DataEng S23: Project Assignment 3

Data Integration

Team Name: Data Engineers

Team Members:

- Vijayalaxmi Pujar
- Chitradevi Maruthavanan
- Susan Onesky
- Prachi Kashyap

Visualization 1. A visualization of speeds for a single trip for any bus route that crosses the US-26 tunnel. You choose the day, time and route for your selected trip. To find a trip that traverses this tunnel, consider finding a trip that includes breadcrumb sensor points within this bounding box: [(45.506022, -122.711662), (45.516636, -122.700316)]. Any bus trip that includes breadcrumb points within that box either drove across the tunnel or teleported across!

Answer:

Query used to find the trip:

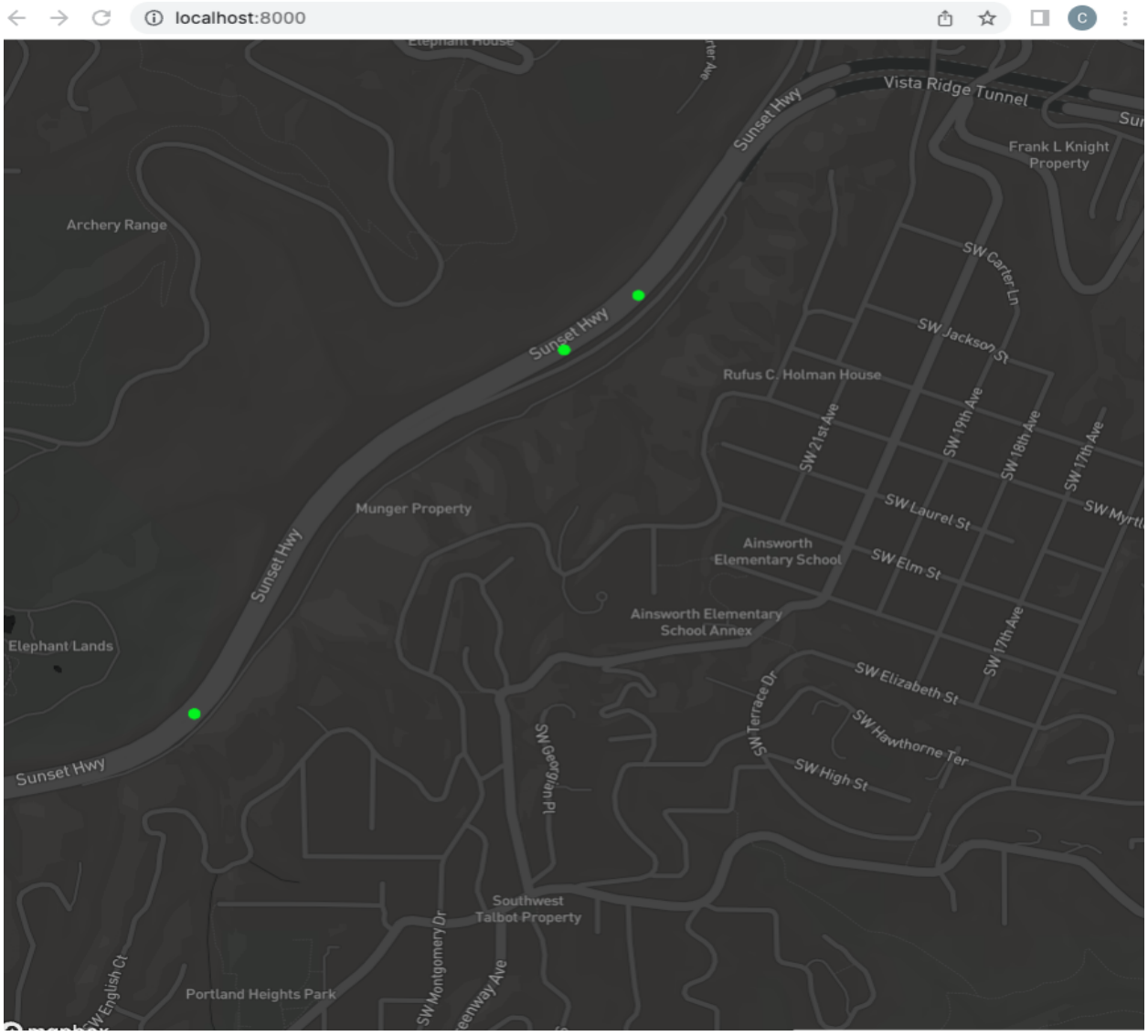
```
SELECT trip_id, COUNT(*) AS event_count
FROM breadcrumb
WHERE latitude BETWEEN 45.506022 AND 45.516636
AND longitude BETWEEN -122.711662 AND -122.700316
GROUP BY trip_id
ORDER BY event_count DESC
LIMIT 3;
```

select * from breadcrumb where trip_id = 232545534;

Trip_id: 232545534

PGPASSWORD=password psql -h localhost -p 5432 -U postgres -d postgres -A -F $'\t' -c "SELECT latitude, longitude, speed FROM breadcrumb WHERE trip_id = '232545534' AND latitude BETWEEN 45.506022 AND 45.516636 AND longitude BETWEEN -122.711662 AND -122.700316 ORDER BY tstamp;" -o output3.tsv

Bus Speed for a single trip that crosses the US-26 tunnel on Thursday,(Jan 5, 2023) with trip_id 232545534



Visualization 2. All outbound trips that occurred on route 65 on any Friday (you choose which Friday) between the hours of 4pm and 6pm.

