"El producto más valioso que conozco es la información"

05

TALLER DE BASE DE DATOS - INF 272 SEMESTRE 2/2022

Consultar varias tablas (JOIN)

Un join es una operación que relaciona dos o más tablas para obtener un resultado que incluya datos (campos y registros) de ambas; las tablas participantes se combinan según los campos comunes a ambas tablas.

Hay tres tipos de combinaciones:

1) combinaciones internas (inner join o join),
2) combinaciones externas y
3) combinaciones cruzadas.

Natural JOIN

Es el más sencillo une las tablas por medio de las columnas que se llamen igual, es decir busca todas las columnas de la tabla A que se llamen igual en la tabla B y las une o mezcla:

```
select *
 from TABLA1
 natural join TABLA2
```

Ejemplo:

```
SELECT * FROM DEPARTAMENTOS
NATURAL JOIN EMPLEADOS
ORDER BY DEPNRO;
```

Clausula USING

En este caso se especifica sobre que columnas haga el join, para evitar resultados incoherentes:

```
select *
 from TABLA1
 join TABLA2
 using (campo);
```

Ejemplo:

```
SELECT nom_dep,nomemp,apeemp
FROM EMPLEADOS
JOIN DEPARTAMENTOS
USING (DEPNRO);
```

Combinaciones Internas

La combinación interna emplea "join", que es la forma abreviada de "inner join". Se emplea para obtener información de dos tablas y combinar dicha información en una salida. La sintaxis básica es la siguiente:

```
select CAMPOS
 from TABLA1
 join TABLA2
 on CONDICIONdeCOMBINACION;
```

Ejemplo:

```
SELECT * FROM EMPLEADOS
 JOIN DEPARTAMENTOS
 ON DEPARTAMENTOS.DEPNRO=EMPLEADOS.DEPNRO;
```

SELF JOINS

Uniones consigo mismo, se producen cuando hay una unión dentro de la misma tabla, cuando alguna de las columnas de la misma tabla es una Foreign Key con alguna columna de la tabla:

```
SELECT T.EMPLOYEE_ID,T.FIRST_NAME AS
TRABAJADOR,J.FIRST_NAME AS JEFE,T.MANAGER_ID
FROM HR.EMPLOYEES T JOIN HR.EMPLOYEES J
ON T.MANAGER_ID = J.EMPLOYEE_ID;
```

Saca el nombre del empleado y el nombre de su jefe

Combinación Externa Izquierda

Las combinaciones externas combinan registros de dos tablas que cumplen la condición, más los registros de la segunda tabla que no la cumplen; es decir, muestran todos los registros de las tablas relacionadas, aún cuando no haya valores coincidentes entre ellas.
Se emplea una combinación externa izquierda para mostrar todos los registros de la tabla de la izquierda. Si no encuentra coincidencia con la tabla de la derecha, el registro muestra los campos de la segunda tabla seteados a "null".

```
select CAMPOS
  from TABLAIZQUIERDA
  left join TABLADERECHA
  on CONDICION;
```

```
select * from departamentos
      left join empleados
on departamentos.depnro=empleados.depnro;
```

Combinación Externa Derecha

Una combinación externa derecha ("right outer join" o "right join") opera del mismo modo sólo que la tabla derecha es la que localiza los registros en la tabla izquierda..

```
select CAMPOS
  from TABLAIZQUIERDA
  right join TABLADERECHA
  on CONDICION;
```

```
select * from empleados
      right join departamentos
on departamentos.depnro=empleados.depnro;
```

Es FUNDAMENTAL tener en cuenta la posición en que se colocan las tablas en los "outer join". En un "left join" la primera tabla (izquierda) es la que busca coincidencias en la segunda tabla (derecha); en el "right join" la segunda tabla (derecha) es la que busca coincidencias en la primera tabla (izquierda).

Combinación Externa Completa

Una combinación externa completa ("full outer join" o "full join") retorna todos los registros de ambas tablas. Si un registro de una tabla izquierda no encuentra coincidencia en la tabla derecha, las columnas correspondientes a campos d**e la tabla derecha aparecen seteadas a "null", y si la tabla de la derecha no encuentra correspondencia en la tabla izquierda, los campos de esta última aparecen conteniendo "null".**

```
select CAMPOS
 from TABLAIZQUIERDA
 full join TABLADERECHA
 on CONDICION;
```

```
select * from empleados
full join departamentos
on
departamentos.depnro=empleados.depnro;
```

Combinación Cruzada Cross

Las combinaciones cruzadas (cross join) muestran todas las combinaciones de todos los registros de las tablas combinadas. Para este tipo de join no se incluye una condición de enlace. Se genera el producto cartesiano en el que el número de **filas del resultado es igual al número de registros de la primera tabla multiplicado por el número de registros de la segunda tabla, es decir, si hay 3 registros en una tabla y 4 en la otra, retorna 12 filas. La sintaxis básica es ésta:**

```
select CAMPOS
from TABLA1
cross join TABLA2;
```

