

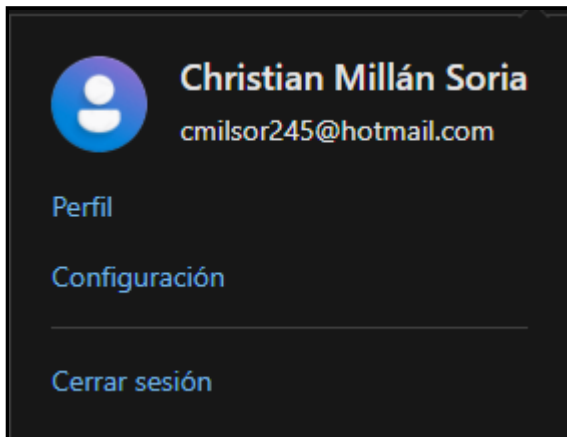
# SEGUNDO CURSO

CHRISTIAN MILLÁN SORIA

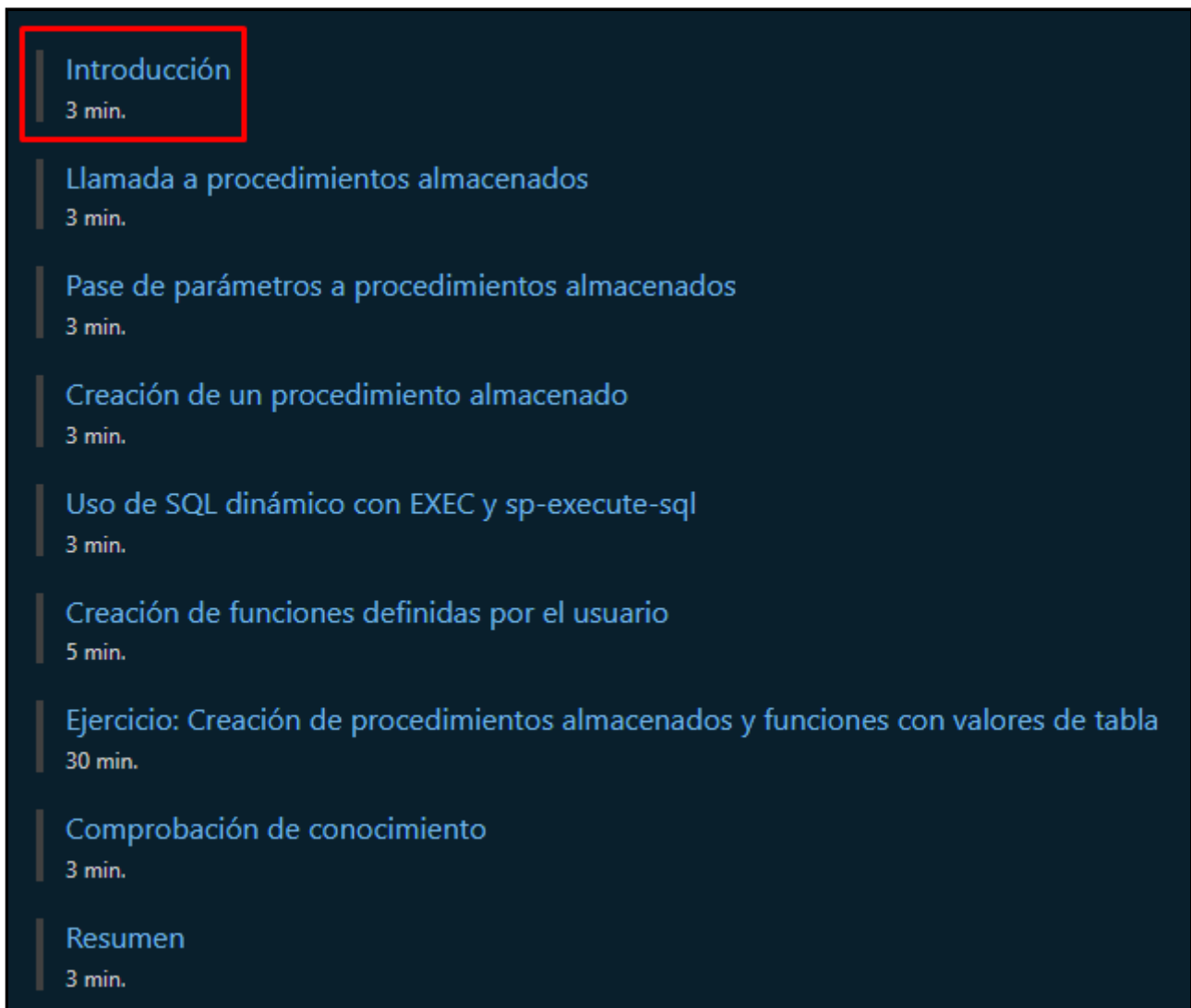
---

Comenzamos entrando en [este enlace](#).

Iniciamos sesión con una cuenta de Hotmail.



Una vez tenemos una cuenta, bajamos y encontramos una serie de apartados. Entramos en el primero.



## INTRODUCCIÓN

Los procedimientos almacenados son grupos con nombre de instrucciones Transact-SQL (T-SQL) que se pueden usar y reutilizar siempre que se necesiten. Los procedimientos almacenados pueden devolver resultados, manipular datos y realizar acciones administrativas en el servidor. Es posible que tenga que ejecutar procedimientos almacenados que otra persona haya creado, o bien crear los suyos propios.

El procedimiento almacenado puede contener comandos de definición de datos y comandos de manipulación de datos, lo que proporciona una interfaz limpia entre una base de datos y una aplicación.

El uso de procedimientos almacenados tiene ventajas, entre las que se incluyen:

- Reutilización del código. El procedimiento almacenado se puede escribir, probar y reutilizar según sea necesario. Esto ayuda a eliminar errores y a reducir el tiempo de desarrollo.
- Seguridad. Los procedimientos almacenados permiten a los usuarios y programas realizar determinadas operaciones en objetos de base de datos, sin conceder permisos a las tablas subyacentes. Esto le permite controlar qué procesos y actividades se permiten, lo que mejora la seguridad.
- Mejora de la calidad. También puede incluir el código de control de errores adecuado y asegurarse de que cada procedimiento almacenado se prueba correctamente antes de usarse en un entorno de producción.