Reason to change the route id from 65 to 72

Due to an oversight, we mistakenly fetched the wrong column for the route_id in the stopevent data for Friday's data on (May 22,2023). As a result, we missed out on the data for the route_id on route 65. After finding this error the following day, we identified the mistake and started the collection of the correct route_id column in the dataset.

```
postgres=# select b.tstamp ,t.route_id ,t.service_key
from breadcrumb b
join trip t on b.trip_id=t.trip_id where b.tstamp='2023-01-13 14:00:00';
       tstamp        | route_id | service_key
---------------------+----------+-------------
 2023-01-13 14:00:00 |    50040 | Weekday
 2023-01-13 14:00:00 |    47220 | Weekday
 2023-01-13 14:00:00 |       -1 |
 2023-01-13 14:00:00 |       -1 |
 2023-01-13 14:00:00 |    50100 | Weekday
(5 rows)

postgres=#
```

We used the same visualization 2 query and changed the route id from 65 to 72

Visualization 2. All outbound trips that occurred on route 72 on any Monday (validated as weekday) between the hours of 4pm and 6pm.

Answer

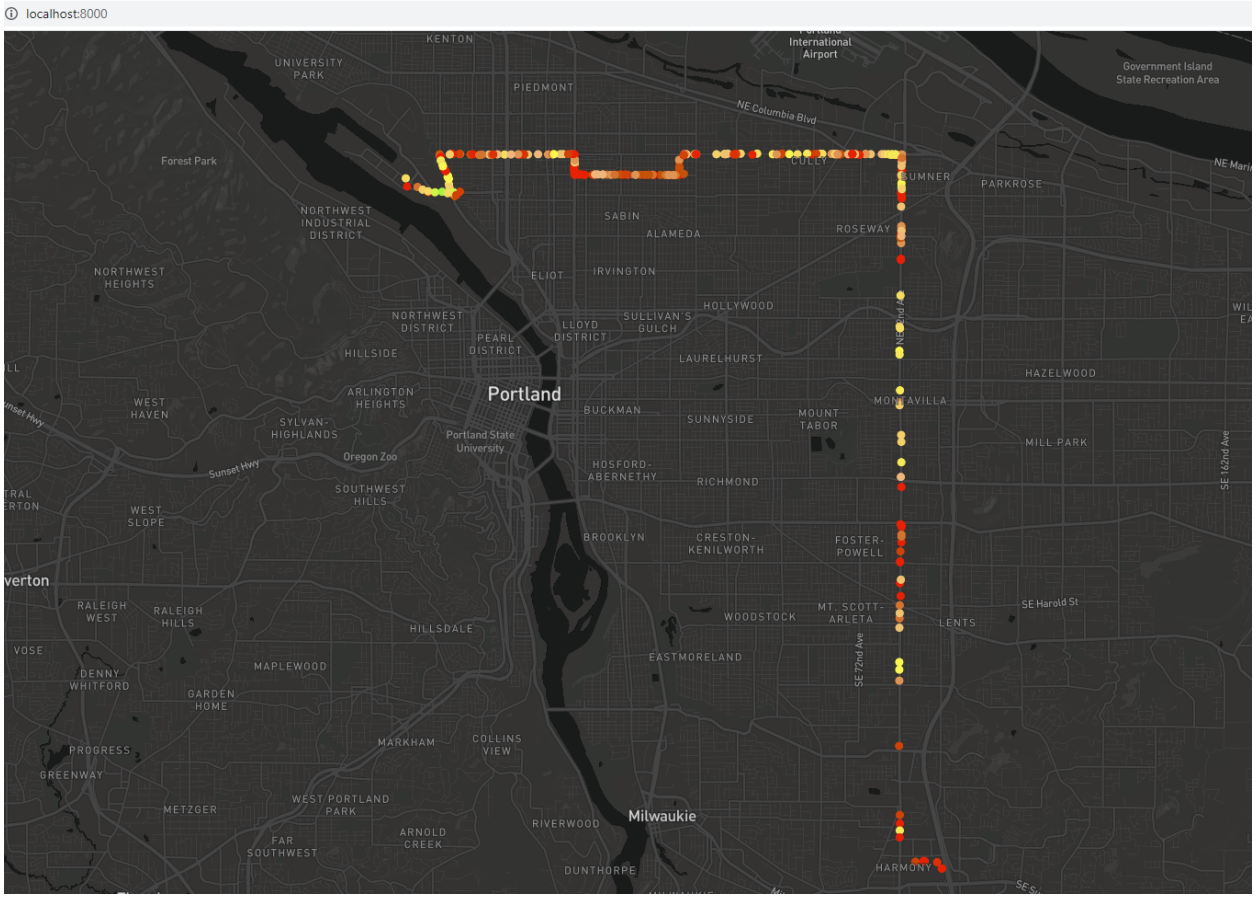
Query used to find the route_id:

```
SELECT route_id, COUNT(*) AS count
FROM breadcrumb
JOIN trip ON breadcrumb.trip_id = trip.trip_id
WHERE breadcrumb.tstamp BETWEEN '2023-01-16 04:00:00' AND '2023-01-16 06:00:00'
GROUP BY route_id
ORDER BY count DESC
LIMIT 2;
```

