**פרויקט קורס בינה עסקית**

**קורס מס' 40205**

**מטלה מספר2**

**מרצה: מר אור פרץ**

**שמות מגישים:**  
עומר שמולביץ 313320061

דור שטיינברג 208931220

**חלק 1**

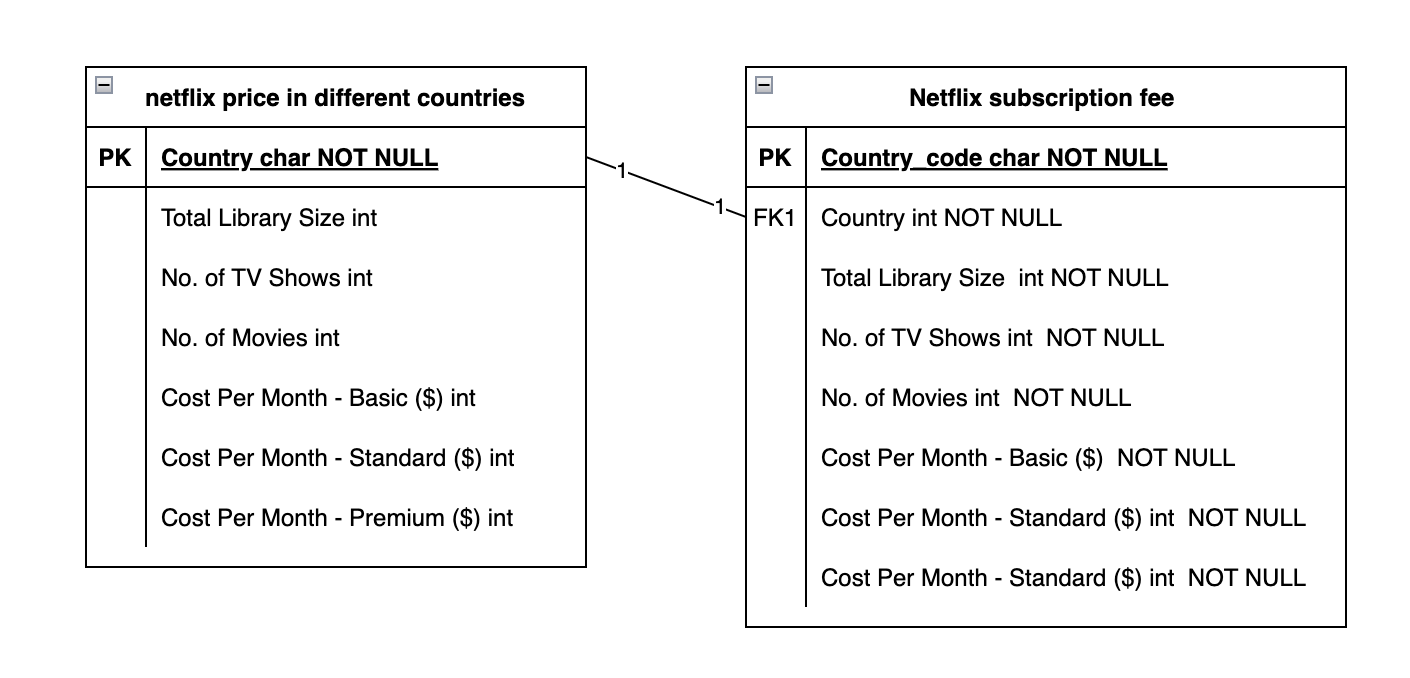
https://github.com/dor2berg/Bi-Project.git

**חלק 2**

1. סכמת star , סכמה קלאסית כל ישות (טבלה) היא בפני עצמה .

בחרנו בסכמת כוכב בכדי לענות על השאלה העסקית , ישנה טבלה מרכזית אחת שמורכבת משתי טבלאות.

1. אנחנו רוצים לדעת בקרב כל המדינות כמה הכי הרבה כסף הוציאו לכל חודש על תוכנית ספיציפית - כלומר איזה תוכנית הכי רווחית , הבחירה הזו מפחיתה לי joins ושליפה מהירה יותר.



