
Universidad
Tecmilenio®

Universidad Tecmilenio

Consulta en Microsoft SQL Server®

Subconsultas con T-SQL

Semana 3

Introducción

El propósito de una consulta es lograr extraer datos de cierta manera y con ciertas condiciones de acuerdo con el escenario en el que sea necesario extraer datos de una tabla, pero para lograr esto, necesitamos datos de otra tabla, para lo que se aplica una subconsulta.

Universidad Tecmilenio

Explicación

Introducción a las subconsultas

Una subconsulta es una incepción de consultas, ya que se escribe una consulta dentro de la misma. Esta subconsulta puede ser integrada en varias partes de la consulta original, es decir, ya sea para crear una columna y antes del FROM, o bien, dentro de la cláusula WHERE (Microsoft, 2022).

Tienen ciertas características que las identifican.

Las subconsultas seleccionan una columna para traer el valor de esta.

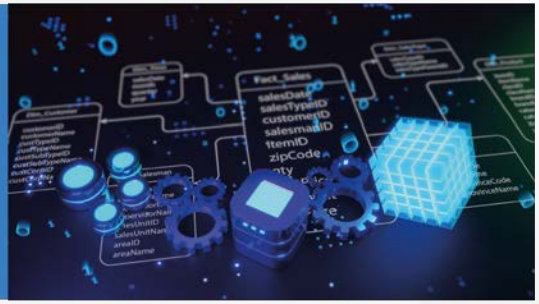
```
SELECT Columna1
   (SELECT MAX (Precio)
    FROM Tabla1
    WHERE Precio > '300') AS PrecioMax
FROM Tabla2
```

Universidad Tecmilenio

Explicación

Subconsulta en espacio de condición **WHERE**

```
SELECT Columna
FROM Tabla
WHERE Columna NOT IN
   (SELECT Columna2
    FROM Tabla2
    WHERE Columna2 = 90);
```



Consulta utilizando **subconsultas**

```
SELECT [Columna]
FROM Tabla
WHERE EjemploPrecio =
   (SELECT EjemploPrecio
    FROM Tabla
    WHERE [Columna] = 'Valor');
```

Consulta utilizando **JOIN**

```
SELECT Columna
FROM Tabla AS NombreTabla1
    JOIN Tabla AS NombreTabla2
      ON (NombreTabla1.EjemploPrecio =
NombreTabla2.EjemploPrecio)
WHERE NombreTabla2[Columna] = 'Valor';
```

Explicación

Subconsultas escalares

La premisa básica de una subconsulta escalar es aquella que solo **regresa** un valor, es decir, **una fila con una columna**.

```
SELECT Columna
FROM tabla
WHERE expresión =
(SELECT DISTINCT Columna2
FROM tabla2 WHERE Precio = 1000);
```

De acuerdo con AWS (2022), en caso de que el resultado de la subconsulta no regrese filas, entonces deberá regresar un resultado **nulo**.

Este tipo de subconsultas se utiliza principalmente para extraer un dato que después sirva de valor de ingreso y/o condición para desplegar otros datos.

Explicación

Subconsultas anidadas

Existen instrucciones u operadores no numéricos que regularmente no se usan para otros tipos de consultas. En la siguiente tabla podemos ver los operadores más comunes:

All	• Compara los registros de la consulta. • Si cumplen, se completa la instrucción.
Any	• Realiza una comparación. • Si es cierta, se completa toda la instrucción.
In	• Verifica el valor si se encuentra en el resultado de la subconsulta.
Not In	• Verifica un valor en específico si no se encuentra en la subconsulta.

Actividad

Con base en lo descrito en el tema, reflexiona sobre las siguientes preguntas:

01 ¿El uso de subconsultas afecta el rendimiento de la base de datos?

02 Menciona un ejemplo donde se puedan aplicar subconsultas de manera optimizada.

Cierre

Las bases de datos en lenguaje SQL hacen uso de tablas y de lógica relacional, por lo que es necesario conocer técnicas como las subconsultas o JOINS para poder explicar al máximo la estructura de las bases de datos y del lenguaje.

Por último, también es necesario recordar que las subconsultas tienen reglas y condiciones que se deben cumplir para lograr la extracción de datos deseada.