Combinaciones y agrupamiento

Podemos usar "group by" y las funciones de agrupamiento con combinaciones de tablas. Por ejemplo para ver la cantidad de empleados en cada departamento:

```
select nom_dep as departamento,
 count(*) as cantidad
 from departamentos
 join empleados
 on empleados.depnro=departamentos.depnro
 group by departamentos.nom_dep;
```

Subconsultas

Subconsultas (Subquerys), es una consulta normal que devuelve valores que pueden ser utilizados por otros:

Ejemplo: Mostrar los empleados que ganan un salario similar al salario más alto:

```
select max(salario) from empleados;
```

Nos da una respuesta 20000

```
Select nomemp,apeemp,salario
from empleados
Where salario = 20000;
```

```
Select nomemp,apeemp,salario
from empleados
Where salario = (select max(salario)
from empleados);
```

WHERE – HAVING - FROM

Subconsultas WHERE

Recomendación: Probar primero la subconsulta y ver si saca el resultado requerido.

Todos los empleados que trabajan en la unidad de "Sistemas"

```
select nomemp,apeemp,depnro from empleados
where depnro=(select depnro from departamentos where nom_dep='Sistemas')
```

Todos los empleados que ganan un salario por encima del salario promedio y además trabajan en la unidad de Sistemas.

```
select nomemp,apeemp,salario,depnro from empleados
where salario > (select avg(salario) from empleados)
and depnro=(select depnro from departamentos where nom_dep='Sistemas')
```

Subconsultas en HAVING

Listar todos los códigos de departamentos cuyo salario promedio esta por encima del salario promedio de toda la empresa.

```
select depnro,round(avg(salario)) AS MEDIA
from empleados
group by depnro
having  avg(salario) > (select avg(salario) from empleados);
```

Subconsultas en IN

Listar los nombres, apellidos, salario y numero de departamento, de todos los empleados cuyo salario es similar al salario máximo de su departamento.

```
select nomemp,apeemp,salario,depnro
from empleados where salario IN
(select max(salario) from empleados group by depnro);
```

```
select nomemp,apeemp,salario,depnro
from empleados where (depnro,salario) IN
(select depnro,max(salario) from empleados group by depnro);
```

Subconsultas con ANY-ALL

Que empleados ganan mas que los empleados de la unidad de "Sistemas" cuyo DEPNRO es 3, sin incluir a los empleados de la unidad mencionada.

```
select nomemp,apeemp,salario
from empleados
where salario > ANY (select salario from empleados where depnro=3)
and depnro<>3;
```

```
select nomemp,apeemp,salario
from empleados
where salario > ALL (select salario from empleados where depnro=3)
and depnro<>3;
```

Subconsultas Sincronizadas

Se denomina así cuando la consulta principal y la subconsulta secundaria dependen una de la otra. Cuando se ejecuta la primaria con el valor obtenido la secundaria cambia.

Ejemplo; Sacar un listado de los empleados que ganan el salario mas alto en su departamento.

```
select depnro,nomemp,apeemp,salario
from empleados emp
where salario = (select max(salario) from empleados where
depnro=emp.depnro);
```

Subconsultas EXISTS

Ejemplo: Sacar un listado de los departamentos que no tengan empleados.

```
select depnro,nom_dep
from departamentos dep
where not exists (select * from empleados where depnro=dep.depnro);
```

Que tengan empleados

```
select depnro,nom_dep
from departamentos dep
where exists (select * from empleados where depnro=dep.depnro);
```

Operaciones de Conjuntos

Las operaciones de conjuntos combinan los resultados de dos o más consultas "select" en un único resultado.
Se usan cuando los datos que se quieren obtener pertenecen a distintas tablas y no se puede acceder a ellos con una sola consulta.
Es necesario que las tablas referenciadas tengan tipos de datos similares, la misma cantidad de campos y el mismo orden de campos en la lista de selección de cada consulta.
Hay cuatro operadores de conjuntos: union, unión all, intersect y minus.

UNION / UNION ALL

Unión permite unir registros de dos tablas sin duplicados. La sintaxis para unir dos consultas con el operador "union" es la siguiente:

```
SENTENCIA CONSULTA1
union
SENTEENCIA CONSULTA2;
```

Recuerde que las consultas DEBEN tener el mismo numero de valores retornados y los valores deben ser del mismo tipo.

```
select nombre,salario,cargo from empleados
union
select nombre,salario,cargo from vendedores;
```

Si queremos que se incluyan TODOS los registros, aún duplicados, debemos emplear "union all".

INTERSECCION

"intersect" devuelve la intersección de las consultas involucradas; es decir, el resultado retornará los registros que se encuentran en la primera y segunda consulta (y demás si las hubiere), o sea, los registros que todas las consultas tienen en común.

```
SENTENCIA SELECT1
 intersect
 SENTENCIA SELECT2;
```

MINUS

"minus" (diferencia) devuelve los registros de la primera consulta que no se encuentran en segunda consulta, es decir, aquellos registros que no coinciden.

```
SENTENCIA SELECT1
 minus
 SENTENCIA SELECT2;
```