Name: Harie Vashini
Avinash Varma
Kaushik
Jetha Harsha

DATA ENGINEERING

Project Part - 2

A. Data Review

No.	Attribute	Datatype	Min Value	Max value	Description
1	EVENT_NO_TRIP	Number	167112203	167722889	Trip Start ID
2	EVENT_NO_STOP	Number	167112208	167722891	Trip Stop ID
3	OPD_DATE	Date	04-SEP-20	11-SEP-20	Trip date
4	VEHICLE_ID	Number	1776	1298380	Vehicle number
5	METERS	Number	0	573674	Meters covered by the vehicle
6	ACT_TIME	Number	15040	90712	Actual time when the reading was recorded
7	VELOCITY	Number	0	75	Velocity of the vehicle when reading was recorded
8	DIRECTION	Varchar2	0	359	Geo orientation(degrees)
9	RADIO_QUALITY	Number	-	-	Signal strength (No Data)
10	GPS_LONGITUDE	Number	-122.299463	-122.708243	Longitude of vehicle
11	GPS_LATITUDE	Number	45.494395	45.86686	Latitude of vehicle
12	GPS_SATELLITES	Number	0	12	Count of satellites

					capturing the vehicle location
13	GPS_HDOP	Number	0.2	25.4	Accuracy of GPS quality strength
14	SCHEDULE_DEVIATION	Number	-2245	5693	Deviation in the schedule

C. Data Validation

1. VEHICLE_ID - Existence Assertion
   Each record should have vehicle ID

2. EVENT_NO_TRIP - Existence Assertion
   Every record has Trip ID

3. DIRECTION - Limit Assertion
   Record values range between 0 to 359

4. VELOCITY - Inter record check Assertion
   Velocity should not exceed 30 meters per second

5. GPS_LATITUDE - limit Assertion
   GPS_LATITUDE should range between 45.766234 and 45.480231

6. GPS_LONGITUDE - Limit assertion
   GPS_LONGITUDE should range between -122.743269 and -122.350367

7. VELOCITY - Statistical distribution assertions
   Velocity is Binomially distributed over ACT_TIME

8. OPT_DATE - Referential assertion
   Every trip records OPT_DATE and ACT_TIME

9. EVENT_NO_STOP - Referential assertion
   Every event_no_trip should have an event_no_stop

10. EVENT_NO_TRIP - Inter record check Assertion
    A single EVENT_NO_TRIP will take a unique VEHICLE_ID

11. GPS_HDOP - Limit Assertion
    Ranges between 0 to 30

12. EVENT_NO_TRIP - Summary Assertion
    Every trip has a unique trip ID

13. GPS_LATITUDE and GPS_LONGITUDE - Inter record check assertion
    For the same time stamp no two buses should be at the same latitude and longitude.

14. OPD_DATE - Inter record check assertion
    The OPD_DATE should be a known date field.

15. VEHICLE_ID - Inter record check assertion
    The VEHICLE_ID should be a known vehicle ID with a valid format.

16. METERS - Statistical distribution assertion
    Meters are exponentially distributed over the Vehicle ID.

17. GPS_SATELLITES- Limit check assertion
    The GPS_SATELLITES should receive signals in the range between 0 and 12 satellites.

18. SCHEDULE_DEVIATION - Limit check Assertion
    SCHEDULE_DEVIATION should not exceed 7200 secs

19. GPS_HDOP - statistical distribution assertion
    GPS_HDOP is uniformly distributed over GPS_SATELLITES

20. ACT_TIME - Inter record check assertion
    The ACT_TIME should be a known field

We considered the highlighted 10 assertions to do data validation.

E. Data Transformation

The transformations we did to fit the data according to target schema design are:
i) Combined the opd_date and act_time to get the tstamp field.
ii) For Service_key we used opd_date to insert according to enum type.
iii) For Speed we used velocity m/s to miles per hr.
iv) To handle the tstamp values which are greater than 86399 seconds we did transformation to subtract that time to 86400 seconds to insert into the database.

F. Example Queries

Answer the following questions about the C-Tran system using your sensor data database. In your submission document include your query code, number of rows in each query result (if applicable) and first five rows of the result (if applicable).

1. How many vehicles are there in the C-Tran system?

   Select count(distinct(vehicle_id)) from trip;

```
redhots=# Select count(distinct(vehicle_id)) from trip;
 count
-------
    99
(1 row)
```

2. How many bread crumb reading events occurred on October 2, 2020?

   Select count(*) from breadcrumb where date(tstamp) = '2020-10-02';

```
redhots=# Select count(*) from breadcrumb where date(tstamp) ='2020-10-02';
 count
--------
 342002
(1 row)
```

3. How many bread crumb reading events occurred on October 3, 2020?

   Select count(*) from breadcrumb where date(tstamp) = '2020-10-03';

```
redhots=# Select count(*) from breadcrumb where date(tstamp) ='2020-10-03';
 count
--------
 166938
(1 row)
```

4. On average, how many bread crumb readings are collected on each day of the week?

   SELECT AVG(rowsPerDay) AS avgPerDay

```
FROM ( SELECT
      COUNT(*) AS rowsPerDay
    FROM breadcrumb
    GROUP BY DATE(tstamp)
  ) AS  a;
```

```
       avgperday
----------------------
 274953.266666666667
(1 row)
```

5. List the C-Tran trips that crossed the I-5 bridge on October 2, 2020. To find this, search for all trips that have bread crumb readings that occurred within a lat/lon bounding box such as [(45.620460, -122.677744), (45.615477, -122.673624)].

