
4.- VISUALIZACIÓN DE DATOS A PARTIR DE VARIAS TABLAS.

4.1 .- JOINS y producto cartesiano

4.2 .- Tipos de JOINS

4.1.-JOINS Y PRODUCTO CARTESIANO

Cuando se necesitan datos de más de una tabla en la base de datos, se utiliza la condición JOIN. Las filas de una tabla pueden unirse a filas de otra, siempre y cuando, exista un nexo de unión entre las tablas para que se verifique la condición de JOIN, debe ir especificada la cláusula WHERE. En caso contrario no será JOIN sino un producto cartesiano.

Sintaxis:

```
SELECT tabla.columna, tabla.columna...  
FROM tabla1, tabla2  
WHERE tabla1.columna1= tabla2.columna2;
```

Ejemplo 1:

Visualizar los campos ENAME, DEPTNO de la tabla empleados y DNAME de la tabla departamento.

```
SELECT EMP.ENAME, EMP.DEPTNO, DEPT.DNAME  
FROM EMP, DEPT  
WHERE EMP.DEPTNO = DEPT.DEPTNO;
```

Se establece producto cartesiano cuando se omite la condición de JOIN, cuando la definición de condición dada es inválida, y/ o cuando interesa combinar las filas de una tabla con las de la otra.

Ejemplo 2:

```
SELECT EMP.ENAME, EMP.DEPTNO, DEPT.DNAME  
FROM EMP, DEPT;
```

4.2 TIPOS DE JOINS

EQUIJOINS

Es una consulta que se realiza en dos tablas, siempre y cuando exista una tabla con clave principal y otra con clave extranjera que sea clave principal de la primera, es decir, que ambas estén relacionadas.

Los registros en las equijoins se recuperan de la siguiente manera:

- En la cláusula SELECT se especifican los nombres de las columnas a recuperar, si el nombre de la columna es el mismo en varias tablas habrá que especificar nb_tabla. nb_columna.

- En la cláusula FROM se especifica el nombre de las tablas que van a formar parte de la join.
- En la cláusula WHERE se especifica la condición de unión.

Ejemplo 1:

Seleccionar en la tabla EMP: ename, empno, deptno y en DEPT: depton y loc, cuando los números de departamento sean iguales.

```
SELECT ENAME,EMPNO, EMP.DEPTNO, DEPT.DEPTNO, DEPT.LOC  
FROM EMP,DEPT  
WHERE EMP.DEPTNO= DEPT.DEPTNO;
```

Calificación de nombres de columna ambiguos

Se recomienda usar como prefijo los nombres de las tablas para calificar los nombres de las columnas que están en varias tablas.

El rendimiento se mejora si se utiliza el prefijo de la tabla. Es conveniente distinguir columnas que tienen nombres idénticos pero pertenecen a diferentes tablas usando alias de columnas.

Ejemplo 2:

Visualizar el número del empleado KING, el nombre, el número de departamento y la localidad:

```
SELECT EMPNO, ENAME, EMP.DEPTNO, LOC  
FROM EMP , DEPT  
WHERE EMP. DEPTNO = DEPT. DEPTNO AND ENAME = 'KING';
```

Uso de alias en las tablas

Calificar las tablas con los nombres de estas puede llevar mucho tiempo si el nombre de la tabla es largo.

Los alias de tabla se utilizan para abreviar la escritura de la sentencia SELECT.

Del ejemplo anterior los alias serian:

```
SELECT E.EMPNO, E.ENAME,E.DEPTNO,D.LOC  
FROM EMP E, DEPT D  
WHERE E.DEPTNO= D.DEPTNO AND E.ENAME= 'KING';
```

NON EQUIJOINS

Los **NON EQUIJOIN** son los casos en los que ninguna columna de una tabla se corresponde directamente con los de otra, aunque sí indirectamente

Ejemplo:

Obtener usando alias de tablas el nombre del empleado, el salario y el grado.

```
SELECT E.ENAME, E.SAL, S.GRADE
FROM EMP E, SALGRADE S
WHERE E.SAL BETWEEN S.LOSAL AND S.HISAL;
```

OTHERS JOINS

Si una fila no satisface la condición de JOIN no aparecerá el resultado en la consulta. La fila o filas que faltan pueden ser recuperadas mediante el operador (+), este operador se coloca en la WHERE en aquella columna deficiente de información.

Ejemplo:

En el ejemplo anterior no aparece el departamento 40, porque en el no hay empleados, si añadimos este operador saldrá el departamento pero sin empleados.

```
SELECT EMPNO, ENAME ,D.DEPTNO, D.LOC
FROM EMP E, DEPT D
WHERE E.DEPTNO(+) =D.DEPTNO;
```

SELF JOINS

Consiste en relacionar una tabla consigo misma.

Ejemplo:

Realizar una consulta en la que se especifique: 'ename' trabaja para 'mgr'

```
SELECT T.ENAME || ' trabaja para ' || J.ENAME "Empleado y Jefe"
FROM EMP T, EMP J
WHERE T.MGR = J.EMPNO;
```