

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**DISEÑO E IMPLEMENTACIÓN
DE BASE DE DATOS**

Tema

**CREACIÓN DE BASE DE DATOS EN SQL
SERVER**

1. CREACIÓN DE BASE DE DATOS EN SQL SERVER

Como muchas instrucciones Transact-SQL, la instrucción [CREATE DATABASE](#) tiene un parámetro obligatorio: el nombre de la base de datos. CREATE DATABASE también tiene muchos parámetros opcionales, como la ubicación de disco donde se quieren colocar los archivos de la base de datos. Si se ejecuta CREATE DATABASE sin los parámetros opcionales, SQL Server usa los valores predeterminados para muchos de estos parámetros.

1. En una ventana del Editor de consultas, escriba el código siguiente, pero no lo ejecute:

```
CREATE DATABASE TestData  
GO
```

2. Use el puntero para seleccionar las palabras CREATE DATABASEy, a continuación, presione **F1**. Debería abrirse el artículo CREATE DATABASE. Puede usar esta técnica para buscar la sintaxis completa de CREATE DATABASE y de otras instrucciones que se usan.
3. En el Editor de consultas, presione **F5** para ejecutar la instrucción y crear una base de datos con el nombre *TestData*.

Al crear una base de datos, SQL Server realiza una copia de la base de datos *model* y cambia el nombre de la copia por el nombre de la base de datos. Esta operación solo debería tardar algunos segundos, a menos que especifique un tamaño inicial grande de la base de datos como un parámetro opcional.

Nota

La palabra clave *GO* separa las instrucciones cuando se envían varias instrucciones en un solo lote. *GO* es opcional cuando el lote solo contiene una instrucción.

2. CREACIÓN DE LA TABLA

En el mismo script donde creo la Base de datos escriba y ejecute el código siguiente para crear una tabla denominada *Products*. Las columnas de la tabla son *ProductID*, *ProductName*, *Price* y *ProductDescription*. La columna *ProductID* es la clave principal de la tabla. *int*, *varchar(25)*, *money* y *varchar(max)* son todos los tipos de datos. Solo las columnas *Price* y *ProductDescription* pueden no tener datos cuando se inserta o cambia una fila. Esta instrucción contiene un elemento opcional (dbo.) denominado esquema. El esquema es el objeto de base de datos propietario de la tabla. Si es un administrador, dbo es el esquema predeterminado. dbo hace referencia al propietario de la base de datos.

```
CREATE TABLE dbo.Products
(
    ProductID int PRIMARY KEY NOT NULL,
    ProductName varchar(25) NOT NULL,
    Price money NULL,
    ProductDescription varchar(max) NULL
)
GO
```

3. INSERCIÓN Y ACTUALIZACIÓN DE DATOS DE UNA TABLA

Ahora que ha creado la tabla *Products*, ya está listo para insertar datos en la tabla mediante la instrucción INSERT. Después de insertar los datos, cambiará el contenido de una fila con una instrucción UPDATE. Usa la cláusula WHERE de la instrucción UPDATE para restringir la actualización a una sola fila. Las cuatro instrucciones introducen los datos siguientes.

ProductID	ProductName	Precio	ProductDescription
1	Clamp	12,48	Workbench clamp
50	Screwdriver	3,17	Flat head
75	Tire Bar		Tool for changing tires.
3000	Corchete de 3 mm	0,52	

La sintaxis básica es: INSERT, nombre de tabla, lista de columnas, VALUES y, a continuación, una lista de los valores que se van a insertar. Los dos guiones delante de una línea indican que la línea es un comentario y el compilador ignora el texto. En este caso, el comentario describe una variación permitida de la sintaxis.

4. INSERCIÓN DE DATOS EN UNA TABLA

- Ejecute la instrucción siguiente para insertar una fila en la tabla *Products* que se ha creado en la tarea anterior debajo de todo el script.

```
-- Standard syntax
INSERT dbo.Products (ProductID, ProductName, Price, ProductDescription)
    VALUES (1, 'Clamp', 12.48, 'Workbench clamp')
GO
```

- La instrucción siguiente muestra cómo se puede cambiar el orden en que se proporcionan los parámetros modificando la situación de *ProductID* y *ProductName* en la lista de campos (entre paréntesis) y en la lista de valores.

```
-- Changing the order of the columns
INSERT dbo.Products (ProductName, ProductID, Price, ProductDescription)
    VALUES ('Screwdriver', 50, 3.17, 'Flat head')
GO
```

- La instrucción siguiente demuestra que los nombres de las columnas son opcionales, siempre y cuando los valores se enumeren en el orden correcto. Esta sintaxis es habitual, pero no se recomienda porque podría ser difícil para otros comprender su código. NULL se especifica para la columna Price porque el precio de este producto no se conoce todavía.

```
-- Skipping the column list, but keeping the values in order
INSERT dbo.Products
    VALUES (75, 'Tire Bar', NULL, 'Tool for changing tires.')
GO
```

- d. El nombre de esquema es opcional mientras tenga acceso a una tabla del esquema predeterminado y la modifique. Puesto que la columna *ProductDescription* permite valores *NULL* y no se ha proporcionado ningún valor, el nombre de columna y el valor de *ProductDescription* se pueden quitar por completo de la instrucción.

```
-- Dropping the optional dbo and dropping the ProductDescription column
INSERT Products (ProductID, ProductName, Price)
    VALUES (3000, '3 mm Bracket', 0.52)
GO
```

5. ACTUALIZACIÓN DE LA TABLA DE PRODUCTOS

Escriba y ejecute la siguiente instrucción UPDATE para cambiar el *ProductName* del segundo producto de Screwdriver a Flat Head Screwdriver.

```
UPDATE dbo.Products
    SET ProductName = 'Flat Head Screwdriver'
    WHERE ProductID = 50
GO
```

Bibliografía:

<https://learn.microsoft.com/es-es/sql/t-sql/lesson-1-creating-database-objects?view=sql-server-ver16>

