CS689 Data Warehouse Final Project

JiaLiang Ma U90151356

**Project Scope**

The topic is about building a full-stack data warehouse with ETL pipelines, and visualization plots or queries based on the questions we want to answer.

In this project, I plan to use the datasets of NYC Taxi Trip Duration and NYC Airbnb Open Date. These two data are deeply relative to New York City’s tourism activity. The goal is to explore the relationship between taxi passengers, trip duration and Airbnb rentals in NYC and provide insights by using the data warehouse. Finding the correlation between rental rate and taxi taking rate and so on.

Business Questions:

1. Does people who rent in high average rental rate region also pay high fare for their taxi rides?
2. What types of Airbnb rooms generate the highest revenue and how much in the top 5 neighborhoods with the highest average rental rates?
3. What is the most common pick-up and drop-off locations for taxis in NYC?
4. What is the relationship between Airbnb density and the taxi taking rate?
5. What are the top months with the highest month-over-month revenue growth rate for the NYC taxi industry compared to the last year?

**Datasets**

Train.csv: (columns: 11, rows: 1458644)

Background pattern

Description automatically generated

Table

Description automatically generated

Test.csv: (columns: 9, rows: 625134)

Table

Description automatically generated

1. New York City Airbnb Open Data

AB\_NYC\_2019.csv (columns: 16, rows: 48895)

A picture containing website

Description automatically generated

Table

Description automatically generated

* What is the URL or location of the data?

1. NYC Taxi Trip Duration: [**https://www.kaggle.com/c/nyc-taxi-trip-duration/data**](https://www.kaggle.com/c/nyc-taxi-trip-duration/data)
2. NYC Airbnb Open Data: [**https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data**](https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data)

**What information does this data provide that will help answer one or more of the above questions?**

This data provides valuable insights that can help answer several of the above questions. The latitude, longitude, and neighborhood information from the NYC Airbnb dataset allows us to determine the locations of Airbnb properties. Additionally, the drop-off longitude, latitude, and drop-off datetime in the NYC Taxi Trip Duration dataset reveal the most common destinations for passengers, as well as the times they arrive. By analyzing the locations, we can explore potential relationships between Airbnb properties and taxi usage rates. This information may also enable us to assess the convenience and accessibility of various neighborhoods, which can further aid in understanding the factors influencing taxi and Airbnb demand in NYC.

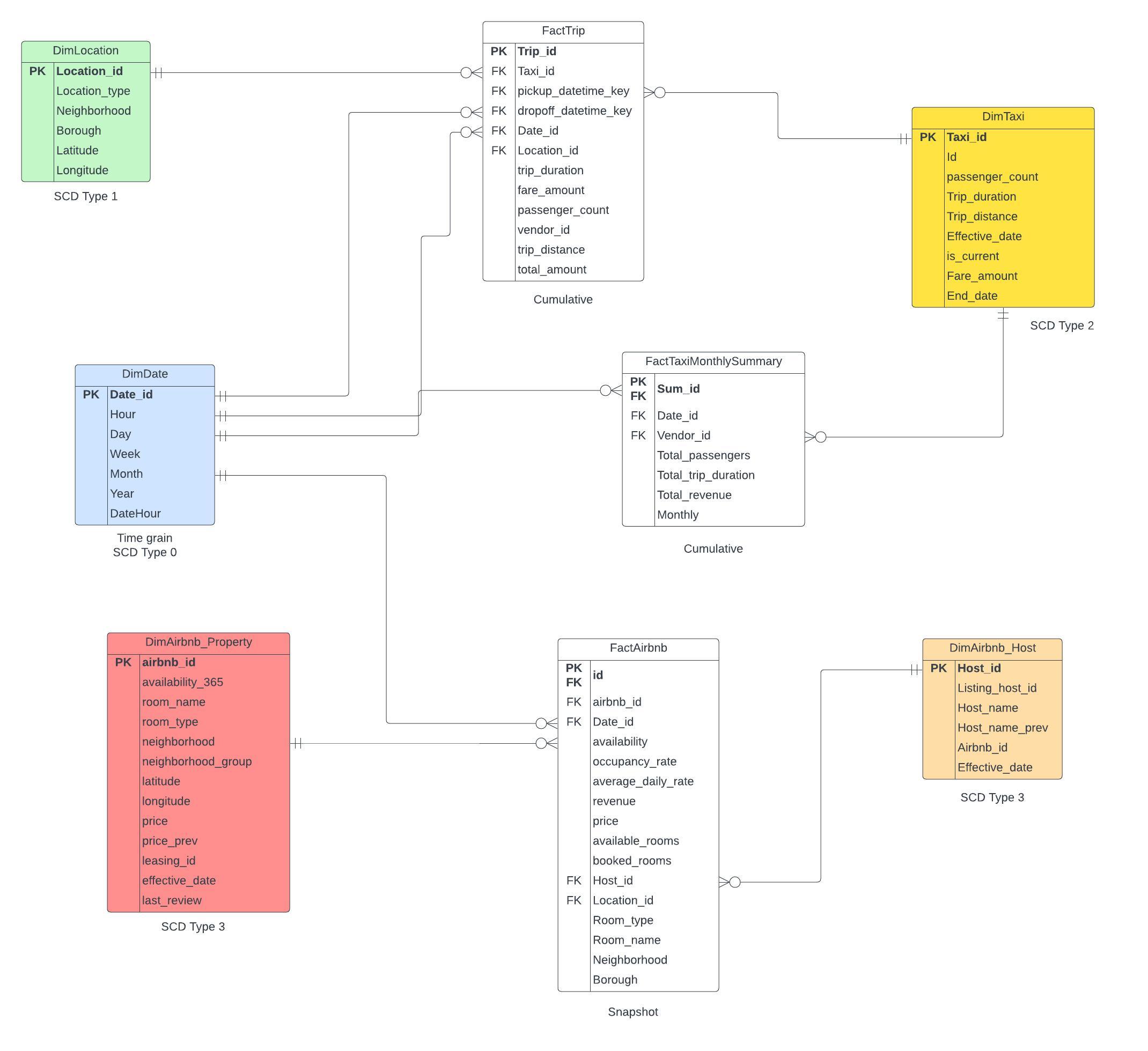
Also, for the detailed information of Airbnb, we can further analyze the room preferences and locations for the renters.

