Michael Robertson
Data Engineering
CS410
Bruce Irvin
Genevieve Lalonde

Project Part 2 Components

Date	Day of Week	# of Sensor Readings	# of Updates/Insertions into database
4/4/2022 = 7SEP2020	MONDAY	133480	131956
4/5/2022 = 8SEP2020	TUESDAY	367212	362917
4/6/2022 = 9SEP2020	WEDNESDAY	356603	352572
4/7/2022 = 10SEP2020	THURSDAY	368613	364812
4/8/2022 = 11SEP2020	FRIDAY	363247	359699
4/9/2022 = 12SEP2020	SATURDAY	169275	167592
4/10/2022 = 13SEP2020	SUNDAY	130746	128953
4/11/2022 = 14SEP2020	MONDAY	365453	360649
4/12/2022 = 15SEP2020	TUESDAY	355478	351372
4/13/2022 = 16SEP2020	WEDNESDAY	364780	360869
4/14/2022 = 17SEP2020	THURSDAY	25528	362393
4/15/2022 = 18SEP2020	FRIDAY	380365	376046
4/16/2022 = 19SEP2020	SATURDAY	180829	178983
4/17/2022 = 20SEP2020	SUNDAY	138083	136397
4/18/2022 = 21SEP2020	MONDAY	3871822	382797
4/19/2022 = 22SEP2020	TUESDAY	366615	362478
4/20/2022 = 23SEP2020	WEDNESDAY	381855	377786
4/21/2022 = 24SEP2020	THURSDAY	380520	376099
4/22/2022 = 25SEP2020	FRIDAY	384541	380397

4/23/2022 = 26SEP2020	SATURDAY	175812	174124
4/24/2022 = 27SEP2020	SUNDAY	134609	132897
4/25/2022 = 28SEP2020	MONDAY	378022	373759
4/26/2022 = 29SEP2020	TUESDAY	371614	367503
4/27/2022 = 30SEP2020	WEDNESDAY	370453	366441
4/28/2022 = 1OCT2020	THURSDAY	366507	362667
4/29/2022 = 2OCT2020	FRIDAY	375777	371608
4/30/2022 = 3OCT2020	SATURDAY	176418	174657
5/1/2022 = 4OCT2020	SUNDAY	135161	133637
5/2/2022 = 5OCT2020	MONDAY	381365	376571
5/3/2022 = 6OCT2020	TUESDAY	376845	372196
5/4/2022 = 7OCT2020	WEDNESDAY	370922	367028
5/5/2022 = 8OCT2020	THURSDAY	373162	369039
5/6/2022 = 9OCT2020	FRIDAY	385209	381316
5/7/2022 = 10OCT2020	SATURDAY	173850	172041
5/8/2022 = 11OCT2020	SUNDAY	135366	133905

Breadcrumb data field descriptions:

EVENT_NO_TRIP: This is an integer that represents the value of the trip that the vehicle is taking that day. This value starts as it leaves its original starting location then stays the same until it arrives at its end destination. The value that next trip will be 2 digits increased from the last EVENT_NO_STOP of the previous trip. FOR EXAMPLE: Say the first EVENT_NO_TRIP is 167311316 and the last stop for the value is 3167311331. The next EVENT_NO_TRIP will be 3167311333, 2 values increased from the previous trip's last stop number.
EVENT_NO_STOP: These numbers are the stops that the trip takes throughout the vehicle's journey. The first value will be 1 number increased from the EVENT_NO_TRIP. The last number will be the EVENT_NO_TRIP + the number of stops that vehicle will take in its journey.
OPD_DATE: This is the date that the activity was recorded on, which is from the year 2020, so it would be roughly 1 year, 6 months, and 27 days behind the current date.
VEHICLE_ID: This value is the identification number of the vehicle that is being recorded. Typically it is a 4 digit number, but there are some Ids that have 7 digits in its id.
METERS: This field is the odometer for the vehicle. While there may be a new vehicle being entered into the records, that does not guarantee that the data field will be 0.
ACT_TIME: This data field is the "Actual Time", which means it is the time in the day that the record was taken, but it is an integer that is registered in seconds. For example, 33014 seconds = 09:10:14.
VELOCITY: This field is the recorded speed that the vehicle is going. This field has a minimum of 0. As it is in the city, I am assuming that the max value for this would be around 30-40.
DIRECTION: This is the direction that the vehicle was moving in with the field range being 0-359.
RADIO_QUALITY: This data is in relation to the strength of the radio for the vehicle. This information also is not included in the data tables.
GPS_LONGITUDE: This is the relative longitude position that the GPS communicates when the message is sent out. For this map the longitude should be around -122 with a decimal point.
GPS_LATITUDE: This is the relative latitude position that the GPS communicates when the message is sent out. For this map the latitude should be around 45 with a decimal point.
GPS_SATELLITES: This is the count of the satellites that are in position to capture the location of the vehicle during the time of the record.
GPS_HDOP: This is the horizontal dilution of precision.
SCHEDULE_DEVIATION: This is the difference between the expected schedule time of the trip compared to the actual time that the vehicle is at during that location of the trip. This can be a negative or positive value, as the vehicle could be behind or ahead of schedule.