Route_id: 72

```
PGPASSWORD=pwd psql -h localhost -p 5432 -U postgres -d postgres -A -F $'\t' -c "
SELECT b.longitude ,b.latitude ,b.speed  FROM breadcrumb b JOIN trip t ON b.trip_id = t.trip_id
WHERE t.route_id=72 AND t.service_key='Weekday'
AND t.direction='Out' AND b.tstamp BETWEEN '2023-01-16 14:00:00' AND '2023-01-16 16:00:00';" -o output7.tsv
```

Bus Speed for all outbound trips on route 72 on Monday (Jan 16, 2023) between 4pm and 6pm



<u>Visualization 3</u>. All trips that travel to and from PSU campus on any Sunday morning (you choose which Sunday) between 9am and 11am.

<u>Answer:</u>

Query used:

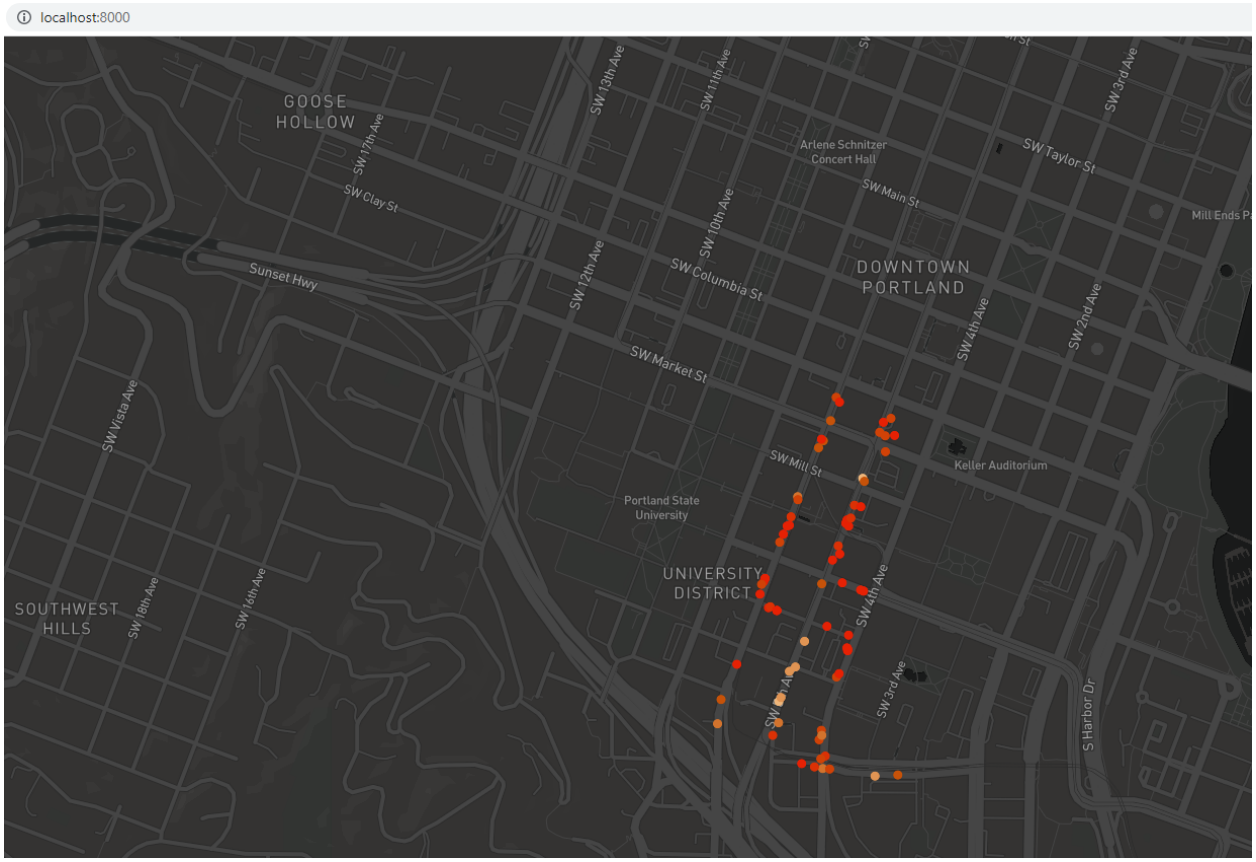
```
SELECT b.longitude, b.latitude, b.speed
FROM breadcrumb b
JOIN trip t ON b.trip_id = t.trip_id
WHERE b.latitude BETWEEN 45.514297 AND 45.509282
AND b.longitude BETWEEN -122.688863 AND -122.681311
AND t.service_key = 'Sunday'
AND b.tstamp BETWEEN '2023-01-15 09:00:00' AND '2023-01-15 11:00:00';
```

In google maps, we found the closest PSU latitude and longitude in the box that we used in our query

PGPASSWORD=pwd psql -h localhost -p 5432 -U postgres -d postgres -A -F $'\t' -c "SELECT b.longitude, b.latitude, b.speed
FROM breadcrumb b
JOIN trip t ON b.trip_id = t.trip_id

WHERE b.latitude BETWEEN 45.514297 AND 45.509282
AND b.longitude BETWEEN -122.688863 AND -122.681311
AND t.service_key = 'Sunday'
AND b.tstamp BETWEEN '2023-01-15 09:00:00' AND '2023-01-15 11:00:00';" -o output3.tsv

Bus Speed for all trips that are traveling to and from PSU on Sunday,(Jan 15, 2023) between 4pm and 6pm



<u>Visualization 4</u>. The longest (as measured by time) trip in your entire data set. Indicate the date, route #, and the trip ID of the trip along with a visualization showing the entire trip.

