

SPRINT 8. Visualitzacions en Python: Scripts en PowerBI

Aquesta tasca consisteix en l'elaboració d'un informe de Power BI, aprofitant les capacitats analítiques de Python. S'utilitzaran els scripts de Python creats prèviament en la Tasca 1 per a generar visualitzacions personalitzades amb les biblioteques Seaborn i Matplotlib. Aquestes visualitzacions seran integrades en l'informe de Power BI per a oferir una comprensió més profunda de la capacitat del llenguatge de programació en l'eina Power BI.

Nivell 1

Per a començar a exportar els scripts de Python del Sprint 8.1 a PowerBI primer hem d'obrir l'arxiu de Python on tenim la connexió que vam fer servir per a connectar amb MySQL. Mitjançant la biblioteca PyMYSQL podem importar arxius de MySQL al nostre IDE de Python (Visual Studio) i, posteriorment exportar-los a PowerBI amb el mateix connector. L'únic canvi que necessitem fer per a què la exportació sigui un èxit, segons l'script emprat en l'Sprint 8.1 és modificar el nom del diccionari on emmagatzemarem els dataframes de 'dataframes' a 'datasets', ja que aquesta és la nomenclatura que reconeix PowerBI.

Per tant, en primer lloc executem els dos primers camps del document de Python a VSCode.

```
import pymysql
import pandas as pd

# Conectar a la base de datos
connection = pymysql.connect(
    host='localhost',
    user='root',
    password='Gael2023',
    database='sales'
)

cursor = connection.cursor()
cursor.execute("SHOW TABLES")
tabla = [t[0] for t in cursor.fetchall()]

#Modifiquem el nom de dataframes a datasets per a no tenir
incompatibilitats de nomenclatura amb PowerBI
```

```

datasets = {}

for nombre_tabla in tabla:
    df = pd.read_sql_query(f"SELECT * FROM {nombre_tabla}", connection)
    datasets[nombre_tabla] = df
    print(f"Tabla {nombre_tabla} cargada exitosamente.")

cursor.close()
connection.close()

```

```

df_transactions = datasets['transactions']
df_products = datasets['products']
df_users = datasets['users']
df_prod_numbers = datasets['products_numbers']

```

Gràcies a aquest comandament podem extreure únicament les taules amb les que pensem treballar a PowerBI sense importar totes les taules de la nostra database. Un cop verifiquem que les taules han estat carregades i extretes del diccionari correctament, passem a treballar amb PowerBI.

```

    cursor.close()
    connection.close()
[1] ✓ 0.9s
...
Tabla cards_status cargada exitosamente.
Tabla companies cargada exitosamente.
Tabla credit_card cargada exitosamente.
Tabla products cargada exitosamente.
Tabla products_numbers cargada exitosamente.
Tabla transactions cargada exitosamente.
Tabla users cargada exitosamente.
C:\Users\Cristina\AppData\Local\Temp\ipykernel_6856\4211847509.py:20: UserWarning: pandas only supports
df = pd.read_sql_query(f"SELECT * FROM {nombre_tabla}", connection)

