**Assignment-2**

Assignment üç tane dosyadan oluşur:

* Db\_connect.py: Postgresql database fonksiyonlarının yazılı olduğu dosya.
* Main.py: Ana işlemlerin yapıldığı, verinin process edildiği dosya.
* Configure.txt: Database bağlantısı yapılması için gereken bilgileri tutan .txt dosyası.
* Dockerfile: Dockerization işleminin yapıldığı dosya.

**db\_connect.py**

İlk olarak, database’e bağlanmak için gerekli olan bilgileri configure.txt dosyasından okur. Okunan değerleri dictionary’e atar. Bu dosyadaki her fonksiyon database connection işlemi yapar. Bu işlemi yaparken de elde edilen configuration bilgilerini kullanır.

“create\_table” fonksiyonu, database üzerinde yeni tablo oluşturulur. Bu işlemi yapmak için fonksiyon parametre olarak table oluşturacak script alınır.

“insert\_to\_table” fonksiyonu, belirtilen tabloya veri insert eder. Parametre olarak insert edilecek bilgilerin tutulduğu dataframe’i, insert için kullanılacak script’i ve table’ın last\_transaction table olup olmadığını kontrol edecek bir parametre alır.

“select\_from\_table” fonksiyonu, assignment’da bizden veri içinden çekmemiz istenen verileri çeken işlemleri yapar. Parametre olarak bulunmak istenen sonucu getirecek query’leri alır. Fonksiton bu query’nin sonucunu bir dataframe haline getirip return eder.

**main.py**

Dört fonksiyondan oluşur:

1.)get\_data\_from\_API: Parametre olarak link alır. Fonksiyon bu linki kullanarak requests kütüphanesinden de faydalanarak API üzerinden veri çeker. Bu veriyi json kütüphanesini kullanarak dictionary’e çevirdikten sonra return eder.

2.)get\_specific\_array: Parametre olarak önceki fonksiyondaki dictionary haline çevrilmiş veriyi, veri içinde istenen kısmın adını(buyers mı, sellers mı yoksa last transaction mı?), para birimini (TRY ya da USDT) ve fonksiyon içinde işlemlerden geçtikten sonra verinin atılacağı listeyi alır.

İlk olarak verinin istenen kısmı dictionary’den alınır ve dataframe’e çevrilir. Veri buyers veya sellers tarafından ise dataframe’e ekstra buyers\_or\_sellers column’u eklenir ve bu değer kaydedilir. Sonrasında da currency adında yeni bir column oluşturulur. Verideki para birimini bu column’a kaydeder. En son işlem ise, parametredeki listeye dataframe’i insert eder.

3.)script\_for\_select: Parametre olarak verinin para birimi, table adı ve select işlemi için kullanılacak iki column’un adını alır. Assignment’ta istenen max, min, avg ve volume bulma işlemlerini yapacak query’i oluşturur ve return eder. Zamanı geldiğinde bu fonksiyon script yazmada kullanılacaktır.

4.)main: Yukardaki fonksiyonların çağrıldığı yerdir. "get\_data\_from\_API” işleminden elde edilen veri ile “get\_specific\_array” fonksiyonu çağrılır. Bu fonksiyon hem buyers\_and\_sellers hem de last\_transaction verisi için çağrılır ve iki verinin farklı para birimli versiyonlarını kendi içinde aynı listeye atar. Bu listeler concat işlemi için kullanılır. Tüm buyers\_and\_sellers verileri ve last\_transaction verileri kendi içinde aynı dataframe’e atar.

db\_connection üzerinden “create\_table” fonksiyonu çağrılır.

db\_connection üzerinden “insert\_to\_table” fonksiyonu çağrılır.

db\_connection üzerinden “select\_from\_table” fonksiyonu çağrılır. Select işlemi için gereken script’i ise “script\_for\_select” fonksiyonu ile alırız. Sonucunda dönen veriyi dataframe’e atarız. Currency ve table türüne göre farklı dört tane dataframe oluşturulur ve sonuçlar bu dataframe’lere atılır. Sonra bu dataframe’lerden para birimi TRY olanlar ayrı bir dataframe’de, USDT olanlar ise başka bir dataframe’de birleştirilir.

Dört ayrı dataframe kullanmamızın sebebi elde edilen verinin iki farklı excel’e yazılması gerektiğindendir. Bu excel’lerden biri sadece TRY para birimine ait olan verileri barındırması gerekirken, diğer excel dosyasının ise sadece USDT para birimine ait olan veriliri barındırması gerekir.

Excel’e atma işlemleri pandas kütüphanesi sayesinde yapılıyor. Df.to\_excel() fonksiyonu çağrılır.

**Dockerfile**

Dockerize işlemi yaparken aldığım bir hatadan dolayı, bu süreç yarım kalmıştır.

Image oluşturup, sonrasında bu image’ı çalıştırırken "can't find '\_\_main\_\_' module in './main.py'" hatası alıyorum.