<u>Answer:</u>

Query used to find the longest trip:

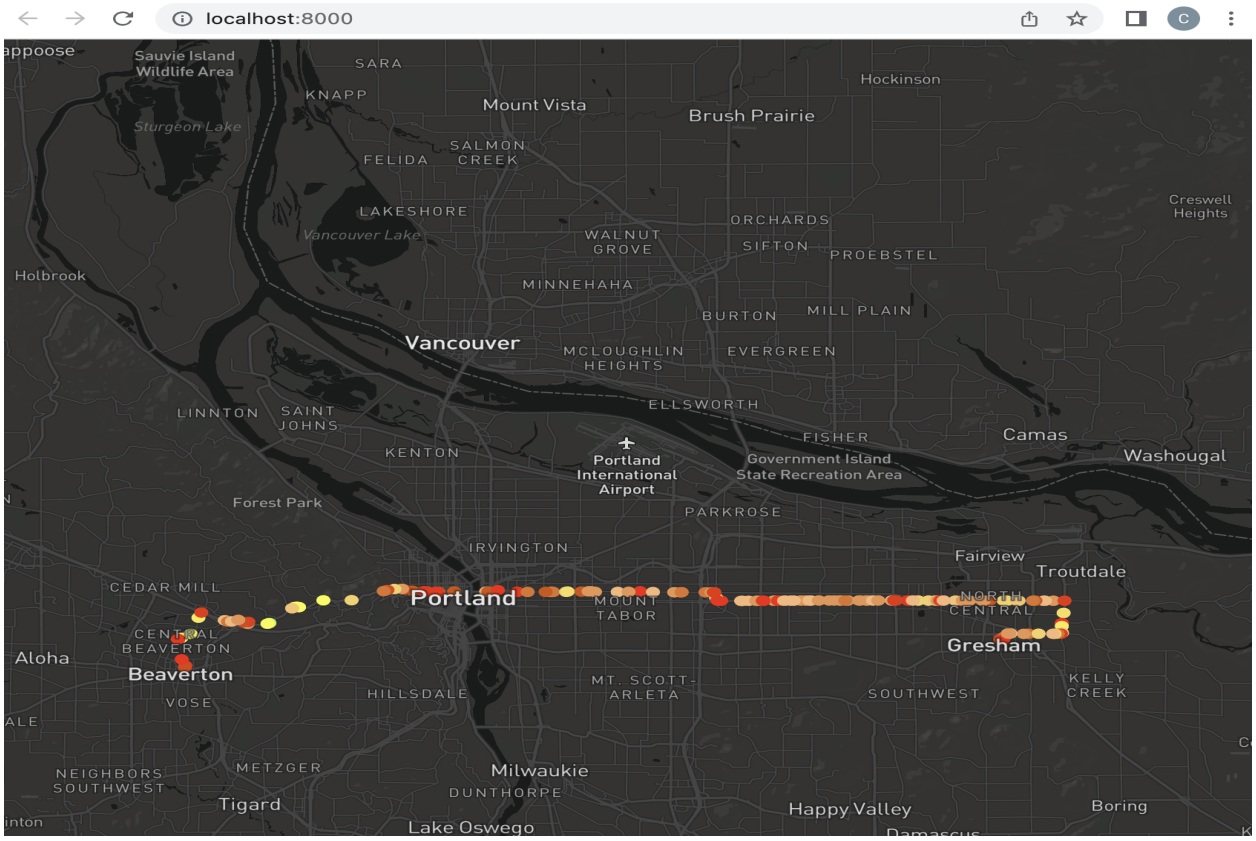
select trip_id, max(tstamp) - min(tstamp) as duration from breadcrumb group by trip_id order by duration desc limit 2;

Part- A(First-longest trip)

Date: Jan 2, 2023
Trip_id = 230596277

PGPASSWORD=password psql -h localhost -p 5432 -U postgres -d postgres -A -F $'\t' -c "select longitude,latitude,speed from breadcrumb where trip_id = 230596277;" -o output5.tsv

Bus speed for the longest trip in the trip id is 230596277 on Monday (Jan 2, 2023) from Gresham to Beaverton



