TRANSACT SQL

Comentarios:

Comentar sólo una línea

--

Comentar un párrafo

/* principio del párrafo

fin del párrafo*/

DECLARAR VARIABLES

DECLARE @nomVariable tipoDato

DECLARE @nomVariable1 tipoDato1, .. , @nomVariableN tipoDatoN

ASIGNAR VALORES A UNA VARIABLE

- Mediante SET con un valor

SET @nomVariable= ValorTipoDatoCompatible

Para imprimir el valor de una variable: PRINT @nomVariable

Imprimir un texto: PRINT 'HOLA MUNDO!'

- Mediante SET con el resultado de una select
  Ejemplo:

  -- Asignar el nombre del paciente cuyo código de paciente es 1 a la variable @nomPaciente:

  -- Declaramos variable:

  DECLARE @ nomPaciente Nvarchar(30)

  -- Asignamos

  SET @ nomPaciente=(SELECT NOMBRE FROM PACIENTES WHERE CODIGO=1)

  -- Mostramos el valor almacenado en la variable

  PRINT @ nomPaciente

- Mediante una select asignar valores a varias variables:

(Debemos tener en cuenta que el resultado de la select deberá ser sólo un registro, si la consulta devolviera más de un registro, los valores que quedarían asignados son los de la última fila.)

Ejemplo BD clínica:

-- Asignar el nombre y la ciudad del paciente cuyo código de paciente es 1 a las variables nomP y ciud.

-- Declaramos variable:

DECLARE @nomP Nvarchar(30), @ciud NVARCHAR(15)

-- Asignamos directamente con la select

SELECT @nomP=NOMBRE,@ciud=CIUDAD FROM PACIENTES WHERE CODIGO=1

-- Mostramos el valor almacenado en la variable

PRINT @ nomP+' ('+ @ ciud+')'

OPERADORES

- Asignación **=**
- Operadores aritméticos**: +**,**-**,*****,**/**,******(exponente), **%**(módulo)
- Operadores relacionales o de comparación: **=** (igual a)

  <> ó **!=** (distinto de o no igual a)

  > (mayor que) **<** (menor que) **>=** (mayor o igual que) **<=** (menor o igual que)

  !> (no mayor a) **!<** (no menor a)

- Operadores lógicos: **AND**(y lógico), **NOT**(negación), **OR**(o lógico)
- Operador de concatenación: **+**
- OTROS:
  - **ALL**(devuelve TRUE si el conjunto completo de todas las comparaciones es TRUE)
  - **ANY**(devuelve TRUE si cualquier elemento del conjunto de comparaciones es TRUE)
  - **BETWEEN**(devuelve TRUE si el operando está dentro del intervalo)
  - **EXISTS**(devuelve TRUE si el resultado de una subconsulta contiene filas)
  - **IN**(devuelve TRUE si operando está en la lista)
  - **LIKE**(devuelve TRUE si el operando coincide con un patrón)
  - **NOT**(invierte el valor de cualquier operador booleano)
  - **SOME**(devuelve TRUE si alguna de las comparaciones de un conjunto es TRUE)

ESTRUCTURAS DE CONTROL

Estructura condicional IF

Permite evaluar una expresión booleana(de resultado Sí(1) o No(0)), y ejecutar las operaciones contenidas en el bloque formado por BEGIN END.

Simple:

```
IF (<expresion>)
    BEGIN
    …
    END
```

Compuesta:

```
IF (<expresion>)
    BEGIN
    …
    END
ELSE
    BEGIN
    …
    END
```

ANIDADA:

```
IF (<expresion>)
    BEGIN
    …
    END
ELSE IF (<expresion>)
            BEGIN
            …
            END
     ELSE
            BEGIN
            …
            END
```

