

Ejercicios 6 de Programación en Transact-SQL

Seguiremos con la base Gestion8 y practicaremos ahora con cursores.

1.- Queremos que en la pestaña de Mensajes aparezca una lista de empleados por oficina con el siguiente formato:

```
-----
11 Valencia
-----
    106 Luis Antonio
-----
12 Alicante
-----
    101 Antonio Viguer
    104 José González
    103 Juan Rovira
-----
13 Castellon
-----
    114 Pablo Moreno
    105 Vicente Pantalla
-----
21 Badajoz
-----
    102 Alvaro Jaumes
    103 ... ..
```

Como queremos que aparezca el resultado en la pestaña de mensajes tenemos que utilizar un **PRINT** para visualizar cada línea.

Como queremos hacer un PRINT por cada fila necesitamos poder tratar cada fila individualmente, por lo que necesitamos un **cursor**.

¿Qué filas necesitamos? Necesitamos el número y nombre de cada empleado así como su nº de oficina y ciudad de la oficina. Además las filas tienen que estar ordenadas por oficina y dentro de cada oficina por nombre del empleado. La SELECT que da respuesta a esto será la SELECT asociada al cursor.

¿Qué tenemos que hacer con el cursor? Leer cada empleado, cuando empiece una nueva oficina escribir las tres líneas de cabecera de oficina y mientras el empleado sea de esa oficina escribiré los datos del empleado, cuando cambie la oficina volveré a escribir la cabecera de oficina y luego sus empleados. Así hasta que se acaben los empleados.

```
USE Gestion8;
DECLARE cur INSENSITIVE CURSOR FOR
    SELECT E.oficina, ciudad, numemp, nombre
    FROM Oficinas O INNER JOIN Empleados E ON O.oficina=E.oficina
    ORDER BY oficina, nombre;
-- Declaramos las variables que van a recoger los valores del FETCH
DECLARE @ofi AS INTEGER, @num INTEGER, @nom AS VARCHAR(30);
DECLARE @ciu VARCHAR(30), @nom AS VARCHAR(30);
DECLARE @gofi INTEGER; -- Para controlar cuando cambia de oficina
OPEN cur;
FETCH NEXT FROM cur INTO @ofi, @ciu, @num, @nom;
WHILE @@FETCH_STATUS = 0
    BEGIN
        -- empieza una nueva oficina
        PRINT '-----';
        PRINT STR(@ofi,2) + ' ' + @ciu;
        PRINT '-----';
        SET @gofi = @ofi; -- Nos guardamos la oficina
        WHILE @gofi = @ofi AND @@FETCH_STATUS = 0
            -- Mientras estemos en la misma oficina y queden empleados
            BEGIN
                PRINT '    ' + STR(@num,3) + ' ' + @nom; --visualizamos el emp
                FETCH NEXT FROM cur INTO @ofi, @ciu, @num, @nom; -- leemos el siguiente
            END;
        -- Hemos leído un empleado de otra oficina o no quedan empleados
    END;
CLOSE cur;
DEALLOCATE cur;
GO
```

2.- Queremos obtener de todos los empleados de la oficina 12 su número (numemp), nombre (del empleado) y cuántos pedidos ha realizado, pero utilizando el procedimiento almacenado CuantosPedidosPorRep que creamos en un ejercicio anterior.

Resolución:

El procedimiento almacenado CuantosPedidosPorRep tiene un parámetro de entrada que recoge el nº del empleado y devuelve la cantidad de pedidos realizados en un parámetro de salida.

Por lo tanto tendremos que obtener el código de los empleados de la oficina 12 (con una SELECT numemp FROM Empleados WHERE oficina = 12), y por cada empleado llamar al procedimiento y después escribir el resultado.

Como tenemos que tratar **cada fila de la SELECT por separado** necesitamos un **cursor**.

¿Qué filas necesitamos tratar? "SELECT numemp FROM Empleados WHERE oficina = 12" El cursor tendrá asociado esa SELECT.

¿Qué haremos con el cursor? Recorrerlo y por cada fila ejecutar el procedimiento y hacer un PRINT del resultado.

```
DECLARE cur INSENSITIVE CURSOR FOR
    SELECT numemp FROM Empleados WHERE oficina = 12;
-- Declaramos las variables que van a recoger los valores del FETCH
DECLARE @num INTEGER;
-- Declaramos las variables que van a recoger los valores devueltos por el
procedimiento
DECLARE @pedidos INTEGER;
-- Tratamos el cursor
OPEN cur; -- Lo abrimos
FETCH NEXT FROM cur INTO @num; -- Leemos la primera fila (el primer numemp)
WHILE @@FETCH_STATUS = 0
    BEGIN
        -- ejecutamos el procedimiento, el valor con el que rellenar el parámetro
de entrada lo tenemos en @num
        -- recogemos el resultado devuelto en @pedidos
        EXEC CuantosPedidosPorRep @num, @pedidos OUTPUT
        -- Escribimos el resultado en la pestaña Mensajes,
        -- recuerda que los números se tienen que pasar a string
        PRINT STR(@num,3)+ ' ' + STR(@pedidos,3);
        FETCH NEXT FROM cur INTO @num; -- leemos el siguiente
    END;
-- No quedan más empleados
CLOSE cur;
DEALLOCATE cur;
GO
```

3.- Ahora vamos a añadir una variante al ejercicio 1. Tienen que aparecer todos las oficinas incluso las que no tienen empleados y en este caso, en vez de la lista de empleados se escribirá el texto *La oficina no tiene empleados*.

```
-----
21 Badajoz
-----
      102 Alvaro Jaumes
      108 Ana Bustamante
      111 Juan Gris
-----

22 A Coruña
-----
      107 Jorge Gutiérrez|
-----

23 Madrid
-----

La oficina no tiene empleados
```

El razonamiento es el mismo que en el ejercicio 1.

