



Introducción a Transact-SQL

En este capítulo veremos los elementos que forman el lenguaje Transact-SQL, la implementación de Microsoft para el estándar ANSI SQL.

Esta página se ha dejado en blanco intencionalmente.

Capítulo 6

Introducción a Transact-SQL

Contenido

- *Elementos de Transact-SQL*
 - ✓ *Declaraciones del Data Manipulation Language (DML)*
 - ✓ **Ejercicio 39:** *Declaración DML*
 - ✓ *Declaraciones del Data Definition Language (DDL)*
 - ✓ **Ejercicio 40:** *Declaración DDL*
 - ✓ *Declaraciones del Data Control Language (DCL)*
 - ✓ **Ejercicio 41:** *Declaración DCL*
 - ✓ *Variables*
 - *Variables locales (las instrucciones DECLARE y SET)*
 - **Ejercicio 42:** *Uso de variable local*
 - *Variables globales*
 - **Ejercicio 43:** *Uso de variable global*
 - ✓ *Operadores*
 - *Operadores aritméticos*
 - *Operador de concatenación*
 - *Operadores de fecha*
 - *Operadores de comparación ó relacionales*
 - *Operadores lógicos*
 - *Precedencia de los operadores*
 - **Ejercicio 44:** *Uso de operadores*

Esta página se ha dejado en blanco intencionalmente.

Introducción a Transact-SQL

Transact-SQL es la implementación Microsoft del estándar ANSI SQL que define elementos del lenguaje SQL que pueden ejecutarse desde cualquier aplicación frontal. Además de los elementos del estándar ANSI SQL, Transact-SQL tiene elementos del lenguaje que son exclusivos de él conocidos como **extensiones Transact-SQL**. Estos, son mejoras a algunas instrucciones del estándar ó añadidos que proporcionan nuevas funcionalidades.

Elementos de Transact-SQL

Al escribir y ejecutar sentencias Transact-SQL usará:

- Declaraciones del **Data Manipulation Language (DML)** que se usan para consultar y modificar los datos.
- Declaraciones del **Data Definition Language (DDL)** que se usan para crear los objetos en la base de datos.
- Declaraciones del **Data Control Language (DCL)** que se utilizan para determinar quién ve o modifica los datos.
- Elementos adicionales del language como: variables, operadores, funciones, sentencias de control de flujo, y comentarios.

Declaraciones del Data Manipulation Language (DML)

Las declaraciones DML trabajan con los datos de la base de datos e incluyen a las siguientes instrucciones:

Instrucción	Descripción
SELECT	Selecciona filas y columnas de una ó más tablas de la base de datos.
INSERT	Añade una nueva fila a una tabla.
UPDATE	Modifica los datos existentes en una tabla.
DELETE	Elimina filas de una tabla.

Ejercicio 39: Declaración DML

La siguiente instrucción lee y muestra las columnas **idAlumno**, **nombreAlumno** y **direccion** de la tabla **Alumno** de la base de datos **ExtPro**.

```
USE ExtPro
go

SELECT idAlumno, nombreAlumno, direccion
FROM Alumno
go
```

Declaraciones del Data Definition Language (DDL)

Las declaraciones del DDL permiten crear bases de datos, tablas, y tipos de datos definidos por usuarios. También se usan para manejar los objetos de la base de datos.

Algunas declaraciones de DDL son:

Instrucción	Descripción
CREATE <i>nombre_objeto</i>	Crea un objeto de la base de datos.
ALTER <i>nombre_objeto</i>	Modifica la definición de un objeto de la base de datos.
DROP <i>nombre_objeto</i>	Elimina un objeto de la base de datos.

Ejercicio 40: Declaración DDL

El siguiente ejercicio crea una tabla de nombre **Laboratorio** en la base de datos **ExtPro**. La tabla está formada por las columnas **idLaboratorio**, **local**, **numMaquinas**.

```
USE ExtPro
go

CREATE TABLE Laboratorio(
    idLaboratorio char(4) PRIMARY KEY,
    local int not null,
    numMaquinas int null)
go
```

Declaraciones del Data Control Language (DCL)

Las declaraciones del DCL se usan para cambiar los permisos o roles asociados con un usuario de la base de datos. La tabla siguiente describe las declaraciones de DCL.