Bibliografía

• AWS. (2022). *Subconsultas escalares*. Recuperada de https://docs.aws.amazon.com/es_es/redshift/latest/dg/r_scalar_subqueries.html

• Microsoft. (2022). *Subqueries (SQL Server)*. Recuperada de https://docs.microsoft.com/en-us/sql/relational-databases/performance/subqueries?view=sql-server-ver15

Universidad Tecmilenio

Consulta en Microsoft SQL Server®

Funciones integradas

Semana 3

Introducción

Las funciones integradas forman parte del lenguaje SQL y T-SQL y son ampliamente utilizadas para procesos de naturaleza numérica.

Asimismo, forman parte de SQL Server y se usan para instrucciones en Transact-SQL.

Estas funciones regresan diferentes tipos de datos, según el tipo de función que se utilice.

Existe la posibilidad de crear funciones definidas por el usuario, es decir, crear funciones hechas a la medida para realizar un procedimiento específico.

Universidad Tecmilenio

Explicación

Introducción a las funciones integradas

Las funciones integradas ya vienen por *default* en SQL Server, no se pueden modificar y se utilizan para diversas operaciones. Hay tres tipos de funciones, son las únicas que están preestablecidas en el sistema y no se pueden alterar.

- AVG ()
- SUM ()
- COUNT ()
- DATETIME ()
- CONCAT ()
- LOWER ()
- UPPER ()
- MAX ()
- MIN ()

Funciones integradas enfocadas al tiempo

```
SELECT GETDATE() as FechaActual;
```

Results | Messages

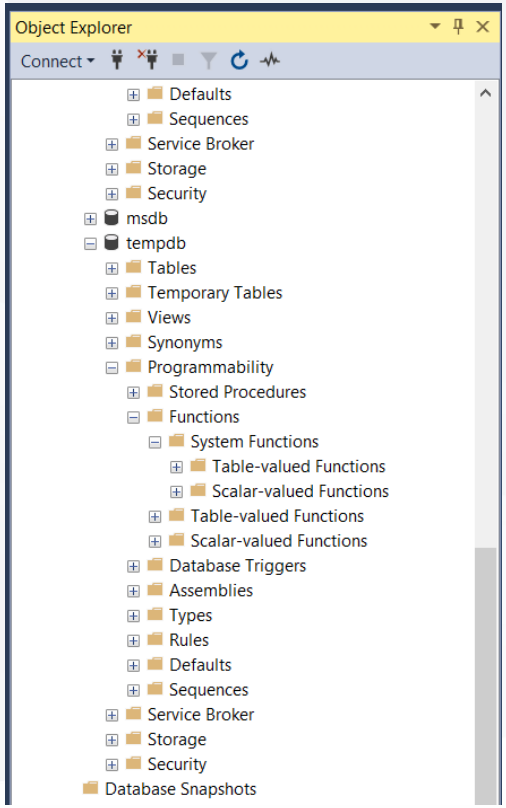
FechaActual
2022-05-15 16:54:12.327

- GETDATE()
- DATEPART()
- DATEADD()
- DATEDIFF()

Explicación

Pasos para encontrar las funciones integradas:

Para obtener una fecha como resultado, pero en un formato en específico, hay que crear una función que arroje la fecha y el tiempo en el formato deseado, utilizando el siguiente código:




```
CREATE FUNCTION FormatoFecha
(@ValorFecha as DATETIME)
RETURNS Varchar (MAX)
AS
BEGIN
RETURN
DATENAME (DiaSemana, @ValorFecha) + ', '+
DATENAME (Dia, @ValorFecha) + ' '+
DATENAME (Mes, @ValorFecha) +', '+
DATENAME (Anio, @ValorFecha)
END
```

