



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.



Ejercicio guiado

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



Para un desarrollo detallado de este ejercicio, se recomienda revisar la **Lectura 01**.

Subquery

- 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 temporales que sólo estarán para poder complementar el resultado de nuestra consulta principal.

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);
```

Ejercicio guiado

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



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;
```

Ejercicio guiado

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



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.



Quiz



20 minutos

{desafío}
latam_



Cierre



30 minutos

Preguntas de cierre




¿Cómo se puede relacionar dos tablas?



¿Para qué nos sirve INNER JOIN?



¿Para qué nos sirve LEFT JOIN?



¿Qué nos permite hacer FULL JOIN?



*Academia de
talentos digitales*

www.desafiolatam.com



/DesafioLatam



/DesafioLatam



/DesafioLatam



/DesafioLatam