```

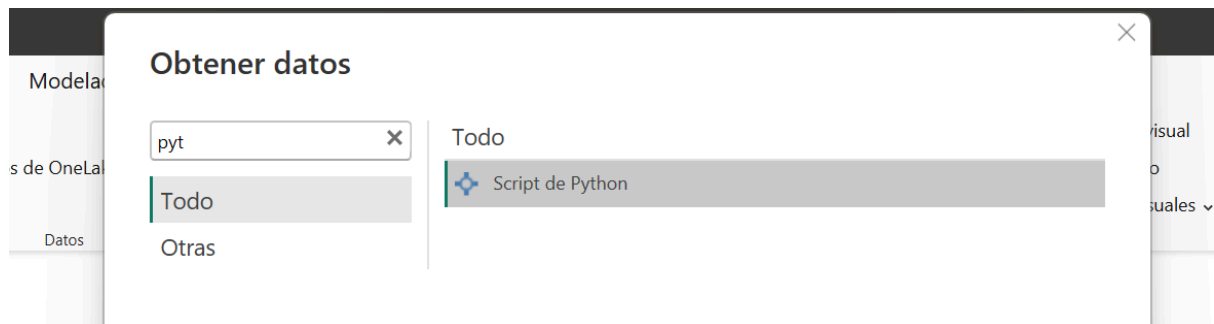
Abans però de començar a treballar amb PowerBI hem extret les taules del diccionari amb les que volem treballar.

```

df_transactions = datasets['transactions']
df_products = datasets['products']
df_users = datasets['users']
df_prod_numbers = datasets['products_numbers']
[2] ✓ 0.0s

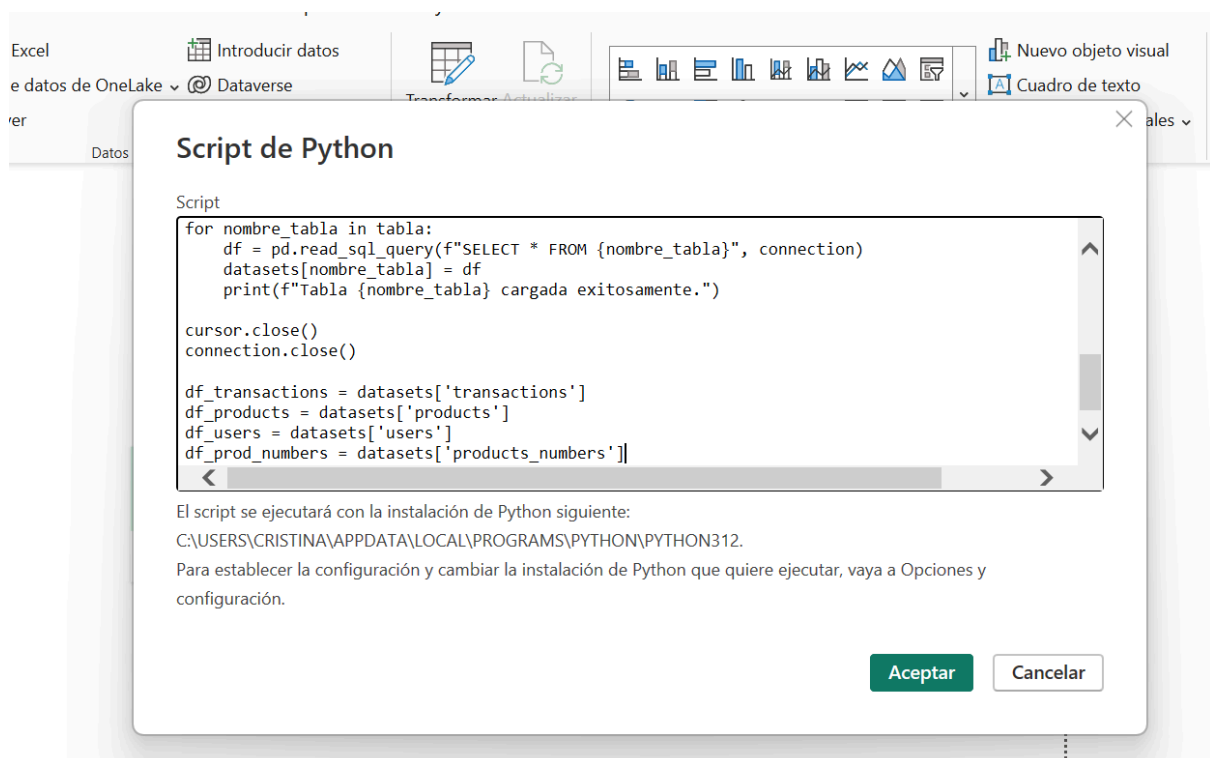
```

En obrir la interfície de PowerBI, seleccionem un document en blanc per a elaborar el nostre dashboard i cliquem a l'opció Obtenir dades.



La opció d'script de Python no apareix entre les principals opcions per defecte que ens presenta el menú d'obtenció de dades de PowerBI, per tant, fem una cerca per a que aparegui i la seleccionem.

Aquesta opció ens obre un pop-up anomenat 'Script de Python' amb una capsa on hem de deixar el codi de la nostra Script. Per tal que apareguin les dades de les taules que hem cridat al Visual Studio Code, copiem la primera script on es troben les connexions i la segona script amb les taules extretes del diccionari.



Un cop copiada l'script, premem el botó Aceptar. Ens apareixen les taules que es troben recollides a Python. Seleccionem les 4 taules que hem fet servir per a l'sprint 8.1 (product_numbers, products, transactions i users). Un indicador de que la càrrega de dades ha estat exitosa és la preview de les dades que apareix en seleccionar cadascuna de les taules.

