

SUB CONSULTAS/JOINS

Sistemas de Bases de Datos 1

Sección B

SUB CONSULTA

- Una sub consulta es una instrucción SELECT anidada dentro de una instrucción DML.
- Se puede utilizar cualquier cantidad de sub consultas en una instrucción DML.

SUB CONSULTA

- Se pueden comparar los resultados de una sub consulta por medio de operadores de comparación regulares (=, <>, etc.).
- El numero de parámetros que se comparan en la consulta principal debe ser igual al numero de parámetros devueltos por la sub consulta.

SUB CONSULTA

Ejemplo:

```
select * from Empleado where sueldo > (  
    SELECT AVG(monto) from Pago  
    where mes = 'Marzo'  
)
```

```
select * from EMPLEADO where fecha_nacimiento > (  
    SELECT fecha_nacimiento, from EMPLEADO where  
    nombre = 'FELIPE SANTOS'  
)
```

SUB CONSULTA

- Se pueden realizar comparaciones sobre una lista de resultados por medio de las instrucciones ALL, ANY, SOME.
 - ALL: Indica que la comparación debe cumplirse para todos los elementos de la lista.
 - ANY: Indica que la comparación debe cumplirse para por lo menos un elemento de la lista.
 - SOME: Es un sinónimo de la cláusula ANY.

SUB CONSULTA

Ejemplo:

```
select * from Empleado where sueldo > ALL (  
    select sueldo from Empleado where DATEPART(YEAR,  
    fecha_nacimiento) = 1990  
)
```

En el ejemplo anterior, la sub consulta devuelve un listado de sueldos de personas que nacieron en 1990.

HAVING

- Especifica una condición de búsqueda para un grupo o agregado (SUM, AVG, COUNT, etc.)
- HAVING se coloca después de las condiciones de tipo WHERE, y de existir cláusulas GROUP BY, después de estas.

HAVING

Ejemplo:

```
select SUM(e.sueldo), d.nombre  
from Departamento d, Empleado e  
GROUP BY d.nombre  
HAVING SUM(e.sueldo) > 7000
```

En la siguiente consulta se busca los departamentos en los que la suma de sueldos de todos sus empleados de mas de 7000.

IN/ NOT IN

- Permite indicar una condición que será verdadera si un campo se encuentra en un grupo de valores, ya sea dados directamente o por medio de una sub consulta.
- Puede ser usado en cualquier sentencia DML (INSERT, DELETE, UPDATE, SELECT)

IN/ NOT IN

Ejemplo:

```
SELECT * from Empleado where codigo IN(  
select codigo from Pago where Mes = 'Marzo'  
)
```

```
SELECT * from Empleado where SUELDO NOT IN (2000,  
3000, 4000)
```

EXISTS/ NOT EXISTS

- Especifica una sub consulta para probar la existencia de filas.
- Por lo general, se realiza un JOIN en la sub consulta de la tabla utilizada en la consulta principal y la consulta utilizada en la sub consulta.
- Puede ser usado en cualquier sentencia DML (INSERT, DELETE, UPDATE, SELECT)

EXISTS / NOT EXISTS

Ejemplo:

```
select * from Empleado where EXISTS (  
    select * from Pago where Empleado.codigo = Pago.cempleado  
    and mes = 'Marzo'  
);
```

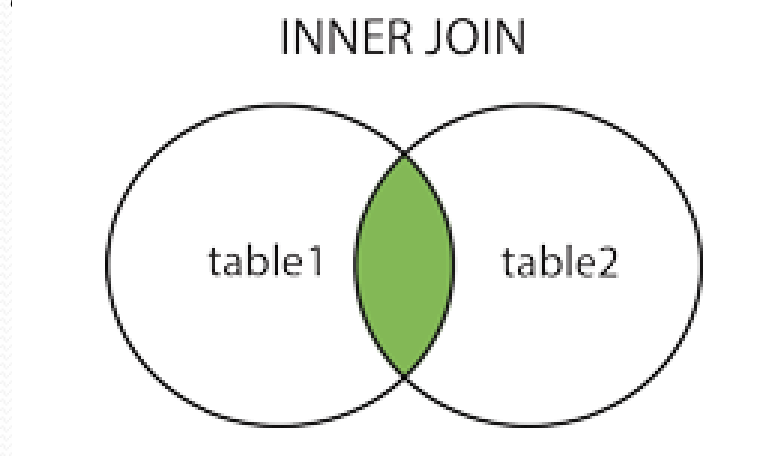
JOIN

- Una clausula JOIN es usada para combinar filas en dos tablas, basándose en un campo en común entre estas.
- Existen diferentes tipos de SQL JOINS.

