

**ASIGNATURA:** SISTEMAS GESTORES DE BASES DE DATOS **PROFESOR:** JUAN RAMÓN ALEMANY

#### EJERCICIO

Realizar un cursor que liste los campos CustomerID, FirstName, MiddleName, LastName, PhoneNumber de las tablas Sales.Customer, Person.Person y Person.PersonPhone, de tal modo que el cursor debe listar TODOS los customers, pero con saltos de 5 en 5, es decir, primero el customer número 1, luego el número 6, luego el número 11, el número 16, el número 21 y así hasta el final, SABIENDO QUE existe un tipo de FETCH que permite dar saltos de “n” en “n” valores.

Este comando es: “FETCH RELATIVE 5 FROM NombreDelCursor”, pero para ello, habrá que declarar el cursor con la cláusula “SCROLL”.

DECLARE NombreDelCursor SCROLL CURSOR FOR SELECT .....

#### EJERCICIO

En la bases de datos AdventureWorks crear una nueva tabla llamada PERSONTOEXPORT, que tenga los campos Title, FirstName, MiddleName y LastName, del mismo tipo que la tabla PERSON.PERSON.

Realizar un cursor tal que, para cada registro de la tabla PERSON.PERSON lo inserte en la nueva tabla PERSONTOEXPORT, pero decodificando a castellano el campo Title, y manteniendo el resto de la información con los mismos valores. Si el Campo Title es NULL, entonces pondremos la palabra “Sr./Sra.”.

#### EJEMPLO DE TIPOS DE FETCH EN CURSORES

```
USE AdventureWorks2012;
GO
-- Execute the SELECT statement alone to show the
-- full result set that is used by the cursor.
SELECT LastName, FirstName FROM Person.Person
ORDER BY LastName, FirstName;

-- Declare the cursor.
DECLARE contact_cursor SCROLL CURSOR FOR
SELECT LastName, FirstName FROM Person.Person
ORDER BY LastName, FirstName;

OPEN contact_cursor;

-- Fetch the last row in the cursor.
FETCH LAST FROM contact_cursor;

-- Fetch the row immediately prior to the current row in the cursor.
FETCH PRIOR FROM contact_cursor;

-- Fetch the second row in the cursor.
FETCH ABSOLUTE 2 FROM contact_cursor;

-- Fetch the row that is three rows after the current row.
FETCH RELATIVE 3 FROM contact_cursor;

-- Fetch the row that is two rows prior to the current row.
FETCH RELATIVE -2 FROM contact_cursor;

CLOSE contact_cursor;
DEALLOCATE contact_cursor;
GO
```

SOLUCION COMPLETA DEL EJERCICIO PRIMERO

```
DECLARE @CustomerID INT
DECLARE @FIRSTNAME NVARCHAR(50)
DECLARE @MIDDLENAME NVARCHAR(50)
DECLARE @LASTNAME NVARCHAR(50)
DECLARE @PHONENUMBER NVARCHAR(25)

DECLARE CUSTOMER_CURSOR SCROLL CURSOR FOR SELECT Sales.Customer.CustomerID,
Person.Person.FirstName, Person.Person.MiddleName, Person.Person.LastName,
Person.PersonPhone.PhoneNumber FROM ((Sales.Customer INNER JOIN Person.Person ON
Sales.Customer.PersonID = Person.Person.BusinessEntityID) INNER JOIN
Person.PersonPhone ON Person.Person.BusinessEntityID =
Person.PersonPhone.BusinessEntityID)

OPEN CUSTOMER_CURSOR
FETCH NEXT FROM CUSTOMER_CURSOR INTO @CUSTOMERID,@FIRSTNAME,
@MIDDLENAME,@LASTNAME,@PHONENUMBER
WHILE (@@FETCH_STATUS = 0)
BEGIN
    PRINT CAST(@CUSTOMERID AS NVARCHAR(50))+ ' ' +
    @FIRSTNAME + ' ' +
    CASE WHEN @MIDDLENAME IS NULL THEN '*' ELSE @MIDDLENAME END + ' ' +
    @LASTNAME + ' ' + @PHONENUMBER
    FETCH RELATIVE 5 FROM CUSTOMER_CURSOR INTO @CUSTOMERID,@FIRSTNAME,
    @MIDDLENAME,@LASTNAME, @PHONENUMBER
END
CLOSE CUSTOMER_CURSOR
DEALLOCATE CUSTOMER_CURSOR
```