Navegador

Opciones de presentación ▾

- Python [5]
 - ☐ df
 - ☒ df_prod_numbers
 - ☒ df_products
 - ☒ df_transactions
 - ☒ df_users

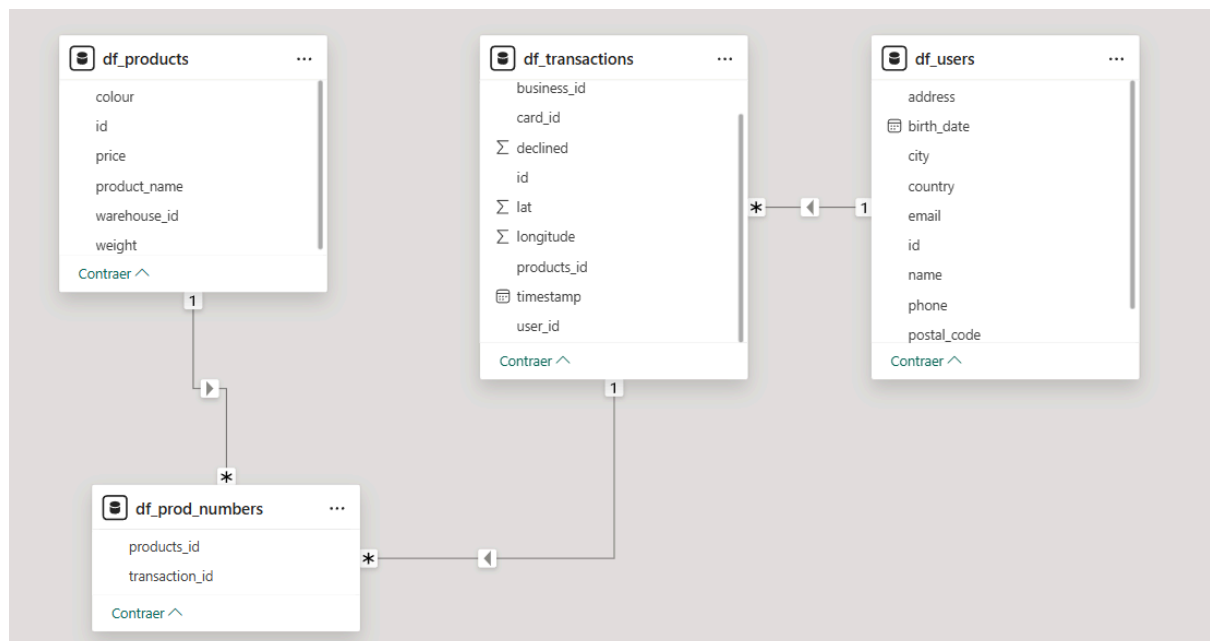
df_users

id	name	surname	phone	email
1	Zeus	Gamble	1-282-581-0551	interdum.enim@protonr
2	Garrett	Mcconnell	(718) 257-2412	integer.vitae.nibh@prote
3	Ciaran	Harrison	(522) 598-1365	interdum.feugiat@aol.or
4	Howard	Stafford	1-411-740-3269	ornare.egestas@icloud.e
5	Hayfa	Pierce	1-554-541-2077	et.malesuada.fames@hc
6	Joel	Tyson	(718) 288-8020	gravida.nunc.sed@yahoo
7	Rafael	Jimenez	(817) 689-0478	eget@outlook.ca
8	Nissim	Franks	(692) 157-3469	egestas.aliquam.fringilla
9	Mannix	Mcclain	(590) 883-2184	aliquam.nisl@outlook.co
10	Robert	McCarthy	(324) 746-6771	fermentum@protonmail

Un cop carregades les taules, PowerBI ens obre les dades dins del menú lateral de Dades. A partir d'aquí ens trobem amb un informe en blanc i podem consultar la informació de les dades dins del submenú de Vista de Taula. Però abans de començar a crear objectes, hem d'entrar a la Vista de Model on ens apareixen les taules carregades de forma independent. PowerBI no és capaç d'importar les connexions entre les taules ni de crear el model, per tant, hem de connectar manualment les taules.

Vinculem la taula de dimensions 'Users' a la taula de fets 'Transactions' per mitjà de la PK/FK User-Id/ID. La relació entre ambdues taules es d'1 a molts. Connectem l'altra taula de dimensions 'Products' amb la taula 'Transactions' per mitjà de la taula pont 'Prod_Numbers'. Aquesta taula ens permetia vincular la taula de fets i la taula de dimensions amb una relació de molts a molts entre la taula pont.

El model resultant és un model d'estrella amb una taula pont vinculant.



