Analyzing NYC Taxi Trips Data

For Big Data and Hadoop for Beginners - with Hands-on!

Instructor: Andalib Ansari

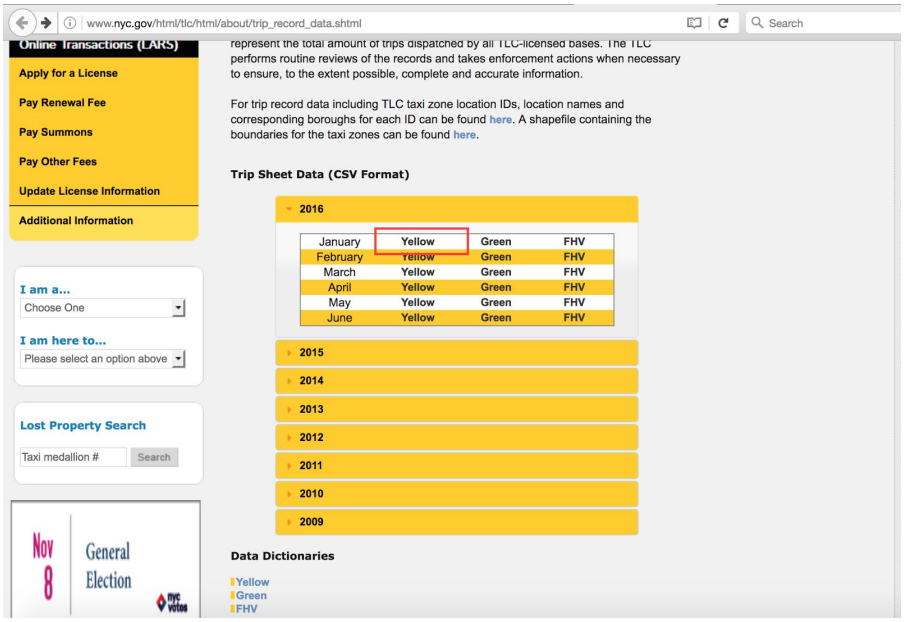
Data Source: This is a public data sets provided by NYC Taxi. The yellow and green taxi trip records include fields capturing pick-up and drop-off dates/times, pick-up and drop-off locations, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts.

URL: http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

We will be analyzing **Yellow Taxi** data. Below is a sample trip records:

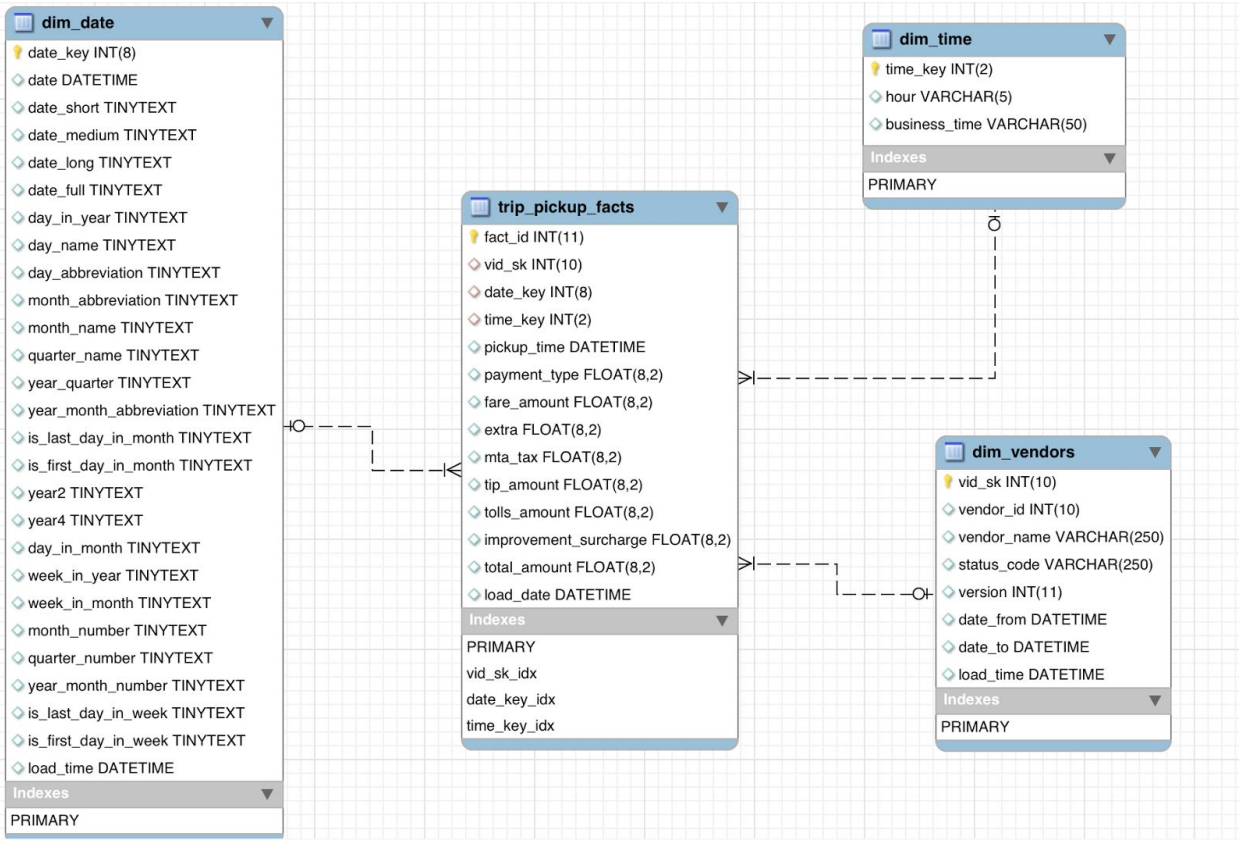
VendorID,tpep_pickup_datetime,tpep_dropoff_datetime,passenger_count,trip_distance,pickup_longitude,pickup_latitude,RatecodeID,store_and_fwd_flag,dropoff_longitude,dropoff_latitude,payment_type,fare_amount,extra,mta_tax,tip_amount,tolls_amount,improvement_surcharge,total_amount
2,2016-01-0100:00:00,2016-01-01
00:00:00,2,1.10,-73.990371704101563,40.734695434570313,1,N,-73.981842041015625,40.732406616210937,2,7.5,0.5,0.5,0,0,0.3,8.8