**Do you see any issues in the data that will require transformation.**

The length (amount) of rows in these two datasets has a huge difference. Therefore, we may need to sample the taxi dataset for better combination. Luckily, both datasets have latitude and longitude columns, which make the analysis more easily.

Also, in taxi dataset, train and test datasets have the different columns, which will need to be integrated.

**Constellation Schema**



**Dimensions - Review the data and the business questions from part 2.**

* What fields (attributes) are in the data that will be used for the dimensions.

Based on the questions above, we can identify the attributes that can be used for the dimensions from both datasets. In the preliminary stage of selecting attributes, we will focus on

NYC Taxi Trip Duration:

* + pickup\_datetime
  + passenger\_count
  + pickup\_longitude
  + dropoff\_longitude
  + trip\_duration
  + fare\_amount
  + vendor\_id
  + store\_and\_fwd\_flag

NYC Airbnb Open Data:

* + longitude
  + latitude
  + host\_id
  + neighbourhood
  + neighbourhood\_group
  + room\_type
  + price
  + host\_name
  + calculated\_host\_listings\_count
  + name
  + last\_review
  + review\_per\_month
  + availability\_365
* Determine the dimension tables. There should be at least two non-date dimensions and one date dimension for each fact table.

Dimension table:

DimDate:

Date\_id (PK)

Hour

Day

Week

Month

Year

DateHour

DimLocation:

Location\_id (PK)

Location\_type

Neighborhood

Borough

Latitude

Longitude

DimTaxi:

Taxi\_id (PK)

Id (taxi\_id in taxi\_staging)

Passenger\_count

Trip\_distance

Trip\_duration

Fare\_amount

Effective\_date

End\_date

Is\_current

DimAirbnb\_Property:

Airbnb\_id (PK)

Leasing\_id (original Airbnb\_id)

Room\_name

Room\_type

Price

Neighborhood

Neighborhood\_group

Latitude

Longitude

Availability\_365

Last\_review

Price\_prev

Effective\_date

DimAirbnb\_Host:

Host\_id (PK)

Listing\_host\_id (original Host\_id)

Airbnb\_id (FK)

Host\_name

Host\_name\_prev

Effective\_date

Fact table:

FactTrip (Cumulative)

Trip\_id (PK)

Taxi\_id (FK)

Location\_id (FK)

Date\_id (FK)

Pickup\_datetime\_key (FK to DimDate)

Dropoff\_datetime\_key (FK to DimDate)

Passenger\_count

Trip\_distance

Trip\_duration

Fare\_amount

Total\_amount

Vendor\_id

FactAirbnb (Snapshot)

Id (PK)

Airbnb\_id (FK)

Date\_id (FK)

Host\_id (FK)

Location\_id (FK)

Room\_name

Room\_type

Neighborhood

Borough

Availability

Price

Revenue

Available\_rooms

Booked\_rooms

Occupancy\_rate

Average\_daily\_rate

FactTaxiMonthlySummary (Cumulative)

Sum\_id (PK)

Date\_id (FK)

Taxi\_id (FK)

Total\_passengers

Total\_trip\_duration

Total\_revenue

Monthly

**At least one (non-date) dimension in your design should have a hierarchy.**

For the hierarchy, we design a parent-child relationship between boroughs and neighborhoods in DimLocation. The boroughs are the higher-level grouping and neighborhoods are the lower-level grouping.

(e.g., {Boroughs: Manhattan}, {Neighborhoods: Battery Park City})

Also, in DimDate, we have a hierarchy of time from Year, month, day to time.

**What are the attributes that will be tracked via slowly changing dimensions?**

For DimAirbnb\_Property table, the price could be tracked by using SCD Type 3 with new column for the price to capture any changes over time.

Also, in the DimAirbnb\_Host table, we can use SCD Type 3 to capture the host name changes.

For DimTaxi table, we added effective date and end date as SCD Type 2 to record any changes in vendor id.

DimLocation using SCD Type 1 would be enough.

**What attributes within the dimensions will need transformation before they are loaded into the dimension, for example it could be to build consistency or any other issues? This is where for example you might build case statements in your code to handle various scenarios. Two to three examples showing some sample data and what you think the transformation will be during your ETL would be helpful here.**

1. Date parsing: In the DimDate table, the pickup\_datetime or dropoff\_datetime attribute may need to be parsed into separate columns to aggregate data by different period.
2. Dealing with missing values in some tables can be beneficial to our analysis.

Graphical user interface, table

Description automatically generated Table

Description automatically generated with medium confidence

1. Name standardization: In the DimTaxi table, the vendor’s name attribute may need to be standardized to ensure consistency.
2. Address normalization: In the DimAirbnb table, the address attribute may need to be normalized to make sure the address format is consistency. Also, normalized address can help avoiding duplication. For this problem, I would like to try deriving a new column called pickup\_neighborhood based on the latitude and longitude.
3. In the DimAirbnb table, I found some values in the neighborhood column have the different format.
4. Date parsing: In the DimDate table, the pickup\_datetime or dropoff\_datetime attribute may need to be parsed into separate columns to aggregate data by different period.
5. Dealing with missing values in some tables can be beneficial to our analysis.

Graphical user interface, table

Description automatically generated Table

Description automatically generated with medium confidence

1. Name standardization: In the DimTaxi table, the vendor’s name attribute may need to be standardized to ensure consistency.
2. Address normalization: In the DimAirbnb table, the address attribute may need to be normalized to make sure the address format is consistency. Also, normalized address can help avoiding duplication. For this problem, I would like to try deriving a new column called pickup\_neighborhood based on the latitude and longitude.
3. In the DimAirbnb table, I found some values in the neighborhood column have the different format.
4. Longitude and Latitude in Taxi\_df that need to transform to borough and neighborhoods to simpler the analysis and queries. (Not working, due to big data size. Using reverse method instead)

**Facts – Review the data and the business questions from step 1.**

* What measurements are in the data that will be used for the fact tables?

FactAirbnb:

* + - 1. Revenue
      2. Available\_rooms
      3. Booked\_rooms
      4. Occupancy\_rate
      5. Average\_daily\_rate
      6. Availability

FactTaxiMonthlySummary:

* + - 1. Total\_passengers
      2. Total\_trip\_duration
      3. Total\_revenue
      4. Monthly

FactTrip:

1. Passenger\_count
2. Trip\_distance
3. Trip\_duration
4. Fare\_amount
5. Total\_amount

DimTaxi:

1. Trip\_distance
2. Fare\_amount

* What measures will you be calculating (i.e. using an aggregate function, or some other transformation – recall as an example some of the aggregation you did in assignment 1A)

Availability: availability\_365/ 365.0

Revenue: availability\_365 \* price

Booked\_rooms = 365 - availability\_365

Occupacy\_rate = (365 - availability\_365) / 465.0

Trip\_duration: (SET Trip\_duration =

CASE

WHEN Taxi\_staging.trip\_duration IS NOT NULL THEN Taxi\_staging.Trip\_duration

ELSE DATEDIFF(second, pickup\_datetime, dropoff\_datetime)

END

FROM DimTaxi

JOIN Taxi\_staging ON DimTaxi.Id = Taxi\_staging.taxi\_id;)

Trip\_distance: (CASE

WHEN ts.Trip\_duration IS NOT NULL THEN ts.trip\_duration \* 0.2

ELSE 0

END,)

Fare\_amount: (CASE

WHEN ts.Trip\_duration IS NOT NULL THEN ts.trip\_duration \* 0.05 + 10

ELSE 0

END)

Total\_passenger: SUM(Passenger\_count)

Total\_trip\_duration: SUM(Trip\_duration)

Total\_revenue/ Total\_amount: SUM(Fare\_amount)

Monthly: FORMAT(d.DateHour, 'yyyy-MM') AS Monthly