- Mejorar el rendimiento. Cuando se ejecutan por primera vez los procedimientos almacenados, se crea un plan de ejecución. Ese plan de ejecución se puede reutilizar cuando se vuelve a ejecutar el procedimiento almacenado. Esto suele ser más rápido que crear un plan de ejecución cada vez que se ejecuta el código.
- Mantenimiento inferior. Los procedimientos almacenados proporcionan una interfaz a la capa de datos. Cuando los cambios en los objetos de base de datos subyacentes son diferentes, solo se actualizan los procedimientos, lo que proporciona una separación limpia entre los datos y las capas de aplicación.

Hay tres tipos de procedimientos almacenados:

- Procedimientos almacenados definidos por el usuario.
- Procedimientos almacenados temporales.
- Procedimientos almacenados del sistema.

Este módulo le mostrará cómo llamar a un procedimiento almacenado, pasar un parámetro a un procedimiento almacenado y crear y modificar procedimientos almacenados. También aprenderá a construir SQL dinámico y escribir funciones con valores de tabla insertadas.

Después de completar este módulo, podrá:

- Devuelva los resultados mediante la ejecución de procedimientos almacenados.
- Pase de parámetros a procedimientos.
- Cree procedimientos almacenados simples que encapsulen una instrucción SELECT.

- Construya y ejecute SQL dinámico con EXEC y sp\_executesql.
- Cree funciones definidas por el usuario y escriba consultas con ellas.

---

## LLAMADA A PROCEDIMIENTOS ALMACENADOS

Una aplicación o un usuario pueden llamar a los procedimientos almacenados. También se los puede llamar al iniciarse SQL Server.