There are more than **100 Million+** records in each month, and, on an average, approximate file size of a single month is **1.6+ GB**. Make sure you have good internet connectivity before you start downloading the data. I have downloaded 2016 January data as shown below.



DataDictionary:
http://www.nyc.gov/html/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf

Objective: We will design data Data Warehouse schema in Hive. The designed DWH schema will be based on Star Schema where we can slice and dice data based on Date dimensions and Business Timings. This has been designed in MySQL workbench, and you need to design the same Model in **Hive**. Ignore **dim_vendors** dimension table as it is a SCD Type-2 Dimension table.



Note: For Data Warehouse basics, refer below link:
http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-techniques/

Trip Pickup Facts: This is a transactional fact table where each record represents a business event (pick-up event). It is having three dimension tables namely; Date Dimension, Time Dimension and Vendors Dimension. Time Dimension's keys hold hour's numbers i.e. 0,1,2,..,23. To give flexibility on minute level aggregation, I have also included pick-up timestamp field (pickup_time) in the fact table.

Time Dimension: Below is how I have defined business timings. This is a one time generated dimension table. Whenever we will be inserting data into the fact table, we will be extracting hour numbers from the pick-up timestamp as a time_key.

time_key	hour_name	Business Timing
0	12am	Late Night

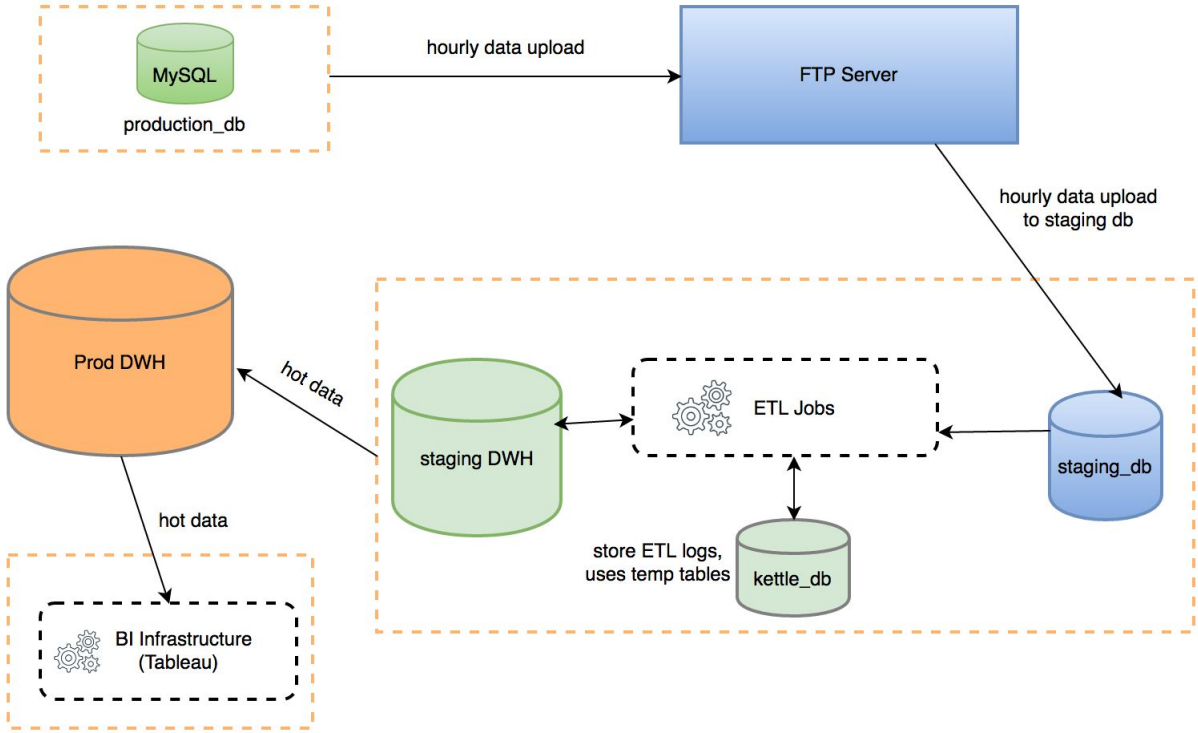
1	1am	Late Night
2	2am	Late Night
3	3am	Early Morning
4	4am	Early Morning
5	5am	Early Morning
6	6am	Early Morning
7	7am	AM Peak
8	8am	AM Peak
9	9am	AM Peak
10	10am	Mid Morning
11	11am	Mid Morning
12	12pm	Lunch
13	1pm	Lunch
14	2pm	Mid Afternoon
15	3pm	Mid Afternoon
16	4pm	Mid Afternoon
17	5pm	Evening
18	6pm	Evening
19	7pm	PM Peak
20	8pm	PM Peak
21	9pm	PM Peak
22	10pm	Night
23	11pm	Night

Date Dimension: The date dimension corresponds to many derived date attributes (as shown in above schema). I have a generated a Date lookup (date_lookup) table which is having 10 years of future lookup dates (this is can be generated as per the business need). During each ETL process new date attributes will be inserted (if the date attributes does not exist in the dimension table). ETL process will lookup the date_lookup table to pull all, given date_key, date attributes which will be inserted into the data_dimension table.

