Administración de base de datos

Índice

1.- Desventajas de los enlaces en el WHERE

Los comandos básicos (SELECT – FROM – WHERE) de una consulta SQL se realizan por pasos:

a) Producto cartesiano entre las tablas indicadas en el FROM
b) Producto cruz con las condiciones del WHERE (enlaces)
c) Aplicar más condiciones (selección)

Ejemplo 1
Título de las películas del genero terror.

```
SELECT TITULO
FROM TGENEROS G, TPELICULAS P
WHERE G.CODGENERO = P.CODGENERO AND
G.NOMGENERO= “TERROR”
```

Si el contenido de las tablas incluidas en el FROM es el siguiente:

tpeliculas

codpel	titulo	codgenero	codclasif	codtipo	cantidad	status
1	x	1	1	1	12	1
2	y	2	1	1	20	1
3	z	3	1	1	23	1

tgeneros

codgenero	nomgenero	status
1	suspenso	1
2	acción	1
3	terror	1
4	comedia	1

Figura 1

El resultado de cada paso para la anterior consulta es:
a) Producto cartesiano entre las tablas indicadas en el FROM (figura 2)
b) Producto cruz con las condiciones del WHERE (figura 3)
c) Aplicar más condiciones (figura 4)

Como no existen películas del genero “comedia” este fué eliminado desde la aplicación del paso “b”.

tpeliculas / tgeneros

codpel	titulo	codgenero	codclasif	codtipo	cantidad	status	codgenero	nomgenero	status
1	x	1	1	1	12	1	1	suspenso	1
2	y	2	1	1	20	1	1	suspenso	1
3	z	3	1	1	23	1	1	suspenso	1
1	x	1	1	1	12	1	2	acción	1
2	y	2	1	1	20	1	2	acción	1
3	z	3	1	1	23	1	2	acción	1
1	x	1	1	1	12	1	3	terror	1
2	y	2	1	1	20	1	3	terror	1
3	z	3	1	1	23	1	3	terror	1
1	x	1	1	1	12	1	4	comedia	1
2	y	2	1	1	20	1	4	comedia	1
3	z	3	1	1	23	1	4	comedia	1

Figura 2

tpeliculas / tgeneros

codpel	titulo	codgenero	codclasif	codtipo	cantidad	status	codgenero	nomgenero	status
1	x	1	1	1	12	1	1	suspenso	1
2	y	2	1	1	20	1	2	acción	1
3	z	3	1	1	23	1	3	terror	1

Figura 3

tpeliculas / tgeneros

codpel	titulo	codgenero	codclasif	codtipo	cantidad	status	codgenero	nomgenero	status
3	z	3	1	1	23	1	3	terror	1

Figura 4

El JOIN producido en el comando WHERE por los enlaces elimina los registros que no cumplen las restricciones. (G.CODGENERO = P.CODGENERO)

Pero en algunas consultas es importante no perder los registros que no cumplen con las restricciones de los enlaces.

<u>Ejemplo 2</u>
Cuantas películas hay de cada genero.

La tabla resultado (según los datos de la figura 1) es la mostrada en la figura 5.

nomgenero	cantidad
suspenso	1
acción	1
terror	1
comedia	0

Figura 5

La consulta:

```
SELECT NOMGENERO, COUNT(*) AS CANTIDAD
FROM TGENEROS G, TPELICULAS P
WHERE G.CODGENERO = P.CODGENERO
GROUP BY NOMGENERO
```

Elimina el genero “comedia” (ver figura 3) y al formar los grupos la tabla resultado es la siguiente:

nomgenero	cantidad
suspenso	1
acción	1
terror	1

Figura 6

Es necesario forzar la aparición del genero “comedia” e indicar que no existen películas (cero)

2.- JOIN Externo

Existen 3 tipos de JOIN Externo:

- Join externo por la izquierda
- Join externo por la derecha
- Join externo completo

Sintáxis

FROM TABLA_IZQ **LEFT OUTER JOIN** TABLA_DER **ON** *condición*

FROM TABLA_IZQ **RIGHT OUTER JOIN** TABLA_DER **ON** *condición*

FROM TABLA_IZQ **FULL OUTER JOIN** TABLA_DER **ON** *condición*

Donde *condición* indica la condición – el enlace – que une las tablas.

El join externo obliga a incluir los registros de la tabla que esta a la derecha, la izquierda o ambas en el resultado de la consulta.

Con las tablas de la figura 7 se presentan los tres joins externos.

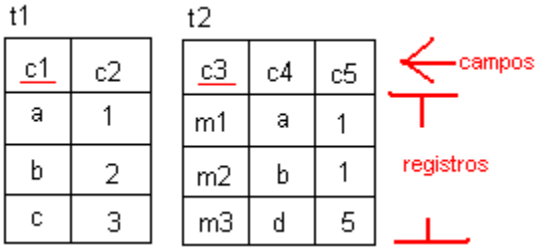

Figura 7

Tipo de JOIN	Descripción / consulta	Figura
JOIN EXTERNO POR LA IZQUIERDA	SELECT * FROM T1 LEFT OUTER JOIN T2 ON C1= C4 Incluye los registros de la tabla t1, cumplan o no la condición. En la figura 8 el último registro está formado por nulos en los campos de la tabla t2 que son unidos con la tabla t1.	t1 t2 c1 c2 c3 c4 c5 a 1 m1 a 1 b 2 m2 b 1 c 3 null null null Figura 8
JOIN EXTERNO POR LA DERECHA	SELECT * FROM T1 RIGHT OUTER JOIN T2 ON C1= C4 Incluye los registros de la tabla t2, cumplan o no la condición. En la figura 9 el último registro está formado por nulos en los campos de la tabla t1 que son unidos con la tabla t2.	t1 t2 c1 c2 c3 c4 c5 a 1 m1 a 1 b 2 m2 b 1 null null m3 d 5 Figura 9
JOIN EXTERNO COMPLETO	SELECT * FROM T1 FULL OUTER JOIN T2 ON C1= C4 Obliga a incluir todos los registros de las tablas t2 y t1, cumplan o no la condición. La figura 10 muestra los últimos registros formados por nulos en los campos de la tabla t1 y t2 que no coinciden.	t1 t2 c1 c2 c3 c4 c5 a 1 m1 a 1 b 2 m2 b 1 c 3 null null null null null m3 d 5 Figura 10

