



30  
años

Licenciada por  
SUNEDU  
para que puedas  
salir adelante

## SESIÓN 12:

Gestionar Sistemas de Información para satisfacer necesidades organizacionales de forma innovadora, respondiendo a estándares de calidad



# Resultados de Aprendizaje

- ✓ Diseña una Base de Datos Relacional para un caso en una empresa y usa el lenguaje SQL como herramienta de manipulación de datos.
- ✓ Aplica los Procedimientos Almacenados, en SQL Server.
- ✓ Crea proyectos en Visual Basic .Net para aplicar el uso de Procedimientos Almacenados.

## TEMARIO:

### TEMA 7: USO DE JOINS Y SUBCONSULTAS.

#### Contenido

- 🔍 Join . Definición ,Tipos
- 🔍 Consultas SQL Server





## JOIN

El propósito del **JOIN** es unir información de diferentes tablas, para no tener que repetir datos en diferentes tablas. Existen 3 tipos de JOINS: **JOIN interno (inner)**, **JOIN externo (outer)** y **JOIN cruzado (cross)**.



# SQL

## UNIR TABLAS





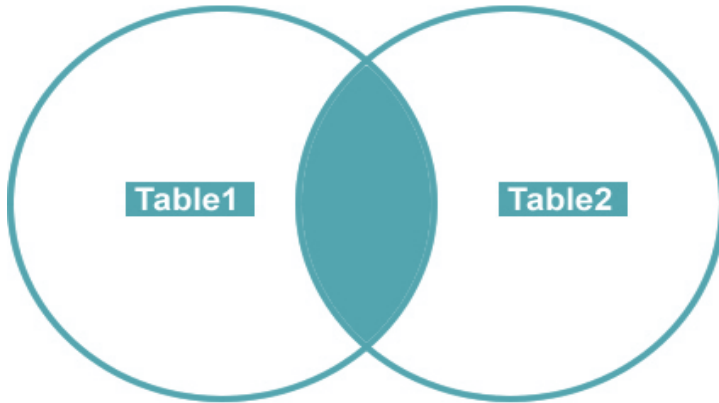
## INNER JOIN

Pregrado

Ingeniería de  
Sistemas

Es la sentencia **JOIN** por defecto, y consiste en combinar cada fila de una tabla con cada fila de la otra tabla, seleccionando aquellas filas que cumplan una determinada condición.

INNER JOIN



```
SELECT *  
FROM Table1 t1  
INNER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

### SINTAXIS:

**SELECT \***

**FROM** tabla1 **INNER JOIN** tabla2

**ON** tabla1.columna1 = tabla2.columna1





# LEFT JOIN

Pregrado

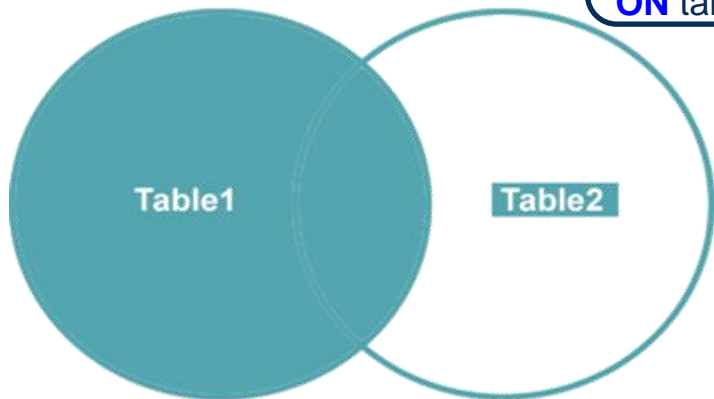
Ingeniería de  
Sistemas

Combina los valores de la primera tabla con los valores de la segunda tabla. **Siempre devolverá las filas de la primera tabla**, incluso aunque no cumplan la condición.

## SINTAXIS:

```
SELECT *  
FROM tabla1 LEFTJOIN tabla2  
ON tabla1.columna1 = tabla2.columna1
```

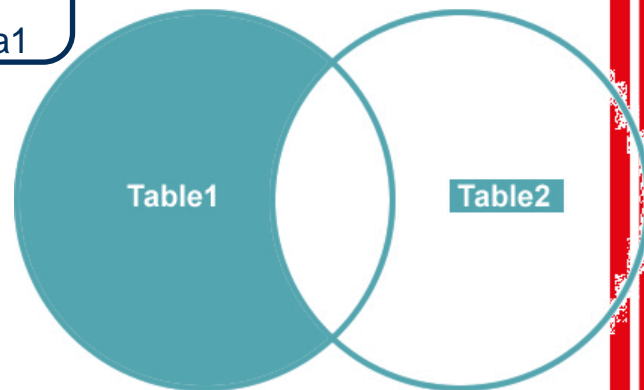
### LEFT OUTER JOIN



```
SELECT *  
FROM Table1 t1  
LEFT OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```



### LEFT OUTER JOIN - WHERE NULL



```
SELECT *  
FROM Table1 t1  
LEFT OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1  
WHERE t2.Col1 IS NULL
```



# RIGHT JOIN

Pregrado

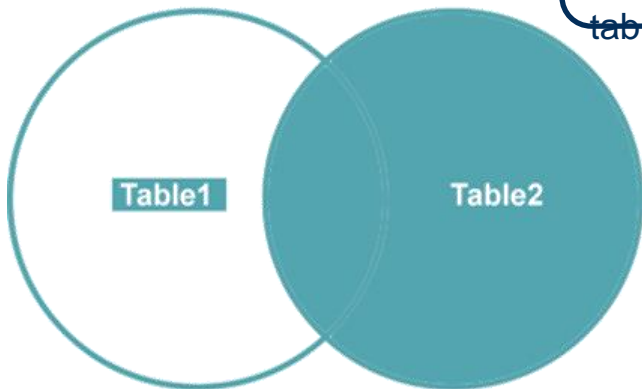
Ingeniería de  
Sistemas

Combina los valores de la primera tabla con los valores de la segunda tabla. **Siempre devolverá las filas de la segunda tabla**, incluso aunque no cumplan la condición.

**SINTAXIS:**

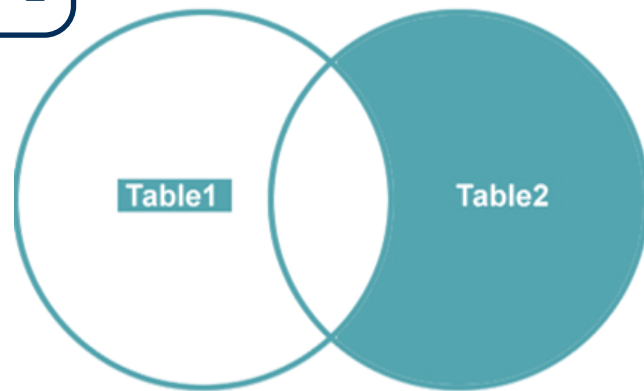
```
SELECT *  
FROM tabla1 RIGHT JOIN tabla2  
ON      tabla1.columna1 =  
        tabla2.columna1
```

RIGHT OUTER JOIN



```
SELECT *  
FROM Table1 t1  
RIGHT OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

RIGHT OUTER JOIN - WHERE NULL