INNER JOIN

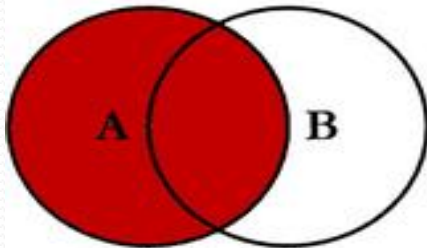
- La clausula INNER JOIN selecciona las filas de ambas tablas siempre y cuando halla un emparejamiento entre las columnas de ambas tablas.

```
SELECT column_name(s) FROM table1 INNER JOIN table2  
ON column_name = column_name;
```

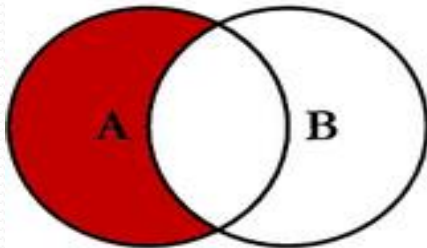


OTROS TIPOS DE JOIN

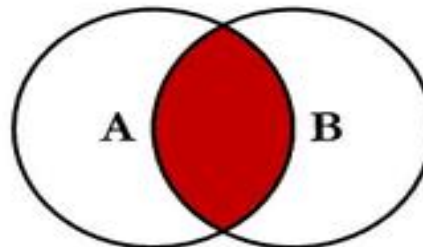
SQL JOINS



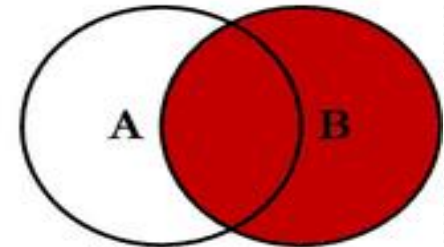
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



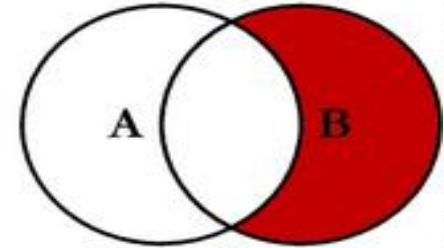
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



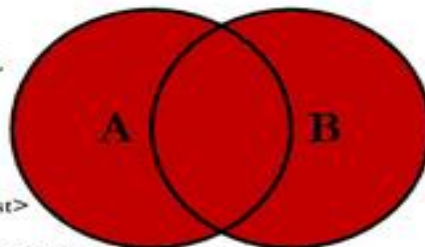
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



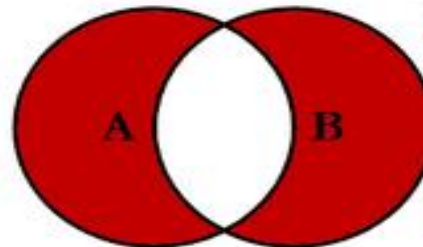
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

Tarea 4

- Realizar las consultas SQL de las tablas presentadas en la siguiente diapositiva:
 - Todos los empleados que tienen salario mayor al promedio.
 - Nombre cada empleado y el nombre de su jefe.
 - Nombre de los empleados que tiene un sueldo mayor al de todos los del área de ciencias.
 - Nombre de los departamentos donde la suma de los sueldos de los empleados pertenecientes es igual o mayor 3000
 - Empleados que pertenecen a los departamentos de ciencias y defensa.

Tablas

- Empleado

codigo	nombre	sueldo	departamento	jefe
1	tony	2000	10	4
2	steve	1800	20	4
3	bruce	1000	10	1
4	nick	1200	30	
5	natasha	1500	20	2
6	clint	1500	20	2

Region

codigo	nombre
1	norte
2	sur
3	este
4	oeste

- Departamento

codigo	nombre	region
10	ciencias	1
20	defensa	2
30	planeamiento	2

Instrucciones

- Crear las tablas en cualquiera de los 4 DBMS previamente instalados.
- Insertar los datos.
- Realizar las consultas solicitadas (deberán de enviar el screenshot de la consulta y el resultado).
- Tomar en cuenta que las consultas deberán de ser pensadas para que funcionen independientemente si se ingresan más departamentos y más empleados.
- Último día de entrega 09/09/2015
- Enviar al correo, asunto [BD1]Tarea4
- Nombre del archivo Tarea4_#carnet.pdf
- No suponer, cualquier duda al correo con asunto [BD1]Duda