**Part 1**

**100. satırdaki x = 4 yerine x = 230** yazılır ve program çalıştırılır.

**autoscout1.py** dosyasının **210.** satırındaki bilgiler, kullanılan pc'deki ayarlar ile değiştirilir.

**autoscout1.py** çalıştırıldıktan sonra database’e \***carlist\_de\_mileage\_desc**\* tablosu oluşur.

**Part 2**

\* **carlist\_de\_mileage\_desc** \* tablosu içerisindeki **link** sütunu **read** edilir.

**autoscout2.py** dosyasının, 45-46-47. ve 275. satırlarındaki bilgiler, kullanılan pc'deki ayarlar ile değiştirilir.

**80 ve 81. satırlardaki gelen x ve y için ;**

Örneğin ;

X=0 ve Y=5 için

sırayla : \***carlist\_de\_mileage\_desc\* tablosundan** id = 1-2-3 ve 4’e karşılık gelen **link**ler, web driver da aynı anda açılır. (Pc eğer 4 işlemcili ise 4 link de aynı anda web driver da açılır)

Böylelikle id=1 ile id=4 aralığındaki modellerin ilan linklerini (ad\_links) toplamış olacağız.

( id=1, id=2, id=3, id=4 )’den gelen linkler –yani model olarak;

Audi : 100, 200, 50, 80

Gelen veriler database’deki \***adlinks\_de\_mileage\_desc**\* tablosuna kayıt olunur.

**Not: *Part2 bir kere çalıştırıldıktan sonra, 297. satırdaki kod, yorum satırına dönüştürülmelidir.***

**Part 3**

\***adlinks\_de\_mileage\_desc**\* tablosu içerisindeki **ad\_link** sütunu **read** edilir.Toplamış olduğumuz ilan linklerinden (ad\_links’lerden) verileri çekeceğimiz kısma gelmiş bulunmaktayız.

Yine part 2 de olduğu gibi, **autoscout3.py** dosyasının, 37-38-39. ve 626. satırlarındaki bilgiler, kullanılan pc'deki ayarlar ile değiştirilir.

**74 ve 75. satırlardaki x-y mantığı part 2 de anlatıldığı gibidir.**

Gelen veriler \***cars**\* tablosuna kayıt olunur.

**Not: *Part3 bir kere çalıştırıldıktan sonra, 695. satırdaki kod, yorum satırına dönüştürülmelidir.***

**Diğer Bilgiler**

Elimizde Part 1 den gelen 4397 adet mark-model linki vardır.

Elimizde Part 2 den gelen 324.460 adet ilan linki vardır. (*carlist\_de\_***mileage\_desc** şeklinde)

Part 2 için **mileage\_asc, price\_asc, power\_asc** vs. şeklinde filtremeler yapıp autoscout24 sitesinde bulunan tüm ilan linkleri elde edilmeye çalışılacaktır.

Sadece ***Country: Germany*** filtresi yapıldığında şuan için toplam ilan sayısı 685.700 olarak gözükmektedir.

Ayrıca Part 3 için, Part 2 den gelen ilan linkleri (ad\_links) açılıp, araç bilgileri adına herşey çekilmeye başlanmıştır. Toplamda **324.640** adet araç bilgisinin tümü yakın zamanda elimizde olacaktır. Şuanda autoscout3.py dosyası hala çalışır durumdadır. Firefox Web Driver açık halde ve elimize yaklaşık **300.000** aracın bilgileri geçmiş durumdadır.

Database : **autoscout**

Tablo : **cars**

içerisine her ort. 6,5 saniyede bir, \*8 ilan içerisindeki tüm bilgiler aktarılıyor durumdadır. (\*8 çekirdekli bir pc kullanıyor olduğumuzu varsayarsak)

**Geliştirmeler**

+ Multiple WebDriver kullanımı entegre edildi.

- Tablolar birbirlerine bağlanacak.

**Dipnotlar :**

- **geckodriver.exe, python.exe’nin olduğu dosya konumuna taşınmalıdır. s**

**Örneğin : (**C:\Users\\*\*\*\ AppData\Local\Programs Python\Python39)

- \***cars**\* tablosuna, link column dan sonra yeni kolonlar (fields) eklenmiştir ve Part 3 deki bu kod yapısı yaklaşık id = 8105 ile birlikte değişmiş olup (tarih olarak 2022\_09\_29\_10:25), buraya kadarki gelen veriler o kolonlarda bu yüzden NULL olarak gözükmektedir.

**- autoscout.sql dosyası rar içerisinden çıkartılıp db’ye aynı isimde bir database oluşturularak onun içine direkt import edilebilir. (USE AUTOSCOUT)**

**- Veritabanındaki her tabloya, db oluşturulduktan sonra ‘*id’* kolonu manuel olarak eklenmelidir !**

**(id kolonu oluşturmayı sona bırakmamın sebebi, primary key olarak eklenmesini gerektiğindendir)**