Görevler

1. **FastAPI Oluşturma**
   FastApi ile bir WebAPI oluşturulacak. Api de aşağıdaki methodlar(route) yer alıyor olmalı;
   - PUT(SendEvent)
     Tek seferde 1 object alabilir; alınacak object aşağıdaki gibi olmalı;
     - UserId
     - SessionId
     - EventName(PageVisited, AddedBasket, CheckedProductReviews)
     - TimeStamp
     - Attributes
       - ProductId
       - Price
       - Discount
   - POST(PurchasedItems)
     Bu method birden fazla nesne alıyor olabilri, yani methodun parametresi Array cinsinden olmalı, tek seferde birden fazla kayıt gönderebilmeliyim.
     - SessionId
     - TimeStamp
     - UserId
     - TotalPrice
     - OrderId
     - Products[Array]
       - ProductId
       - ItemCount
       - ItemPrice
       - ItemDiscount
     - PaymentType

   Yukarıdaki methodlara gelen kayıtlar Apache Kafka'da oluşturulacak UserEvents ve PurchasedItem isimli iki farklı Topic'e iletilmesi gerekiyor.

2. **Data Generator Oluşturma**
   Bir Python projesi oluşturulmalı. Projede **Faker** isimli kütüphane kullanılarak 1. adımda oluşturduğunuz WebAPIdeki iki methoda fake veriler gönderen bir yapı kurmalısınız. Her saniye yeni bir kitle oluşturup göndermelisiniz. Her bir kayıt Faker ile üretilmeli.

3. **Airflow DAG Oluşturma**
   Airflow da çalışacak bir DAG oluşturlmalısınız. DAG'te oluşturacağınız Task'lar her 2 dakikada bir çalışmalı ve Kafka'daki UserEvents(SendEvent methoduna gelen veriler) Topic'ne Subscribe olarak kuyrukta bekleyen işlerin hepsini topluca alıp Mongo DB'ye oluşturacağınız bir Collection'a insert etmeli. İlk task tamamlandıktan sonra sonraki Task ise Mongo DB'ye kaydettiğiniz UserEvents'lerden aşağıdaki aggregation sonucunu hesaplayarak başka bir Collectiondaki verileri güncellemeli. Veri şeması aşağıdaki gibi olmalı;
   - UserId

- EventName
- EventCount

Yani; her bir UserId için X eventName'inden kaç tane kayıt var bilgisini ilgili collection'a kaydetmeniz gerekiyor. aynı UserId ve EventName'den zaten varsa da EventCount'u güncellemeniz gerekiyor.

4. **PySpark için Python Projesi**
   Bir Python projesi oluşturmalısınız. Proje çalışmaya başladığında PySpark Session'ı oluşturmalı(Lokal makinanıza Pyspark'ı pip ile eklemelisiniz). Spark ReadStream ile Kafka'daki PurchasedItem isimli Topic'e subscribe olmalısınız. Gelen verileri Hadoop HDFS üstünde istediğiniz Path'e WriteStream ile raw haliyle yazdırmalısınız(Verileri Parquet formatlı yazdırın). Proje çalıştığı sürece Kafka'ya gönderilen PurchasedItems'lar hdfs'e kayıt ediliyor olmalı(Burada WriteStream anlık yazma yapmaz, biriktirerek yazma yapar, denemelerinizde sabırlı olmalısınız).

5. **JupyterNotebook üstünde analiz**
   Notebook üstünde Hadoop HDFS üstüne kaydettiğiniz PurchasedItems kayıtlarını normal şekilde okuyun(spark.read.parquest("hdfs://…/") -> sadece üst dizini vermeniz yeterli, o altındaki tüm dosyaları okuyacaktır).

   Analiz etmenizi istediğim bilgiler aşağıdadır;
   - En çok satılan ürünler hangileri
   - En çok tercih edilen ödeme tipi nedir
   - Son 1 saatte en yüksek tutarlı siparişi veren top 10 müşteriler hangileri
   - (****)Aynı ürünü birden çok kez satın alan müşteriler ve birden çok aldıkları ürünler

   **** ile başlayan analizin çıktısınız Postgres üstünde oluşturacağınız veritabanındaki bir tabloya istediğiniz formatta tabloyu önce truncate sonra insert veya önce tüm kayıtları delete sonra insert yaparak kaydetmelisiniz.

Ayrıca başka bir notebook oluşturarak orada da Postrges'e kaydettiğiniz verilerden aşağıdaki sonucu bulmanız gerekmektedir;

- En çok tekrar tekrar satın alınan en popüler ilk 10 ürün hangisidir sonucunu veren SQL sorgusunu yazınız.

Yukarıdaki maddelerdeki tüm isterleri yapmalısınız fakat takıldığınız noktada çalışacağını düşündüğünüz kodları yazarakta ilerlenebilsiniz. Herkes kendi ödevini kendisi yapmalı ama kolaylık olsun diye sadece size 1 adet jokeri hakkı veriyorum 🙂, bu 1 joker ise yukarıdaki 5 sorudan sadece 1 tanesini o soruyu yapan başka bir arkadaşınızdan destek isteyerek çözebilirsiniz.

Yukarıdaki maddelerde öğrenmediğimiz kısımlar var(ama zor olmaya şeyler), burada amaç araştırma çabanızı ve bilgi edinebileceğiniz web sayfalarının farkına varmanızı sağlamak. O yüzden olabildiğince her soruyu elinizden geldiğince kendiniz yapmaya çalışın ;)

Takıldığınız yerlerde herzaman bana yazabilirsiniz ;) Başarılar…