**SCRIPT AND EXECUTION:**

There is one python script for performing the ETL of raw data (userDetails.json and placeDetails.json). It also contains the methods for performing the operations required for the given analytical questions 1-3.

The script uses spark distributed computing framework for performing the ETL. It can be scaled up depending on the size of the raw data being processed.

The script can be run with the following command:

spark-submit --master yarn --deploy-mode client --files ./etl.prop rest.py etl.prop

rest.py is the main script.

The name of the properties file must be passed as the first argument.

etl.prop contains the necessary configs required for performing the operations. Sample properties file:

[conf] #required. Please use the same section name.

spark.restetl.baselocation=/tmp/sony/ #required

spark.restetl.top3.execute=true #optional

spark.restetl.top3.fromdate=2020-01-01 #optional

spark.restetl.top3.untildate=2020-05-01 #optional

spark.restetl.nthtop.execute=true #optional

spark.restetl.nthtop.n=5 #required if above config nthtop.execute=true

spark.restetl.nthtop.fromdate=2020-01-01 #optional

spark.restetl.nthtop.untildate=2020-05-01 #optional

spark.restetl.avghours.execute=true #optional

spark.restetl.avghours.date=2020-04-10 #required if above config avghours.execute=true

spark.restetl.baselocation contains the raw data files. The script saves the processed data to the same location into different folders.

The script does not perform the analytical operations unless it is specified in the etl.prop file like given above.

The date configs for questions 1,2 (spark.restetl.top3.fromdate, spark.restetl.top3.untildate, spark.restetl.nthtop.fromdate, spark.restetl.nthtop.untildate) are optional. All or none of these can be provided in the etl.prop file. If none of them are provided, the script runs the operations on the entire data.

**DATA MODEL:**

Besides the analytical operations, the script reads the raw data (userDetails.json and placeDetails.json), performs some transformations like separating out the facts and dimensions, renaming columns, changing data types of columns and saving the raw data into 3 datasets. The three datasets are:

**visits:**

|-- food\_rating: integer (nullable = true)

|-- place\_id: string (nullable = true)

|-- rest\_rating: integer (nullable = true)

|-- sales\_amount: double (nullable = true)

|-- service\_rating: integer (nullable = true)

|-- visit\_date: timestamp (nullable = true)

|-- user\_id: string (nullable = true)

Visits will be the facts table with all the numerical values. It contains user\_id and place\_id dimensions as references to user\_details and place\_details dimension tables. User\_details and place\_details contain all the descriptive values of users and restaurants respectively.

**User\_details**:

|-- activity: string (nullable = true)

|-- ambience: string (nullable = true)

|-- birth\_year: string (nullable = true)

|-- budget: string (nullable = true)

|-- dress\_preference: string (nullable = true)

|-- drink\_level: string (nullable = true)

|-- fav\_cuisine: array (nullable = true)

| |-- element: string (containsNull = true)

|-- height: string (nullable = true)

|-- children: string (nullable = true)

|-- interest: string (nullable = true)

|-- latitude: string (nullable = true)

|-- longitude: string (nullable = true)

|-- marital\_status: string (nullable = true)

|-- personality: string (nullable = true)

|-- religion: string (nullable = true)

|-- smoker: string (nullable = true)

|-- transport: string (nullable = true)

|-- user\_id: string (nullable = true)

|-- user\_payment\_methods: array (nullable = true)

| |-- element: string (containsNull = true)

|-- weight: string (nullable = true)

**Place\_details**:

|-- accepted\_payments: array (nullable = true)

| |-- element: string (containsNull = true)

|-- open\_hours: array (nullable = true)

| |-- element: struct (containsNull = true)

| | |-- days: string (nullable = true)

| | |-- hours: string (nullable = true)

|-- parking\_type: array (nullable = true)

| |-- element: string (containsNull = true)

|-- place\_id: string (nullable = true)

|-- served\_cuisines: array (nullable = true)

| |-- element: string (containsNull = true)

To keep the historical data, current date (named ‘day’) and hour of processing (named ‘hour’) are being added as extra columns to all the datasets. These columns are used to maintain partitions for all the datasets. So, the max(day) and max(hour) in each dataset will contain the latest data for that dataset.

**OUTPUTS OF ANALYTICAL FUNCTIONS:**

1. Top 3 restaurants per cuisine by sales:

|-- place\_id: string (nullable = true)

|-- total\_amount: double (nullable = true)

|-- served\_cuisine: string (nullable = true)

|-- rank: integer (nullable = true)

|-- from\_date: string (nullable = true)

|-- until\_date: string (nullable = true)

|-- day: string (nullable = false)

|-- hour: string (nullable = false)

1. Nth top restaurant for all the cuisines by sales:

|-- place\_id: string (nullable = true)

|-- total\_amount: double (nullable = true)

|-- served\_cuisine: string (nullable = true)

|-- rank: integer (nullable = true)

|-- from\_date: string (nullable = true)

|-- until\_date: string (nullable = true)

|-- day: string (nullable = false)

|-- hour: string (nullable = false)

1. avg hours between consecutive visits to any restaurants on a given date:

|-- user\_id: string (nullable = true)

|-- avg\_diff\_in\_hours: double (nullable = true)

|-- date\_ran: string (nullable = false)

|-- day: string (nullable = false)

|-- hour: string (nullable = false)

All the datasets are being saved in parquet file format. For the sample files that are uploaded to the repository, the results for 1st and 2nd questions don’t have a time range. They are generated from the whole data. N value for 2nd question is N=5. For the 3rd question, the date used is ‘2020-05-10’.

I have used combinations of aggregations like average, sum, max with group by operation along with windowing functions like row\_number, lag for the calculations of analytical questions. Going into the details of every step in this document might not be clear. I can explain the details in a verbal discussion.