```
SELECT *  
FROM Table1 t1  
RIGHT OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1  
WHERE t1.Col1 IS NULL
```





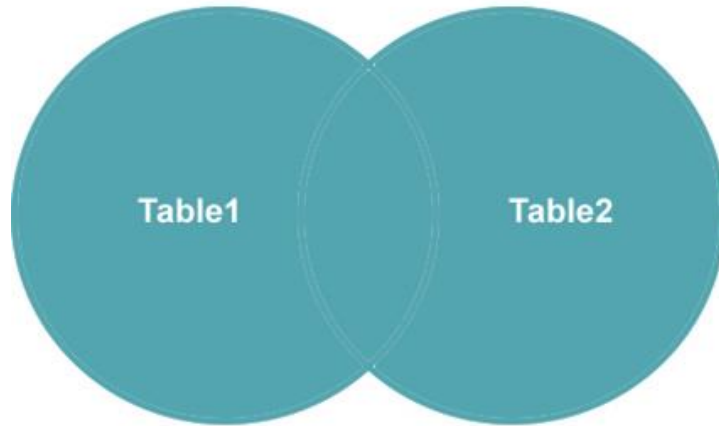
## FULL JOIN

Pregrado

Ingeniería de  
Sistemas

Combina los valores de la primera tabla con los valores de la segunda tabla. Siempre devolverá las filas de las dos tablas, aunque no cumplan la condición.

FULL OUTER JOIN



```
SELECT *  
FROM Table1 t1  
FULL OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

**SINTAXIS:**

**SELECT \***

**FROM** tabla1 **FULL JOIN** tabla2

**ON** tabla1.columna1 = tabla2.columna1



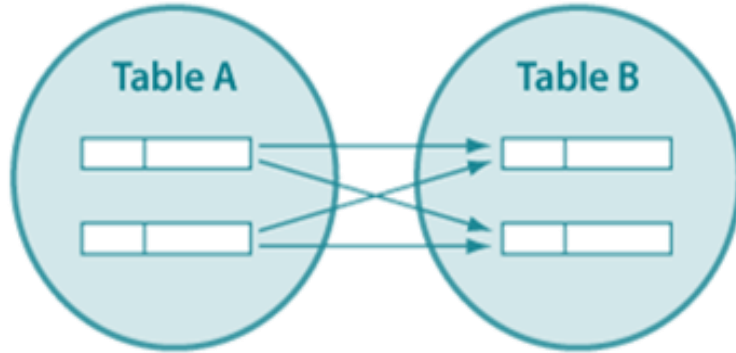


# CROSS JOIN

Pregrado

Ingeniería de  
Sistemas

Retorna todos los registros de todas las tablas implicadas en la unión, devuelve el producto cartesiano.



## SINTAXIS:

**SELECT \***

**FROM** tabla1 **CROSS JOIN** tabla2

**ON** tabla1.columna1 = tabla2.columna1







Es una sentencia **SELECT** que aparece dentro de otra sentencia **SELECT** que llamaremos consulta principal. Se puede encontrar en la lista de selección, en la cláusula **WHERE** o en la cláusula **HAVING** de la consulta principal.



Tiene la misma sintaxis que una sentencia **SELECT** normal exceptuando que:

- ✓ Aparece encerrada **entre paréntesis**
- ✓ **No** puede contener la cláusula **ORDER BY**
- ✓ **No** puede ser la **UNION** de varias sentencias **SELECT**



Tiene algunas restricciones en cuanto a número de columnas según el lugar donde aparece en la consulta principal. Cuando se ejecuta una consulta que contiene una subconsulta, **la subconsulta se ejecuta por cada fila de la consulta principal.**



Se aconseja **no utilizar campos calculados en las subconsultas**, ralentizan la consulta. Las consultas que utilizan subconsultas suelen ser más fáciles de interpretar por el usuario.



# SUBCONSULTA

Pregrado

Ingeniería de  
Sistemas

A menudo, es necesario, dentro del **cuerpo de una subconsulta**, hacer referencia al valor de una columna en la fila actual de la consulta principal, ese nombre de columna se denomina **referencia externa**.



Una **referencia externa** es un nombre de columna que estando en la **subconsulta**, no se refiere a ninguna columna de las tablas designadas en la **FROM** de la **subconsulta** sino a una columna de las tablas designadas en la **FROM** de la consulta principal. Como la **subconsulta** se ejecuta por cada fila de la consulta principal, el valor de la **referencia externa** irá cambiando.

