FUNDAMENTOS DE BASE DE DATOS.

Tema Nº13:Introducción a la programación en SQL SERVER IV.

Indicador de logro Nº13:Implementa cursores para la gestión de datos.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº13:**

Introducción a la programación en SQL SERVER IV.

**Subtema 13.1:**

Declaración de cursores, apertura de un cursor, lectura y recorrido de un cursor, cierre de un cursor y eliminación de un cursor.

Es como que, si Fuese una tabla, pero se almacena en la memoria de la computadora. Son usados para almacenar estructuras de datos en la secuencia de un proceso de negocio. Se recomienda usarlos en las situaciones que amerite el caso.

Son estructuras que se crean en la memoria del ordenador, capaces de albergar gran cantidad de datos, esto dependerá del tamaño de la memoria RAM o memoria disponible.

Los cursores son muy eficientes para utilizarlos en Job de la base de datos que realicen alguna operación donde necesitemos modificar alguna información dentro de un bucle. Los cursores demandan mucho del servidor de base datos, por lo tanto, no es recomendable abusar del mismo, ya que necesitan bastantes recursos para su ejecución

**Un cursor** es una herramienta de SQL Server que nos permite recorrer el resultado de una consulta SQL y realizar operaciones con estos resultados dentro de un bucle de datos.

* Un cursor es una variable que nos permite recorrer con un conjunto de resultados obtenidos a través de una sentencia SELECT fila por fila.
* El uso de los cursores es una técnica que permite tratar fila por fila el resultado de una consulta, contrariamente al SELECT SQL que trata a un conjunto de fila.
* Los cursores pueden ser implementador por instrucciones TRANSACT-SQL (cursores ANSI-SQL) o por la API OLE-DB.

**¿Cómo se crea un CURSOR?**

* Cuando trabajemos con cursores debemos seguir los siguientes pasos.
  + Declarar el cursor, utilizando DECLARE
  + Abrir el cursor, utilizando OPEN
  + Leer los datos del cursor, utilice FETCH …INTO
  + Cerrar el cursor, utilizando CLOSE
  + Liberar el cursor, utilizando DEALLOCATE

**ESTRUCTURA DE UN CURSOR:**

**-- Declaración del cursor**

DECLARE <nombre\_cursor> CURSOR FOR <sentencia\_sql>

**-- Apertura del cursor**

OPEN <nombre\_cursor>

**-- Lectura de la primera fila del cursor**

FETCH <nombre\_cursor> INTO <lista\_variables>

WHILE (@@FETCH\_STATUS = 0)

BEGIN

-- Procesos …

**-- Lectura de la siguiente fila de un cursor**

FETCH <nombre\_cursor> INTO <lista\_variables> ...

END -- Fin del bucle WHILE

**-- Cierra el cursor**

CLOSE <nombre\_cursor>

**-- Libera los recursos del cursor**

DEALLOCATE <nombre\_cursor>

**Subtema 13.2:**

Sentencias a emplear: DECLARE CURSOR, OPEN, FETCH-WHILE, CLOSE, DEALLOCATE.

**DECLARACIÓN DE CURSOR**

**DECLARE [**Nombre Cursor**] CURSOR FOR <**SQL Select, . . . . .**>**

El primer paso constará de la declaración del cursor, donde se indicarán (junto con el nombre del cursor) la consulta que el mismo representará y algunas otras características bastante interesantes.

**Un ejemplo de declaración de cursor es el siguiente:**

Donde **ProdInfo** representará al nombre del cursor y la sentencia “**SELECT ProductNumber, Name FROM Producto**” será el conjunto de datos del mismo, como comentamos previamente, es posible en este paso definir algunas características del comportamiento del cursor, por ejemplo, la sentencia:

**DECLARE** ProdInfo **CURSOR FOR SELECT** Name

**FROM** Producto

Indicará que el cursor será de solo lectura, más adelante veremos en detalle las opciones disponibles, por el momento nuestro objetivo es crear un cursor lo más simple posible.

