

Conexión SQL Server y Operaciones con Python

Juan David Ruiz Olmos

November 16, 2024

1 Paso 1: Crear un Servidor Local en SQL Server

Listing 1: Crea un servidor local en SSMS

2 Paso 2: Crear un Login de SQL Authentication

```
1 USE master;  
2 GO  
3 CREATE LOGIN MyUserLogin  
4 WITH PASSWORD = 'SecurePass123';  
5 GO  
6  
7 CREATE USER MyUser FOR LOGIN MyUserLogin;  
8 GO  
9 ALTER SERVER ROLE sysadmin ADD MEMBER MyUser;  
10 GO
```

Listing 2: Crear un login SQL Authentication

3 Paso 3: Crear una Base de Datos en SSMS

1. Haz clic derecho en **Databases** y selecciona **New Database**.
2. Asigna un nombre, por ejemplo, **MyDatabase**.
3. Haz clic en **OK** para crear la base de datos.

4 Paso 4: Conectar a la Base de Datos usando Python

Asegúrate de tener instalado el paquete pyodbc.

```

1 import pyodbc
2
3
4 conn = pyodbc.connect(
5     'DRIVER={ODBC Driver 17 for SQL Server};'
6     'SERVER=localhost;'
7     'DATABASE=MyDatabase;'
8     'UID=MyUserLogin;'
9     'PWD=SecurePass123;'
10 )
11 print("Conexión exitosa!")
12 cursor = conn.cursor()

```

Listing 3: Código para conectarse a SQL Server

5 Paso 5: Crear una Tabla con Mínimo 10 Registros

```

1 CREATE TABLE Employees (
2     ID INT PRIMARY KEY,
3     Name VARCHAR(50),
4     Position VARCHAR(50)
5 );
6
7 INSERT INTO Employees (ID, Name, Position) VALUES
8 (1, 'John Doe', 'Manager'),
9 (2, 'Jane Smith', 'Developer'),
10 (3, 'Samuel Johnson', 'Analyst'),
11 (4, 'Michael Brown', 'Designer'),
12 (5, 'Linda White', 'Sales'),
13 (6, 'Chris Green', 'HR'),
14 (7, 'Sarah Black', 'Marketing'),
15 (8, 'Paul Blue', 'Support'),
16 (9, 'Laura Red', 'Finance'),
17 (10, 'Alice Grey', 'Consultant');

```

Listing 4: Crear una tabla e insertar registros

6 Paso 6: Crear un Catálogo para la Tabla

```

1 CREATE FULLTEXT CATALOG EmployeeCatalog AS DEFAULT;
2 GO
3 CREATE FULLTEXT INDEX ON Employees(Name)
4 KEY INDEX PK_Employees ON EmployeeCatalog;
5 GO

```

Listing 5: Crear un catálogo

7 Paso 7: Crear un Índice para el Catálogo

```
1 CREATE NONCLUSTERED INDEX idx_position
2 ON Employees(Position);
```

Listing 6: Crear un índice

8 Paso 8: Crear un Stored Procedure Dinámico

```
1 CREATE PROCEDURE sp_GetEmployeesByPosition
2     @Position VARCHAR(50)
3 AS
4 BEGIN
5     SET NOCOUNT ON;
6     DECLARE @query NVARCHAR(MAX);
7     SET @query = 'SELECT * FROM Employees WHERE Position = ''' +
8         @Position + '''';
9     EXEC sp_executesql @query;
10 END;
11 GO
12 -- Ejemplo de uso:
13 EXEC sp_GetEmployeesByPosition 'Developer';
```

Listing 7: Stored Procedure Dinámico

9 Paso 9: Crear y Utilizar Cursores

```
1 DECLARE EmployeeCursor CURSOR FOR
2 SELECT Name, Position FROM Employees;
3
4 DECLARE @Name VARCHAR(50), @Position VARCHAR(50);
5
6 OPEN EmployeeCursor;
7 FETCH NEXT FROM EmployeeCursor INTO @Name, @Position;
8
9 WHILE @@FETCH_STATUS = 0
10 BEGIN
11     PRINT 'Nombre: ' + @Name + ', Cargo: ' + @Position;
12     FETCH NEXT FROM EmployeeCursor INTO @Name, @Position;
13 END;
14
15 CLOSE EmployeeCursor;
16 DEALLOCATE EmployeeCursor;
```

Listing 8: Uso de cursores para consultas

10 Conclusión

En este documento, se ha demostrado cómo crear un servidor local en SQL Server, configurar un login de autenticación, crear una base de datos, y realizar diversas operaciones avanzadas como índices, procedimientos almacenados, y cursores, usando tanto SQL Server como Python.