Instrucción	Descripción
GRANT	Crea una entrada en la seguridad del sistema que le permite a un usuario trabajar con los datos o ejecutar ciertas sentencias Transact-SQL.
DENY	Crea una entrada en la seguridad del sistema negando un permiso de una cuenta de seguridad y evita que el usuario, grupo, o rol herede el permiso a través de su grupo y rol.
REVOKE	Quita un permiso previamente concedido o negado.

Ejercicio 41: Declaración DCL

Este ejemplo concede permiso para leer la tabla **Curso** de la base de datos **ExtPro** al rol ó función **public** de dicha base de datos.

```
USE ExtPro
go

GRANT SELECT ON Curso TO public
go
```


Variables

Las variables son los elementos del lenguaje que se definen en la memoria del sistema a los que se puede asignar valores para luego usarlos posteriormente. Pueden ser locales o globales.

Variables locales (las instrucciones DECLARE y SET)

Una variable local es creada por el usuario (el programador) y se define en una sentencia DECLARE, se le asigna un valor inicial en una declaración SET, y se utiliza dentro de la declaración, batch, o procedimiento en la que fue declarada. Una variable local se identifica con un símbolo @ que precede a su nombre.

Sintaxis

```
DECLARE @nombre_variable tipo_dato , ...
```

```
SET @nombre_variable = expresión
```

Ejercicio 42: Uso de variable local

Este ejercicio declara una variable local **@curso**, le asigna un valor, y consulta la tabla **Curso** para buscar las filas que contengan el valor de la variable en la columna **idCurso**.

```
USE ExtPro
go

DECLARE @curso char(4)
SET @curso = 'PB1A'

SELECT nombreCurso, profesor
FROM Curso
WHERE idCurso = @curso
go
```

Variables globales

Las variables globales son predefinidas y mantenidas por SQL Server. El usuario no puede asignar o cambiar directamente los valores de las variables globales. Muchas de las variables globales reportan al sistema la actividad que ha tenido lugar desde la última vez que SQL Server fue iniciado, otras reportan información sobre una conexión. Una variable global es identificada con dos símbolos @ precediendo su nombre.

Las variables globales son útiles para verificar la seguridad o condiciones del entorno actual de SQL Server. Por ejemplo, si quiere verificar el número de versión de SQL Server, puede realizar la consulta a la variable global @@version. La consulta devuelve el número de la versión del SQL Server que está ejecutando.

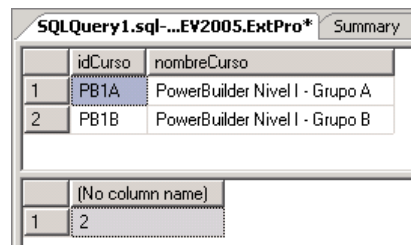
Ejercicio 43: Uso de variable global

Este ejercicio utiliza la variable global @@rowcount para averiguar el número de filas recuperadas por la instrucción SELECT.

La variable global @@rowcount devuelve el número de filas afectadas por la última sentencia SQL ejecutada.

```
SELECT idCurso, nombreCurso FROM Curso
      WHERE LEFT(idCurso, 2) = 'PB'
```

```
SELECT @@rowcount
go
```



SQLQuery1.sql - ...EY2005.ExtPro*		Summary
	idCurso	nombreCurso
1	PB1A	PowerBuilder Nivel I - Grupo A
2	PB1B	PowerBuilder Nivel I - Grupo B

	(No column name)
1	2

Operadores

Los operadores son símbolos que indican la ejecución de operaciones matemáticas, encadenamientos de cadenas, y comparaciones entre las columnas, constantes y variables.

Los operadores pueden combinarse y utilizarse en las expresiones de búsqueda. Cuando los combina, el orden en que los operadores se procesan se basa en una jerarquía predefinida.

Operadores aritméticos

Los operadores aritméticos realizan cálculos con las columnas numéricas o las constantes.