   Select distinct(trip_id) from breadcrumb where (latitude between 45.615477 and 45.620460) and (longitude between -122.677744 and -122.673624) LIMIT 5;

```
redhots=# Select distinct(trip_id) from breadcrumb where latitude between 45.615477 and  45.
620460 and  longitude between -122.677744 and -122.673624 LIMIT 5;
  trip_id
-----------
 167112212
 167112228
 167112232
 167112246
 167112462
(5 rows)
```

6. List all bread crumb readings for a specific portion of Highway 14 (bounding box: [(45.610794, -122.576979), (45.606989, -122.569501)]) during Mondays between 4pm and 6pm. Order the readings by tstamp. Then list readings for Sundays between 6am and 8am. How do these two time periods compare for this particular location?

   Select query1.day,query1.trip_id from (select extract(dow from date(bc.tstamp)) as day,trip_id from breadcrumb bc where cast(bc.tstamp as time) between '16:00:00' and '18:00:00' and bc.latitude between 45.606989 and 45.610794 and bc.longitude between -122.576979 and -122.569501 group by day,trip_id) as query1 where query1.day=1 Limit 5;

```
 day |  trip_id
-----+-----------
   1 | 167314331
   1 | 167840568
   1 | 167841668
   1 | 167841684
   1 | 167841845
```

(1 represents Monday)

Total number of rows returned = 32

select query1.day,query1.trip_id from (select extract(dow from date(bc.tstamp)) as day,tr
ip_id from breadcrumb bc where cast(bc.tstamp as time) between '06:00:00' and '08:00:00' and bc.lat
itude between 45.606989 and 45.610794 and bc.longitude between -122.576979 and -122.569501 group by
day,trip_id) as query1 where query1.day=0;

```
 day |  trip_id
-----+-----------
   0 | 167270386
   0 | 167775241
   0 | 167781569
   0 | 168346024
   0 | 168906840
```

(0 represents Sunday)

Total number of rows returned = 7

7. What is the maximum velocity reached by any bus in the system?

   Select max(speed) from breadcrumb;

```
redhots=# select max(speed) from breadcrumb;
   max
----------
 165.5306
(1 row)
```

8. List all possible directions and give a count of the number of vehicles that faced precisely that direction during at least one trip. Sort the list by most frequent direction to least frequent.

Select t.vehicle_id, bc.direction, count(*) from breadcrumb bc ,trip t where bc.trip_id=t.trip_id group by t.vehicle_id, bc.direction, date(bc.tstamp) = "2020-09-04" order by bc.direction, count(*) desc Limit 5;

```
 vehicle_id | direction | count
------------+-----------+-------
       2284 |         0 | 12596
       2263 |         0 | 11612
       2289 |         0 | 11294
       2286 |         0 |  9325
       2290 |         0 |  9082
(5 rows)
```

9. ~~Which is the longest (in terms of distance) trip of all trips in the data?~~ (ignore question 9)
10. Which is the longest (in terms of time) trip of all trips in the data?

select age(max(tstamp),min(tstamp)) as time, trip_id from breadcrumb group by trip_id, dat
e(tstamp) order by time desc LIMIT 1;

```
   time    |  trip_id
-----------+-----------
 23:59:56 | 169276194
(1 row)
```

11. ~~Which vehicle is the fastest? "Fastest" in this case should be measured in miles per hour averaged from the beginning of a trip to the end of the trip. That is, the total distance of the trip divided by the total time of the trip. This then should be averaged over all trips that each vehicle serviced.~~ (ignore question 11)
12. Devise three new, interesting questions about the C-Tran bus system that can be answered by your bread crumb data. Show your questions, their answers, the SQL you used to get the answers and the results of running the SQL queries on your data (the number of result rows, and first five rows returned).

    1. What is the minimum speed for each trip?

    select min(bc.speed), bc.trip_id from breadcrumb bc where bc.speed is not null group by bc.trip_id order by min(bc.speed) desc limit 5;

min	trip_id
55.9225	167873878
55.9225	167624710
53.6856	167481922
53.6856	169104560
53.6856	168982173

(5 rows)

2. Gather the average /speed covered by each bus for each day

select avg(bc.speed), tr.vehicle_id,date(bc.tstamp) from trip tr, breadcrumb bc where tr.trip_id = bc.trip_id group by tr.vehicle_id, date(bc.tstamp) Limit 5;

avg	vehicle_id	date
38.3560722790699	1776	2020-09-04
27.2734907340554	1776	2020-09-08
33.0784061946902	1776	2020-09-10
32.7246573670445	1776	2020-09-11
37.449109532539	1776	2020-09-14

(5 rows)

3. How many trips a particular vehicle covers on a particular day?

Select date(bc.tstamp), vehicle_id, count(*) from trip t, breadcrumb bc where bc.trip_id = t.trip_id group by date(bc.tstamp), vehicle_id order by date(bc.tstamp), count(*) DESC Limit 5;

date	vehicle_id	count
2020-09-04	6004	7809
2020-09-04	6003	7739
2020-09-04	4006	7298
2020-09-04	4007	7268
2020-09-04	4003	7012

(5 rows)

Date	Day of Week	# Sensor Readings	# updates/insertions into your database Note: First value is breadcrumb table insertion count Second value is the Trip table insertion count
2020-01-15	Friday	358546	319581, 1503
2020-01-16	Saturday	170828	161242, 824
2020-01-17	Sunday	133973	128116, 679
2020-01-18	Monday	133479	125171, 658
2020-01-19	Tuesday	367211	333943,1546
2020-01-20	Wednesday	356602	323361,1541
2020-01-21	Thursday	368612	338746, 1572
2020-01-22	Friday	363246	333824,1544
2020-01-23	Saturday	169274	159273, 787
2020-01-24	sunday	130745	121266,665