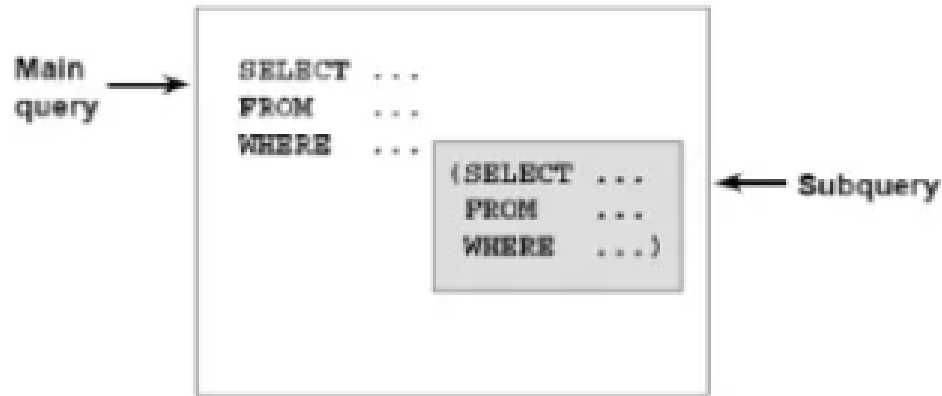


# Sub Consulta

## *¿Qué es una sub consulta?*

Una sub consulta es una sentencia **SELECT** que es incrustada en una cláusula de otra sentencia SQL, llamada sentencia padre.

La sub consulta (consulta interna) obtiene un valor que es usado por la sentencia padre. Usar una sub consulta anidada es equivalente a ejecutar dos consultas secuenciales y utilizar el resultado de la consulta interna como valor de búsqueda en la consulta externa (consulta principal).





# Sub Consultas

```
SELECT nombre_empleado, paga  
FROM empleados  
WHERE paga < 2500
```

(**SELECT** paga **FROM** empleados  
**WHERE** nombre\_empleado='Martina')

```
AND paga < 1870
```

(**SELECT** paga **FROM** empleados  
**WHERE** nombre\_empleado='Luis')



## EJEMPLO 1

```
SELECT numemp, nombre, (SELECT MIN(fechapedido) FROM pedidos WHERE rep = numemp)  
FROM empleados;
```

- ✓ La **consulta principal** es SELECT... FROM empleados.
- ✓ La **subconsulta** es (SELECT MIN(fechapedido) FROM pedidos WHERE rep = numemp ).
- ✓ En esta **subconsulta** tenemos una referencia externa ( numemp ) es un campo de la tabla empleados (origen de la consulta principal).





## ¿QUÉ PASA CUANDO SE EJECUTA LA CONSULTA PRINCIPAL?

### EJEMPLO 1

**SELECT** numemp, nombre, (**SELECT MIN**(fechapedido) **FROM** pedidos **WHERE** rep = numemp)  
**FROM** empleados;

- ✓ Se coge el primer empleado y se **calcula la subconsulta** sustituyendo numemp por el valor que tiene en el primer empleado. La subconsulta obtiene la fecha más antigua en los pedidos del rep = 101.
- ✓ Se coge el segundo empleado y se **calcula la subconsulta** con numemp = 102 (numemp del segundo empleado)... y así sucesivamente hasta llegar al último empleado. Al final obtenemos una lista con el número, nombre y fecha del primer pedido de cada empleado.
- ✓ Si quitamos la cláusula **WHERE** de la subconsulta obtenemos la fecha del primer pedido de todos los pedidos no del empleado correspondiente.