### EJECUCIÓN DE UN PROCEDIMIENTO ALMACENADO POR PARTE DE UN USUARIO

Cuando una aplicación o un usuario ejecuta un procedimiento almacenado, se usa el comando EXECUTE o su acceso directo, EXEC, seguido del nombre de dos partes del procedimiento. Por ejemplo:

```
EXEC dbo.uspGetEmployeeManagers
```

También se llama a los procedimientos almacenados del sistema mediante la palabra clave EXECUTE o EXEC. La intercalación de base de datos de llamada se usa al comparar los nombres de los procedimientos del sistema. Si la intercalación de base de datos distingue mayúsculas de minúsculas, debe ejecutar el procedimiento almacenado con las letras mayúsculas y minúsculas correctas del nombre del procedimiento.

Si el procedimiento almacenado es la primera instrucción del lote de T-SQL, el procedimiento se puede ejecutar sin la palabra clave EXECUTE o EXEC.

Para comprobar los nombres exactos de los procedimientos del sistema, use las vistas de catálogo:

```
sys.system_objects
```

```
sys.system_parameters
```

Los procedimientos almacenados del sistema tienen el prefijo sp\_. Los usuarios no crean procedimientos almacenados del sistema, pero forman parte de todas las bases de datos definidas por el usuario y por el sistema. No requieren que se ejecute un nombre completo, pero es un procedimiento recomendado incluir el nombre de esquema sys. Por ejemplo:

```
EXEC sys.sp_who;
```

### EJECUCIÓN AUTOMÁTICA DE UN PROCEDIMIENTO ALMACENADO

Puede ejecutar un procedimiento almacenado cada vez que se inicia SQL Server. Es posible que desee llevar a cabo operaciones de mantenimiento de bases de datos o ejecutar un procedimiento como un proceso en segundo plano. Los procedimientos almacenados que se ejecutan automáticamente no pueden contener parámetros de entrada o salida.

Use `sp_procoption` para ejecutar un procedimiento almacenado cada vez que se inicia una instancia de SQL Server. La sintaxis es:

```
sp_procoption [ @ProcName = ] 'procedure'  
    , [ @OptionName = ] 'option'  
    , [ @OptionValue = ] 'value'
```

Por ejemplo:

```
EXEC sp_procoption @ProcName = myProcedure  
    , @OptionName = 'startup'  
    , @OptionValue = 'on';
```

Para ejecutar varios procedimientos que no necesitan ejecutarlos en paralelo, use un procedimiento como procedimiento de inicio y llame a los demás procedimientos desde el procedimiento de inicio. Esto usará solo un subproceso de trabajo.

Los procedimientos de inicio deben estar en la base de datos maestra.

---

## PASE DE PARÁMETROS A PROCEDIMIENTOS ALMACENADOS

Una de las ventajas de usar procedimientos almacenados es que puede pasarles parámetros en tiempo de ejecución. Los parámetros de entrada se pueden usar para filtrar los resultados de la consulta, como en el predicado de una cláusula `WHERE` o el valor de un operador `TOP`. Los parámetros de procedimiento también pueden devolver valores al programa de llamada si el parámetro se marca como `OUTPUT`. También puede asignar un valor predeterminado a un parámetro.

### PARÁMETROS DE ENTRADA

Los procedimientos almacenados declaran sus parámetros de entrada por nombre y tipo de datos en el encabezado de la instrucción `CREATE PROCEDURE`. A continuación, el parámetro se usa como una variable local dentro del cuerpo del procedimiento. Puede declarar y usar más de un parámetro en un procedimiento almacenado. Los parámetros de entrada son el tipo predeterminado de parámetro.

Los nombres de parámetro deben ir precedidos del carácter `@` y ser únicos en el ámbito del procedimiento.

Para pasar un parámetro a un procedimiento almacenado, use la sintaxis siguiente:

```
EXEC <schema_name>.<procedure_name> @<parameter_name> = 'VALUE'
```

Por ejemplo, un procedimiento almacenado denominado ProductsBySupplier en el esquema de productos se ejecutaría con un parámetro denominado supplierid con el código siguiente:

```
EXEC Products.ProductsBySupplier @supplierid = 5
```

Es un procedimiento recomendado pasar valores de parámetro como pares nombre-valor. Varios parámetros se separan con comas. Por ejemplo, si el parámetro se denomina customerid y el valor que se va a pasar es 5, use el código siguiente:

```
EXEC customers.customerid @customerid=5
```

También puede pasar parámetros por posición, omitiendo el nombre del parámetro. Sin embargo, los parámetros se deben pasar por nombre o por posición; no se puede mezclar la forma en que se pasan los parámetros al procedimiento. Si los parámetros se pasan por orden, deben estar en el mismo orden en el que aparecen en la instrucción CREATE PROCEDURE.