**DECLARE** ProdInfo **CURSOR READ\_ONLY FOR**

**SELECT** Name **FROM** Producto

**APERTURA DEL CURSOR**

**OPEN** [nombre de cursor]

La apertura del cursor ejecutará la consulta definida en el paso previo y cargará los datos en el mismo. La función **OPEN** de T-SQL permitirá efectuar esta terea, para continuar con el ejemplo previo la forma de abrir el cursor será la siguiente:

**Open** ProdInfo

**CIERRE DEL CURSOR**

**CLOSE** [nombre de cursor]

En el cierre del cursor se liberarán los registros tomados por el mismo. Una vez que el cursor es cerrado ya no podrá recorrerse el conjunto de resultados hasta que el mismo sea reabierto, la sentencia CLOSE cerrará un cursor abierto y la sintaxis puede verse a continuación:

**Close** ProdInfo

**RECORRIDO DEL CURSOR Y ACCESO A LOS DATOS**

Este paso constará de recorrer los resultados del cursor, la instrucción FETCH permitirá efectuar dicha operación. Las filas leídas podrán copiarse a variables utilizando la sentencia INTO en combinación con la sentencia FETCH, por ejemplo, la sentencia:

**FETCH NEXT FROM [**nombre de cursor**]**

FETCH NEXT FROM **ProdInfo** INTO **@Description**

**Tomará la siguiente fila de resultados del cursor y lo alojará en la variable @Description.**

Un detalle a comentar es que en la sentencia INTO (como puede verse en el ejemplo anterior) el mapeo entre columnas del cursor y variables se realizará implícitamente, asignándose la primera columna a la primera variable, la segunda columna a la segunda variable y así sucesivamente. Esto implica que deberán crearse tantas variables como columnas se definan en la declaración del cursor y las mismas deberán ubicarse en el mismo orden que se encuentran definidas las columnas en la sentencia SELECT de la declaración.

Como cada sentencia FETCH leerá un registro, una pregunta interesante que podríamos hacernos es, ¿de qué manera podremos saber si existe un próximo o previo registro, o si hemos llegado al límite (ya sea superior o inferior)?. La respuesta se encontrará en una variable de SQL Server llamada @@FETCH\_STATUS que tomará el valor 0 si la lectura del registro ha sido correcta.

En este punto será también posible modificar o eliminar las filas que se van recorriendo, como veremos al final del artículo.

**LEER EL CURSOR**

Gracias a este bucle, se podrán leer todas las filas del cursor.

**WHILE** **(@@FETCH\_STATUS**=0)

**BEGIN**

**. . . . . .**

**FETCH NEXT FROM** [Nombre del Cursor]

**END**

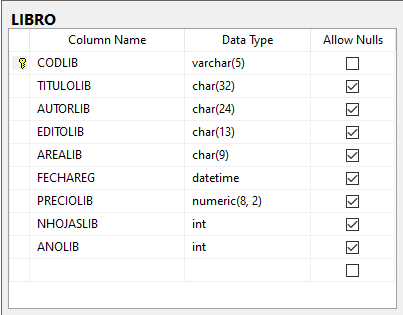
**DESHACER o DESALOJAR EL CURSOR**

DEALLOCATE Nombre de Cursor

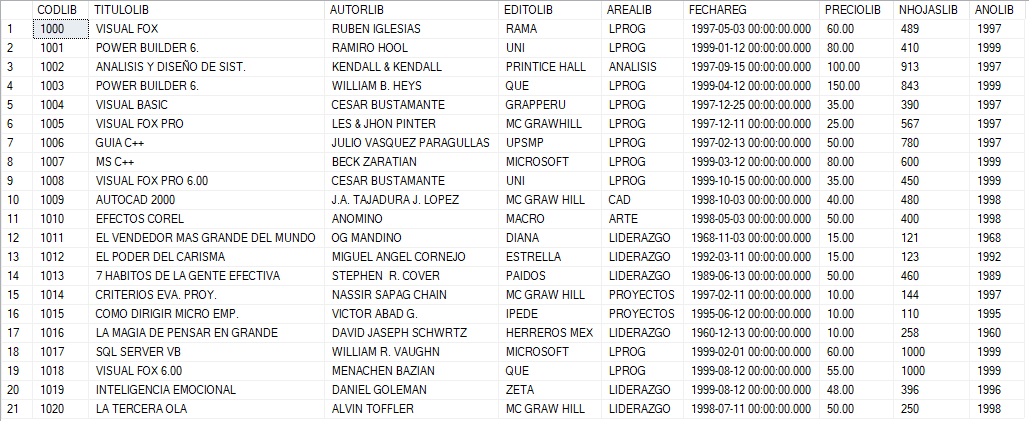
**DEALLOCATE** ProdInfo

Este paso eliminará la referencia al cursor definido previamente, por lo que ya no será posible realizar una reapertura del mismo.