## SUBCONSULTA

Pregrado

Ingeniería de  
Sistemas

- ✓ Las subconsultas pueden **anidarse** de forma que **una subconsulta aparezca en la cláusula WHERE** (por ejemplo) **de otra subconsulta** que a su vez forma parte de otra consulta principal.
- ✓ En la práctica, una consulta consume mucho más tiempo y memoria cuando se incrementa el número de niveles de anidamiento.
- ✓ La consulta resulta también más difícil de leer , comprender y mantener cuando contiene más de uno o dos niveles de **subconsultas**.

### EJEMPLO 2

```
SELECT numemp, nombre
FROM empleados
WHERE numemp = (SELECT rep
                FROM pedidos WHERE clie = (SELECT numclie
                                           FROM clientes
                                           WHERE nombre = 'Julia Antequera'))
```



En este **ejemplo**, por cada línea de pedido se **calcula la subconsulta** de clientes, y esto se repite por cada empleado, en el caso de tener 10 filas de empleados y 200 filas de pedidos (tablas realmente pequeñas), la subconsulta más interna se ejecutaría 2000 veces (10 x 200).





## Subconsultas correlacionadas

Una subconsulta correlacionada es una subconsulta que utiliza los valores de la consulta exterior en su cláusula WHERE. La subconsulta se evalúa una vez por cada fila procesada por la consulta externa. Puede usar subconsultas correlacionadas para filtrar o modificar los datos al consultar tablas virtuales en un servicio de datos SQL.

idprodu...	nombreProducto	idProvee...	idCategoria	cantidadPorUnidad	precioUnidad	unidadesEnExisten...	unidadesEnPed...
1	Té Dharamsala	1	1	10 cajas x 20 bolsas	18	39	0
2	Cerveza tibetana Barley	1	1	24 - bot. 12 l	19	17	40
3	Sirope de regaliz	1	2	12 - bot. 550 ml	10	13	70
4	Especias Cajun del chef Anton	2	2	48 - frascos 6 l	22	53	0
5	Mezcla Gumbo del chef Anton	2	2	36 cajas	21	0	0
6	Mermelada de grosellas de la abuela	3	2	12 - frascos 8 l	25	120	0
7	Peras secas orgánicas del tío Bob	3	7	12 - paq. 1 kg	30	15	0
8	Salsa de arándanos Northwoods	3	2	12 - frascos 12 l	40	6	0
9	Buey Mishi Kobe	4	6	18 - paq. 500 g	97	29	0
10	Pez espada	4	8	12 - frascos 200 ml	31	31	0
11	Queso Cabrales	5	4	paq. 1 kg	21	22	30
12	Queso Manchego La Pastora	5	4	10 - paq. 500 g	38	86	0
13	Algas Konbu	6	8	caja 2 kg	6	24	0



# Sub Consulta No Correlacional

Ejemplos de subconsulta **no correlacional**:

Mostrar los productos cuyo precio es mayor al promedio de todos los precios de los productos:

```
SELECT * FROM productos WHERE precioUnidad > (SELECT avg(precioUnidad) FROM productos)
```

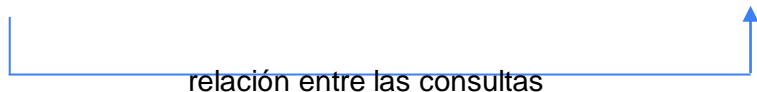


# Sub Consulta Correlacional

Ejemplos de subconsulta **correlacional**:

Mostrar los productos cuyo precio es mayor al promedio de todos los precios de los productos de su categoría:

```
SELECT * FROM productos P WHERE precioUnitario > (SELECT avg(precioUnitario) FROM  
productos R WHERE P.idCategoría = R.idCategoría)
```





## ¿QUÉ HEMOS APRENDIDO HOY?



Para que reflexionen y entiendan la importancia de los temas tratados y el mejoramiento de su propio proceso de aprendizaje.



Universidad **César Vallejo**

Licenciada por Sunedu  
para que puedas salir adelante