Ejercicio de clase:

```
-- UTILIZAR SENTENCIAS CONDICIONALES
-- Utilizando la BD clinicasql
/* Realizar un script, que realice lo siguiente:
Si existe el paciente cuyo código es 99, que le actualice el nombre de
paciente a 'PEPITO LOPEZ SANCHEZ', si no existe, que inserte este
registro en la BD
    */
    Solución:
-- UTILIZAR SENTENCIAS CONDICIONALES
-- Utilizando la BD clinicasql
/* Realizar un script, que realice lo siguiente:
Si existe el paciente cuyo código es 99, que le actualice el nombre de
paciente a 'PEPITO LOPEZ SANCHEZ', si no existe, que inserte este
```

```
registro en la BD
*/
-- Declaramos variable:
DECLARE @cod smallint, @nom nvarchar(30)
-- Asignamos con set
SET @cod=99
SET @nom='PEPITO LOPEZ SANCHEZ'

IF(EXISTS(SELECT * FROM PACIENTES WHERE CODIGO=@cod))
	BEGIN --SI EXISTE EL PACIENTE 99, LE CAMBIAMOS NOMBRE
		PRINT 'ACTUALIZAMOS'
		UPDATE PACIENTES SET NOMBRE=@nom WHERE CODIGO=@cod

	END
 ELSE
	BEGIN --SI NO EXISTE EL PACIENTE 99, LO INSERTAMOS
		PRINT 'INSERTAMOS'
		INSERT INTO PACIENTES(CODIGO,NOMBRE) VALUES (@cod,@nom)
	END


UTILIZAR SENTENCIAS CONDICIONALES
-- Utilizando la BD clinicasql
/* Realizar un script, que realice lo siguiente:
Si hay pacientes diabéticos en el turno 2 que nos muestre todos los datos y si no
lo hay que muestre 'NO HAY PACIENTES DIABÉTICOS EN EL TURNO 2"
*/
```

Ejercicios para casa:

Realizar una consulta con transactSQL que muestre lo siguiente:
Pacientes que han visitado 4 o más veces la clínica:

(Todos los datos de los pacientes)

Pacientes que han visitado 3 o más veces la clínica:

(Todos los datos de los pacientes)

Pacientes que han visitado 2 o más veces la clínica:

ESTRUCTURAS DE CONTROL II

Estructura condicional CASE

Permite evaluar una expresión y devolver un valor u otro.

Sintaxis general (con expresión)

CASE(<expresion>)

WHEN (<valor_expresion1>) THEN (<valor_devuelto1>)

…..

WHEN (<valor_expresionN>) THEN (<valor_devueltoN>)

ELSE (<valor_devuelto>) --Valor por defecto

Estructura condicional CASE (que me permite evaluar diferentes expresiones)

CASE

WHEN (<expresion1>)=(<valor_expresion1>) THEN (<valor_devuelto1>)

…..

WHEN (<expresionN>)=(<valor_expresionN>) THEN (<valor_devueltoN>)

ELSE (<valor_devuelto>) --Valor por defecto

```
DECLARE @dia smallint, @nomDia varchar(20)
-- Asignamos con set
SET @dia=8

SET @nomDia=
   (CASE @dia
       WHEN (1) THEN ('LUNES')
       WHEN (2) THEN ('MARTES')
       WHEN (3) THEN 'MIERCOLES'
       WHEN (4) THEN ('JUEVES')
       WHEN (5) THEN ('VIERNES')
       WHEN (6) THEN ('SABADO')
       WHEN (7) THEN ('DOMINGO')
   ELSE 'VALOR NO TIPIFICADO' --Valor por defecto
   END)
   PRINT @nomDia
```

--Como podemos ver, el uso de los paréntesis en las expresiones de WHEN y THEN, delante de CASE y detrás de END es opcional.

```
DECLARE @dia smallint, @nomDia varchar(20)
-- Asignamos con set
SET @dia=5

SET @nomDia=
   CASE @dia
       WHEN 1 THEN 'LUNES'
       WHEN 2 THEN 'MARTES'
       WHEN 3 THEN 'MIERCOLES'
       WHEN 4 THEN 'JUEVES'
       WHEN 5 THEN 'VIERNES'
       WHEN 6 THEN 'SABADO'
       WHEN 7 THEN 'DOMINGO'
   ELSE 'VALOR NO TIPIFICADO' --Valor por defecto
   END
   PRINT @nomDia
```

Getdate()

Year(Getdate())

Month()

Day()

Conversiones de tipo.

Cast (@nomVar as tipoNuevo)

cast(@mes as varchar)

ESTRUCTURAS DE CONTROL III

Estructura condicional WHILE

Permite realizar bucles, es decir, un bucle WHILE se repite mientras la expresión se evalúe a verdadera, mientras tanto ejecutará las operaciones contenidas en el bloque formado por BEGIN END.

Caso más simple:

```
WHILE (<expresion>)
    BEGIN
    …
    END
```

Modificadores que podemos emplear en el bucle:

CONTINUE: nos permite pasar a la siguiente operación.

BREAK: Nos permite salir del bucle.