



import pandas as pd

from sqlalchemy import create\_engine

# Crear la cadena de conexión

engine =

create\_engine('mysql+mysqlconnector://usuario:contraseña@nombre\_del\_servidor/nombre\_de\_base\_de\_datos') # Leer datos de una tabla MySQL df\_mysql = pd.read\_sql("SELECT \* FROM nombre\_de\_tabla", engine) print(df\_mysql)



import pandas as pd

import pyodbc

conn\_str = "DRIVER={ODBC Driver 17

for SQL Server};

SERVER=nombre\_del\_servidor;

DATABASE=nombre\_de\_base\_de\_datos;

UID=usuario;PWD=contraseña" conn = pyodbc.connect(conn\_str)

df = pd.read\_sql("SELECT \* FROM nombre\_de\_tabla", conn)

conn.close()



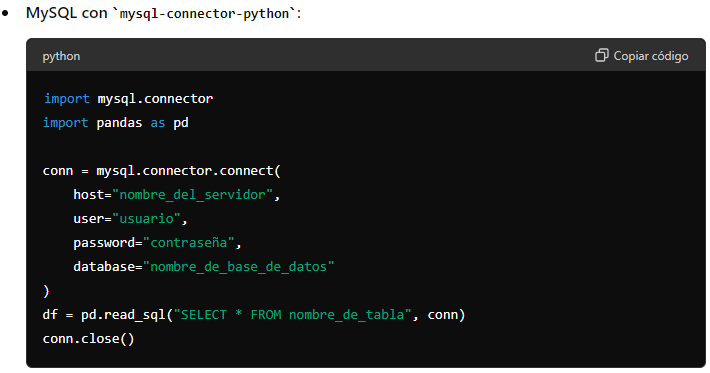
from sqlalchemy import create\_engine

import pandas as pd

engine =

create\_engine('mssql+pyodbc://usuario:contraseña@nombre\_del\_servidor/nombre\_de\_base\_de\_datos?driver=ODBC+Driver+17+for+SQL+Server')

df = pd.read\_sql("SELECT \* FROM nombre\_de\_tabla", engine)





from sqlalchemy import create\_engine

import pandas as pd

engine =

create\_engine('mysql+mysqlconnector://usuario:contraseña@nombre\_del\_servidor/nombre\_de\_base\_de\_datos') df = pd.read\_sql("SELECT \* FROM nombre\_de\_tabla", engine)

Estos ejemplos te permiten conectar y cargar datos desde SQL Server y MySQL a un DataFrame de Pandas usando las bibliotecas más comunes en Python para trabajar con bases de datos.