Uso de Cursores, se tiene la siguiente tabla:



Cuyos datos son los siguientes:

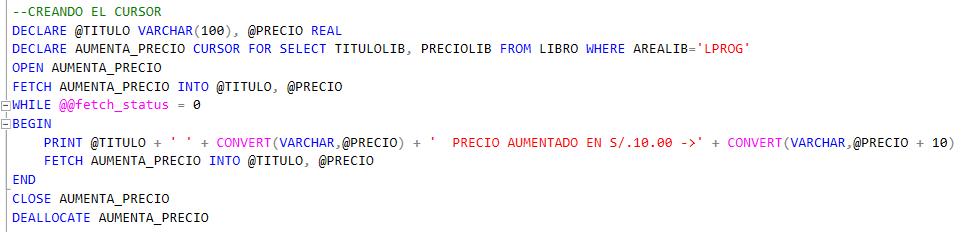


Puede descargar el Script de base de datos SEMANA13\_01.sql.

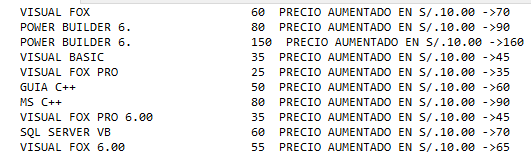
**EJEMPLO 01:**

Crear un cursor el cual muestre el **título del libro y su respectivo precio**, **y en una columna separada el precio aumentado en 10 soles más**, de aquellos libros que pertenezcan al área **‘LPROG’** para ellos debe contar con la siguiente tabla de la base de datos **LIBRERÍA**.

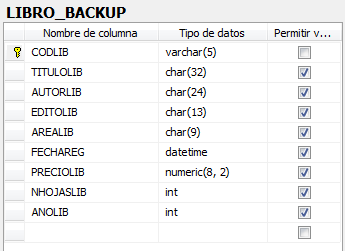
**SOLUCIÓN:** este código debe ser ejecutado en una sola selección.



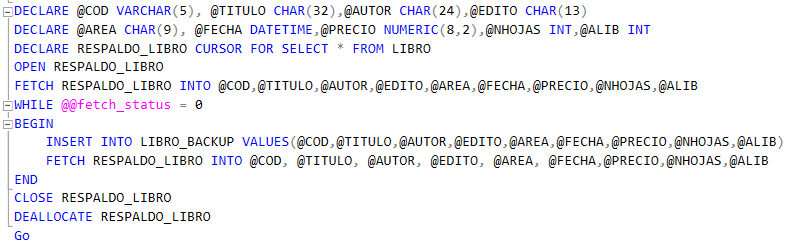
**RESULTADO DE LA EJECUIÓN DEL CURSOR:**



**EJEMPLO 02:**

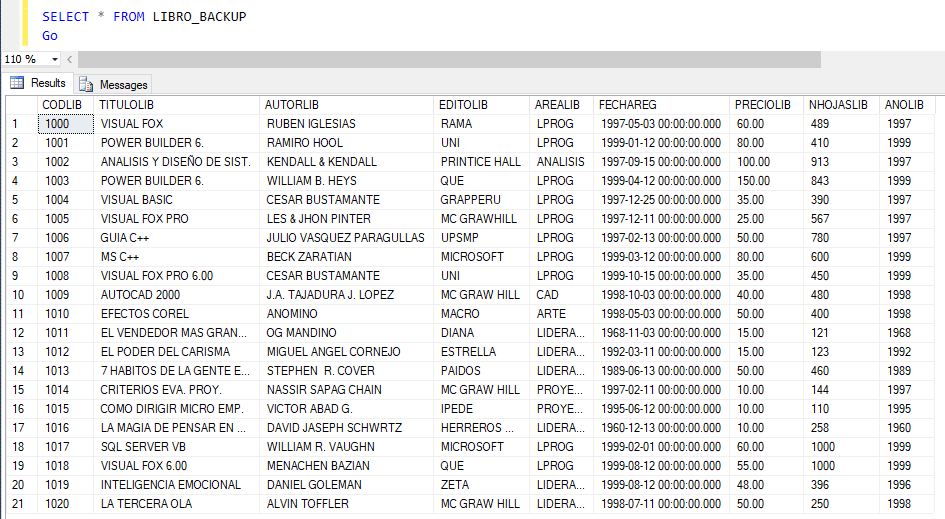
Mediante un cursor obtener una copia de seguridad de todos los datos de la tabla libro, para ellos debe contar con una nueva tabla con la misma estructura de la tabla libro, con el nombre **LIBRO\_BACKUP** (agregar esta nueva tabla a la base de datos).

**SOLUCIÓN:**



Ejecute este código en una sola selección.