Universidad Tecmilenio

Funciones escalares

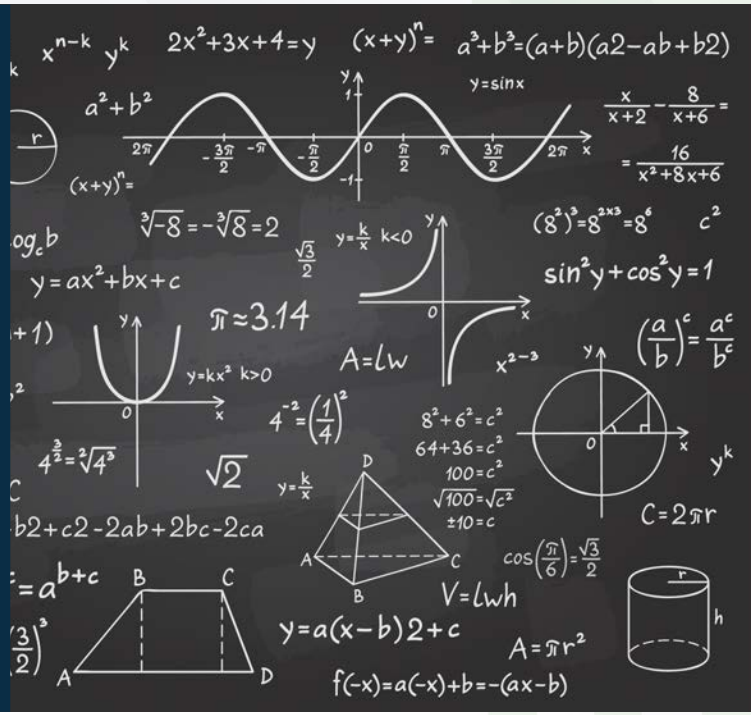
De acuerdo con Microsoft (2022a), una función escalar tiene que devolver un valor único y, para esto, se tiene que definir el tipo de valor que regresará la función.

Explicación

Por otro lado, se le conoce como ”determinista” a la función, y devuelve el mismo valor para el mismo estado de entrada y base de datos, es decir, que opere sobre un mismo valor y devuelva otro mismo valor.

Un ejemplo de esto sería utilizar la función integrada ROUND, la cual es intrínseca de T-SQL y se utiliza para redondear un valor.

ROUND (2.1, 0)



Explicación

Funciones con valor de tabla

Esta función también es definida por el usuario y lo que busca es regresar una tabla como valor de respuesta. En este caso, la tabla es el conjunto de resultados de una sola instrucción SELECT (Microsoft, 2022b).

```
Create Function HorarioNoDisp(@nodisp int)
 Returns Table
 AS
 Return (Select * From Horarios Where nodisponible=@nodisp)
 Select * From dbo.ProdSuspendido(1)
```

Actividad

Con base en lo descrito en el tema, reflexiona sobre las siguientes preguntas:

01 ¿Cuál es la principal razón por la que un programador T-SQL decide crear una función que regrese valor de tabla?

02 ¿Cuál es un buen ejemplo de aplicación de una función de usuario en un *e-commerce*?

Cierre

Las funciones se utilizan ampliamente en la creación de consultas para extracciones de datos.

Trabajar en un ambiente de datos, o bien, de bases de datos, nos exige estar preparados y conocer las funciones y herramientas necesarias para manejar estos datos que, en su gran mayoría, resultan ser numéricos, por lo que es de suma importancia reconocer todos los diferentes tipos de funciones y aplicarlos para obtener el resultado deseado.

Bibliografía

• Microsoft. (2022a). *Funciones escalares de ODBC (Transact-SQL).* Recuperado de https://docs.microsoft.com/es-es/sql/t-sql/functions/odbc-scalar-functions-transact-sql?view=sql-server-ver15

• Microsoft. (2022b). *Funciones con valores de tabla en CLR.* Recuperado de https://docs.microsoft.com/es-es/sql/relational-databases/clr-integration-database-objects-user-defined-functions/clr-table-valued-functions?view=sql-server-ver15

Universidad Tecmilenio

Consulta en Microsoft SQL Server®

Manipulación de datos

Semana 3

Un **analista o científico de datos** a menudo **manipula valores** para varios fines, por ejemplo, los ingenieros en calidad de software hacen pruebas para verificar que las tablas de una base de datos estén almacenando los datos de manera correcta, o bien, usando valores correctamente.

Es necesario **emplear** ciertas **técnicas** y cláusulas que ya forman parte del lenguaje T-SQL **para alterar datos e incluso borrarlos**, y poder continuar con las pruebas. Asimismo, es esencial aplicar técnicas para combinar datos entre varias tablas.

Explicación

Actualización de datos

El comando UPDATE cambia datos existentes de una tabla por nuevos datos que sean definidos dentro de la consulta (Microsoft, 2022a).

```
UPDATE TABLA
SET COLUMNA = 'NUEVO VALOR'
WHERE CONDICION (OTRA COLUMNA)
```

- UPDATE: va a seguida de la tabla que se desea actualizar.
- SET: esta sentencia va a definir específicamente cuáles serán los valores que se actualizarán.
- WHERE: va al final de la sentencia para indicar filtrado de datos.