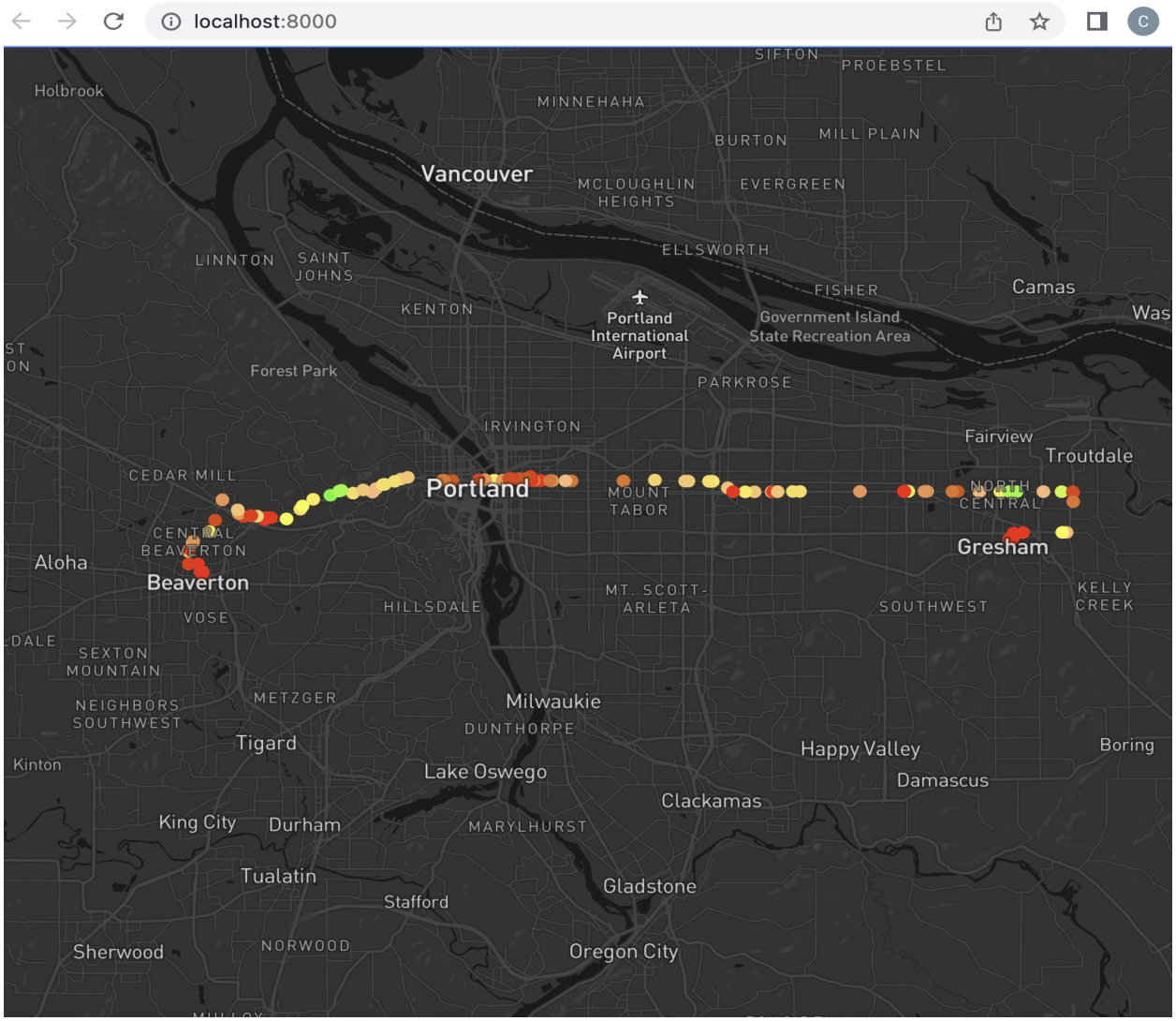
PART-B (Second longest trip)

Date: Jan 13, 2023
Trip_id = 237812592

```
select tstamp
from breadcrumb
where trip_id = 237812592;
```

```
PGPASSWORD=password psql -h localhost -p 5432 -U postgres -d postgres -A -F $'\t' -c "select
longitude,
latitude,
speed
from breadcrumb
where trip_id = 237812592;" -o output6.tsv
```

Bus speed for the second longest trip in the trip id is 237812592 on Friday(Jan 13,2023)



Visualization 5a, 5b, 5c, …. Three or more additional visualizations of your choice. Indicate why you chose each particular visualization.

5a.Visualization for the most popular route.

We wanted to visualize this as we were curious to know which is the most used route in portland.From the visulas we can see popular routes are in Portland downtown.

Query

Answer:

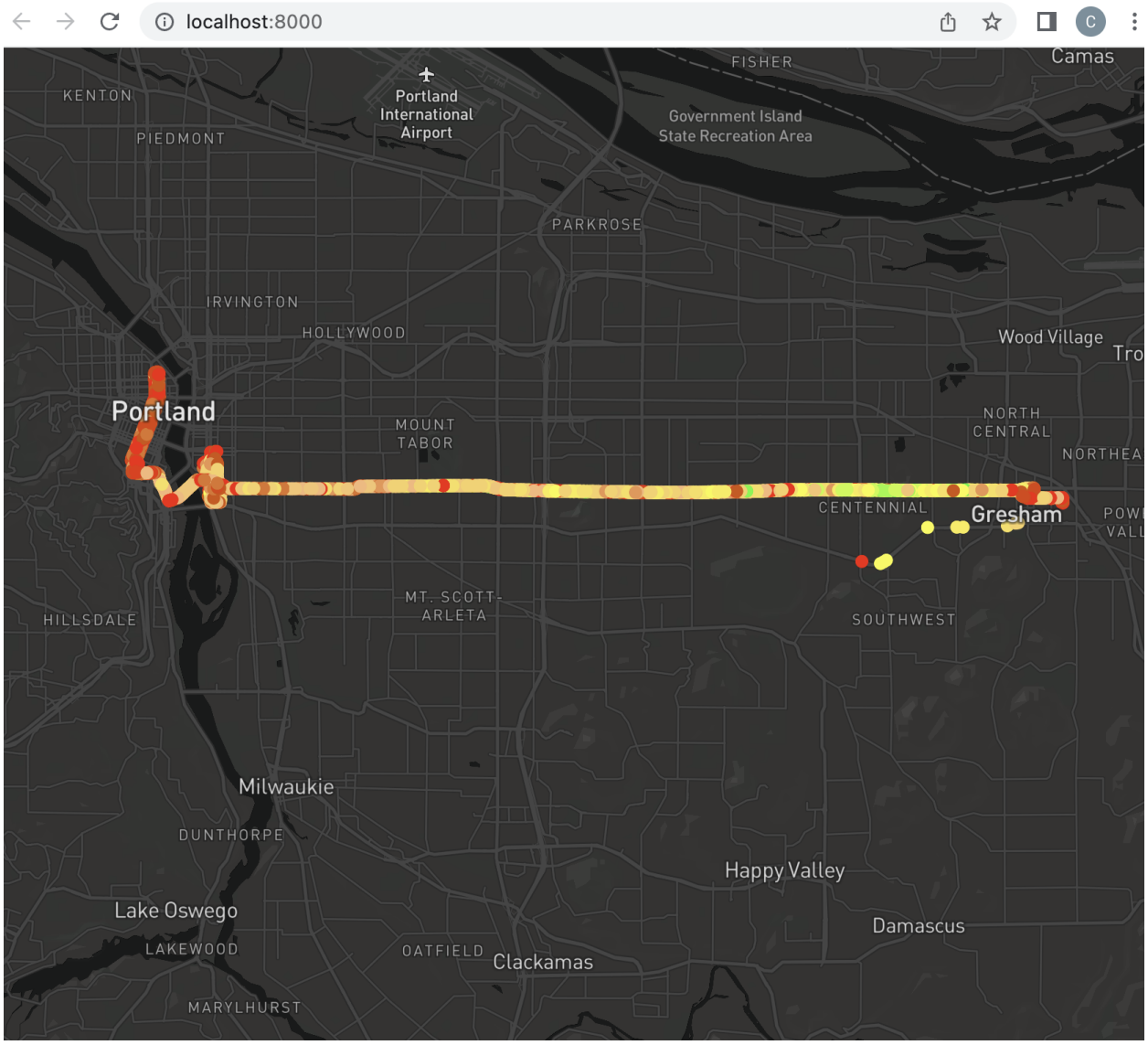
Query used to find the most popular route_id:

select route_id, count(*) total from trip group by route_id order by total desc;

PART-1(most popular route)

PGPASSWORD=password psql -h localhost -p 5432 -U postgres -d postgres -A -F $'\t' -c "SELECT b.longitude, b.latitude, b.speed
FROM breadcrumb b
JOIN trip t ON b.trip_id = t.trip_id
WHERE t.route_id = 2;" -o output7.tsv

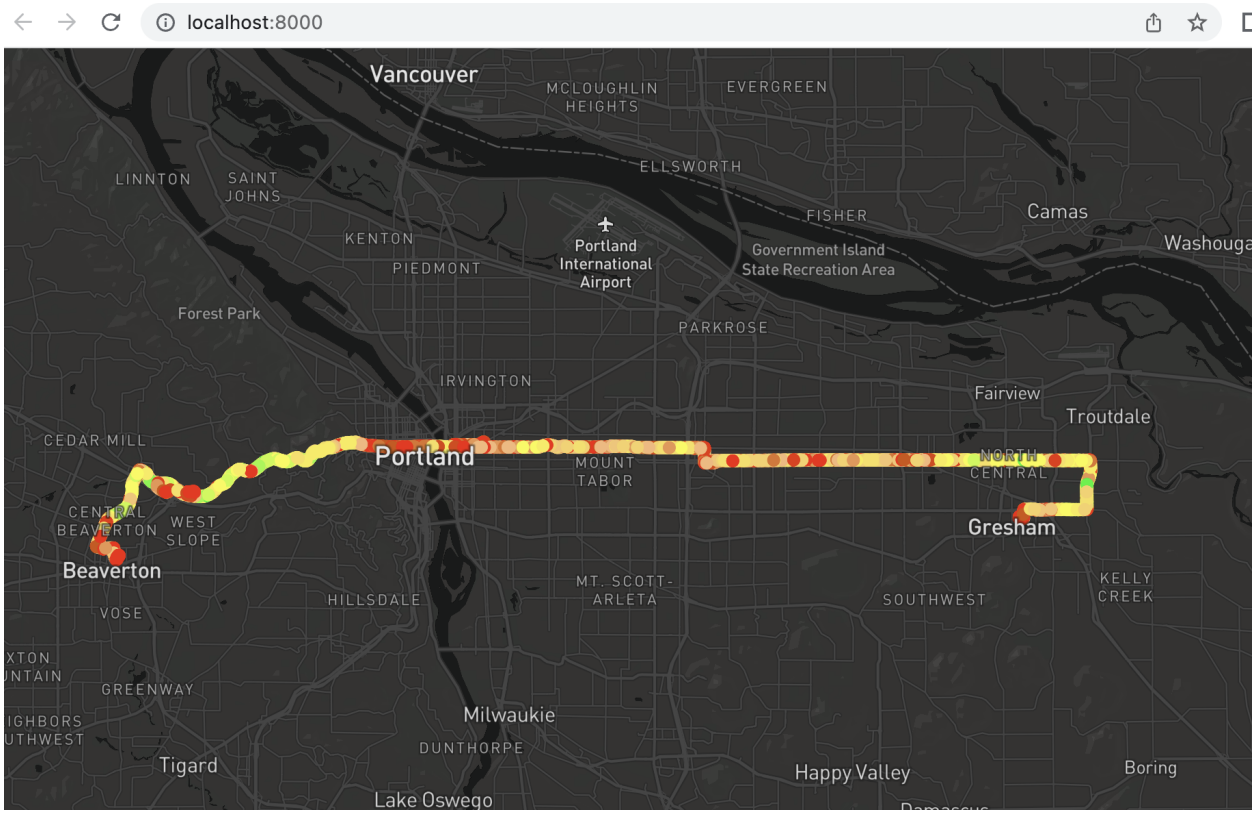
Route id 2 is the most popular route.



