**DataEng Project Assignment 2 Submission Document**

Construct a table showing each day for which your pipeline successfully, automatically processed one complete day’s worth of sensor readings. The table should look like this:

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Day of Week | # Sensor Readings | # rows added to your database |
| 06-05-2024 | Monday | 300532 | 300532 |
| 07-05-2024 | Tuesday | 221762 | 221762 |
| 08-05-2024 | Wednesday | 238366 | 238366 |
| 09-05-2024 | Thursday | 288229 | 288229 |
| 10-05-2024 | Friday | 341791 | 341791 |
| 11-05-2024 | Saturday | 304287 | 304287 |
| 12-05-2024 | Sunday | 322807 | 322807 |

Documentation of Each of the Original Data Fields

For each of the fields of the breadcrumb data, provide any documentation or information that you can determine about it. Include bounds or distribution data where appropriate. For example, for something like “Vehicle ID”, say something more than “It is the identification number for the vehicle”. Instead, add useful information such as “the integers in this field range from <min val> to <max val>, and there are <n> distinct vehicles identified in the data. Every vehicle is used on weekdays but only 50% of the vehicles are active on weekends.”

**EVENT\_NO\_TRIP**

This field represents the event number of the trip. It is a unique identifier for each trip in the data. It is always an 8 digit integer. The current data in tables have it ranging from 228871317 to 233137897. There are a total of 3990 distinct trips identified in the data.

**EVENT\_NO\_STOP**

This field represents the event number of the trip. It is a unique identifier for each trip in the data. It is always a 9 digit integer. The current data in tables have it ranging from 228871319 to 233137914. There are hundreds of thousands of distinct stops identified in the data.

**OPD\_DATE**

This field represents the date of the operation. It provides the date on which the trip occurred. The values in this field are in the format 'DDMMYYYY: HH:MM:SS'. The current data in tables have it ranging from 30DEC2022:00:00:00 to 05JAN2023:00:00:00. This does not give any information about time, the value is 00:00:00 for all readings.

**VEHICLE\_ID**

This field represents the identification number of the vehicle. It uniquely identifies each vehicle in the data. It is always a 4 digit integer. The current data in tables have it ranging from 2910 to 4303. There are a total of 66 distinct vehicles identified in the data.

**METERS**

This field represents the distance covered in meters. It indicates the distance traveled between breadcrumb readings. The values in this field can be integers. The values in this field in the current data may range from 0 to 500,000. The number does not indicate the total distance covered by the vehicle in a single trip, but rather helps further in calculating the distance covered by it in each trip or over the course of many trips during the whole day.

**ACT\_TIME**

This field represents the time of the breadcrumb reading. It provides information on when the breadcrumb reading was recorded. It is not a timestamp field by itself. It represents the number of seconds which have passed since 12 AM of a particular date. The values are always integers. The values in this field may range from 0 to 100,000 where any value greater than 86,400 means that it has crossed 24 hours from OPD\_DATE of that reading. This value can further be used to calculate the actual timestamp of the breadcrumb reading.

**GPS\_LONGITUDE**

This field represents the longitude coordinates obtained from GPS. It indicates the east-west position of the vehicle at the time of the breadcrumb reading. The values in this field are in decimal degrees format. The values in this field for the data in the tables range between -124 and -122, providing the bounds of the geographic area covered by the data. The negative sign in the value suggests that the vehicle is in the western hemisphere.

**GPS\_LATITUDE**

This field represents the longitude coordinates obtained from GPS. It indicates the north-south position of the vehicle at the time of the breadcrumb reading. The values in this field are in decimal degrees format. The values in this field for the data in the tables range between 45 and 46, providing the bounds of the geographic area covered by the data. The positive sign in the value suggests that the vehicle is in the northern hemisphere.

**GPS\_SATELLITES**

This field represents the number of GPS satellites used for obtaining the breadcrumb reading. It indicates the strength of the GPS signal during the reading. The number of satellites used can vary for different breadcrumb readings. The values in this field can be integers. The values for the current data in the tables range from 0 to 12.

**GPS\_HDOP**

This field represents the Horizontal Dilution of Precision (HDOP) obtained from GPS. The values in this field can be floats or decimals. It indicates the accuracy of the GPS horizontal positioning. The values in this field can vary, and a lower value indicates higher accuracy. The values for current data in the table range from 0 to 26.

Data Validation Assertions

List 20 or more data validation assertion statements here. These should be English language sentences similar to “The speed of a TriMet bus should not exceed 100 miles per hour”. You will only implement a subset of them, so feel free to write assertions that might be difficult to evaluate.  Create assertions for all of the fields, even those, like GPS\_HDOP, that might not be used in your database schema.

Assertion 1: Check that all speed values are non-negative and reasonable.

