Project 2
Big Data Processing Systems

João Camacho, N. 56861, Analysis and Engineering of Big Data
Simão Gonçalves, N. 54896, Analysis and Engineering of Big Data

I. Introduction

This project intends to use Hive to create indexes that answer 4 questions about the dataset.

II. Dataset

The dataset is a 220Mb csv with about 1.4 million rows regarding taxi rides in a certain city. The data includes columns such as: pickup and dropoff times and locations (location is in latitude and longitude), the distance of the ride, the passenger count and weather conditions.

A. Dataset preprocessing

We realised that some calculated fields were being used in multiple exercises, therefore we created the table *taxis_preprocessed* that performs those operations to save computational resources later on. These calculations are:

- rounding pickup/dropoff longitude and latitude
- creating pickup/dropoff area by concating longitude and latitude
- Fetching the weekday

We realise this table could just be a view so not to spend more storage resources, but given the small dataset at hand we decided to leave it as a table.

III. Exercise 1

We are asked to create an index that answer the following question: *What are the 10 most frequent routes (pickup-area, dropoff-area), given a weekday (0=sun, 6=sat) and hour (no minutes)?*

We begin by creating the view *P1_preprocessed_view* which contains the operations:

- concatenating the pickup area and dropoff area together to represent the route;
- getting the hour of each pickup datetime;

After this we created groups on the fields: week_day, hour and route, and computed the number of routes for each group. This view, *P1_answer_all_routes_view*, contains the desired index of the exercise with the exception that we still need to filter in only the top 10 routes of each group: week_day, hour.

To filter the top 10 routes for each weekday and hour of the day, we used a functionality of Hive called ranking window functions which can rank rows inside each group according to the value of each row. This is useful because we can then query only the rows of each group whose rank value is below a threshold n, in order to get the top n rows of each group.

IV. Exercise 2

This exercise asks us to create the following index: *What is the expected duration and distance of a taxi ride, given the pickup area, the weekday (0=sun, 6=sat) and time? Consider time intervals of 15 minutes.*

We first created the view *time_preprocessed_view* which converts all the pickup times to the nearest 15 minute bin interval and also compute here the duration of the ride in minutes. It also fetches the distance of the rides and pickup area.

And secondly (and lastly) we create the table with the final answer, which groups the data by the desired columns: pickup_area, week_day and time (note that the time is in 15 minute bins), and computes the average of the distance and duration of the rides for each group.

V. Exercise 3

In this exercise we were asked to make an index that answers the following question: *What is the factor of additional clients when it is raining or snowing (merge these conditions into one) versus dry weather, given the pickup-area?* To solve this we do the following steps:

1) We create a simple view with all the columns that are gonna be used to subset the data and simplify the problem. We picked the following columns
   - precipitation
   - snow_fall
   - pickup area

   The pickup area was obtained by concatenating the pickup latitude and longitude , similarly to what we did in the previous exercises. The precipitation and snow_fall we add together for simplicity, we called this variable *(*weather), with this we can determine if there is any kind of precipitation happening.
2) We create two additional views. One with the pickup_area and the rows with bad weather $weather$. We select these rows using a threshold of 0.05 for the weather column. Similarly for the second view, we select the rows with values above the threshold. This threshold is where we consider there is an reasonable amount of precipitation to the point where people would be willing to pickup a taxi to get to their destination (for example).

3) In this last step we do the final select with the two previous views we group by the key $\boldsymbol{pickup\_area}$ and calculate the factor of the counts from both views:

$$Factor = \frac{N_{rain}}{N_{notrain}} \tag{1}$$

We get the counts from both views by doing a inner join with both of them and joining them where the pickup_area is equal for both.

VI. Exercise 4

In this exercise we made our own index that answers the following question: *What is the fastest taxi company per route?*. This is motivated by the fact that we noticed there are only two taxi services in this dataset so it would be interesting to compare both of them in terms of speed in the same routes.

1) We create an initial view with all the columns that we need
   - travel_time (in seconds)
   - route
   - vendor_id
2) After that get the average travel time by route and vendor
3) Then we separate into two views the information of each vendor
4) Lastly we join the information in one table so that each row represents a route and we can compare the avg travel time of each vendor in the two columns.