

**PRACTICO#5**

**TP5-T0 (Teoría sobre Programación en TRANSACT SQL)**



**ALUMNO:** Esrom Obed Andrade Ortiz

**REGISTRO:** 218003319

**DOCENTE:** UBALDO PEREZ FERREIRA

**FECHA:** 15/10/21

### Desarrolle las siguientes actividades:

- 1.Cuál es la diferencia entre el Lenguaje SQL y el Lenguaje Transact SQL

R. SQL es un lenguaje de consulta para sistemas de base de datos relacionales, pero no posee la potencia de los lenguajes de programación.

Transact SQL(T SQL), es el lenguaje de programación que proporciona al SQL los elementos característicos de programación.

2. Que se puede programar con Transact SQL

- Scripts
- Funciones
- Procedimientos almacenados
- Triggers

3. Liste los tipos de datos que maneja Transact SQL

- Datos numéricos
- Datos de carácter
- Datos de fecha
- Datos binarios
- Datos XML

4. Como se declara una variable en Transact SQL, cite ejemplos

Se declara de la siguiente manera:

```
DECLARE @VARIABLE CHAR(100)
```

```
DECLARE @NUMERO INT
```

```
DECLARE @CODIGO INT
```

5. Como se asigna valor a una variable en Transact SQL, cite ejemplos

```
SELECT @CODIGO =CALM
```

```
SELECT @NOMBRE=NOMA
```

6. Como asignar valores obtenidas de una consulta a una variable en Transact SQL, cite ejemplos

```
SET @NOMRE = (SELECT NOMB FROM PROV
               WHERE CPRV = 1)
```

## 7. Que es un CURSOR en Transact SQL

Los cursores en SQL Server pueden definirse como un elemento que representará a un conjunto de datos determinado por una consulta T-SQL. Los cursores permiten recorrer fila a fila, leer y eventualmente modificar dicho conjunto de resultados.

## 8. Para que sirve el uso de CURSOR en Transact SQL

Se pueden asignar variables con el fetch de un cursor y recorrer el conjunto de filas de una consulta SELECT.

## 9. Como se declare un CURSOR en Transact SQL

```
DECLARE CDATOS CURSOR FOR SELECT NOMP,COLO FROM PROD
```

## 10. Como se abre un CURSOR en Transact SQL

```
OPEN <NOMBRE CURSOR>
```

## 11. Para que sirve la instrucción FETCH en Transact SQL

Las instrucciones FETCH devuelven el valor de la columna especificada en DECLARE.

## 12. Para que sirve la variable @@FETCH\_STATUS, que valores devuelve

Devuelve el valor de la columna que se declaró.

## 13. Como se cierra un CURSOR en Transact SQL

```
CLOSE <NOMBRE CURSOR>
```

## 14. Cite un ejemplo de cómo leer todas las filas de una tabla en Transact SQL

```
DECLARE @NOMBRE VARCHAR (20)
DECLARE CDATOS CURSOR
FOR SELECT NOMP FROM PROD
OPEN CDATOS
FETCH CDATOS INTO @NOMBRE
WHILE (@@FETCH_STATUS = 0)
BEGIN
    PRINT @NOMBRE
    FETCH CDATOS INTO @NOMBRE
END
CLOSE CDATOS
DEALLOCATE CDATOS
```

15. Liste los principales operadores utilizados en Transact SQL

Operadores:

Orden de evaluación	Operador
1	( ) Parentesis
2	*, /, % (Multiplicación, División, Modulo)
3	+, - (Añadir / Positivo / Concatenar, Substracción / Negativo)
4	=, <, >, <=, >=, <>, !=, !>, !< (Comparación)
5	NOT
6	AND
7	BETWEEN, IN, LIKE, OR
8	= (Asignación)

16. Como funciona la estructura de control IF THEN, cite ejemplo

Permite expresar una expresión booleana (resultado VERDADERO/FALSO), y ejecuta las instrucciones contenidas dentro del formato BEGIN END. Sintaxis:

if()

begin

end

ELSE

Begin

End

EJEMPLO: Verificar si el proveedor con código 1 ha suministrado algún product

IF(select count(\*) from sumi where cprv=1) > 0 PRINT 'El proveedor 1 a sumistrado producto'

ELSE PRINT 'El proveedor 1 NO a sumistrado producto'

17. Como funciona la estructura de control CASE, cite ejemplo

Permite evaluar una expresión y devolver un valor u otro, SINTAXIS :

CASE <expresión>

WHEN <expresión> THEN <valor devuelto>

WHEN <expresión> THEN <valor devuelto>

WHEN <expresión> THEN <valor devuelto>

ELSE <valor devuelto>

END

Ejemplo: Programa para clasificar a un proveedor en base al importe de todos sus productos suministrado

```
DECLARE @clase char(40),@impt float
select @impt=isnull(sum(impt),0) from sumi where cprv=3
SET @clase=(CASE
WHEN (@impt>0 and @impt<=50) THEN 'Proveedor
Minoritas'
WHEN (@impt>51 and @impt<=200) THEN 'Proveedor
Intermedio'
WHEN (@impt>200) THEN 'Proveedor Mayoritas'
ELSE 'Proveedor sin Clasificar'
END)
PRINT @clase + 'Iporte' + cast(@impt as char(10))
```

#### 18. Como funciona la estructura de control WHILE, cite ejemplo

El bucle while se repite mientras la expresión se evalúe como verdadero, es el único bucle dentro de Transact SQL. SINTAXIS:

WHILE

BEGIN

END

EJEMPLO:Listar los productos suministrados por el proveedor 3, en la lista se debe mostrar: el nombre del producto,color,cantidad y fecha de suministro

```
DECLARE @nombre char(40),@color varchar(15), @cant FLOAT, @ftra
DATE
DECLARE CDATOS CURSOR
FOR SELECT nomb, colo, cant, ftra FROM sumi, prod where
sumi.cprd=prod.cprd
OPEN CDATOS
FETCH CDATOS INTO @nombre, @color, @cant, @ftra
WHILE (@@FETCH_STATUS = 0)
BEGIN
PRINT @nombre + @color + cast(@cant as char(5)) + cast(@ftra
as char(12))
FETCH CDATOS INTO @nombre, @color, @cant, @ftra
END
CLOSE CDATOS
DEALLOCATE CDATOS
```

## 19. Como se hace el control de errores en Transact SQL, cite ejemplo

A través de las instrucciones TRY y CATCH se proporciona el control de errores. La sintaxis es la siguiente:

```
BEGIN TRY
```

```
...
```

```
END TRY
```

```
BEGIN CATCH
```

```
...
```

```
END CATCH
```

Ejemplo: Programa que detecta que hay división por 0

```
BEGIN TRY
DECLARE @divisor int, @dividiendo int, @resultado int
SET @dividiendo = 100
SET @divisor = 0
--esta línea provoca un error de division entre 0
SET @resultado = @dividiendo/@divisor
PRINT 'no hay error'
END TRY
BEGIN CATCH
PRINT 'Se ha producido un error'
END CATCH
```

## 20. Para que sirve la variable @@ERROR y que valores devuelve

Almacena el número de error producido por la ultima sentencia Transact SQL ejecutada

```
DECLARE @divisor int, @dividiendo int, @resultado int
SET @dividiendo = 100
SET @divisor = 0
--esta línea provoca un error de division entre 0
SET @resultado = @dividiendo/@divisor
IF(@@ERROR = 0)
PRINT 'no hay error'
ELSE
PRINT 'Se ha producido un error'
```