Assertion 2: Ensure SPEED column is in the correct format.

Assertion 3: For 'EVENT\_NO\_TRIP' format: 9 digits long and starts with '2'.

Assertion 4: For 'VEHICLE\_ID' format: must be an int and exactly 4 digits.

Assertion 5: Existence Check for Essential Columns

Assertion 6: Valid GPS Coordinate Check Intra-Record Check

Assertion 7: Distribution Check

Assertion 8: tstamp is a string and in the format DD MMM YYYY: HH:MM:SS

Assertion 9: Each event\_no\_trip should have only one unique vehicle\_id.

Assertion 10: Meters covered by any vehicle cannot be negative.

Assertion 11: There cannot exist a trip without a vehicle id.

Assertion 12: At Least 50 trips for every vehicle in a day.

Assertion 13: Direction of each trip which took place cannot be anything other than 0 or 1.

Assertion 14: More than 50% of vehicles speed will be under 15 mph.

Assertion 15: Ensure that service\_key is one of the specified valid values ('Weekday', 'Saturday', or 'Sunday')

Assertion 16: GPS\_HDOP values are always decimals within the range of 0 to 24.

Assertion 17: Hundreds of trips take place each day.

Assertion 18: Every trip will have different stops which recorded details at different locations (latitude and longitude).

Assertion 19: Year should be within 2022 and 2023.

Assertion 20: Data completeness check i.e. whether data contains any null values.

Data Transformations

Describe any transformations that you implemented either to react to validation violations or to shape your data to fit the schema. For each, give a brief description of the transformation along with a reason for the transformation.

* We have transformed a few columns from the consumed data like OPD\_DATE and ACT\_TIME and combined them to get a new column TIMESTAMP in a meaningful way and calculated the speed of each vehicle using METERS and TIMESTAMP values.
* Populated speed column by 0 by default before actual speed calculation to avoid NaN speed value for the first record of every trip.
* Used math. inNan to make sure we do not divide by NaN or null value to avoid undefined or infinite values in speed columns.

Example Queries

Provide your responses to the questions listed in Section F above. For each question, provide the SQL you used to answer the questions along with the count of the number of rows returned (where applicable) and a listing of the first 5 rows returned (where applicable).

1. How many breadcrumb reading events occurred on January 1, 2023?

**SELECT COUNT(\*) AS Jan1st2023 FROM breadcrumb WHERE DATE(tstamp)='2023-01-01';**

A black screen with white text

Description automatically generated

1. How many breadcrumb reading events occurred on January 2, 2023?

**SELECT COUNT(\*) AS Jan1st2023 FROM breadcrumb WHERE DATE(tstamp)='2023-01-02';**

A black screen with white text

Description automatically generated

1. On average, how many breadcrumb readings are collected on each day of the week?

**SELECT COUNT (\*)/7 AS avg\_readings FROM breadcrumb WHERE DATE(tstamp) BETWEEN '2022-12-30' AND '2023-01-05';**

A screen shot of a computer

Description automatically generated

1. List the TriMet trips that traveled a section of I-205 between SE Division and SE Powell on January 1, 2023. To find this, search for all trips that have breadcrumb readings that occurred within a lat/long bounding box such as [(45.497805, -122.566576), (45.504025, -122.563187)].

**SELECT DISTINCT t.trip\_id FROM trip t**

**JOIN breadCrumb b ON t.trip\_id = b.trip\_id**

**WHERE DATE(b.tstamp) = '2023-01-01'**

**AND b.latitude BETWEEN 45.497805 AND 45.504025**

**AND b.longitude BETWEEN -122.566576 AND -122.563187;**

A computer screen shot of a black screen

Description automatically generated

1. List all breadcrumb readings on a section of US-26 west side of the tunnel (bounding box: [(45.506022, -122.711662), (45.516636, -122.700316)]) 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 tstamp) = 1**

**AND EXTRACT (HOUR FROM tstamp) BETWEEN 16 AND 18**

**AND latitude BETWEEN 45.506022 AND 45.516636**

**AND longitude BETWEEN -122.711662 AND -122.700316**

**ORDER BY tstamp;**

A screen shot of a computer screen

Description automatically generated

A screen shot of a computer

Description automatically generated

**SELECT \* FROM breadcrumb**

**WHERE EXTRACT (DOW FROM tstamp) = 0**

**AND EXTRACT (HOUR FROM tstamp) BETWEEN 6 AND 8**

**AND latitude BETWEEN 45.506022 AND 45.516636**

**AND longitude BETWEEN -122.711662 AND -122.700316**

**ORDER BY tstamp;**

A screen shot of a computer screen

Description automatically generated

A screenshot of a computer screen

Description automatically generated

**This location has 7 different trips going through it between 4 PM and 6 PM on Monday, whereas 4 trip between 6 AM and 8AM on Sunday. Hence the first query is giving out over twice as many readings as the second.**

1. What is the maximum speed reached by any bus in the system?

**SELECT MAX(speed) as max\_speed FROM breadcrumb;**

A black screen with white text

Description automatically generated

1. List all speeds and give a count of the number of vehicles that move precisely at that speed during at least one trip. Sort the list by most frequent speed to least frequent.