Operador	Propósito	Ejemplo
+ -	Operadores unarios: utilizados para representar valores numéricos positivos y negativos. Para valores positivos, el símbolo + es opcional.	-347.25
+	Suma: usado para sumar dos valores ó expresiones numéricas.	37 + 45
-	Resta: usado para obtener la diferencia entre dos valores ó expresiones numéricas.	77.1 – 56
*	Producto: usado para multiplicar dos valores ó expresiones numéricas.	56 * 7
/	División: usado para dividir dos valores ó expresiones numéricas.	76.17 / 13
%	Módulo: usado para obtener el resto ó residuo de la división entre dos valores ó expresiones numéricas enteras	75 % 4

Operador de concatenación

El operador de concatenación de cadenas (+) encadena los valores de tipo cadena.

Ejemplo	Resultado
'PERU' + 'Hermoso y Pujante'	'PERUHermoso y Pujante'
'PERU ' + 'Hermoso y Pujante'	'PERU Hermoso y Pujante'

Operadores de fecha

Los operadores + y – también se pueden aplicar a los datos de tipo fecha-hora.

Operador	Resultado
<i>fecha</i> + A	Devuelve una nueva fecha añadiéndole A días a <i>fecha</i> .
<i>fecha</i> – A	Devuelve una nueva fecha quitándole A días a <i>fecha</i> .

Operadores de comparación ó relacionales

Estos operadores comparan dos expresiones. Las comparaciones pueden realizarse entre las variables, columnas, y expresiones de tipo similar. Los operadores de comparación incluyen a: menor que (<), mayor que (>), igual a (=), y todas las combinaciones posibles.

Operador	Resultado
Na < Nb Fa < Fb Ca < Cb	Devuelve TRUE si: el valor numérico Na es menor que el valor numérico Nb; la fecha Fa es anterior a la fecha Fb, la cadena Ca está antes que la cadena Cb cuando las cadenas se ordenan.
Na > Nb Fa > Fb Ca > Cb	Devuelve TRUE si el valor a la derecha del operador es mayor (posterior) al valor de la derecha del operador.
Na = Nb Fa = Fb Ca = Cb	Devuelve TRUE si los valores comparados son iguales.
Na <= Nb Fa <= Fb Ca <= Cb	Devuelve TRUE si el valor a la izquierda del operador es menor ó igual (anterior ó igual) al valor de la derecha del operador.
Na >= Nb Fa >= Fb Ca >= Cb	Devuelve TRUE si el valor a la izquierda del operador es mayor ó igual (posterior ó igual) al valor a la derecha del operador.
Na <> Nb Fa <> Fb Ca <> Cb	Devuelve TRUE si los valores comparados son distintos.

Operadores lógicos

Los operadores lógicos AND, OR y NOT se utilizan para construir expresiones lógicas (expresiones cuyo resultado es verdadero o falso), y se utilizan mucho en las operaciones de búsqueda.

Operador	Resultado
NOT A	Devuelve TRUE si A es FALSE, ó devuelve FALSE si A es TRUE.
A AND B	Devuelve TRUE solo si las expresiones lógicas A y B son ambas TRUE. En cualquier otro caso devuelve FALSE.
A OR B	Devuelve FALSE solo si las expresiones lógicas A y B son ambas FALSE. En cualquier otro caso devuelve TRUE.

Precedencia de los operadores

Cuando una expresión compleja tiene muchos operadores, la precedencia de los operadores determina la secuencia en la que las operaciones se ejecutan. El orden de ejecución de las operaciones puede afectar el resultado de la expresión.

- Operadores unarios: + (positivo), - (negativo)
- * (multiplicación), / (división), % (módulo)
- + (suma), (+ concatenación), - (resta)
- =, >, <, >=, <=, <>, !=, !>, !< (operadores de comparación)
- NOT
- AND
- OR
- = (asignación)

Ejercicio 44: Uso de operadores

Este ejercicio muestra el uso de algunos operadores disponibles en Transact-SQL.

```
-- operadores aritméticos
SELECT 45 + 56
go

SELECT 78.1 - 15
go

SELECT 45 * 13
go

SELECT 78.25 / 13
go

SELECT 75 % 12
go

-- Operador de concatenación
SELECT 'PERU' + 'Hermoso y Pujante'
go

SELECT 'PERU ' + 'Hermoso y Pujante'
go
```

Esta página se ha dejado en blanco intencionalmente.