Explicación

Eliminación de datos

El proceso y la sintaxis son similares a las del UPDATE, pero existen tres sentencias dentro de la eliminación de datos:

Sentencia DELETE:

- Van las dos sentencias juntas: DELETE y FROM.
- La tabla va definida después del FROM.

Sentencia DROP:

- Se utiliza para borrar toda la tabla.
- Borra todo (y los datos que se incluyan).

Sentencia TRUNCATE:

- Se usa de manera intercambiable con la de DELETE sin condición.
- Borra todos los valores de la tabla, sin borrar la tabla.

Explicación

Valores automáticos

IDENTITY

T-SQL puede generar valores automáticos para llenar espacios vacíos en columnas, es decir, aquellas columnas que son necesarias de rellenar en forma automática se pueden completar usando dos cláusulas diferentes (Microsoft, 2022b).

SEQUENCE

Esta propiedad necesita de un objeto para crear una secuencia, y es necesario para que se haga un incremento.

Explicación

Combinación de datos de varias tablas

El tema anteriormente se trató de la cláusula JOIN, la cual, por naturaleza, crea combinaciones entre tablas para poder tener mejor visión de alguna relación en específico.

Combinaciones

T-SQL también cuenta con la cláusula **PIVOT**, que es similar al software MS Excel. Esta cláusula hace un transpuesto de los valores de una columna a nuevas columnas en otra tabla.

Actividad

Con base en lo descrito en el tema, reflexiona sobre las siguientes preguntas:

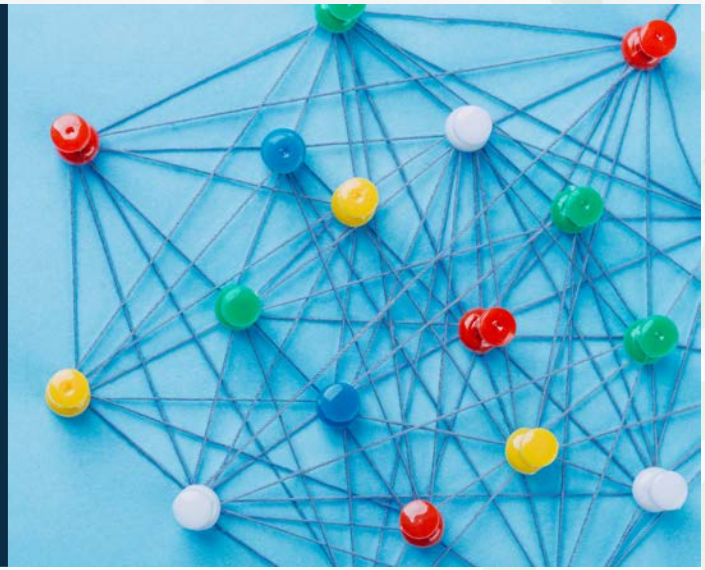
01 ¿Hay ventajas en definir un IDENTITY a un campo de tipo llave primaria?

02 ¿De qué manera se pueden evitar afectaciones en los datos antes de realizar un UPDATE o DELETE?

Cierre

Es importante señalar que, a pesar de que el lenguaje T-SQL por naturaleza es estructurado, también le permite al usuario manipular los datos de diferentes maneras. Por un lado, se manipulan directamente alterándolos, o bien, eliminándolos por completo.

El lenguaje T-SQL no solo nos permite modificar los datos de los registros, sino también combinar, ordenar y mover columnas y registros para una mayor interpretación.



Bibliografía

• Microsoft. (2022a). *UPDATE (Transact-SQL)*. Recuperado de https://docs.microsoft.com/en-us/sql/t-sql/queries/update-transact-sql?view=sql-server-ver15

• Microsoft. (2022b). *CREATE TABLE (Transact-SQL) IDENTITY (Property)*. Recuperado de https://docs.microsoft.com/en-us/sql/t-sql/statements/create-table-transact-sql-identity-property?view=sql-server-ver16