Data Validations:

Existence:

1. Every EVENT_NO_TRIP has a value
2. Every ACT_TIME has a value
3. Every OPD_DATE has a value
4. Every GPS_LATITUDE has a value
5. Every GPS_LONGITUDE has a value
6. Every DIRECTION has a value

7. Every VELOCITY has a value
8. Every VEHICLE_ID has a value

Limit:

9. The VELOCITY should be between 0-60.79
10. The BreadCrumb direction should be between 0 and 359.
11. The ACT_TIME is between 0-86,399.
12. Every record is from the year 2020.
13. Every record is from the months of September, October, and November.
14. Every GPS_LONGITUDE is between -123 and -122.
15. Every GPS_LATITUDE is between 45 and 46.

Intra-record Check

16. Every EVENT_NO_STOP has a corresponding EVENT_NO_TRiP

Inter-record Check

17. Every EVENT_NO_TRIP has at least 5 unique EVENT_NO_STOP numbers.

Summary:

18.

Referential Integrity:

19. Every EVENT_NO_TRIP follows a known bus route.

Distribution:

20. It's likely that at least once during every EVENT_NO_TRIP the SCHEDULE_DEVIATION is at least 1.

Data transformations

The data transformation that occurred were in support of following the data validations that are listed above that are underlined (#1-8 and 10-11). The violations that occurred resulted in the data not being inserted into the postgresql table. This was done so that the necessary information that would be filling the table wouldn't be empty. Currently the only data has not been entered is in the Trip table where the direction and route_id is not entered and is currently set to Null.

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).

select DATE(tstamp) from breadcrumb group by DATE;

NOTE: ALL QUERIES HAVE BEEN RUN BEFORE BREADCRUMB DATA OF 8MAY2022 HAS BEEN ENTERED

1. How many vehicles are there in the C-Tran system?
   SELECT DISTINCT COUNT(vehicle_id) FROM Trip;
   47544 vehicles

SELECT COUNT(result.vehicle_id) FROM (SELECT DISTINCT (vehicle_id) FROM Trip) AS result;
103 vehicles

2. How many bread crumb reading events occurred on October 2, 2020?
SELECT COUNT(tstamp), DATE(tstamp) FROM breadcrumb WHERE DATE(tstamp) = '2020-10-02' GROUP BY DATE;
371608 Rows
3. How many bread crumb reading events occurred on October 3, 2020?
SELECT COUNT(tstamp), DATE(tstamp) FROM breadcrumb WHERE DATE(tstamp) = '2020-10-03' GROUP BY DATE;
174657 Rows
4. On average, how many bread crumb readings are collected on each day of the week?
SELECT AVG(results.COUNT), EXTRACT(dow FROM results.DATE) AS Day_Of_Week FROM (SELECT COUNT(tstamp), DATE(tstamp) FROM breadcrumb GROUP BY DATE) AS results GROUP BY Day_Of_Week ORDER BY Day_Of_Week;

```
        avg         | day_of_week
--------------------+-------------
 132971.000000000000 |           0
 325146.400000000000 |           1
 363293.200000000000 |           2
 364939.200000000000 |           3
 367002.000000000000 |           4
 373813.200000000000 |           5
 173479.400000000000 |           6
(7 rows)
```

