accenturetechnology

DATABRICKS SUPER-30
DATABRICKS DATA ENGINEERING
CASE STUDY

CASE STUDY

accenturetechnology

NEW YORK CITY TAXI DATA

accenturetechnology

Index

2 Background

- New York City has two types of taxies: yellow and green; they are widely recognizable symbols of the city.

- Taxis painted **yellow** (medallion taxis) can pick up passengers from anywhere in the five boroughs.

- Those painted apple **green** (street hail livery vehicles, commonly known as "boro taxis"), which began to appear in August 2013, are allowed to pick up passengers in Upper Manhattan, the Bronx, Brooklyn, Queens (excluding LaGuardia Airport and John F. Kennedy International Airport), and Staten Island.

- Both taxi types have the same fare structure.

- Taxicabs are operated by private companies and licensed by the ***New York City Taxi and Limousine Commission (TLC).***

- It also oversees over 40,000 other **for-hire vehicles (FHVs)**, including "black cars" like Uber, commuter vans, and ambulettes.

- All types of taxis are licensed by the ***TLC*** which oversees for-hire vehicles, taxis, commuter vans, and paratransit vehicles.

- **Accenture** is responsible for developing and maintaining the data and analytical systems for New York City taxi.

3 Challenges

- Things were smooth until the recent arrival of FHVs in the scene. Though the system was working well, the challenge started when Accenture started collecting, storing, and processing the data for FHVs.

- FHVs are of multiple types:
  - Community cars, Black cars, Luxury limousines
  - High volume for-hire services which include app-based companies like Uber and Lyft. These dispatches are more than 10,000 trips per day.
- The request for FHVs by passengers is accepted by bases, and then the bases dispatch the request to the cab drivers.
- There are more than 750 bases and 100000 FHVs, and all these different types of FHVs operate in different ways.
- The number of FHVs are much higher than yellow and green taxis, which led to exponential increase in data volumes.
- The schema for FHV data is a mix of various data formats like CSV, TSV and JSON formats. The sources of this data are quite disparate.

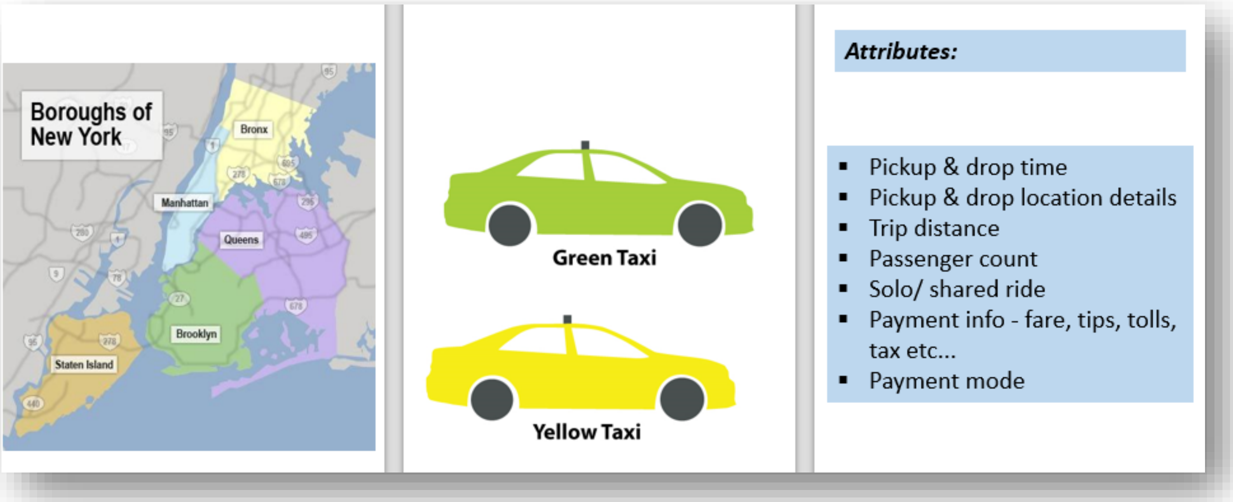
4 Business Need

- The requirement is to stage, process and store the data of all types of taxis irrespective of their sources and formats and transform it to the analytical needs.
- Finally build reports/visualizations which provide actionable insights.
- Most of this is needed in real time. There is an increase in demand for stream processing data as well due to higher trip rate.
- The requirement is to ingest and process the data at very high frequencies. Finally, none of the data should be discarded. In fact, it should be preserved for enabling use-cases related to regulatory compliance, passenger safety, insurance, targeted ads/promotions/offers etc.,
- TLC wants Accenture to build a common platform to store all data related to trips, cabs and passengers in order to analyze the data for better business insights like Revenue by taxi type, location, Total trips, max trips by regions etc.,

5 Proposed Solution

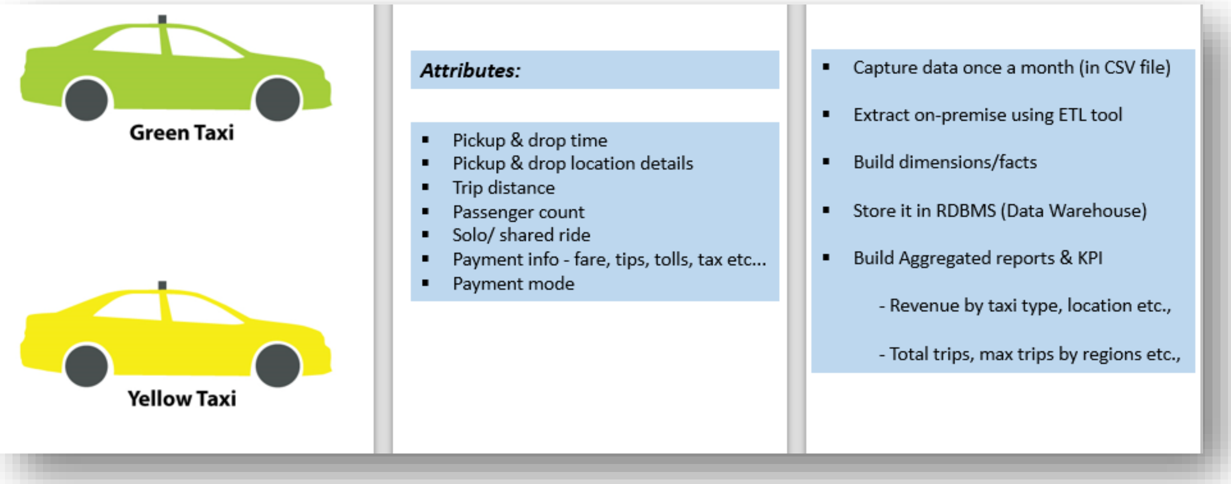
- Considering the Volume, Velocity and Veracity aspects of the data, Accenture has decided to build a Data Lake using Databricks Lake House Platform which is going to be a single store for raw, intermittent and processed data.
- Accenture has analyzed various available options and delivered some PoC's. In a one-year contract with TLC, Accenture will build a Spark based Data Lake augmented by a Cloud based Data Lake on Azure.
- The solution will implement the latest features and concepts of Databricks Lake House Platform.

6 Attributes and Sample Data



Attributes:

- Pickup & drop time
- Pickup & drop location details
- Trip distance
- Passenger count
- Solo/ shared ride
- Payment info - fare, tips, tolls, tax etc...
- Payment mode



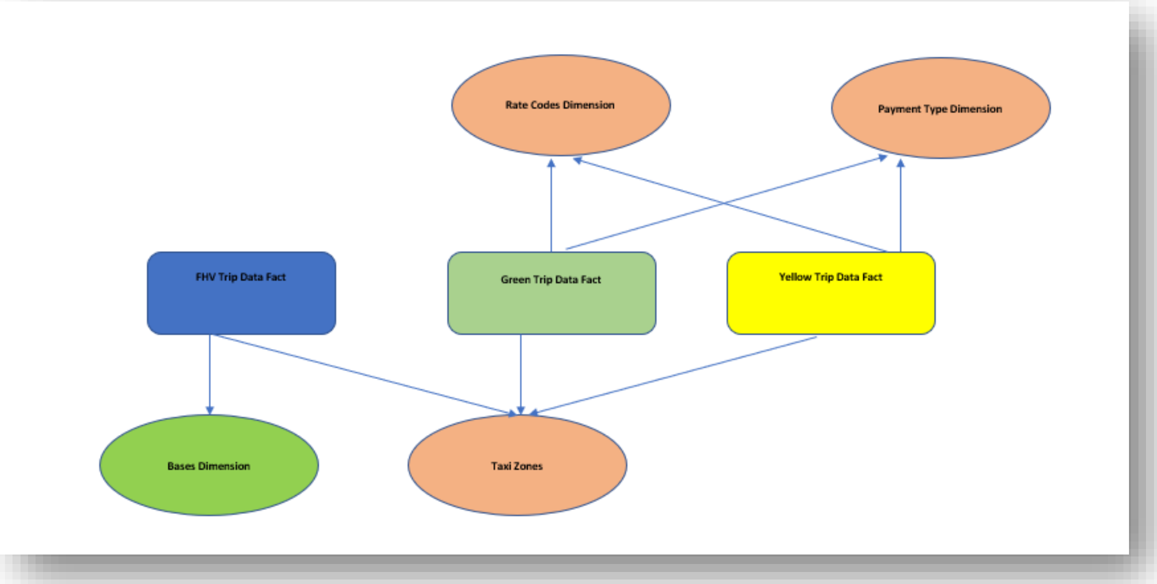
Attributes:

- Pickup & drop time
- Pickup & drop location details
- Trip distance
- Passenger count
- Solo/ shared ride
- Payment info - fare, tips, tolls, tax etc...
- Payment mode

- Capture data once a month (in CSV file)
- Extract on-premise using ETL tool
- Build dimensions/facts
- Store it in RDBMS (Data Warehouse)
- Build Aggregated reports & KPI

  - Revenue by taxi type, location etc.,

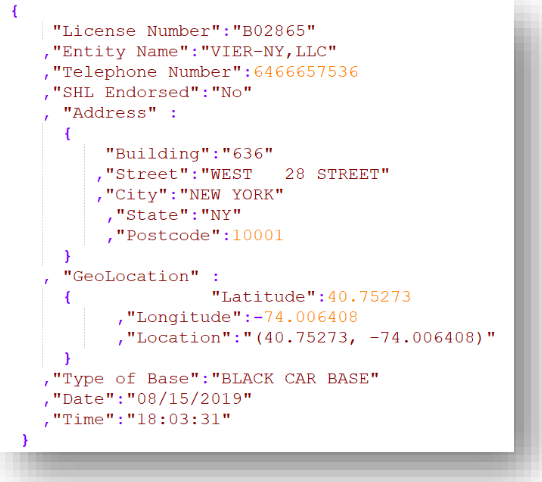
  - Total trips, max trips by regions etc.,



Sample Data: fhv_tripdata_2019-05.csv/ fhv_tripdata_2019-06.csv

dispatching_base_num	B00013
pickup_datetime	2019-06-01 00:51:33
dropoff_datetime	2019-06-01 01:20:07
PULocationID	83
DOLocationID	173
SR_Flag	Null

Sample Data: FhvBases.json



```
{
    "License Number":"B02865"
   ,"Entity Name":"VIER-NY,LLC"
   ,"Telephone Number":6466657536
   ,"SHL Endorsed":"No"
   , "Address" :
     {
        "Building":"636"
       ,"Street":"WEST   28 STREET"
       ,"City":"NEW YORK"
        ,"State":"NY"
        ,"Postcode":10001
     }
   , "GeoLocation" :
     {               "Latitude":40.75273
          ,"Longitude":-74.006408
          ,"Location":"(40.75273, -74.006408)"
     }
   ,"Type of Base":"BLACK CAR BASE"
   ,"Date":"08/15/2019"
   ,"Time":"18:03:31"
 }
```

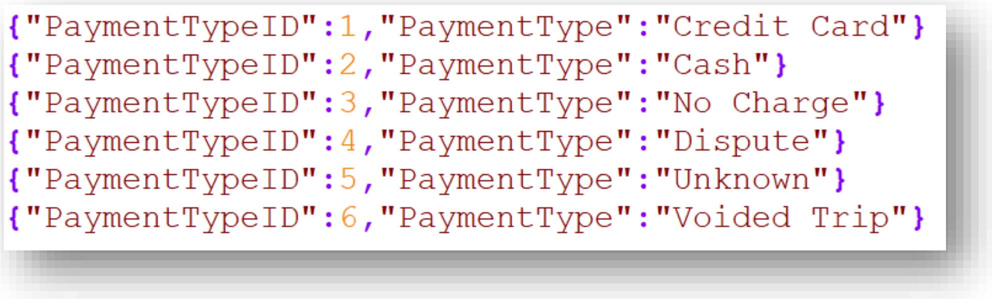
Sample Data: TaxiZones.csv

LocationID	1
Borough	EWR
Zone	Newark Airport
service_zone	EWR

Sample Data: green_tripdata_2019-05.csv/ green_tripdata_2019-06.csv

VendorID	1
lpep_pickup_datetime	2019-05-01 00:48:55
lpep_dropoff_datetime	2019-05-01 00:55:07
store_and_fwd_flag	N
RatecodeID	1
PULocationID	41
DOLocationID	42
passenger_count	1
trip_distance	1.50
fare_amount	7.5
extra	0
tip_amount	0.5
mta_tax	0
tolls_amount	0
ehail_fee	Null
improvement_surcharge	0.3
total_amount	8.3
payment_type	2
trip_type	1
congestion_surcharge	0

Sample Data: PaymentsType.json



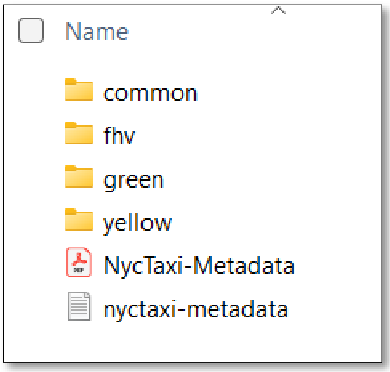
```
{"PaymentTypeID":1,"PaymentType":"Credit Card"}
{"PaymentTypeID":2,"PaymentType":"Cash"}
{"PaymentTypeID":3,"PaymentType":"No Charge"}
{"PaymentTypeID":4,"PaymentType":"Dispute"}
{"PaymentTypeID":5,"PaymentType":"Unknown"}
{"PaymentTypeID":6,"PaymentType":"Voided Trip"}
```

Sample Data: RateCodes.csv

RateCodeID	**1**
RateCode	Standard Rate
IsApproved	Yes

Source Data Files

New York Taxi datasets



☐ Name

common

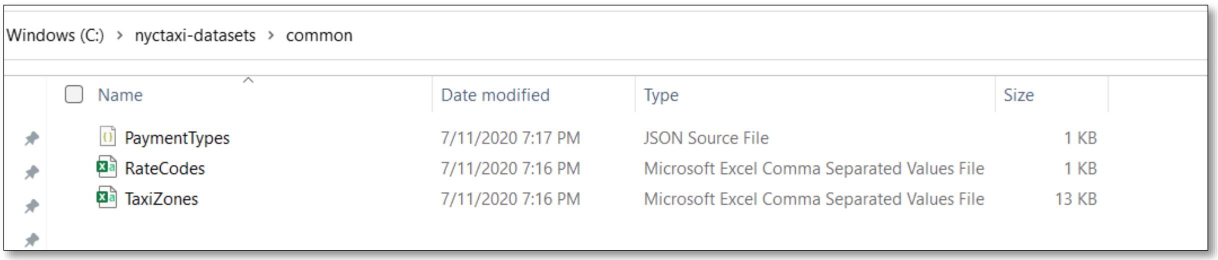
fhv

green

yellow

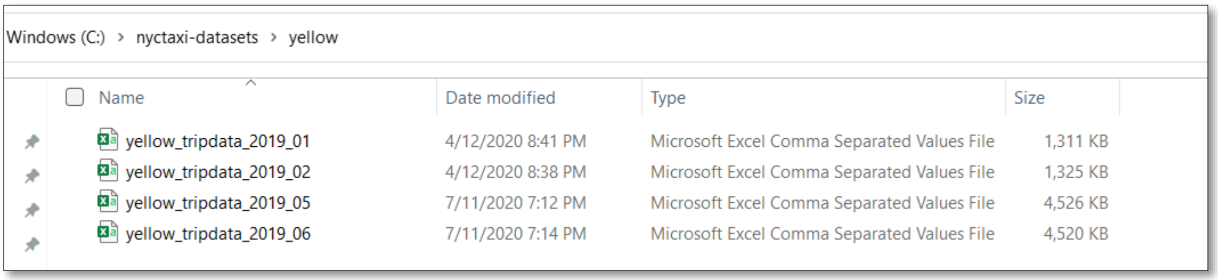
NycTaxi-Metadata

nyctaxi-metadata



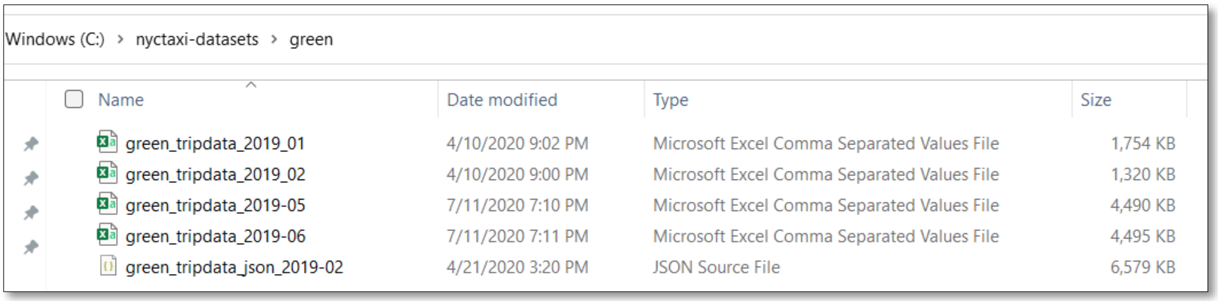
Windows (C:) › nyctaxi-datasets › common

☐ Name	Date modified	Type	Size
PaymentTypes	7/11/2020 7:17 PM	JSON Source File	1 KB
RateCodes	7/11/2020 7:16 PM	Microsoft Excel Comma Separated Values File	1 KB
TaxiZones	7/11/2020 7:16 PM	Microsoft Excel Comma Separated Values File	13 KB



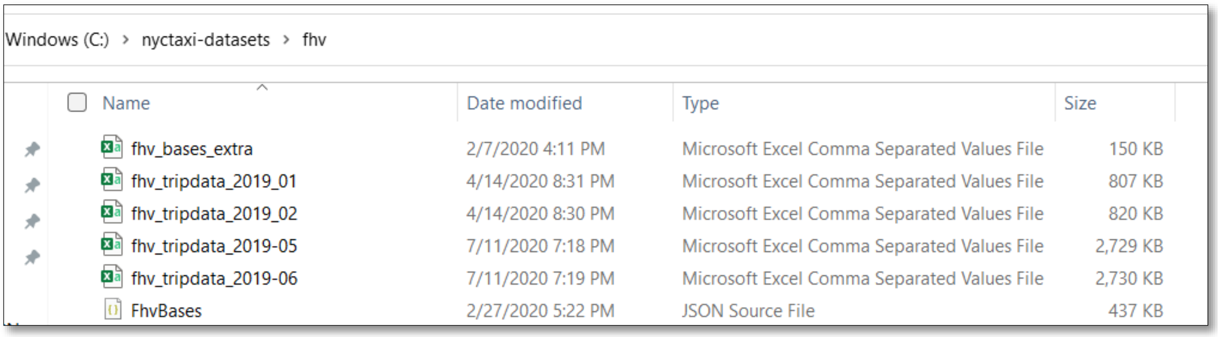
Windows (C:) › nyctaxi-datasets › yellow

☐ Name	Date modified	Type	Size
yellow_tripdata_2019_01	4/12/2020 8:41 PM	Microsoft Excel Comma Separated Values File	1,311 KB
yellow_tripdata_2019_02	4/12/2020 8:38 PM	Microsoft Excel Comma Separated Values File	1,325 KB
yellow_tripdata_2019_05	7/11/2020 7:12 PM	Microsoft Excel Comma Separated Values File	4,526 KB
yellow_tripdata_2019_06	7/11/2020 7:14 PM	Microsoft Excel Comma Separated Values File	4,520 KB



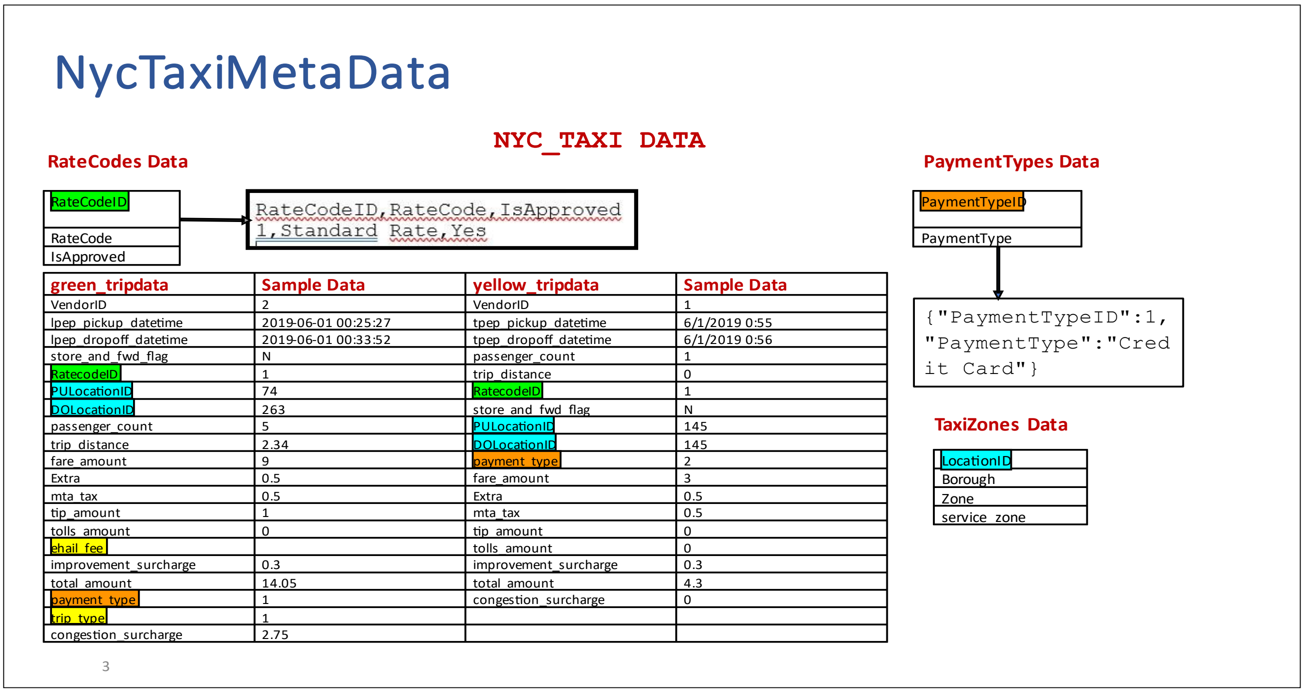
Windows (C:) › nyctaxi-datasets › green

☐ Name	Date modified	Type	Size
green_tripdata_2019_01	4/10/2020 9:02 PM	Microsoft Excel Comma Separated Values File	1,754 KB
green_tripdata_2019_02	4/10/2020 9:00 PM	Microsoft Excel Comma Separated Values File	1,320 KB
green_tripdata_2019-05	7/11/2020 7:10 PM	Microsoft Excel Comma Separated Values File	4,490 KB
green_tripdata_2019-06	7/11/2020 7:11 PM	Microsoft Excel Comma Separated Values File	4,495 KB
green_tripdata_json_2019-02	4/21/2020 3:20 PM	JSON Source File	6,579 KB



Windows (C:) › nyctaxi-datasets › fhv

☐ Name	Date modified	Type	Size
fhv_bases_extra	2/7/2020 4:11 PM	Microsoft Excel Comma Separated Values File	150 KB
fhv_tripdata_2019_01	4/14/2020 8:31 PM	Microsoft Excel Comma Separated Values File	807 KB
fhv_tripdata_2019_02	4/14/2020 8:30 PM	Microsoft Excel Comma Separated Values File	820 KB
fhv_tripdata_2019-05	7/11/2020 7:18 PM	Microsoft Excel Comma Separated Values File	2,729 KB
fhv_tripdata_2019-06	7/11/2020 7:19 PM	Microsoft Excel Comma Separated Values File	2,730 KB
FhvBases	2/27/2020 5:22 PM	JSON Source File	437 KB



NycTaxiMetaData

NYC_TAXI DATA

RateCodes Data

RateCodeID
RateCode
IsApproved

```
RateCodeID,RateCode,IsApproved
1,Standard Rate,Yes
```

PaymentTypes Data

PaymentTypeID
PaymentType

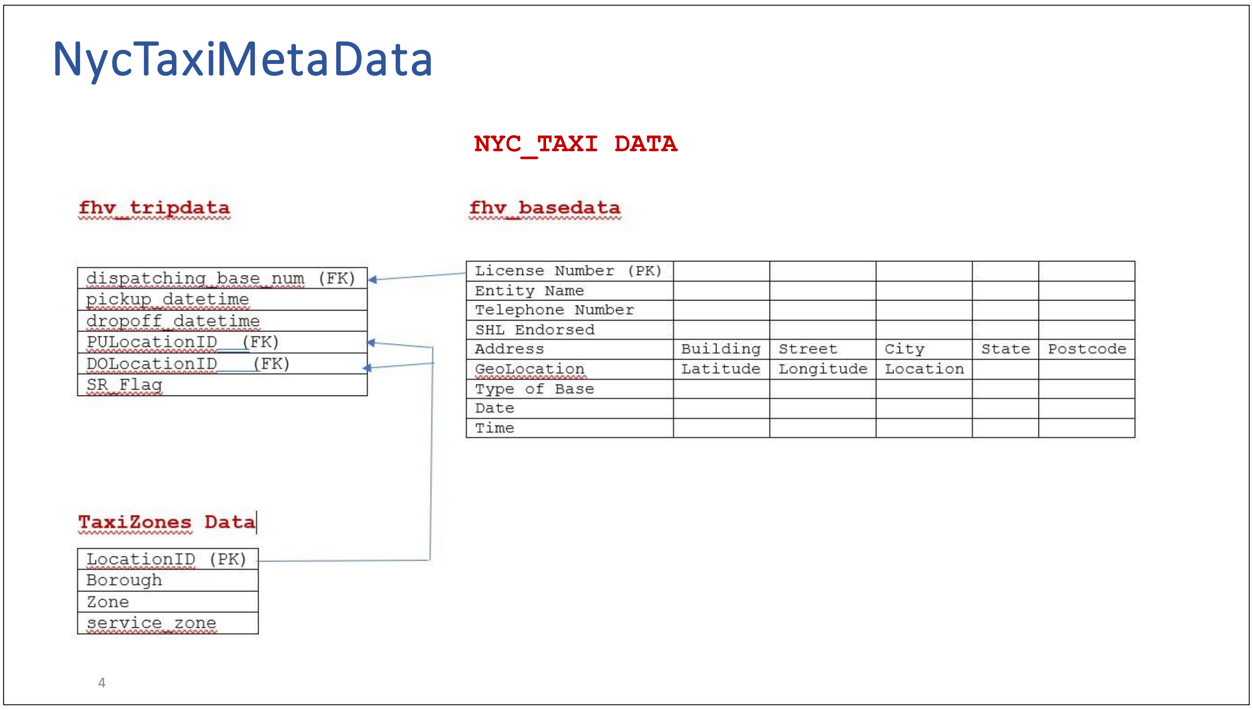
```
{"PaymentTypeID":1,
"PaymentType":"Cred
it Card"}
```

TaxiZones Data

LocationID
Borough
Zone
service_zone

green_tripdata	Sample Data	yellow_tripdata	Sample Data
VendorID	2	VendorID	1
lpep_pickup_datetime	2019-06-01 00:25:27	tpep_pickup_datetime	6/1/2019 0:55
lpep_dropoff_datetime	2019-06-01 00:33:52	tpep_dropoff_datetime	6/1/2019 0:56
store_and_fwd_flag	N	passenger_count	1
RatecodeID	1	trip_distance	0
PULocationID	74	RatecodeID	1
DOLocationID	263	store_and_fwd_flag	N
passenger_count	5	PULocationID	145
trip_distance	2.34	DOLocationID	145
fare_amount	9	payment_type	2
Extra	0.5	fare_amount	3
mta_tax	0.5	Extra	0.5
tip_amount	1	mta_tax	0.5
tolls_amount	0	tip_amount	0
ehail_fee		tolls_amount	0
improvement_surcharge	0.3	improvement_surcharge	0.3
total_amount	14.05	total_amount	4.3
payment_type	1	congestion_surcharge	0
trip_type	1		
congestion_surcharge	2.75		

3



NycTaxiMetaData

NYC_TAXI DATA

fhv_tripdata

dispatching_base_num (FK)
pickup_datetime
dropoff_datetime
PULocationID (FK)
DOLocationID (FK)
SR_Flag

fhv_basedata

License Number (PK)					
Entity Name					
Telephone Number					
SHL Endorsed					
Address	Building	Street	City	State	Postcode
GeoLocation	Latitude	Longitude	Location		
Type of Base					
Date					
Time					

TaxiZones Data

LocationID (PK)
Borough
Zone
service_zone

4

New York Taxi Data Sets Metadata

Green Trip Data Sets Metadata with Data Type (20 columns)

Column Name	*Data type*
VendorID	integer
lpep_pickup_datetime	timestamp
lpep_dropoff_datetime	timestamp
store_and_fwd_flag	string
RatecodeID	integer
PULocationID	integer
DOLocationID	integer
passenger_count	integer
trip_distance	double
fare_amount	double
extra	double
mta_tax	double
tip_amount	double
tolls_amount	double
ehail_fee	string
improvement_surcharge	double
total_amount	double
payment_type	integer
trip_type	integer
congestion_surcharge	double

Yellow Trip Data Sets Metadata with Data Type (18 columns)

Column Name	*Data type*
VendorID	integer
tpep_pickup_datetime	timestamp
tpep_dropoff_datetime	timestamp
passenger_count	integer
trip_distance	double
RatecodeID	integer
store_and_fwd_flag	string
PULocationID	integer
DOLocationID	integer
payment_type	integer
fare_amount	double
extra	double
mta_tax	double
tip_amount	double
tolls_amount	double
improvement_surcharge	double
total_amount	double
congestion_surcharge	double

Taxi Zones Data Set Metadata with Data Type (4 columns)

Column Name	Data type
LocationID	integer
Borough	string
Zone	string
service_zone	string

Payment Types Data Set Metadata with Data Type (2 columns)

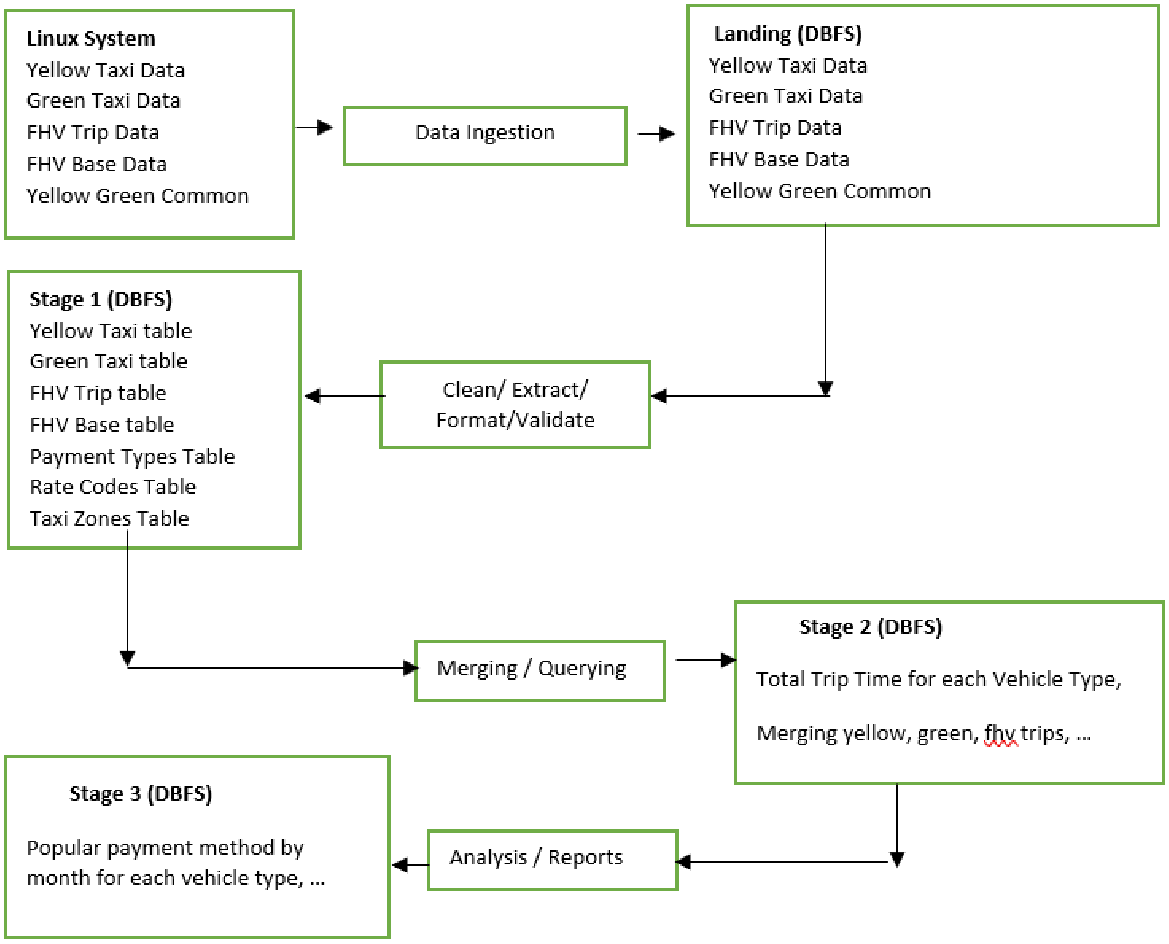
Column Name	Data type
PaymentTypeID	integer
PaymentType	string

Rate Codes Data Set Metadata with Data Type (3 columns)

Column Name	Data type
RateCodeID	integer
RateCode	string
IsApproved	string

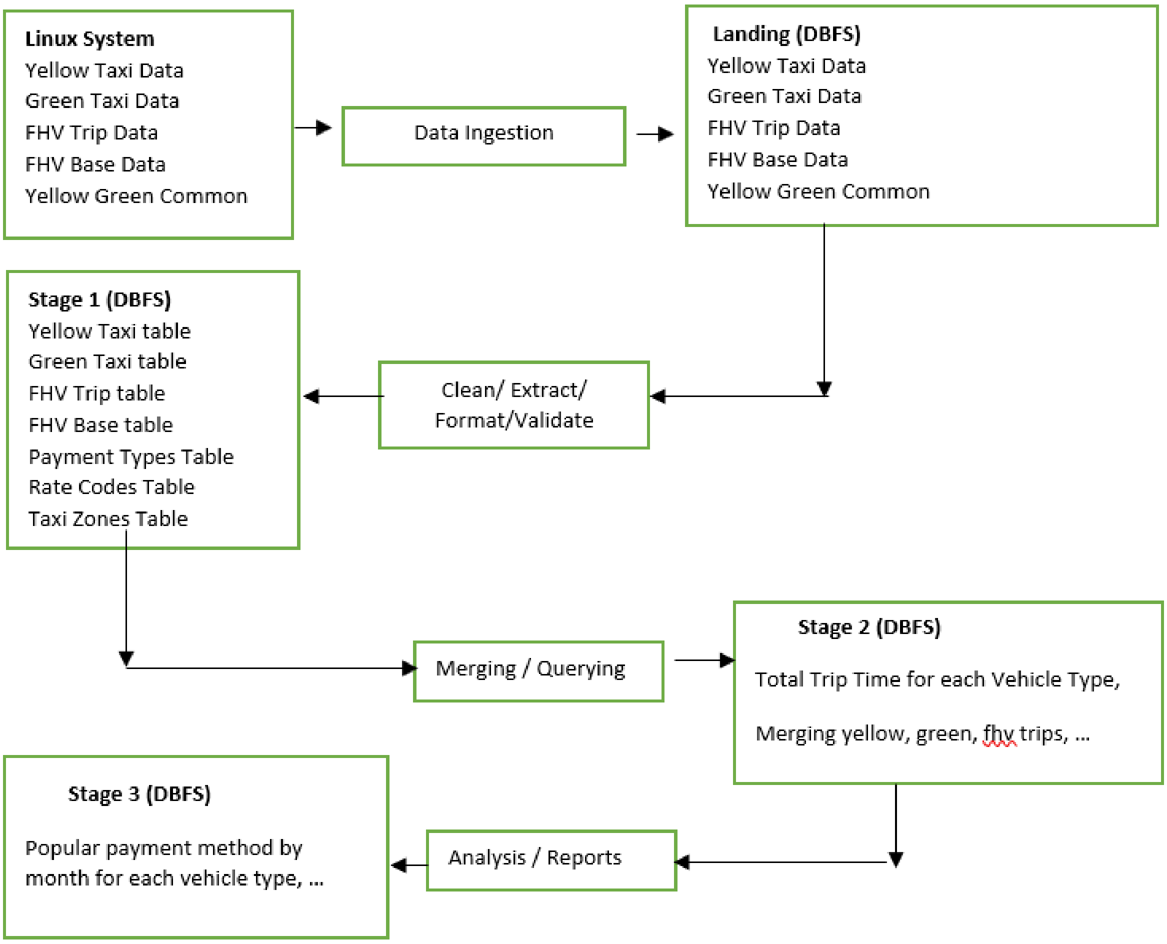
New York Taxi Case Study Dataflow Diagram

Dataflow Diagram:



Distributed Data Processing Using DataFrame and Spark SQL API.

Dataflow Diagram:



1. ETL Processing with Yellow Trip Data Sets

1. Create a raw delta table **"yellowtaxi"** using the yellow taxi data files by reading data and inferring schema from the data.

2. Create a raw delta table "**yellowtaxi_manual_schema**" using the yellow taxi data files by reading data and defining manual schema while creating table.

3. Create a filtered delta table **yellowtrip_filtered** using below column names : (VendorID, pickup_datetime,dropoff_datetime,passenger_count,trip_distance, RatecodeID, PickupLocationID,DropLocationID,payment_type,fare_amount,extra,mta_tax, tip_amount, tolls_amount,improvement_surcharge,total_amount,congestion_surcharge)

   from **yellowtaxi** table and by selecting valid records as per the below filter conditions –
   1. tpep_pickup_datetime and tpep_dropoff_datetime should not be same
   2. record will be considered on for year 2019 only.
   3. passenger_count > 0
   4. trip_distance >0
   5. fare_amount > 0f
   6. total_amount > 0
   7. PULocationID should not be null
   8. DOLocationID should not be null

2. Yellow Trip data analysis

Once the ETL is completed in the above exercises, start analytical processing of the data as below: (Use Spark SQL)

1) Find total number of trips for each vendor.
2) Find total trip distance travelled by for each vendor.
3) Find total fare amount earned by each vendor for each pickup location id.
4) Find total travel time in hour by each vendor for each drop location id.
5) Save the output of query 4 in a table **vendor_travel_time**.

3. ETL Processing with Green Trip Data Sets

1. Create a raw delta table **“greentaxi”** using the green taxi data files by reading data and inferring schema from the data.
2. Create a raw delta table “**greentaxi_manual_schema**” using the green taxi data files by reading data and defining manual schema while creating table.
3. Create a filtered delta table **greentrip_filtered** using the columns below: (VendorID,lpep_pickup_datetime as pickup_datetime, lpep_dropoff_datetime as dropoff_datetime, passenger_count, trip_distance, RatecodeID, PULocationID as PickupLocationID, DOLocationID as DropLocationID, payment_type, fare_amount,extra, mta_tax,tip_amount,tolls_amount, improvement_surcharge, total_amount, congestion_surcharge, ehail_fee,trip_type) from **greentaxi** table and by selecting valid records as per the below filter conditions –
   1. lpep_pickup_datetime and lpep_dropoff_datetime should not be same
   2. record will be considered on for year 2019
   3. passenger_count should not be zero
   4. trip_distance should not be zero
   5. fare_amount should not be zero
   6. total_amount should not be zero
   7. PULocationID should not be null
   8. DOLocationID should not be null

4. Green Trip data analysis

Once the ETL is completed in the above exercises, start analytical processing of the data as below: (Use Spark SQL API)

1) Find total number of trips for each vendor.
2) Find total trip distance travelled by for each vendor.
3) Find total fare amount earned by each vendor for each pickup location id.
4) Find total travel time in hour by each vendor for each drop location id.
5) Save the output of query 4 in a table **greenvendor_travel_time.**

5. Combine Yellow and Green Trip Data Sets

Create a Table “**YellowGreenTripCombinetable”** combining **yellowtrip_filtered_table** and **greentrip_filtered_table** tables with listed columns –

(VendorID, pickup_datetime,dropoff_datetime,passenger_count,trip_distance, RatecodeID,PickupLocationID,DropLocationID,payment_type,fare_amount, extra, mta_tax, tip_amount , tolls_amount, improvement_surcharge, total_amount, congestion_surcharge, **taxiType**)

Where **taxiType** column value will be “Green” for all green trip records and “Yellow” for all yellow trip records.

6. Work with Common Data Sets

6.1 Working with TaxiZones data sets

1. Create a delta table as “**taxizones**” from Taxizones Data.

6.2 Working with RateCodes data sets

1. Create a delta table “**ratecodes**” from rate codes data.

6.3 Working with PaymentTypes data sets

1. Create a delta table “**payments**” from payment types data.

7. Create Report (Process Data) on Yellow Green Combined Data sets (Using Spark SQL API)

1. Generate Report to get total trip time in hours for each taxi type.

2. Generate Report to get taxi-type-wise total trip time in hours for each trip-month.

3. Generate Report to get taxi-type-wise total number of passengers for each trip month.

4. Generate Report to get taxi-type-wise total number of payments for each payment-type.

5. Generate Report to get the total number of light trips for each taxi type (when passenger count for each trip is <=2).

6. Generate Report to get the total number of light trips for each taxi type month-wise (when passenger count for each trip is <=2) and save the results in a table "lightTripsTaxiTypeMonWise".

7. Generate Report to get the total number of fully-loaded trips for each taxi type (when passenger count for each trip is >=4).

8. Generate Report to get the total number of fully-loaded trips for each taxi type month-wise (when passenger count for each trip is >=4) and save the results in a delta table "loadedTripsTaxiTypeMonWise".

9. Generate Report to get the total number of midnight trips for each taxi type (when trips happen between 12AM to 4AM).

10. Generate Report to get the total number of midnight trips for each taxi type month-wise(when trips happen between 12AM to 4AM).

8. Create Report on Yellow Green Combined & Common Data sets (using Spark SQL API)

1. Create a Report to get total trip time in hours for each taxi type.

2. Create a Report to generate Passenger Count for Each Zone
   (Join **yellowgreentripcombinetable** and **taxizones_table** tables for creating report)

   Display and Save the output in table "**ZonePassCountTable**".

3. Create a database "**nyctaxireport**" in Spark SQL.

4. Create a Report to generate Total Trip Count Per Zone, TaxiType, TripMonth, VendorID . Display and save the output in table "**ZoneTaxiMonVendorTripCountTable**" under "**nyctaxireport**" database.

5. Create a Report to generate Total Trip Time Per Borough,TaxiType,TripMonth,VendorID. Display and save the output in table "**BoroughTaxiMonVendorTripCountTable**" under "**nyctaxireport**" database.

6. Create a Report to generate Total Travel Fare Per Service_Zone,TaxiType,TripMonth. Display and Save the output in table "**TotalFareSZoneTaxiMonth**" under "**nyctaxireport**" database.

7. Generate report on Total Different Payment Method count for Each taxi type, each tripMonth, each VendorID.
   (Join **yellowgreentripcombinetable** and **paymenttypes_table** tables for creating report)
   Display and Save the output in table "**PaymentCountTaxiMonthVendor**" under "**nyctaxireport**" database.

8. Create a Report to generate Total Different Payment Method count for Each Zone, Each taxi type, each tripMonth,each VendorID. Display and Save the output in table "**PaymentCountZoneTaxiMonthVendor**" under "**nyctaxireport**" database.

9. Create a report to generate Total Trip count for each vendor for each Zone where "Standard Rate" has been applied. Display and Save the output in table "**StandardRateTripCount**" under "**nyctaxireport**" database.

10. Check "**nyctaxireport**" database and display the records from each table.