DESIGN A DATA WAREHOUSE FOR CAB RIDE SERVICE LIKE UBER, LYFT

Note : While designing any Data Warehouse make sure to cover given below points.
a. Design Fact & Dimension tables
b. Create meaningful Primary & Foreign keys
c. Try to follow Star/SnowFlake Schema Design
d. Try to write few SQL queries to generate insightful business metrics (This is the critical
point because you need to understand the Data & Business both.

Four-Step Dimensional Design Process

➢ Select the business process.
- Drivers rides and performance.
- Customers rides and their information.
- Trips around common destinations such as public areas and airports etc
- Trips cancellation per day
- Rides and average price on peak hours

➢ Declare the grain.
- Provides Individual trip details on transaction level.

➢ Identify the dimensions tables for "who, what, why, when, where and how" associated with event.
- Date , Customers, Drivers , Cars , Documents , Devices ,Locations.

➢ Identify the fact table.
- Trips , Payment

DIMENSIONS

Date dimension:

It is easy to maintain different kind of date elements rather than doing calculations in query.

Elements such as **Week number, Weekends, seasons, fiscal years, holidays and workdays**.

Customers dimension:

Users using this ride sharing app will be maintain In this and used in centre of this model.

The user table shall contains, User_id, Name (First & Last), Email, phone_no, address, joining date,

Membership_date, active_status.

Drivers dimension:

Drivers who are providing services on this ride sharing app will be maintain In this

The user table shall contain, driver_id, Name (First & Last), Email, phone_no, address, joining date,

Car_id, Licence, date, active_status

Documents dimension:

This table contains all the documents uploaded by Drivers, and other information such as

Doc ID, Doc Name, Doc Type, actives tatus, Issuing authority, expiration date, driver id.

cars dimension:

This table stores all cars and drivers registered on the platform, all cars have uniqueid, car id, driver id, year, make, model, licence plate no, car type, base rate, active status.

Locations dimensions:

This table stores all details on popular destination such as parks, airports in city.

Such as Location id, longitude , latitude , landmark type, Landmark name, city, state, country

FACTS

Trips facts:

This table will cover information of all trips created by users, such as trip id, user id, driver id, car id, request time, start and end time, start and end locations, fare, payment id, user and driver ratings and trip status(Completed / In progress / Cancelled).

Payment facts:

This will cover all payments made by user. For every payment following details will be stored, payment id, customer id, payment date, base rate, surge rate, tip amount, taxes , transaction id and payment status.

DATA MODEL : STAR SCHEMA :

Locations table:

- Location_id **(PK)**
- Longitude_and_lat itude
- Landmark_type
- Landmark_name
- city
- state
- country

Users table:

- User_id **(PK)**
- First_name
- Last_name
- Email
- phone_no
- address
- joining date
- Membership_date
- active_status
- payment_id **(FK)**

trips table:

- Trip_id **(PK)**
- User_id **(FK)**
- Driver_id **(FK)**
- Car_id **(FK)**
- Request_time
- Start_time
- End_time
- Start_locations
- end_locations
- fare
- payment_id **(FK)**
- user_rating
- driver_ratings
- trip_status

Drivers table:

- driver_id **(PK)**
- First_name
- Last_name
- Email
- phone_no
- address
- joining_date
- Car_id **(FK)**
- Licence
- date
- active_status

cars table:

- Unique_id
- car_id **(PK)**
- driver_id **(FK)**
- year
- make
- model
- licence_plate_no
- car_type
- base_rate
- active_status

Payments table:

- Payment_id **(PK)**
- user_id **(FK)**
- Payment_date
- Base_rate
- Surge_rate
- Tip_amount
- taxes
- transaction_id
- payment_status

documents table:

- Doc_ID **(PK)**
- Doc_Name
- Doc_Type
- Actives_tatus
- Issuing_authority
- Expiration_date
- Driver_id **(FK)**

PK – Primary keys

FK – Foreign keys

Few SQL queries for Business metrics:-

Trips cancelling for each day in July month

```
SELECT extract(DAY FROM Request_time) as day, count(DISTINCT trip_id)
FROM trips
Group by day
having extract(MONTH from Request_time) = 'JULY';
```

No of trips taken to popular destinations for each day in July month

```
SELECT l.Landmark_name, count(DISTINCT t.trip_id)
FROM trips t JOIN locations l
ON t.end_locations = l. Longitude_and_latitude
Group by day
having extract(MONTH from Request_time) = 'JULY';
```

Monthly Driver performance metrics

```
SELECT d.driver_id, d.First_name, count(DISTINCT t.trip_id),avg(t.driver_ratings),
extract(MONTH from Request_time) as month_1

FROM trips t JOIN Drivers d

ON t. driver_id = d. driver_id

GROUP BY month_1

Having month_1 = 'JULY';
```