**SELECT speed, COUNT(speed) AS count**

**FROM breadcrumb**

**GROUP BY speed**

**ORDER BY COUNT(speed) DESC;**

A screen shot of a computer screen

Description automatically generated

Total: 2791 Rows

1. Which is the longest (in terms of time) trip of all trips in the data?

**SELECT trip\_id, trip\_duration**

**FROM (**

**SELECT trip\_id, MAX(tstamp) - MIN(tstamp) AS trip\_duration**

**FROM breadcrumb**

**GROUP BY trip\_id**

**) AS trip\_durations**

**ORDER BY trip\_duration DESC**

**LIMIT 1;**

A screen shot of a computer

Description automatically generated

1. Are there differences in the number of breadcrumbs between a non-holiday Wednesday, a non-holiday Saturday, and a holiday?  What can that tell us about TriMet’s operations on those types of days?

A screenshot of a computer program

Description automatically generated

**The data suggests that TriMet sees higher trip counts on non-holiday Wednesdays, indicating increased weekday activity. Non-holiday Saturdays show a decrease in trips, typical for weekends. Holidays exhibit a moderate decrease in trips compared to regular weekdays and Saturdays, likely due to altered travel patterns. These patterns reflect varying demand for TriMet's services based on weekdays, weekends, and holidays, highlighting the need for tailored service planning and resource allocation. Analyzing such trends can optimize service efficiency and meet customer needs effectively.**

1. Devise three new, interesting questions about the TriMet bus system that can be answered by your breadcrumb 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).

**Find the 5 trips with the highest average speeds.**

**SELECT trip\_id, AVG(speed) AS average\_speed**

**FROM breadcrumb**

**GROUP BY trip\_id**

**ORDER BY average\_speed DESC**

**LIMIT 5;**

A computer screen with white text

Description automatically generated

**How many vehicles are there in the TriMet system?**

**SELECT COUNT (DISTINCT vehicle\_id) AS veh\_count FROM trip limit 5;**

A black screen with white text

Description automatically generated

Find the average speed and total distance traveled for each vehicle during weekday and weekend

**SELECT t.vehicle\_id, AVG(b.speed) AS avg\_speed, SUM(b.speed) AS distance**

**FROM trip t**

**JOIN breadcrumb b ON t.trip\_id = b.trip\_id**

**WHERE DATE(b.tstamp) = '2023-01-07’**

**GROUP BY t.vehicle\_id**

**ORDER BY avg\_speed DESC;**

A screen shot of a computer

Description automatically generated

**SELECT t.vehicle\_id, AVG(b.speed) AS avg\_speed, SUM(b.speed) AS distance**

**FROM trip t**

**JOIN breadcrumb b ON t.trip\_id = b.trip\_id**

**WHERE DATE(b.tstamp) = '2023-01-07’**

**GROUP BY t.vehicle\_id**

**ORDER BY distance DESC;**

A computer screen shot of a number

Description automatically generated

**SELECT t.vehicle\_id, AVG(b.speed) AS avg\_speed, SUM(b.speed) AS distance**

**FROM trip t**

**JOIN breadcrumb b ON t.trip\_id = b.trip\_id**

**WHERE DATE(b.tstamp) = '2023-01-11'**

**GROUP BY t.vehicle\_id**

**ORDER BY avg\_speed DESC;**

**A screenshot of a computer screen

Description automatically generated**

**SELECT t.vehicle\_id, AVG(b.speed) AS avg\_speed, SUM(b.speed) AS distance**

**FROM trip t**

**JOIN breadcrumb b ON t.trip\_id = b.trip\_id**

**WHERE DATE(b.tstamp) = '2023-01-11’**

**GROUP BY t.vehicle\_id**

**ORDER BY distance DESC;**

A screenshot of a computer program

Description automatically generated

**When comparing the maximum distance travelled on weekdays is more than the weekends but the speed is more on weekends than weekdays.**

Your Code

Provide a reference to the repository where you store your python code. If you are keeping it private then share it with the Professor ([rbi@pdx.edu](mailto:rbi@pdx.edu) or [mina8@pdx.edu](mailto:mina8@pdx.edu)) and TA ([vysali@pdx.edu](mailto:vysali@pdx.edu)).