1. בוצע

**חלק 3**

**שאלה 1**

EXTRACT:

* איתור וזיהוי מקורות המידע הנדרשים למציאת הפופולריות בקרב התוכניות השונות בכל מדינה וחילוץ נתונים מטבלאות המקור (Netflix price in different countries, Netflix subscription fee dec-2021)

TRANSFORM:

* העברת הנתונים שחולצו למודל דו טבלאי. כעת ביצוע סינון של העמודות הרלוונטיות בהתאם לשאלה העסקית, וביצוע ניקוי וסינון הנתונים על מנת להסיר פגומים ונתונים לא רלוונטיים

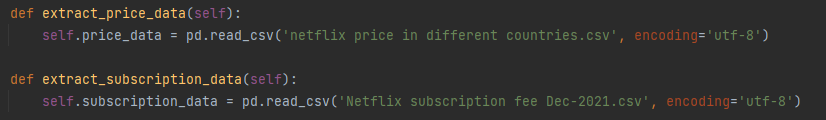
LOAD:

* טעינת הנתונים שחולצו ל- DW

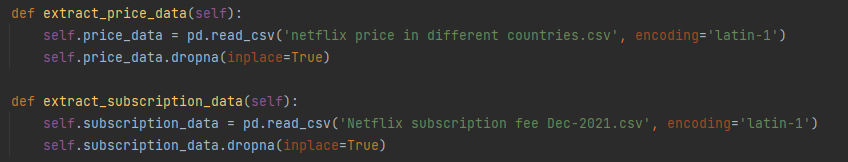
**שאלה 2**

הגדרת תהליך ה Pipeline ETL -:

* שלב 1 – Data Reference :בשלב זה נגדיר את סט הנתונים המורכב מ-2 טבלאות Netflix price in different countries, Netflix subscription fee dec-2021 אותן נגדיר עם השדות הרלוונטיים
* שלב 2 – Reference Data from Extract :בשלב זה חילוץ של הנתונים ע"י קובץ CSV.



* שלב 3 – Validation Data :בשלב זה נוודא כי הנתונים הקיימים מתאימים למטרת הפרויקט, כלומר נוודא שקיימים המאפיינים של מדינה, שם המדינות ומידת הפופולריות שלהן.
* שלב 4 – Data Transformation :בשלב זה נבצע אימות של שילוב המידע מתוך הטבלאות בהן השתמשנו. שלב זה כולל ניקוי של הדאטה, במקרה שלנו לא הופיעו בנתונים ערכים חסרים או חריגים ולכן לא היה צורך

(השארנו ליתר ביטחון) 

* שלב 5 – Stage :בשלב זה הוא שלב ביניים שבו כל הנתונים נמצאים באזור ה- staging בתוכנות כמו Tableau או Airflow, דרכן הנתונים עוברים.
* שלב 6 – Warehouse Data to Publish :בשלב זה העברנו את הנתונים הרלוונטיים לאחר עיבודם למחסן נתונים שמורכב מטבלה אחת בה יהיו המפתחות של השירים והאמנים.

**שאלה 3**

import pandas as pd  
import sqlite3  
  
class NetflixETL:  
 def \_\_init\_\_(self):  
 self.price\_data = None  
 self.subscription\_data = None  
  
 def extract\_price\_data(self):  
 self.price\_data = pd.read\_csv('netflix price in different countries.csv', encoding='latin-1')  
 self.price\_data.dropna(inplace=True)  
 print("Price data column names:")  
 print(self.price\_data.columns)  
  
 def extract\_subscription\_data(self):  
 self.subscription\_data = pd.read\_csv('Netflix subscription fee Dec-2021.csv', encoding='latin-1')  
 self.subscription\_data.dropna(inplace=True)  
 print("Subscription data column names:")  
 print(self.subscription\_data.columns)  
  
 def transform\_data(self):  
 merged\_data = pd.merge(self.price\_data, self.subscription\_data, on='Country')  
 merged\_data['Total Cost per Month'] = merged\_data['Cost Per Month - Basic ($)'] + merged\_data['Cost Per Month - Standard ($)'] + merged\_data['Cost Per Month - Premium ($)']  
 profitable\_program = merged\_data.groupby('Country')['Total Cost per Month'].idxmax()  
 transformed\_data = merged\_data.loc[profitable\_program, ['Country', 'Total Cost per Month']]  
 return transformed\_data  
  
 def load\_data(self):  
 conn = sqlite3.connect('netflix\_profitability.db')  
 transformed\_data = self.transform\_data()  
 transformed\_data.to\_sql('profitability\_data', conn, if\_exists='replace', index=False)  
 conn.close()  
 print("Profitability data successfully loaded to the database.")  
 print(transformed\_data)  
  
 def run\_etl(self):  
 self.extract\_price\_data()  
 self.extract\_subscription\_data()  
  
 self.load\_data()  
  
etl = NetflixETL()  
  
etl.run\_etl()