Verificar en la tabla **LIBRO\_BACKUP**



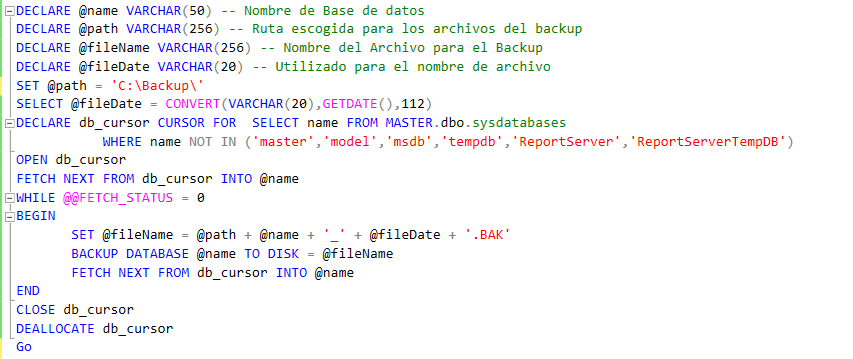
Los datos de la tabla **LIBRO** han pasado a la tabla **LIBRO\_BACKUP**.

**EJEMPLO 03:**

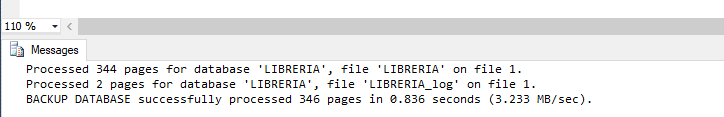
He aquí un ejemplo del cursor de Simple script de copia de seguridad las bases de datos de SQL Server en que se expidan copias de seguridad de manera serial:

**SOLUCIÓN:**

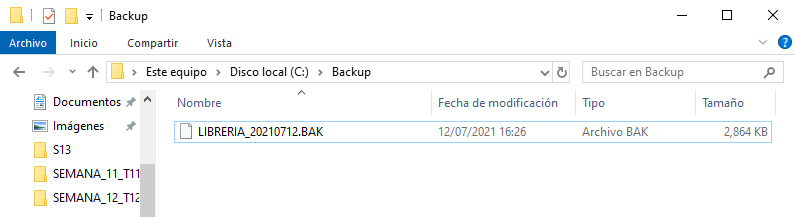
Para que el código funcione, debe crear una carpeta en el disco C:\ con el nombre **Backup**, es decir: **C:\Backup\.**



AL EJECUTAR EL CODIGO, COMO RESULTADO DEBERÁ OBTENER LOS SIGUIENTE:



Y EN EL DISCO C:\

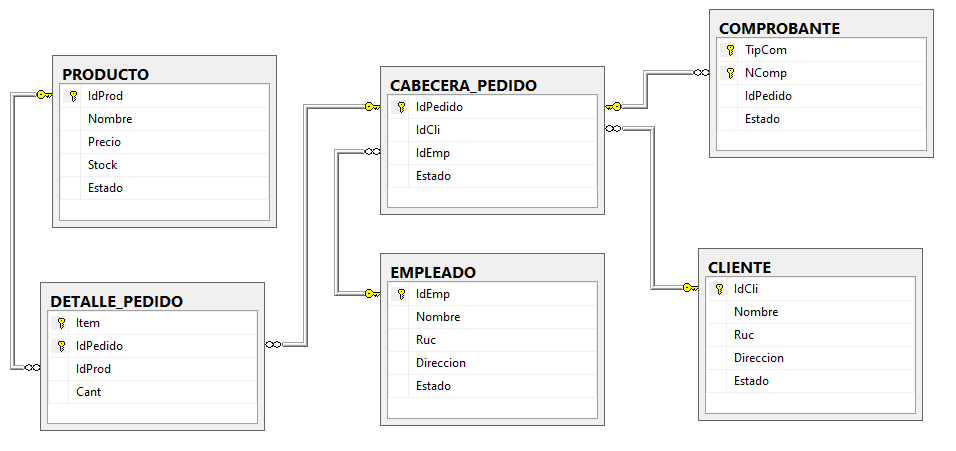


Se observa la copia de seguridad de la base datos que estamos usando.