Ejemplo 2 (Continuación)

Para asegurar la inclusión de todos los generos se necesita una consulta que los muestre y

el join externo forza la aparición de sus registros. (También llamada consulta pivote)

La consulta “SELECT * FROM TGENEROS“ que muestra los generos es colocada a la derecha o izquierda según el join externo a usar.

La otra tabla del join externo presenta la cantidad de películas de los generos que tienen por lo menos una película.

```
SELECT CODGENERO, COUNT(*) AS CANTIDAD
FROM TPELICULAS P
GROUP BY CODGENERO
```

Estas consultas deben compartir un campo en común que permita su unión. (el campo CODGENERO en este ejemplo)

Si se escoge un join externo por la izquierda la siguiente consulta muestra los datos buscados.

```
SELECT T1.NOMGENERO, T2.CANTIDAD
FROM
(SELECT * FROM TGENEROS )  AS T1 'consulta pivote

LEFT OUTER JOIN

    ( SELECT CODGENERO, COUNT(*) AS CANTIDAD
    FROM TPELICULAS P
    GROUP BY CODGENERO ) AS T2

ON T1.CODGENERO = T2.CODGENERO          'Enlace entre t1 y t2
```

Nota: Los comentarios están en letra azul.

Los datos de T1 y T2 son:

t1

codgenero	nomgenero	status
1	suspenso	1
2	acción	1
3	terror	1
4	comedia	1

Figura 11

t2

codgenero	cantidad
1	1
2	1
3	1

Figura 12

El join externo produce la tabla de la figura 13 que contiene un registro con campos nulos donde se obligó la unión.

La anterior consulta produce como resultado la tabla de la figura 14. En el último registro el contenido del campo CANTIDAD es nulo y debe ser reemplazado por un cero usando el

operador CASE.

t1

codgenero	nomgenero	status
1	suspenso	1
2	acción	1
3	terror	1
4	comedia	1

t2

codgenero	cantidad
1	1
2	1
3	1
null	null

Figura 13

nomgenero	cantidad
suspenso	1
acción	1
terror	1
comedia	null

Figura 14

3.- Operador CASE
Este operador se coloca en el comando SELECT y evalua el contenido de un campo en cada registro de la tabla resultado para cambiarlo por otro valor.

Cada operador CASE representa un campo de la tabla resultado de una consulta; pero no genera un nombre se necesita usar la instrucción AS.

Sintaxis

CASE { WHEN *condición* THEN *valor_1* } … ELSE *valor_n* END

Donde
- Las instrucciones encerradas entre llaves indican un grupo y los tres puntos suspensivos la repetición incondicional del grupo encerrado entre llaves.
- La instrucción WHEN indica las condiciones para cada caso.
- *Condicion* es una expresión que obtiene un valor booleano usando operadores relacionales, logicos y el operador IS; se apoya en los campos que tiene la tabla resultado.
- *Valor_1 y valor_n* son los nuevos valores incluidos en la tabla resultado.

Si ninguna condición de los caos (WHEN) se cumple, se aplica el *valor_n* de la instrucción ELSE.

Ejemplo 3
Listado de los nombres de los trabajadores, acompañado de una leyenda que describe su antigüedad en la empresa. Si su antigüedad es:

- Menor a 5 años la leyenda es “principiante en formación”.
- Mayor a 5 años y menor de 10 años, la leyenda es “supervisor en formación”.
- Mayor de 10 años, la leyenda es “Trabajador comprometido”.

```
SELECT NOMBRE,
        CASE WHEN ANIOS<5 THEN “principiante en formación”
             WHEN ANIOS>=5 AND ANIOS <10 THEN “supervisor en formación”
             ELSE “Trabajador comprometido” END AS LEYENDA
FROM TTRABAJADOR
```

nombre	leyenda
y	principiante en formación
z	supervisor en formación
⋮	

Figura 15

Ejemplo 2 (continuación)

El operador CASE quita los nulos y los reemplaza por un cero.

Consulta original

```
SELECT T1.NOMGENERO, T2.CANTIDAD
FROM
  (SELECT * FROM TGENEROS )  AS T1
  LEFT OUTER JOIN
  (SELECT CODGENERO, COUNT(*) AS CANTIDAD
    FROM TPELICULAS P
    GROUP BY CODGENERO ) AS T2
ON T1.CODGENERO = T2.CODGENERO
```

nomgenero	cantidad
suspenso	1
acción	1
terror	1
comedia	null

Figura 14

Consulta con CASE

```
SELECT T1.NOMGENERO,
          CASE WHEN T2.CANTIDAD IS NULL
                    THEN 0
               ELSE T2.CANTIDAD END
         AS CANTIDAD
FROM
 (SELECT * FROM TGENEROS )  AS T1
 LEFT OUTER JOIN
 ( SELECT CODGENERO, COUNT(*) AS CANTIDAD
    FROM TPELICULAS P
    GROUP BY CODGENERO ) AS T2
ON T1.CODGENERO = T2.CODGENERO
```

nomgenero	cantidad
suspenso	1
acción	1
terror	1
comedia	0

Figura 15