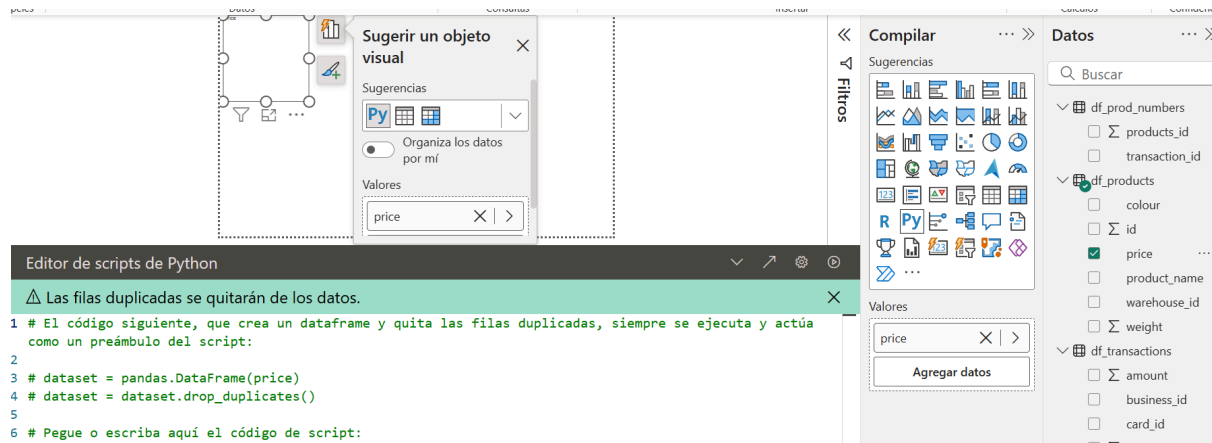
Exercici 1

CREACIÓ DELS OBJECTES AMB PYTHON:

Per a crear un objecte mitjançant una script de Python seleccionem dins del menú **Compilar** l'objecte de Python representat amb el **Py**. Aquest ens apareix com un camp buit amb el logotip de Python i ens obre un editor de scripts de Python. Tanmateix no podem editar ni treballar amb ells fins que dins de la columna **Dades** no triem els camps que afecten l'objecte.

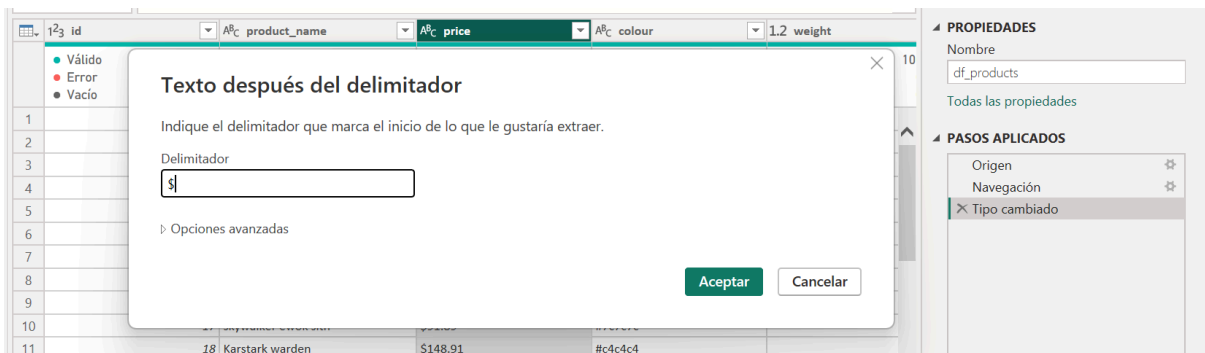


Un cop hem seleccionat un dels camps que volem representar a la nostra visualització, PowerBI ens presenta el següent missatge.