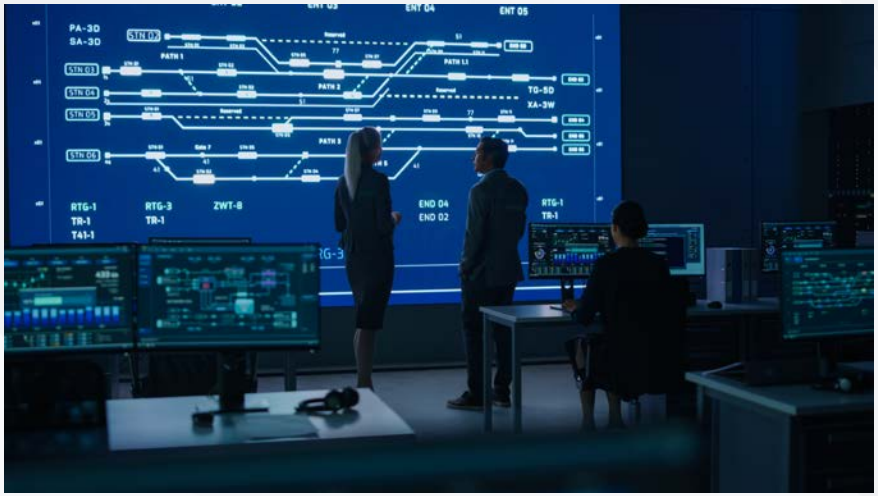
Universidad Tecmilenio

Consulta en Microsoft SQL Server®

Introducción a la programación con T-SQL

Semana 3

Introducción



Transact-SQL (T-SQL) es un lenguaje de programación con bases sólidas que permiten la creación de procesos que demandan tareas complejas sobre una base de datos.

Utilizar variables y bloques para controlar el flujo del programa será fundamental, a fin de implementar lógica y reglas de negocio necesarias con la intención de manipular la información y obtener el resultado deseado.

Explicación

Descripción de T-SQL

“Las variables tienen un nombre (la palabra que se usa para hacer referencia al valor contenido por la variable). Las variables también tienen un tipo de datos que determina el tipo de datos que puede almacenar la variable” (Microsoft, 2022a).

¿Qué es una variable?

Una variable suele funcionar como una caja etiquetada, por ejemplo, “documentos”, “juguetes” o “utensilios de cocina”, de tal manera que cuando exista un juguete nuevo, se pueda guardar en la caja con la etiqueta “juguetes”.

Universidad Tecmilenio

Explicación

Declaración de variables

La instrucción **DECLARE** seguida de un nombre definido por el programador, deberá iniciar con el símbolo de arroba @, sin espacios y que tenga sentido con lo que se vaya a almacenar para una mejor identificación en futuras referencias. No se puede repetir el nombre de una variable en un mismo segmento de código, y después se coloca AS con el tipo de dato que va a contener.

```
--Declarar e inicializar variables.
DECLARE @numero1     AS INT = 10
       ,@numero2     AS INT = 5
       ,@resultado   AS INT = 0;

--Operación con variables.
SET @resultado = @numero1 + @numero2;

--Mostrar el resultado de la suma de dos números enteros.
SELECT @resultado AS 'Resultado'; --Resultado de la suma de dos números enteros.
```

Explicación

Bloques para controlar el flujo del programa

¿Qué es el control de flujo?

Es el rumbo que puede tomar un programa de manera natural, se ejecuta línea por línea, por ejemplo, antes de que se ejecute la línea 10, deberá pasar por nueve líneas, pero no es obligatorio que se ejecuten todas para llegar a una en específico.

Los comandos que se usarán son los siguientes:

Universidad Tecmilenio

Explicación

IF...ELSE

Es importante saber emplear condiciones; el resultado de una condición puede ser verdadero (TRUE) o falso (FALSE) y, de acuerdo con ese resultado, el flujo del programa deberá continuar, como se expresa en el siguiente ejemplo:

```
--Declarar variable.
DECLARE @productoID  AS INT = NULL;

--Evaluar si el código de producto es nulo
IF @productoID IS NULL
BEGIN
	PRINT 'El producto es nulo.'
END
ELSE
BEGIN
	PRINT 'El producto no es nulo, contiene ' + CAST(@productoID AS VARCHAR(10))
END
```

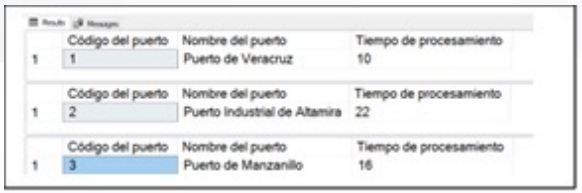
Explicación

WHILE

Establece una condición para la ejecución repetida de una sentencia SQL o un bloque de sentencias. Las declaraciones se ejecutan repetidamente siempre que la condición especificada sea verdadera.