Your Assignment:

1. *Understand raw data, refer data dictionary, and download a single month data. Upload downloaded data into HDFS. Once you perform a sample test with below design, then download other months' data sets.*
2. *Design the same data model in **Hive** (refer above shown Schema Design)*
3. *Design **Date Dimension** table (refer attached files for schema design which I have designed using MySQL Workbench)*
4. *Design Time **Dimension** table (refer attached files for schema design which I have designed using MySQL Workbench)*
5. *Design a **Trip Pickup Fact** table which is having date partitioning, and storage as ORC. Think if you can better optimize this fact table by keeping in mind that this fact table is going to have Billions of rows down the line. While designing you need to consider Data Scan Cost, Query Performance, Storage Cost and etc.. For more details refer below link:*
   https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DDL
   https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DML
6. *Write Hive scripts to load data into Dimension and Fact Tables. Refer attached files to refer sample Date Dimension and Time Dimension data. You can import those SQL files into MySQL to see how Date and Time Dimension looks like, and then you can export those data as a text file and upload them into HDFS to create your Date and Time Dimension tables.*
7. *Run Different Group By, Order By and Join Queries to analyze data.*
8. *Think about what Other Dimension tables you can add in the same schema to give better business insights.*

   ***Hint:** City Area Dimension: Since there are pick-up latitude and longitude, by resolving these data (like hitting some Map API we can get geo location details), we can then define city area dimensions like i.e. South, East, West, North, Central to understand area based business performance.*
9. *Think about Automating this entire process. Assume if you get the data in hourly basis, how you can automate the data download process and data load process to insert new data into Dimension and Fact Tables. For example, you can assume the entire Data Pipeline as below:*



All The Best..!! Comment your questions if you have any doubts while doing this exercise.