Lo que va a cambiar son las filas, ahora necesitamos que aparezcan también las oficinas que no tienen empleados → LEFT en lugar de INNER. ¡Ojo! Al cambiar a LEFT habrá que cambiar algo más.

¿Qué tenemos que hacer con el cursor? Lo mismo que en el ejercicio 1 pero ahora hay que distinguir cuando se ha leído un empleado o una fila de oficina sin empleados para escribir una cosa u otra. Añadir un IF en el PRINT del empleado.

```
USE Gestion8;
DECLARE cur INSENSITIVE CURSOR FOR
    SELECT O.oficina,ciudad, numemp, nombre
    FROM Oficinas O LEFT JOIN Empleados E ON O.oficina=E.oficina
    ORDER BY oficina, nombre;
-- Declaramos las variables que van a recoger los valores del FETCH
DECLARE @ofi AS INTEGER, @num INTEGER;
DECLARE @ciu VARCHAR(30), @nom AS VARCHAR(30);
DECLARE @gofi INTEGER; -- Para controlar cuando cambia de oficina
OPEN cur;
FETCH NEXT FROM cur INTO @ofi, @ciu, @num, @nom;
WHILE @@FETCH_STATUS = 0
    BEGIN
        -- empieza una nueva oficina
        PRINT '----- ';
        PRINT STR(@ofi,2)+ ' ' + @ciu;
        PRINT '----- ';
        SET @gofi = @ofi; -- Nos guardamos la oficina
        WHILE @gofi = @ofi AND @@FETCH_STATUS = 0
            -- Mientras estemos en la misma oficina y queden empleados
            BEGIN
                IF @num IS NULL PRINT ' La oficina no tiene empleados'
                ELSE PRINT '      ' + STR(@num,3)+ ' ' + @nom; --visualizamos el emp
                FETCH NEXT FROM cur INTO @ofi, @ciu, @num, @nom; -- leemos el sig.
            END;
        -- Hemos leído un empleado de otra oficina o no quedan empleados
        END;
CLOSE cur;
DEALLOCATE cur;
GO
```

4.- Queremos obtener empleados con los clientes a los que están asignados, con el siguiente formato:

```
-----
101 Antonio Viguer
-----
      2102 Alvaro Rodríguez
      2105 Antonio Canales
      2115 Vicente Martínez
-----
102 Alvaro Jaumes
-----
      2114 Cristina Bulini
      2123 José Libros
      2120 Juan Malo
      2106 Juan Suárez
```

El razonamiento es el mismo que en el ejercicio1, necesitamos un **cursor**.

¿Qué filas necesitamos? Necesitamos los codigos y nombres de los clientes con los códigos y nombres de los empleados que tienen asignados. La SELECT que da respuesta a esto será la SELECT asociada al cursor.

¿Qué tenemos que hacer con el cursor? Lo mismo que en el ejercicio 1 pero ahora los bucles serán por cada empleado una cabecera , mientras sea el mismo empleado escribir sus clientes y cuando cambie de empleado volver a escribir la cabecera del empleado y seguir... Lo que nos tenemos que guardar es el numemp.

```
USE Gestion8;
DECLARE cur INSENSITIVE CURSOR FOR
    SELECT numemp, E.nombre, numclie, C.nombre
    FROM Clientes C INNER JOIN Empleados E ON repclie = numemp
    ORDER BY repclie, C.nombre;
-- Declaramos las variables que van a recoger los valores del FETCH
DECLARE @cli AS INTEGER, @emp INTEGER;
DECLARE @enom VARCHAR(30), @cnom AS VARCHAR(30);
DECLARE @gemp INTEGER; -- Para controlar cuando cambia de empleado
OPEN cur;
FETCH NEXT FROM cur INTO @emp, @enom, @cli, @cnom;
WHILE @@FETCH_STATUS = 0
    BEGIN
        -- empieza una nueva oficina
        PRINT '-----';
        PRINT STR(@emp,3) + ' ' + @enom;
        PRINT '-----';
        SET @gemp = @emp; -- Nos guardamos el empleado
        WHILE @gemp = @emp AND @@FETCH_STATUS = 0
            -- Mientras estemos en la mismo empleado y queden clientes
            BEGIN
                PRINT '      ' + STR(@cli,4) + ' ' + @cnom; --visualizamos el emp
                FETCH NEXT FROM cur INTO @emp, @enom, @cli, @cnom; -- leemos el
siguiente
            END;
        -- Hemos leído un cliente de otro empleado o no quedan clientes
    END;
CLOSE cur;
DEALLOCATE cur;
GO
```