Puede pasar valores como una constante o como una variable, como:

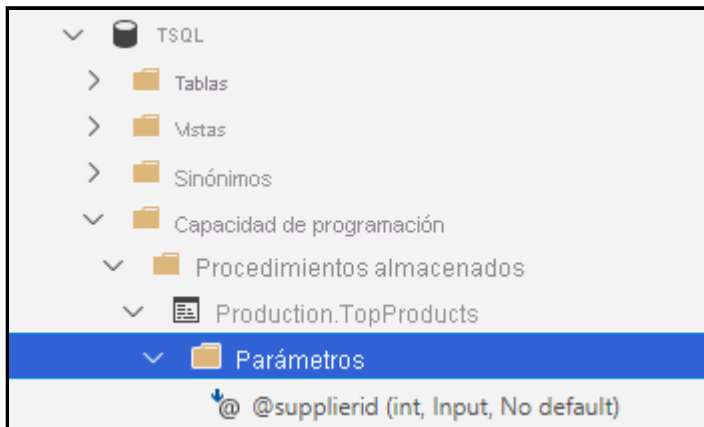
```
EXEC customers.customerid @CustomerID
```

Sin embargo, no se puede usar una función para pasar un parámetro. Por ejemplo, el código siguiente generaría un error:

```
EXEC customers.customerid GETDATE()
```

Compruebe que los parámetros son del tipo de datos correcto. Por ejemplo, si un procedimiento acepta un NVARCHAR, realice el pase en el formato de cadena de caracteres Unicode: N'string'.

Puede ver nombres de parámetros y tipos de datos en Azure Data Studio o SQL Server Management Studio (SSMS). Expanda la lista de objetos de base de datos hasta que vea la carpeta Procedimientos almacenados, debajo de la carpeta Programación.



Se muestran los nombres de dos partes de los procedimientos almacenados, junto con una carpeta Parameters que contiene para cada parámetro:

- Nombre del parámetro.
- Tipo de datos.
- Una flecha de entrada que indica un parámetro de entrada.
- Una flecha de salida que indica un parámetro de salida.

Puede consultar una vista de catálogo del sistema como sys.parameters para recuperar definiciones de parámetros junto con el identificador de objeto.

## VALORES PREDETERMINADOS

Si un parámetro se declaró con un valor predeterminado, no tiene que pasar el valor al ejecutarse el procedimiento almacenado. Si se pasa un valor, se usará, pero si no se pasa ningún valor, se usa el valor predeterminado.

Cuando se crea el procedimiento almacenado, los parámetros reciben valores predeterminados mediante el operador = , como:

```
CREATE PROCEDURE Sales.SalesYTD
-- Set NULL as the default value
@SalesPerson nvarchar(50) = NULL
AS ...
```

## PARÁMETROS DE SALIDA

Ha visto cómo pasar un valor a un procedimiento almacenado, conocido como parámetro de entrada.

Sin embargo, también puede devolver un valor al programa que realiza la llamada. Esto se conoce como parámetro OUTPUT. Use la palabra clave OUTPUT o OUT para especificar un parámetro de salida en la instrucción CREATE PROCEDURE. El procedimiento devuelve el valor actual del parámetro de salida al programa de llamada cuando se abandona el procedimiento.

El programa de llamada también debe usar la palabra clave OUTPUT al ejecutar el procedimiento, a fin de guardar el valor del parámetro en una variable que se pueda usar en el programa de llamada.

En el siguiente fragmento de código de T-SQL, dos parámetros se definen como parámetros OUTPUT, @ComparePrice y @ListPrice.

```
CREATE PROCEDURE Production.uspGetList @Product varchar(40)
    , @MaxPrice money
    , @ComparePrice money OUTPUT
    , @ListPrice money OUT
AS
```

A continuación, los valores se asignan a los parámetros OUTPUT en el cuerpo del procedimiento almacenado, por ejemplo, SET @ComparePrice = @MaxPrice;.