Se usa para ejecutar una o más líneas de código en un ciclo basado en condiciones, es decir, se ejecutará el bloque de código tantas veces se cumplan las condiciones (Microsoft, 2022b).

```
DECLARE @puertoID                       AS INT
              ,@puertoNombre            AS VARCHAR(50)
              ,@puertoTiempoProceso     AS INT
              ,@puertoConteo            AS INT;

--Inicializar variables.
SET    @puertoID = 1;
SELECT @puertoConteo = COUNT([PuertoID]) FROM [dbo].[Puertos];

--Ejecutar bloque de código siempre que la condición se cumpla.
WHILE @puertoID <= @puertoConteo
BEGIN

       SELECT @puertoNombre = [PuertoNombre], @puertoTiempoProceso =
[PuertoTiempoProceso]
       FROM [dbo].[Puertos]
       WHERE [PuertoID] = @puertoID;

       SELECT @puertoID                 AS 'Código del puerto'
              ,@puertoNombre            AS 'Nombre del puerto'
              ,@puertoTiempoProceso     AS 'Tiempo de procesamiento';

       SET @puertoID += 1;
END
```



	Código del puerto	Nombre del puerto	Tiempo de procesamiento
1	1	Puerto de Veracruz	10

	Código del puerto	Nombre del puerto	Tiempo de procesamiento
1	2	Puerto Industrial de Altamira	22

	Código del puerto	Nombre del puerto	Tiempo de procesamiento
1	3	Puerto de Manzanillo	16

Universidad Tecmilenio

Explicación

BREAK

	Código del puerto	Nombre del puerto	Tiempo de procesamiento
1	1	Puerto de Veracruz	10

(1 row affected)
No se puede continuar con el proceso, la parametrización supera al tiempo permitido.

Completion time: 2022-04-19T12:24:43.4139788-05:00

```
DECLARE @puertoID                   AS INT
       ,@puertoNombre               AS VARCHAR(50)
       ,@puertoTiempoProceso        AS INT
       ,@puertoConteo               AS INT;

--Inicializar variables.
SET    @puertoID = 1;
SELECT @puertoConteo = COUNT([PuertoID]) FROM [dbo].[Puertos];

--Ejecución de código hasta que la condición no se cumpla.
WHILE @puertoID <= @puertoConteo
BEGIN

       SELECT @puertoNombre = [PuertoNombre], @puertoTiempoProceso = [PuertoTiempoProceso]
       FROM [dbo].[Puertos]
       WHERE [PuertoID] = @puertoID;

       IF @puertoTiempoProceso >= 20
       BEGIN
              PRINT 'No se puede continuar con el proceso, la parametrización supera al tiempo
permitido.'
              BREAK;
       END

       SELECT @puertoID                    AS 'Código del puerto'
              , @puertoNombre              AS 'Nombre del puerto'
              ,@puertoTiempoProceso        AS 'Tiempo de procesamiento';

       SET @puertoID += 1;
END
```

"Sentencia para salir inmediatamente de un bucle WHILE", y en el siguiente *script* se aprecia mejor la manera de interpretarlo (Microsoft, 2022b).

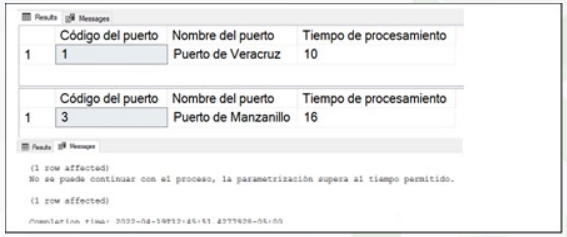
Universidad Tecmilenio

Explicación

CONTINUE

- “La sentencia CONTINUE detiene la iteración actual del ciclo y comienza la nueva” (SQLServerTutorial.net, s.f.).
- Permitirá continuar con las instrucciones.
- Puede emplearse para notificar que se encontró con una parametrización errónea.

```
--Declara variables.
DECLARE @puertoID                       AS INT
              ,@puertoNombre        AS VARCHAR(50)
              ,@puertoTiempoProceso AS INT
              ,@puertoConteo        AS INT;

--Inicializar variables.
SET    @puertoID = 1;
SELECT @puertoConteo = COUNT([PuertoID]) FROM [dbo].[Puertos];

--Ejecución de código hasta que la condición no se cumpla.
WHILE @puertoID <= @puertoConteo
BEGIN

       SELECT @puertoNombre = [PuertoNombre], @puertoTiempoProceso = [PuertoTiempoProceso]
       FROM [dbo].[Puertos]
       WHERE [PuertoID] = @puertoID;

       IF @puertoTiempoProceso >= 20
       BEGIN
              PRINT 'No se puede continuar con el proceso, la parametrización supera al tiempo permitido.'
              SET @puertoID += 1;
              CONTINUE;
       END

       SELECT @puertoID                    AS 'Código del puerto'
              ,@puertoNombre                      AS 'Nombre del puerto'
              ,@puertoTiempoProceso       AS 'Tiempo de procesamiento';

       SET @puertoID += 1;
END
```



