**Conexión a SQL Server con Python y la librería PyODBC (CRUD)**

**Paso 0:**

**Crear una base de datos en Microsoft SQL Sever Management Studio llamada**:

**baseCine**



**Paso 1: Crear las siguientes tablas:**

CREATE TABLE peliculas(

idPel int identity(1,1) primary key,

titulo VARCHAR(255) NOT NULL,

year SMALLINT NOT NULL

)

CREATE TABLE cines(

idCine int identity(1,1) primary key,

nombre VARCHAR(255) NOT NULL,

direc VARCHAR(255) NOT NULL,

)

CREATE TABLE pelicula\_cines(

idPc int identity(1,1) primary key,

idPel int references peliculas,

idCine int references cines,

)

**Paso 2: Ahora maneja de los datos desde Python (CRUD).**

Se debe conectar Python 3 a una base de datos de SQL Server a través del driver ODBC y la librería PyODBC.

**pyodbc** es un módulo de Python de código abierto que simplifica el acceso a las bases de datos ODBC desde Python, implementando el uso de la DB API 2.0 de una forma conveniente para Python.

**pyodbc** también es considerado como un controlador SQL para Python.

Para instalar pyodbc se utiliza el instalador de paquetes de Python:

**pip install pyodbc**

Se puede comprobar que el módulo fue instalado de forma correcta importando el módulo con import:

**Paso 3: >>import pyodbc**

Si no arroja ningún error, entonces el paquete fue instalado de forma correcta.

**Paso 4: hacer la conexión a la base de datos, crear un archivo bd.py**

direccion\_servidor = 'XXXXXXXX'

nombre\_bd = 'baseCine'

nombre\_usuario = 'sa'

password = 'adminadmin2018'

try:

conexion = pyodbc.connect('DRIVER={ODBC Driver 17 for SQL Server};SERVER=' +

direccion\_servidor+';DATABASE='+nombre\_bd+';UID='+nombre\_usuario+';PWD=' + password)

print("\n"\*2)

print("conexión exitosa")

except Exception as e:

print("Ocurrió un error al conectar a SQL Server: ", e)

Nota: es recomendable siempre cerrar la base de datos

**Paso 5: insertar los datos en las tablas**

**Tabla películas**

try:

with conexion.cursor() as cursor:

consulta = "INSERT INTO peliculas(titulo, year) VALUES (?, ?);"

# Podemos llamar muchas veces a .execute con datos distintos

cursor.execute(consulta, ("Volver al futuro 1", 1985))

cursor.execute(consulta, ("Pulp Fiction", 1994))

cursor.execute(consulta, ("It", 2017))

cursor.execute(consulta, ("Ready Player One", 2018))

cursor.execute(consulta, ("Spider-Man: un nuevo universo", 2018))

cursor.execute(consulta, ("Avengers: Endgame", 2019))

cursor.execute(consulta, ("John Wick 3: Parabellum", 2019))

cursor.execute(consulta, ("Toy Story 4", 2019))

cursor.execute(consulta, ("It 2", 2019))

cursor.execute(consulta, ("Spider-Man: lejos de casa", 2019))

except Exception as e:

print("Ocurrió un error al insertar: ", e)

#finally:

# conexion.close()

**Tabla cines**

try:

with conexion.cursor() as cursor:

consulta = "INSERT INTO cines(nombre, direc) VALUES (?, ?);"

# Podemos llamar muchas veces a .execute con datos distintos

cursor.execute(consulta, ("cinemex polanco", "polanco"))

cursor.execute(consulta, ("cinemex loreto", "loreto"))

cursor.execute(consulta, ("cinepolis san angel", "san angel"))

cursor.execute(consulta, ("cinepolis pedregal", "pedregal"))

except Exception as e:

print("Ocurrió un error al insertar: ", e)

#finally:

# conexion.close()

**Table pelicula\_cines**

try:

with conexion.cursor() as cursor:

consulta = "INSERT INTO pelicula\_cines(idPel, idCine) VALUES (?, ?);"

# Podemos llamar muchas veces a .execute con datos distintos

cursor.execute(consulta, (1,1))

cursor.execute(consulta, (2,2))

cursor.execute(consulta, (3,3))

cursor.execute(consulta, (4,4))

cursor.execute(consulta, (5,4))

cursor.execute(consulta, (6,3))

cursor.execute(consulta, (7,2))

cursor.execute(consulta, (8,1))

cursor.execute(consulta, (9,1))

cursor.execute(consulta, (10,1))

except Exception as e:

print("Ocurrió un error al insertar: ", e)

finally:

conexion.close()

**Paso 6: hacer querys**

**Seleccionar todos los datos de la tabla películas**