**EJEMPLO 4:**

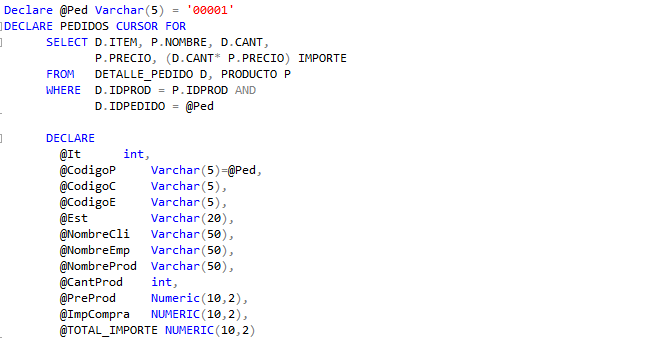
Se pide crear un cursor que sea capaz de mostrar los pedidos realizados, según la tabla **CABECERA\_PEDIDO** este tiene 3 pedidos los cuales son: ‘**00001’**, ‘**00002’** y ‘**00003’**, para ello se debe ingresar el número del pedido (ALMACENAR EN UNA VARIABLE), por intermedio de variables, y brindar la información solicitada, tal como se muestran en las siguientes imágenes

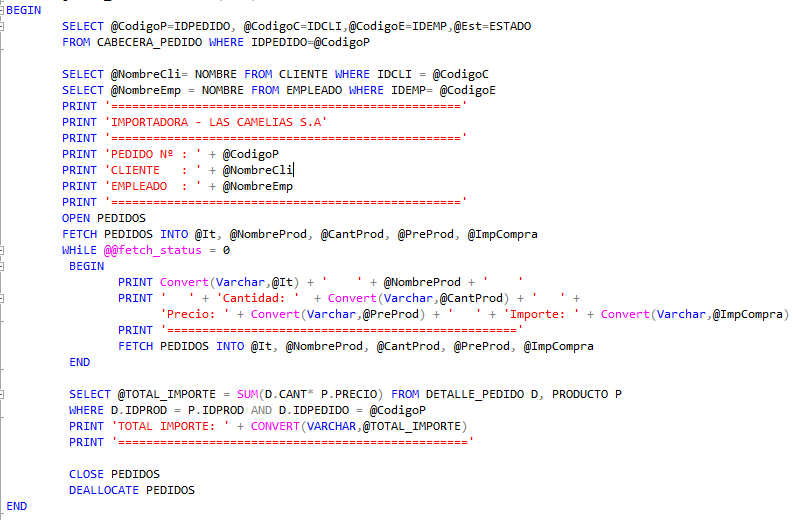
Para realizar este ejercicio, se debe contar con el siguiente modelo relacional:



EL script de base de datos se encuentra en el archivo SEMANA12\_02.sql

**SOLUCIÓN:**





Al ejecutar el código, como resultado obtendrá lo siguiente:



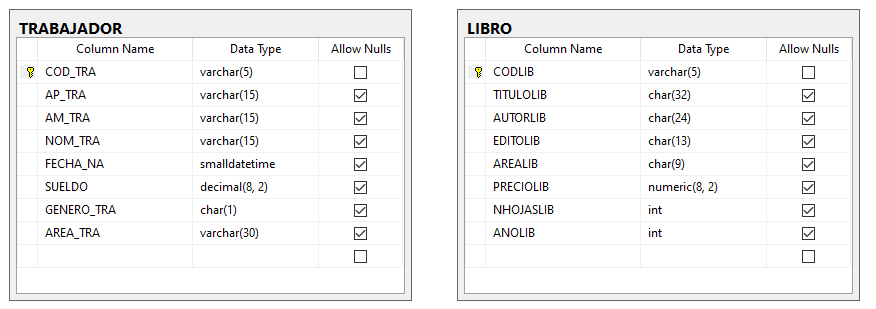
**CONCLUSIONES Y RECOMENDACIONES DE LA EXPERIENCIA**

* Un cursor es una estructura utilizada para el recorrido de los registros que es el resultado de una consulta.
* Un cursor se utiliza para el procesamiento individual de las filas devueltas por el sistema gestor de base de datos para una consulta.
* Un cursor se declara, se abre, lee, recorre, se cierra y se libera de memoria.
* @@FetchStatus es una variable global que devuelve un valor que representa el estado del cursor. El valor correcto es 0.

**ACTIVIDAD VIRTUAL**

Analizar y revisar detenidamente este recurso de aprendizaje y luego responde las preguntas propuestas.

* ¿Qué es un Cursor?
* Importancia de los cursores en las bases de datos.
* Se tiene el siguiente modelo relacional:



* Descargar y ejecutar el script de base de datos llamado **Semana12\_03.sql** y proceda a realizar los siguientes cursores:
  + Obtener el Libro, con el precio más caro.
  + Obtener los sueldos acumulados según el género del trabajador.
  + Obtener el total acumulado en precios según el área del libro.
  + Listado general de Libros.
  + Listado general de trabajadores.
  + Mediante el género, lista a los trabajadores y la sumatoria de los sueldos como resultado de resumen.
  + Mediante el área del libro, se pide mostrar los libros y la sumatoria de los precios, como resultado de resumen.