NOTE: 0 is Sunday

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 DATE(tstamp)='2020-10-02' AND (longitude >= -122.677744 AND longitude <= -122.673624) AND (latitude >= 45.615477 AND latitude <= 45.620460);

```
  trip_id
-----------
 169357292
 169357303
 169357306
 169357318
 169374799
```

263 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 * FROM breadcrumb WHERE EXTRACT(dow FROM DATE(tstamp)) = 0 AND (EXTRACT(hour FROM tstamp) >= 18 AND EXTRACT(hour FROM tstamp) <= 20) AND (longitude >= -122.576979 AND longitude <= -122.569501) AND (latitude >= 45.606989 AND latitude <= 45.610794);
Mondays: 318 Rows

tstamp	latitude	longitude	direction	speed	trip_id
2020-09-07 18:06:43	45.60806	-122.570432	287	28	167319840
2020-09-07 18:06:48	45.608433	-122.572193	287	28	167319840
2020-09-07 18:06:53	45.608807	-122.573953	287	28	167319840
2020-09-07 18:06:58	45.60917	-122.575705	287	28	167319840
2020-09-07 17:56:13	45.608085	-122.570613	287	27	167314331

Sunday: 54 Rows

tstamp	latitude	longitude	direction	speed	trip_id
2020-09-13 18:06:51	45.607992	-122.570088	287	24	167775947
2020-09-13 18:06:56	45.60832	-122.571627	287	24	167775947
2020-09-13 18:07:01	45.608652	-122.573177	287	25	167775947
2020-09-13 18:07:06	45.608982	-122.574733	287	25	167775947
2020-09-13 18:07:11	45.609307	-122.576292	287	25	167775947

7. What is the maximum velocity reached by any bus in the system?
   SELECT MAX(speed) FROM breadcrumb;
   159
   74 (removed any rows that had speed greater than 80)
   I know this is probably a record I should have removed but did not at the time.
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 DISTINCT COUNT(trip_id), direction FROM breadcrumb GROUP BY direction ORDER BY COUNT DESC;

count	direction
657758	0
382642	180
300957	90
296461	270
210913	181

   360 Rows

9. Which is the longest (in terms of time) trip of all trips in the data?
   SELECT trip_id, (MAX(tstamp)::time - MIN(tstamp)::time) AS time FROM breadcrumb GROUP BY trip_id HAVING (MAX(tstamp)::time - MIN(tstamp)::time) = (SELECT MAX(results.time) FROM(SELECT (MAX(tstamp)::time - MIN(tstamp)::time) AS time, trip_id, DATE(tstamp) FROM breadcrumb GROUP BY trip_id, DATE(tstamp)) AS results);
   05:32:21

trip_id	time
169302880	05:32:21

   (1 row)

10. 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).
    a. What is the average speed?

SELECT AVG(speed) FROM breadcrumb;

```
postgres=# SELECT AVG(speed) FROM breadcrumb;
        avg
--------------------
 10.353804165395804
(1 row)
```

b. Which day has the most breadcrumbs,and how many does it have?

SELECT COUNT(tstamp), DATE(tstamp) FROM breadcrumb GROUP BY DATE(tstamp) HAVING COUNT(tstamp) = (SELECT MAX(results.COUNT) FROM (SELECT COUNT(DATE(tstamp)) FROM breadcrumb GROUP BY DATE(tstamp)) AS results);

```
 count  |    date
--------+------------
 382797 | 2020-09-21
(1 row)
```

c. What is the velocity and its count that holds the largest count, where the speed is not 0?

SELECT COUNT(speed), speed FROM breadcrumb WHERE (speed != 0) GROUP BY speed HAVING COUNT(speed) = (SELECT MAX(results.COUNT) FROM (SELECT COUNT(speed) FROM breadcrumb WHERE speed != 0 GROUP BY speed) AS results);

```
 count  | speed
--------+-------
 714928 |    11
(1 row)
```

Code links:

These links are for my producer and consumer programs. The consumer program is where all of my validation and insertions occur.

https://github.com/marobertson11/CS410/blob/main/Project_consumer.py

https://github.com/marobertson11/CS410/blob/main/Project_producer.py