

Documentación M10UF1



Lluc Galceran Jodar
28/05/2023

Handlers:

Los handlers son constructos que te permiten manejar y controlar los errores que pueden ocurrir durante la ejecución de consultas o comandos SQL. Cuando trabajas con una base de datos, es posible que encuentres situaciones en las que las operaciones no se realicen correctamente debido a errores.

Un handler te brinda la capacidad de anticipar esos errores y establecer acciones específicas para manejarlos adecuadamente. Puedes definir un handler para un tipo de error particular o para varios tipos de errores.

Por ejemplo, puedes configurar un handler para capturar un error de división por cero y mostrar un mensaje personalizado al usuario en lugar de que la consulta falle.

También puedes especificar acciones como registrar el error en un archivo de registro, revertir transacciones incompletas o ejecutar comandos alternativos cuando se produce un error.

```
BEGIN TRY
    -- Tu consulta o comando SQL aquí
    SELECT 1/0; -- División por cero para generar un error
END TRY
BEGIN CATCH
    -- Manejar el error capturado
    PRINT 'Se produjo un error: ' + ERROR_MESSAGE();
END CATCH;
```

Usuarios:

En SQL, los usuarios son entidades que tienen acceso a una base de datos y pueden interactuar con ella.

Cada usuario tiene su propio nombre de usuario y una contraseña asociada para autenticarse y acceder a la base de datos. Los usuarios pueden realizar diferentes acciones y operaciones en la base de datos según los privilegios y permisos que se les hayan otorgado.

Por ejemplo, un usuario puede tener permisos para **leer datos, modificar registros, crear tablas o ejecutar consultas**, mientras que otro usuario puede tener **solo permisos de lectura**.

La gestión de usuarios te permite **controlar** quién puede **acceder y manipular los datos de la base de datos**, asegurando que solo las personas o programas autorizados tengan acceso a la información.

Esto es fundamental para la seguridad y la integridad de la base de datos, ya que garantiza que solo los usuarios autorizados puedan realizar cambios y protege la información sensible de accesos no autorizados.

```
-- Crear un nuevo usuario
CREATE USER username
WITH PASSWORD 'password';

-- Conceder permisos al usuario
GRANT SELECT, INSERT, UPDATE, DELETE
ON table_name
TO username;

-- Revocar permisos de usuario
REVOKE INSERT, UPDATE
ON table_name
FROM username;

-- Eliminar un usuario
DROP USER username;
```

Cursores:

En SQL, un cursor es una estructura de programación que te permite recorrer y manipular filas individuales dentro de una tabla. Imagina una tabla como una hoja de cálculo con varias filas.

Con un cursor, puedes seleccionar una fila específica y luego realizar acciones en ella, como leer o modificar los valores de las columnas.

Un cursor te brinda un control más preciso sobre los datos, ya que puedes procesar cada fila de manera individual. Esto es útil en situaciones donde necesitas realizar operaciones detalladas en los datos, como cálculos complejos o actualizaciones condicionales.

Los cursores te permiten avanzar o retroceder a través de las filas de una tabla según tus necesidades y aplicar las operaciones deseadas en cada una.

Por ejemplo supongamos que tenemos una tabla llamada "clientes" con las columnas "id_cliente" y "nombre". Queremos recorrer todos los registros de la tabla y mostrar el nombre de cada cliente.

```
DECLARE @nombre_cliente VARCHAR(100); -- Variable para almacenar el nombre
del cliente

DECLARE cursor_clientes CURSOR FOR
SELECT nombre
FROM clientes;

OPEN cursor_clientes;

FETCH NEXT FROM cursor_clientes INTO @nombre_cliente;

WHILE @@FETCH_STATUS = 0
BEGIN
    -- Realizar acciones en la fila actual
    PRINT 'Nombre del cliente: ' + @nombre_cliente;

    FETCH NEXT FROM cursor_clientes INTO @nombre_cliente;
END

CLOSE cursor_clientes;
```