	Código del puerto	Nombre del puerto	Tiempo de procesamiento
1	1	Puerto de Veracruz	10

	Código del puerto	Nombre del puerto	Tiempo de procesamiento
1	3	Puerto de Manzanillo	16

(1 row affected)
No se puede continuar con el proceso, la parametrización supera al tiempo permitido.

(1 row affected)

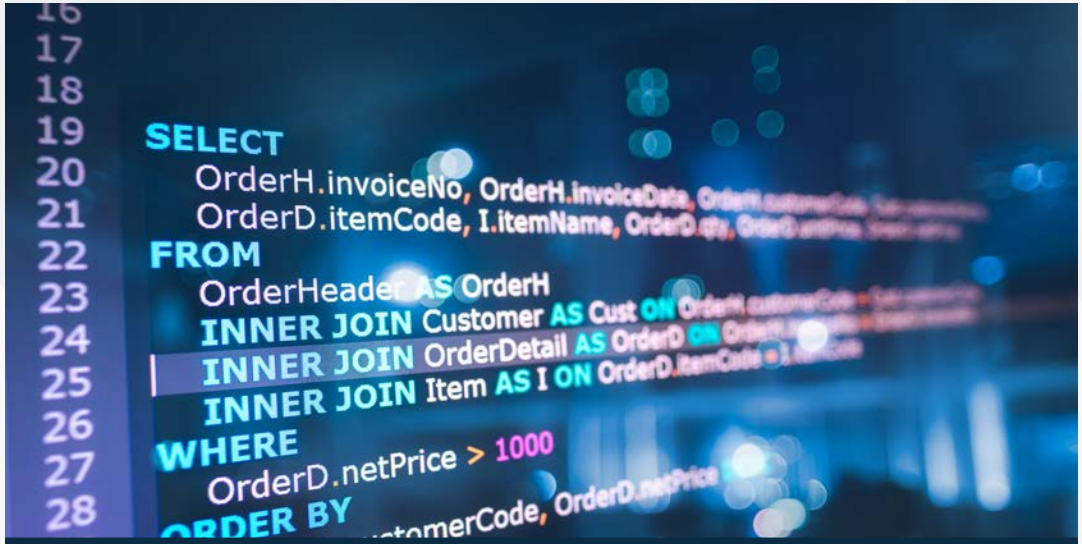
Actividad

Con base en lo descrito en el tema, reflexiona sobre las siguientes preguntas:

01 ¿De las aplicaciones que suelen tener los celulares en la actualidad, ¿cuál es la que usa más variables de tipo número?

02 ¿Se puede generar un ciclo infinito en SQL?, ¿qué consecuencias se tendrían?

Universidad Tecmilenio



Como puedes ver, los comandos extendidos de T-SQL son tan útiles y flexibles que te permitirán controlar procesos laboriosos. Será sencillo aplicar instrucciones con base en la demanda que se tenga por parte de un tercero.

Asimismo, recuerda hacer un buen uso de las variables, además de contemplar la mayoría de las posibilidades en un flujo del programa y, si se desea repetir una tarea, aplicar un bucle de manera optimizada que no tenga fin.

Bibliografía

• Microsoft. (2022a). *Variables en Visual Basic.* Recuperado de https://docs.microsoft.com/es-es/dotnet/visual-basic/programming-guide/language-features/variables/

• Microsoft. (2022b). *WHILE (Transact-SQL)*. Recuperado de https://docs.microsoft.com/en-us/sql/t-sql/language-elements/while-transact-sql?view=sql-server-ver15

• SQLServerTutorial.net. (s.f.). *SQL Server CONTINUE*. Recuperado de https://www.sqlservertutorial.net/sql-server-stored-procedures/sql-server-continue/