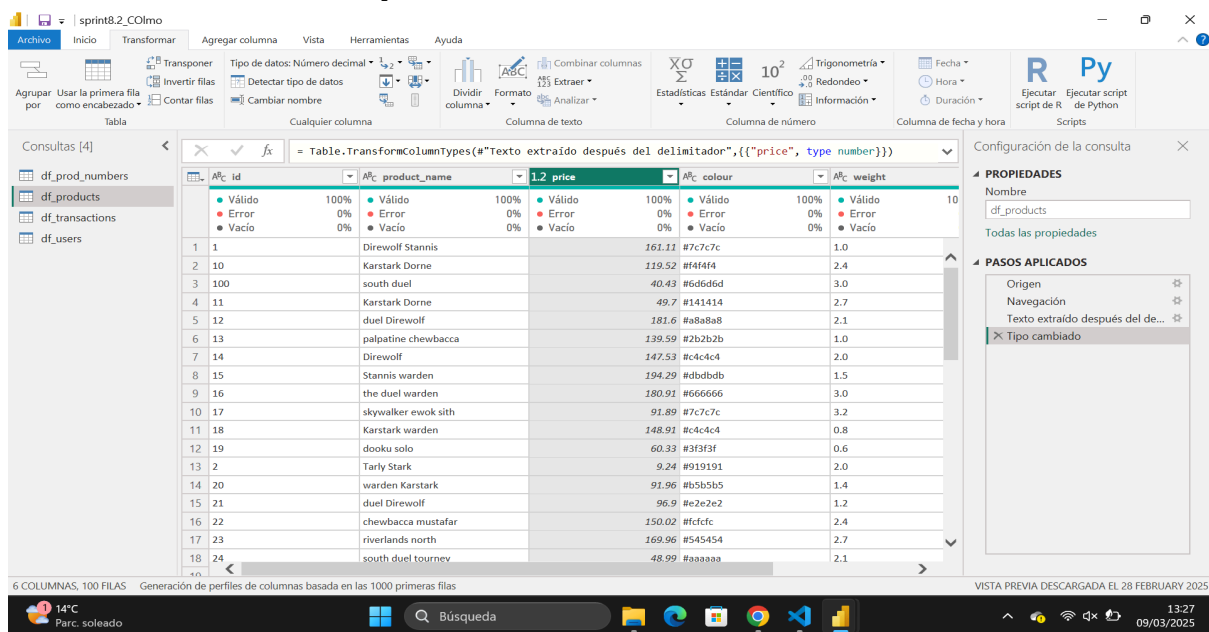


Ara ja podem enganxar l'script de Python de l'exercici de l'Sprint 8.1 dins l'editor de scripts de Python de PowerBI. Tanmateix, en aquest primer exercici vam realitzar una transformació de dades amb una funció de Python per a transformar el preu que havíem emmagatzemat a la nostra base de dades com a 'string' per un 'float'. Malauradament aquesta funció de transformació no aplica quan intentem reproduir l'script d'una visualització, per tant, és necessari fer la transformació addicional en PowerBI.

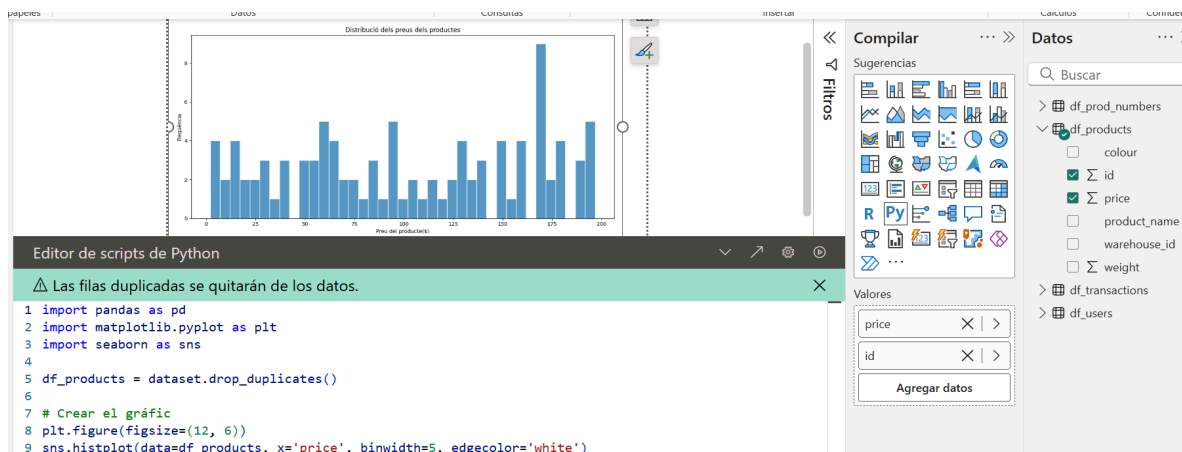
En aquest cas, hem optat per transformar les dades amb PowerQuery. En primer lloc hem extret el símbol del dòlar (\$) com a extracció del primer caràcter.



Posteriorment hem transformat el tipus de dada de text a número decimal. Hem realitzat els canvis en la mateixa columna, per tant, el model no es veu alterat.



Un cop feta la transformació, hem copiat l'script de la visualització de Python. Per a la configuració del dataset, aquest s'ha creat per defecte en seleccionar el camp que hem representat: `df_products > price`. A més hem seleccionat el camp ID dins de la taula 'products' per a evitar que el comandament que apareix per defecte `.drop_duplicates()` ens esborri els productes que tenen un preu igual.



La creació de cadascun dels objectes de Python segueix el mateix procediment.

DASHBOARD FINAL:

