1. Creating taxi_details_str table & loading the whole row as a string

```
CREATE TABLE IF NOT EXISTS chicago_taxis.taxi_details_str (
taxi_trip_details_str String)
STORED AS TEXTFILE
tblproperties("skip.header.line.count"="1");


LOAD DATA INPATH
'/bigdatapgp/common_folder/midproject/taxi_trip_dataset/taxi_trip.csv'
OVERWRITE INTO TABLE chicago_taxis.taxi_details_str;
```

2. Splitting one column in multiple columns and creating taxi_trip_details table

```
CREATE TABLE IF NOT EXISTS chicago_taxis.taxi_trip_details
AS
select split(taxi_trip_details_str, ",")[0] as trip_id,
split(taxi_trip_details_str, ",")[1] as taxi_id,
split(taxi_trip_details_str, ",")[2] as trip_start_time,
split(taxi_trip_details_str, ",")[3] as trip_end_time,
cast(split(taxi_trip_details_str, ",")[4] as int) as trip_seconds,
cast(split(taxi_trip_details_str, ",")[5] as float) as trip_miles,
cast(split(taxi_trip_details_str, ",")[6] as bigint) as pickup_tract,
cast(split(taxi_trip_details_str, ",")[7] as bigint) as dropoff_tract,
cast(split(taxi_trip_details_str, ",")[8] as tinyint) as pickup_community,
cast(split(taxi_trip_details_str, ",")[9] as tinyint) as dropoff_community,
cast(split(taxi_trip_details_str, ",")[10] as float) as trip_fare,
cast(split(taxi_trip_details_str, ",")[11] as float) as tip_amt,
cast(split(taxi_trip_details_str, ",")[12] as float) as toll_amt,
cast(split(taxi_trip_details_str, ",")[13] as float) as extra_amt,
cast(split(taxi_trip_details_str, ",")[14] as float) as trip_total_amt,
split(taxi_trip_details_str, ",")[15] as payment_type,
split(taxi_trip_details_str, ",")[16] as company,
cast(split(taxi_trip_details_str, ",")[17] as double) as pickup_latitude,
cast(split(taxi_trip_details_str, ",")[18] as double) as pickup_longitude,
split(taxi_trip_details_str, ",")[19] as pickup_location,
cast(split(taxi_trip_details_str, ",")[20] as double) as dropoff_latitude,
cast(split(taxi_trip_details_str, ",")[21] as double) as dropoff_longitude,
split(taxi_trip_details_str, ",")[22] as dropoff_location,
split(taxi_trip_details_str, ",")[23] as community_areas
from
chicago_taxis.taxi_details_str;



select split(taxi_trip_details_str, ",")[0] as trip_id,
split(taxi_trip_details_str, ",")[1] as taxi_id,
split(taxi_trip_details_str, ",")[2] as trip_start_time
from
chicago_taxis.taxi_details_str limit 5;
```

3. Numerical Mapping of taxi_id & trip_id to reduce the data volume

3.1.1. Creating a separate table with distinct taxi_id values

```
CREATE TABLE IF NOT EXISTS chicago_taxis.taxi_id_mapping
AS
select distinct taxi_id from taxi_trip_details

CREATE TABLE IF NOT EXISTS chicago_taxis.taxi_id_mapping_with_id
AS
select row_number() over() as id, taxi_id from taxi_id_mapping
```

3.1.2. Joining taxi_id_mapping_with_id table to master table to replace the current taxi_id i.e uuid with a numerical id

```
CREATE TABLE IF NOT EXISTS chicago_taxis.taxi_trip_details_taxi_id_removed
AS
SELECT
trip_id,
id as taxi_id_int,
trip_start_time,
trip_end_time,
trip_seconds,
trip_miles,
pickup_tract,
dropoff_tract,
pickup_community,
dropoff_community,
trip_fare,
tip_amt,
toll_amt,
extra_amt,
trip_total_amt,
payment_type,
company,
pickup_latitude,
pickup_longitude,
pickup_location,
dropoff_latitude,
dropoff_longitude,
dropoff_location,
community_areas
from
taxi_trip_details as a
join
taxi_id_mapping_with_id as b
on
a.taxi_id = b.taxi_id
```

3.1.3. Removing trip_id (uuid) and adding an int id instead

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.taxi_trip_details_taxi_trip_id_removed
```

```
STORED AS ORC
AS
SELECT
row_number() over() as trip_id_int,
taxi_id_int,
trip_start_time,
trip_end_time,
trip_seconds,
trip_miles,
pickup_tract,
dropoff_tract,
pickup_community,
dropoff_community,
trip_fare,
tip_amt,
toll_amt,
extra_amt,
trip_total_amt,
payment_type,
company,
pickup_latitude,
pickup_longitude,
pickup_location,
dropoff_latitude,
dropoff_longitude,
dropoff_location,
community_areas
from
chicago_taxis.taxi_trip_details_taxi_id_removed
```

3.1.4. Cleaning up the temp tables

```
drop table chicago_taxis.taxi_details_str
drop table chicago_taxis.taxi_trip_details
drop table chicago_taxis.taxi_trip_details_taxi_id_removed
```

3.1.5. Casting date fields

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.taxi_trip_details_taxi_trip_id_removed_ts
STORED AS ORC
AS
SELECT
taxi_id_int,
trip_start_time,
trip_end_time,
to_date(from_unixtime(unix_timestamp(split(trip_start_time, " ")[0],
'MM/dd/yyyy'), 'yyyy-MM-dd')) as trip_start_date,
to_date(from_unixtime(unix_timestamp(split(trip_end_time, " ")[0],
'MM/dd/yyyy'), 'yyyy-MM-dd')) as trip_end_date,
trip_seconds,
trip_miles,
pickup_tract,
dropoff_tract,
pickup_community,
dropoff_community,
trip_fare,
tip_amt,
toll_amt,
extra_amt,
trip_total_amt,
payment_type,
company,
pickup_latitude,
pickup_longitude,
pickup_location,
dropoff_latitude,
dropoff_longitude,
dropoff_location,
community_areas
from
chicago_taxis.taxi_trip_details_taxi_trip_id_removed
```

3.1.6. Adding two fields for the trip start & end day of the week

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.taxi_trip_details_processed_with_dayofweek
AS
SELECT
*,
from_unixtime(unix_timestamp(split(trip_start_time, " ")[0], 'MM/dd/yyyy'),
'u') as start_dayofweek,
from_unixtime(unix_timestamp(split(trip_end_time, " ")[0], 'MM/dd/yyyy'),
'u') as end_dayofweek
from
chicago_taxis.taxi_trip_details_taxi_trip_id_removed
```

3.1.7. Adding a weekend field to store whether a day is weekday or weekend

```
CREATE TABLE IF NOT EXISTS chicago_taxis.taxi_trip_details_weekend_encoded
STORED AS ORC
AS
SELECT
*,
CASE
WHEN start_dayofweek in (6,7) THEN 1
WHEN start_dayofweek in (1,2,3,4,5) THEN 0
END AS weekend
from
chicago_taxis.taxi_trip_details_processed_with_dayofweek
```

3.2.1 Data Summary

```
use chicago_taxis;
```

1. What are the total number of trips per year? Present the findings in the below format

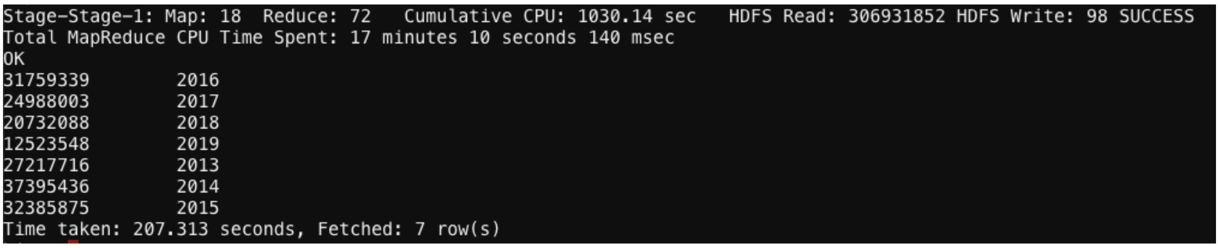
```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_trip_year_month
AS
```

```
select cast(split((split(trip_start_time, " ")[0]),"/")[2] as smallint) as
trip_year,
cast(split((split(trip_start_time, " ")[0]),"/")[0] as tinyint) as
trip_month,
trip_id_int as trip_id
FROM
taxi_trip_details_weekend_encoded;
```

```
select trip_year as `Year`,count(trip_id) as `Total Number of Trips`
from edureka_817479_taxi_trip_year_month
group by trip_year
order by trip_year asc;
```

OR

```
select count(trip_id_int),cast(split((split(trip_start_time, "
")[0]),"/")[2] as smallint) as trip_year
FROM
taxi_trip_details_weekend_encoded group by
cast(split((split(trip_start_time, " ")[0]),"/")[2] as smallint);
```



```
Stage-Stage-1: Map: 18  Reduce: 72   Cumulative CPU: 1030.14 sec   HDFS Read: 306931852 HDFS Write: 98 SUCCESS
Total MapReduce CPU Time Spent: 17 minutes 10 seconds 140 msec
OK
31759339        2016
24988003        2017
20732088        2018
12523548        2019
27217716        2013
37395436        2014
32385875        2015
Time taken: 207.313 seconds, Fetched: 7 row(s)
```

2. Create the same summary for number of trips at monthly level. Present the findings in the below format.

```
select trip_year as `Year`, trip_month as `Month`, count(*) as `Number of
Trips`
from edureka_817479_taxi_trip_year_month
group by trip_year, trip_month
order by trip_year asc, trip_month asc;
```

3. Calculate the percentage of records that contains drop-off community value. Excluding all the NULL records, find out the top 10 communities, where people travel to, based on the drop-off community field and also find its percentage to the total number of trips. Present the findings in the below format.

```
Select
dropoff_community,Community_trips,(Community_trips/total_trips*100)
as percentage
 from (select count(dropoff_community) as
Community_trips,dropoff_community from
taxi_trip_details_weekend_encoded where dropoff_community is not
null group by dropoff_community) a
join (select count(dropoff_community) as total_trips from
taxi_trip_details_weekend_encoded where dropoff_community is not
null) b order by Community_trips desc limit 10;
```

4.Create a table which contains the total number of trips for each drop-off community across each year. Using the above table, find the top 10 records based on number of trips with year and drop_off community. Remove the null record while creating the table to remove inconsistencies.

```
CREATE TABLE IF NOT EXISTS chicago_taxis.edureka_817479_trip_year_drop_off
AS
select cast(split((split(trip_start_time, " ")[0]),"/")[2] as smallint) as
trip_year, dropoff_community
FROM
taxi_trip_details_weekend_encoded
where dropoff_community is not NULL;

select *, count(dropoff_community) as `Trip per community per year`
from edureka_817479_trip_year_drop_off
group by trip_year, dropoff_community
order by `Trip per community per year` desc
limit 10;
```

5.Create a table which contains total number of trips for each drop-off communities across weekdays & weekends to check if there is any sort of pattern visible. After creating the table, find the top 10 drop off communities based on number of trips where people travel on weekdays. Find the same for the weekends. Also find the total number of trips taken on weekdays & weekends and their ratio.

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_trip_week_drop_off
AS
select start_dayofweek, dropoff_community, count(dropoff_community) as
`Trip per community per day`
FROM
taxi_trip_details_weekend_encoded
where dropoff_community is not NULL
group by start_dayofweek, dropoff_community
order by `Trip per community per day` desc;


select dropoff_community, `trip per community per day` from
edureka_817479_taxi_trip_week_drop_off
where start_dayofweek not in('6','7')
order by `trip per community per day` desc
limit 10;

select dropoff_community, `trip per community per day` from
edureka_817479_taxi_trip_week_drop_off
where start_dayofweek in('6','7')
order by `trip per community per day` desc
limit 10;
```

```
select count(dropoff_community) from edureka_817479_taxi_trip_week_drop_off
where start_dayofweek in('6','7');

select count(dropoff_community) from edureka_817479_taxi_trip_week_drop_off
where start_dayofweek not in('6','7');
```

6. Find the distribution of total number of trips based on trip duration, like <1 hr, 1 to 2 hr, 2 to 3, … 22 to 23 hr. Note that this requires converting trip_seconds into trip_hours as pre-processing. Remove the trips that do not contain trip duration

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_trip_duration_hrs
AS
select cast((trip_seconds/3600) as tinyint) as hours
from taxi_trip_details_weekend_encoded
where trip_seconds is not NULL;


select hours, count(hours) as Trip_Duration from
edureka_817479_taxi_trip_duration_hrs
group by hours
order by hours;
```

7. Find the top 10 buckets of the number of trips distribution based on the distance covered. Also round off the trip miles to the nearest integer. Remove the trips that do not contain distance

```
CREATE TABLE IF NOT EXISTS chicago_taxis.edureka_817479_taxi_trip_distance
AS
select cast(round(trip_miles) as smallint) as trip_miles
from taxi_trip_details_weekend_encoded
where trip_miles is not NULL;


select trip_miles, count(trip_miles) as Top_10_Distance_Bucket
from edureka_817479_taxi_trip_distance
group by trip_miles
order by Top_10_Distance_Bucket desc
limit 10;
```

8. Find top 10 buckets of the number of trips distribution based on the trip fare. Also round off the trip fare to the nearest integer. Remove the trips that do not contain trip fare

```
CREATE TABLE IF NOT EXISTS chicago_taxis.edureka_817479_taxi_trip_fare
AS
select cast(round(trip_fare) as smallint) as trip_fare
from taxi_trip_details_weekend_encoded
where trip_fare is not NULL;

select trip_fare, count(trip_fare) as Top_10_Fare_Bucket
from edureka_817479_taxi_trip_fare
```

```
group by trip_fare
order by Top_10_Fare_Bucket desc
limit 10;
```

9.Compute the average trip fare per day. Also compute the average trip fare per trip. Compute the same based on weekdays and weekend days. Find out if there is any substantial difference observed.

```
CREATE TABLE IF NOT EXISTS chicago_taxis.edureka_817479_taxi_day_trip_fare
AS
select cast(round(trip_fare) as smallint) as trip_fare, start_dayofweek
from taxi_trip_details_weekend_encoded
where trip_fare is not NULL and start_dayofweek is not NULL;

select avg(trip_fare) as Average_Trip_Fare_Per_Trip from
edureka_817479_taxi_day_trip_fare;
### 14.127929055625097

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_day_trip_fare_avg
AS
select start_dayofweek, round(avg(trip_fare),2) as Weekday_Average_Fair
from edureka_817479_taxi_day_trip_fare
group by start_dayofweek
order by start_dayofweek;

select avg(Weekday_Average_Fair) as Daily_Average_Fair
from edureka_817479_taxi_day_trip_fare_avg;
### 14.177142857142856

select avg(Weekday_Average_Fair) as WorkDay_Average_Fair
from edureka_817479_taxi_day_trip_fare_avg
where start_dayofweek not in ('6','7');
### 14.193999999999999

select round(avg(Weekday_Average_Fair),2) as WeekendDay_Average_Fair
from edureka_817479_taxi_day_trip_fare_avg
where start_dayofweek in ('6','7');
### 14.14
```

10.Create a table to store the taxi wise total fare & total number of trips for each day. Find the following insights from the table:
a. Find the top 10 taxis based on average trips per day.
b. Find the top 10 taxis based on average fare per day.

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_day_trip_day_fare
AS
select taxi_id_int, split(trip_start_time, " ")[0] as trip_date, trip_fare,
trip_id_int
from taxi_trip_details_weekend_encoded
where trip_fare is not NULL;

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_day_trip_day_fare_SUM
```

```
AS
select taxi_id_int, trip_date, sum(trip_fare) as TaxiWiseTotalFare,
count(trip_id_int) as TotalNumberOfTripsForEachDay
from edureka_817479_taxi_day_trip_day_fare
group by taxi_id_int, trip_date;

select taxi_id_int, round(avg(TotalNumberOfTripsForEachDay)) as
AverageTripsPerDay
from edureka_817479_taxi_day_trip_day_fare_SUM
group by taxi_id_int
order by AverageTripsPerDay desc
limit 10;

select taxi_id_int, round(avg(TaxiWiseTotalFare),2) as AverageFarePerDay
from edureka_817479_taxi_day_trip_day_fare_SUM
group by taxi_id_int
order by AverageFarePerDay desc
limit 10;
```

3.2.2 Data Preparation for Forecasting

1. Daily Summary Table

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Daily_Summary_table_initial
AS
select from_unixtime(unix_timestamp(split(trip_start_time, "
")[0],'MM/dd/yyyy'),'yyyy-MM-dd') as `Date`,
trip_id_int, trip_fare, trip_miles, (trip_seconds/60) as trip_minutes
from taxi_trip_details_weekend_encoded
where trip_seconds is not NULL;

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Daily_Summary_table_intermediate
AS
select `Date`, date_format(`Date`, 'u') as Day_Of_Week, month(`Date`) as
`month`,
year(`Date`) as `year`, count(trip_id_int) as Total_Trip_Count,
round(sum(trip_fare)) as Total_Trip_Fare, round(sum(trip_miles)) as
Total_Trip_Miles,
round(sum(trip_minutes)) as `Total_Trip_Duration(min)`,
round(avg(trip_fare)) as Avg_Trip_Fare,
round(avg(trip_miles)) as Avg_Trip_Miles, round(avg(trip_minutes)) as
`Avg_Trip_Duration(min)`
from edureka_817479_taxi_Daily_Summary_table_initial
group by `Date`
order by `Date`;

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Daily_Summary_table
AS
select `Date`, Day_Of_Week, `month`, `year`,
CASE
WHEN Day_Of_Week in (6,7) THEN 1
WHEN Day_Of_Week in (1,2,3,4,5) THEN 0
END AS `weekend\weekday`,
Total_Trip_Count, Total_Trip_Fare,
Total_Trip_Miles, `Total_Trip_Duration(min)`, Avg_Trip_Fare,
Avg_Trip_Miles, `Avg_Trip_Duration(min)`
```

```
from edureka_817479_taxi_Daily_Summary_table_intermediate;
```

2. Weekly Summary Table

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Weekly_Summary_table_initial
AS
select date_sub(`date`,pmod(datediff(`date`,'1900-01-07'),7)) as Date_From,
date_add(`date`,6 - pmod(datediff(`date`,'1900-01-07'),7)) as Date_To,
trip_id_int, trip_fare, trip_miles, trip_minutes
from edureka_817479_taxi_Daily_Summary_table_initial
order by Date_From;

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Weekly_Summary_table_intermediate
AS
select Date_From, Date_To, count(trip_id_int) as Total_Trip_Count,
round(sum(trip_fare)) as Total_Trip_Fare, round(sum(trip_miles)) as
Total_Trip_Miles,
round(sum(trip_minutes)) as `Total_Trip_Duration(min)`,
round(avg(trip_fare)) as Avg_Trip_Fare,
round(avg(trip_miles)) as Avg_Trip_Miles, round(avg(trip_minutes)) as
`Avg_Trip_Duration(min)`
from edureka_817479_taxi_Weekly_Summary_table_initial
group by Date_From, Date_To;

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Weekly_Summary_table
STORED AS ORC
AS
select weekofyear(Date_From) as Week_No, date_format(date_from,'MM/dd/yy'),
date_format(Date_To,'MM/dd/yy'),
date_format(date_from, 'MMM'), Total_Trip_Count, Total_Trip_Fare,
Total_Trip_Miles, `Total_Trip_Duration(min)`,
Avg_Trip_Fare, Avg_Trip_Miles, `Avg_Trip_Duration(min)`
from edureka_817479_taxi_Weekly_Summary_table_intermediate;
```

3. Monthly Summary Table

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Monthly_Summary_table_initial
AS
select date_add(last_day(add_months(`date`, -1)),1) as Date_From,
last_day(`date`) as Date_To,
trip_id_int, trip_fare, trip_miles, trip_minutes
from edureka_817479_taxi_Daily_Summary_table_initial
order by Date_From;

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Monthly_Summary_table_intermediate
AS
select Date_From, Date_To, count(trip_id_int) as Total_Trip_Count,
round(sum(trip_fare)) as Total_Trip_Fare, round(sum(trip_miles)) as
Total_Trip_Miles,
round(sum(trip_minutes)) as `Total_Trip_Duration(min)`,
round(avg(trip_fare)) as Avg_Trip_Fare,
round(avg(trip_miles)) as Avg_Trip_Miles, round(avg(trip_minutes)) as
`Avg_Trip_Duration(min)`
from edureka_817479_taxi_Monthly_Summary_table_initial
group by Date_From, Date_To;
```

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Monthly_Summary_table
STORED AS ORC
AS
select month(Date_From) as Month_No, date_format(date_from,'MM/dd/yy'),
date_format(Date_To,'MM/dd/yy'),
year(date_from) as `year`, Total_Trip_Count, Total_Trip_Fare,
Total_Trip_Miles, `Total_Trip_Duration(min)`,
Avg_Trip_Fare, Avg_Trip_Miles, `Avg_Trip_Duration(min)`
from edureka_817479_taxi_Monthly_Summary_table_intermediate
order by `year`, Month_No;
```

3.2.3 Data Preparation for Community Summary

1. Pickup Communities

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Daily_Pickup_Community_Summary_table_init
ial
AS
select pickup_community,
from_unixtime(unix_timestamp(split(trip_start_time, "
")[0],'MM/dd/yyyy'),'yyyy-MM-dd') as trip_date,
trip_id_int, trip_fare, trip_miles, trip_seconds
from taxi_trip_details_weekend_encoded
where pickup_community is not NULL and trip_seconds is NOT NULL;

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Daily_Pickup_Community_Summary_table
AS
select pickup_community, trip_date, count(trip_id_int) as DailyTripCount,
round(sum(trip_fare)) as DailyTotalFare, round(sum(trip_miles)) as
DailyTotalDistance,
round(sum(trip_seconds)) as DailyTotalDuration, round(avg(trip_fare)) as
DailyAverageAmount,
round(avg(trip_miles)) as DailyAverageDistance, round(avg(trip_seconds)) as
DailyAverageDuration
from edureka_817479_taxi_Daily_Pickup_Community_Summary_table_initial
group by pickup_community, trip_date
order by pickup_community, trip_date;
```

2. Dropoff Communnities

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Daily_Dropoff_Community_Summary_table_ini
tial
AS
select dropoff_community,
from_unixtime(unix_timestamp(split(trip_start_time, "
")[0],'MM/dd/yyyy'),'yyyy-MM-dd') as trip_date,
trip_id_int, trip_fare, trip_miles, trip_seconds
from taxi_trip_details_weekend_encoded
where dropoff_community is not NULL and trip_seconds is NOT NULL;

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Daily_Dropoff_Community_Summary_table
AS
select dropoff_community, trip_date, count(trip_id_int) as DailyTripCount,
round(sum(trip_fare)) as DailyTotalFare, round(sum(trip_miles),2) as
DailyTotalDistance,
round(sum(trip_seconds)) as DailyTotalDuration, round(avg(trip_fare)) as
DailyAverageAmount,
round(avg(trip_miles),2) as DailyAverageDistance, round(avg(trip_seconds))
as DailyAverageDuration
from edureka_817479_taxi_Daily_Dropoff_Community_Summary_table_initial
group by dropoff_community, trip_date
order by dropoff_community, trip_date;
```

3.2.4 Data Preparation for Origin to Destination Pair Summary

```
1.
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Origin_Destination_Pair_Summary_table_ini
tial
AS
select pickup_community, dropoff_community, trip_id_int, trip_fare,
trip_miles, trip_seconds
from taxi_trip_details_weekend_encoded
where pickup_community is not NULL and dropoff_community is not NULL and
trip_seconds is NOT NULL;

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Origin_Destination_Pair_Summary_table
AS
select pickup_community, dropoff_community, count(trip_id_int) as
TripCount,
round(sum(trip_miles),2) as TotalTripMiles, round(avg(trip_miles),2) as
AverageTripMiles,
round(avg(trip_seconds)) as AverageTripDuration, round(avg(trip_fare)) as
AverageTripFare
from edureka_817479_taxi_Origin_Destination_Pair_Summary_table_initial
group by pickup_community, dropoff_community
order by pickup_community, dropoff_community;

2.
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Origin_Destination_Pair_Summary_table_opt
ional
AS
select pickup_community, dropoff_community, count(trip_id_int) as
TripCount, round(sum(trip_fare),2) as TotalAmount,
round(sum(trip_miles),2) as TotalMiles, round(sum(trip_seconds / 60),2) as
TotalMins, round(avg(trip_miles),2) as AverageMiles,
round(avg(trip_seconds / 60)) as AverageMins, round(avg(trip_fare)) as
AverageAmount
from edureka_817479_taxi_Origin_Destination_Pair_Summary_table_initial
group by pickup_community, dropoff_community
order by pickup_community, dropoff_community;
```

3.2.5 Data Preparation for Company Summary

```
1.
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Daily_Company_Summary_table_initial
AS
select company, from_unixtime(unix_timestamp(split(trip_start_time, "
")[0],'MM/dd/yyyy'),'yyyy-MM-dd') as trip_date,
year(from_unixtime(unix_timestamp(split(trip_start_time, "
")[0],'MM/dd/yyyy'),'yyyy-MM-dd')) as trip_year, trip_id_int, trip_fare,
trip_miles, trip_seconds
from taxi_trip_details_weekend_encoded
where company is not NULL and trip_seconds is NOT NULL;

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Daily_Company_Summary_table
AS
select company, trip_year, count(trip_id_int) as DailyTripCount,
round(sum(trip_fare),2) as DailyTotalFare,
round(sum(trip_miles),2) as DailyTotalDistance, round(sum(trip_seconds /
60),2) as DailyTotalMins, round(avg(trip_miles),2) as DailyAverageDistance,
round(avg(trip_seconds / 60)) as DailyAverageMins, round(avg(trip_fare)) as
DailyAverageAmount
from edureka_817479_taxi_Daily_Company_Summary_table_initial
group by company, trip_year
order by company, trip_year;
```

3.2.6 Data Summary in RDBMS

1.

```
#hive
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Daily_pickup_community_Summary_stage
AS
select pickup_community,
from_unixtime(unix_timestamp(split(trip_start_time, "
")[0],'MM/dd/yyyy'),'yyyy-MM-dd') as trip_date,
year(from_unixtime(unix_timestamp(split(trip_start_time, "
")[0],'MM/dd/yyyy'),'yyyy-MM-dd')) as trip_year, trip_id_int, trip_fare,
trip_miles, trip_seconds
from taxi_trip_details_weekend_encoded
where pickup_community is not NULL and trip_seconds is NOT NULL;

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_pickup_community_Summary_table
AS
select pickup_community, trip_year, count(trip_id_int) as DailyTripCount,
round(sum(trip_fare),2) as DailyTotalFare,
round(sum(trip_miles),2) as DailyTotalDistance, round(sum(trip_seconds /
60),2) as DailyTotalMins, round(avg(trip_miles),2) as DailyAverageDistance,
round(avg(trip_seconds / 60)) as DailyAverageMins, round(avg(trip_fare)) as
DailyAverageAmount
from edureka_817479_taxi_Daily_pickup_community_Summary_stage
group by pickup_community, trip_year
order by pickup_community, trip_year;


#MYSQL

use labuser_database;

CREATE TABLE `edureka_817479_taxi_Daily_pickup_coummunity_table` (

  `pickup_community_id` VARCHAR(255) NOT NULL,

`year` INT NULL,
  `daily_trip_count` INT NULL,

  `daily_total_fare` FLOAT NULL,

  `daily_total_distance` FLOAT NULL,

  `daily_total_duration` FLOAT NULL,

  `daily_avg_fare` FLOAT NULL,

  `daily_avg_distance` FLOAT NULL,

  `daily_avg_duration` FLOAT NULL,

);
```

```
hive -e 'select * from
chicago_taxis.edureka_817479_taxi_Daily_pickup_coummunity_table | sed
's/[\t]/,/g' > pickup_community_summary_daily.csv
```

```
LOAD DATA INFILE 'pickup_community_summary_daily.csv'

INTO TABLE edureka_817479_taxi_Daily_pickup_coummunity_table

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n';
```

2.

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_Daily_dropoff_community_Summary_stage
AS
select dropoff_community,
from_unixtime(unix_timestamp(split(trip_start_time, "
")[0],'MM/dd/yyyy'),'yyyy-MM-dd') as trip_date,
year(from_unixtime(unix_timestamp(split(trip_start_time, "
")[0],'MM/dd/yyyy'),'yyyy-MM-dd')) as trip_year, trip_id_int, trip_fare,
trip_miles, trip_seconds
from taxi_trip_details_weekend_encoded
where dropoff_community is not NULL and trip_seconds is NOT NULL;

CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_taxi_dropoff_community_Summary_table
AS
select dropoff_community, trip_year, count(trip_id_int) as DailyTripCount,
round(sum(trip_fare),2) as DailyTotalFare,
round(sum(trip_miles),2) as DailyTotalDistance, round(sum(trip_seconds /
60),2) as DailyTotalMins, round(avg(trip_miles),2) as DailyAverageDistance,
round(avg(trip_seconds / 60)) as DailyAverageMins, round(avg(trip_fare)) as
DailyAverageAmount
from edureka_817479_taxi_Daily_dropoff_community_Summary_stage
group by dropoff_community, trip_year
order by dropoff_community, trip_year;
```

```
use labuser_database;


CREATE TABLE `edureka_817479_taxi_Daily_dropoff_coummunity_table` (

  `dropoff_community_id` VARCHAR(255) NOT NULL,

   `year` INT NULL,

  `daily_trip_count` INT NULL,

  `daily_total_fare` FLOAT NULL,

  `daily_total_distance` FLOAT NULL,

  `daily_total_duration` FLOAT NULL,

  `daily_avg_fare` FLOAT NULL,

  `daily_avg_distance` FLOAT NULL,

  `daily_avg_duration` FLOAT NULL,

  );

hive -e 'select * from
chicago_taxis.edureka_817479_taxi_Daily_dropoff_coummunity_table | sed
's/[\t]/,/g' > dropoff_community_summary_daily.csv


LOAD DATA INFILE 'dropoff_community_summary_daily.csv'

INTO TABLE edureka_817479_taxi_Daily_dropoff_coummunity_table

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n';
```

3.

```
mysql -u edu_labuser -p edureka -h bserver.edu.cloudlab.com -D labuser_database


#mysql

use labuser_database;

CREATE TABLE `edureka_817479_taxi_Daily_company_table` (

  `company_id` VARCHAR(255) NOT NULL,

`year` INT NULL,
  `daily_trip_count` INT NULL,

  `daily_total_fare` FLOAT NULL,

  `daily_total_distance` FLOAT NULL,

  `daily_total_duration` FLOAT NULL,

  `daily_avg_fare` FLOAT NULL,

  `daily_avg_distance` FLOAT NULL,

  `daily_avg_duration` FLOAT NULL,

);
```

```
hive -e "select * from
chicago_taxis.edureka_817479_taxi_Daily_Company_Summary_table where
company !=''" | sed 's/[\t]/,/
g' > company_summary_daily.csv
```

```
LOAD DATA LOCAL INFILE 'company_summary_daily.csv'

INTO TABLE edureka_817479_taxi_Daily_company_table

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n';
```

3.2.7 Summary Data Mart

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_company_summary_stage
AS
select cast(split((split(trip_start_time, " ")[0]),"/")[1] as smallint) as
trip_date,cast(split((split(trip_start_time, " ")[0]),"/")[2] as smallint)
as trip_year,cast(split((split(trip_start_time, " ")[0]),"/")[0] as
tinyint) as trip_month,company,trip_total_amt
  from chicago_taxis.taxi_trip_details_weekend_encoded;
```

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_company_summary AS
select company, count(*) as total_trip, sum(trip_total_amt) as
total_amount,count(distinct trip_month) as months_count,trip_year
from chicago_taxis.edureka_817479_company_summary_stage
group by company,trip_year;
```

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_community_summary_stage
AS
select cast(split((split(trip_start_time, " ")[0]),"/")[1] as
smallint) as trip_date,cast(split((split(trip_start_time, "
")[0]),"/")[2] as smallint) as
trip_year,cast(split((split(trip_start_time, " ")[0]),"/")[0] as
tinyint) as
trip_month,pickup_community,dropoff_community,trip_total_amt
  from chicago_taxis.taxi_trip_details_weekend_encoded;
```

```
CREATE TABLE IF NOT EXISTS
chicago_taxis.edureka_817479_community_summary AS
select pickup_community, dropoff_community,count(*) as total_trip,
sum(trip_total_amt) as total_amount,count(distinct trip_month) as
months_count,trip_year
from chicago_taxis.edureka_817479_community_summary_stage
group by pickup_community, dropoff_community,trip_year;
```

JAVA Code

```
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.HColumnDescriptor;
import org.apache.hadoop.hbase.HTableDescriptor;
import org.apache.hadoop.hbase.client.HBaseAdmin;
import org.apache.hadoop.hbase.client.HTable;
import org.apache.hadoop.hbase.client.Put;
import org.apache.hadoop.hbase.util.Bytes;

import com.example.hbase.HBaseCRUDTest;

public class HiveToHbase {

	private static Configuration conf = null;
	private static String driverName = "org.apache.hadoop.hive.jdbc.HiveDriver";

	/* Initialization */

	static {
		conf = HBaseConfiguration.create();
		conf.clear();
		conf.set("hbase.zookeeper.quorum", "ip-20-0-21-196.ec2.internal");
		conf.set("hbase.zookeeper.property.clientPort", "2181");
```

```
    }

    /* Create a table */

    public static void creatTable(String tableName, String[] familys) throws Exception {

        HBaseAdmin admin = new ~~HBaseAdmin~~(conf);
        if (admin.tableExists(tableName)) {
            System.out.println("table already exists!");
        }

        else {
            HTableDescriptor tableDesc = new ~~HTableDescriptor~~(tableName);
            for (int i = 0; i < familys.length; i++) {
                tableDesc.addFamily(new HColumnDescriptor(familys[i]));
            }
            admin.createTable(tableDesc);
            System.out.println("create table " + tableName + " ok.");
        }
    }

    /* Put (or insert) a row */

    public static void addRecord(String tableName, String rowKey, String family, String
qualifier, String value)
            throws Exception {
        try {
            HTable table = new ~~HTable~~(conf, tableName);
            Put put = new Put(Bytes.toBytes(rowKey));
            put.~~add~~(Bytes.toBytes(family), Bytes.toBytes(qualifier),
Bytes.toBytes(value));
            table.put(put);
            System.out.println("insert record " + rowKey + " to table " + tableName +
" ok.");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void fetchDataFromHiveForCompany() throws Exception {
        try {
            try {
                Class.forName(driverName);
            } catch (ClassNotFoundException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
                System.exit(1);
            }
            Connection con;

            con = DriverManager.getConnection("jdbc:hive://localhost:10000/default",
"", "");

            Statement stmt = con.createStatement();
            ResultSet res = stmt.executeQuery("select company,sum(total_trip) as
total_trip,sum(months_count) as month,sum(total_amount) as total_amt,count(distinct trip_year)
as year_count from chicago_taxis.edureka_817479_company_summary\n"
                    + "group by company");
            while(res.next()) {

    addRecord("company_summary_table",res.getString("company"),"Trip_Count_Stats","total
_trips",res.getInt("total_trip")+"");

addRecord("company_summary_table",res.getString("company"),"Trip_Count_Stats","monthly_av
g",(res.getInt("total_trip")/res.getInt("month"))+"");

addRecord("company_summary_table",res.getString("company"),"Trip_Count_Stats","yearly_avg"
,(res.getInt("total_trip")/res.getInt("year_count"))+"");

addRecord("company_summary_table",res.getString("company"),"revenue_details","monthly_avg"
,(res.getInt("total_amt")/res.getInt("month"))+"");

addRecord("company_summary_table",res.getString("company"),"revenue_details","yearly_avg",(
res.getInt("total_amt")/res.getInt("year_count"))+"");

addRecord("company_summary_table",res.getString("company"),"revenue_details","total_revenue
",res.getInt("total_amt")+"");
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public static void fetchDataFromHiveForCommunity() throws Exception {
        try {
            try {
                Class.forName(driverName);
            } catch (ClassNotFoundException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
                System.exit(1);
            }
            Connection con;

            con = DriverManager.getConnection("jdbc:hive://localhost:10000/default",
"", "");

            Statement stmt = con.createStatement();
            ResultSet res = stmt.executeQuery("select pickup_community,
dropoff_community,sum(total_trip) as total_trip,sum(months_count) as month,sum(total_amount)
as total_amt,count(distinct trip_year) as year_count \n"
                    + "from
chicago_taxis.edureka_817479_community_summary\n"
                    + "group by pickup_community,dropoff_community");
            while(res.next()) {
                if(res.getString("pickup_community")!=null) {
```

```
    addRecord("community_summary_table",res.getString("community"),"origin_Count_Stats","total_trips",res.getInt("total_trip")+"");

    addRecord("community_summary_table",res.getString("community"),"origin_Count_Stats","monthly_avg",res.getString("company"));

addRecord("community_summary_table",res.getString("community"),"origin_Count_Stats","yearly_avg",res.getString("company"));

                }
                if(res.getString("dropoff_community")!=null) {

    addRecord("community_summary_table",res.getString("community"),"destination_Count_Stats","total_trips",res.getInt("total_trip")+"");

addRecord("community_summary_table",res.getString("community"),"destination_Count_Stats","monthly_avg",res.getString("company"));

addRecord("community_summary_table",res.getString("community"),"destination_Count_Stats","yearly_avg",res.getString("company"));

                }

addRecord("community_summary_table",res.getString("community"),"revenue_summary","monthly_avg",res.getString("company"));

addRecord("community_summary_table",res.getString("community"),"revenue_summary","yearly_avg",res.getString("company"));

addRecord("company_summary_table",res.getString("company"),"revenue_details","total_revenue",res.getInt("total_amt")+"");

            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    /* Delete a table */

    public static void deleteTable(String tableName) throws Exception {
        try {
            HBaseAdmin admin = new ~~HBaseAdmin(conf)~~;
            admin.disableTable(tableName);
            admin.deleteTable(tableName);
            System.out.println("delete table " + tableName + " ok.");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        try {
```

```
			String[] companyColFamilies = { "Trip_Count_Stats","revenue_details" };
			System.out.println("===========create table========");
			HBaseCRUDTest.creatTable("company_summary_table",
companyColFamilies);
			String[] communtyColFamilies = {
"origin_Count_Stats","destination_Count_Stats","revenue_summary" };

			HBaseCRUDTest.creatTable("community_summary_table",
communtyColFamilies);

			System.out.println("=========insert records========");
			fetchDataFromHiveForCommunity();
			fetchDataFromHiveForCompany();


		} catch (Exception e) {
			e.printStackTrace();
		}
	}


}
```