PART-2(Second most popular route)

PGPASSWORD=password psql -h localhost -p 5432 -U postgres -d postgres -A -F $'\t' -c "SELECT b.longitude, b.latitude, b.speed
FROM breadcrumb b
JOIN trip t ON b.trip_id = t.trip_id
WHERE t.route_id = 20;" -o output8.tsv

Route id 20 is the second most popular route.



5b..

Visualization for the least popular route.

As we visualized the most popular route we wanted to visualize the least used route as well in portland.

PART-1(Least popular route)

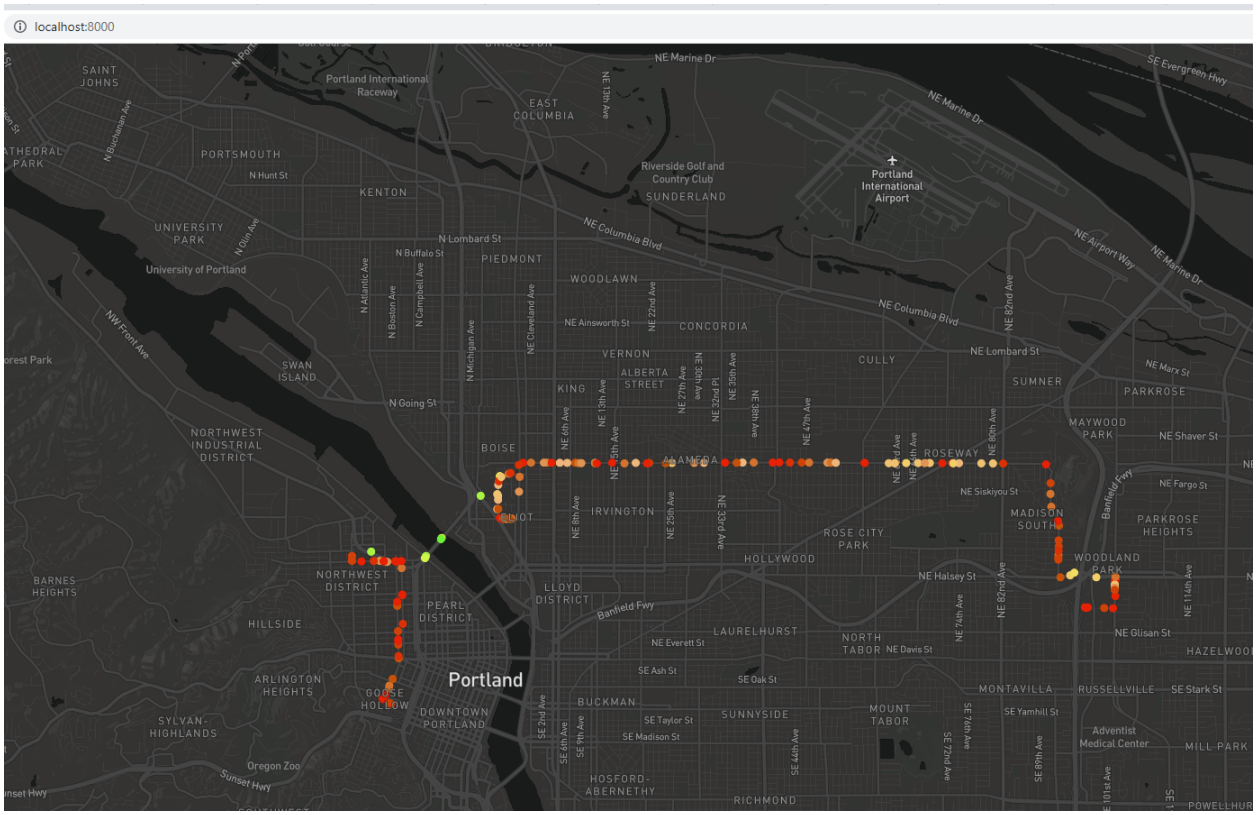
Answer:

Route id: 24

Trip id: 238341663 and 238341734

PGPASSWORD=pwd psql -h localhost -p 5432 -U postgres -d postgres -A -F $'\t' -c "SELECT b.longitude, b.latitude, b.speed
FROM breadcrumb b
JOIN trip t ON b.trip_id = t.trip_id
WHERE t.route_id = 24;
" -o output5.tsv

Route Id 24 has the second least number of trips (trip Ids 238341663 and 238341734)



PART-2(second Least popular route)

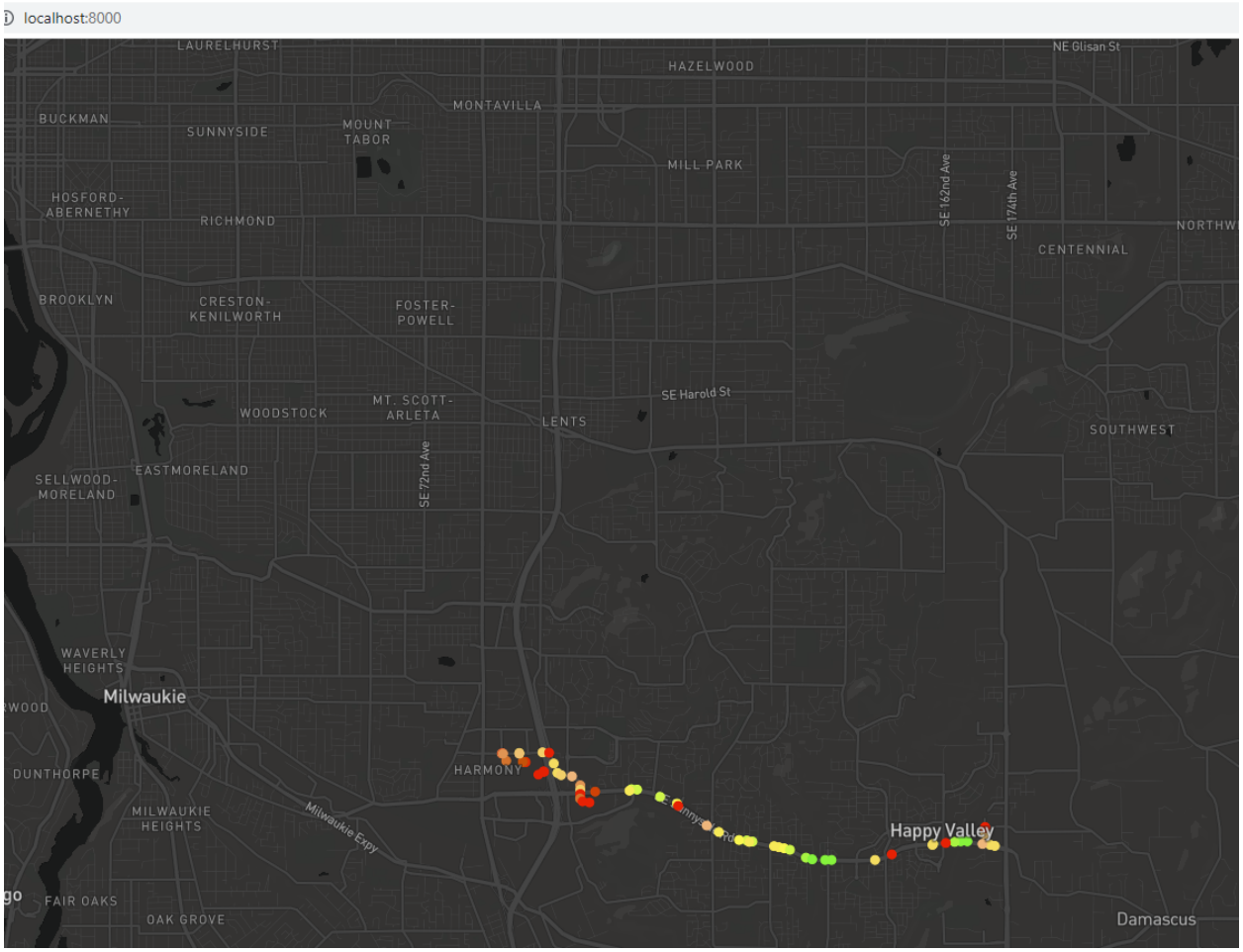
Answer:

Route id: 155

Trip id: 238341663 and 238341734

PGPASSWORD=pwd psql -h localhost -p 5432 -U postgres -d postgres -A -F $'\t' -c "SELECT b.longitude, b.latitude, b.speed
FROM breadcrumb b
JOIN trip t ON b.trip_id = t.trip_id
WHERE t.route_id = 155;" -o output6.tsv

Route Id 155 has the least number of trips (trip Ids 238336566,238336589,239373391 and 239373414)+



5c) The route taken in a trip which has the highest speed

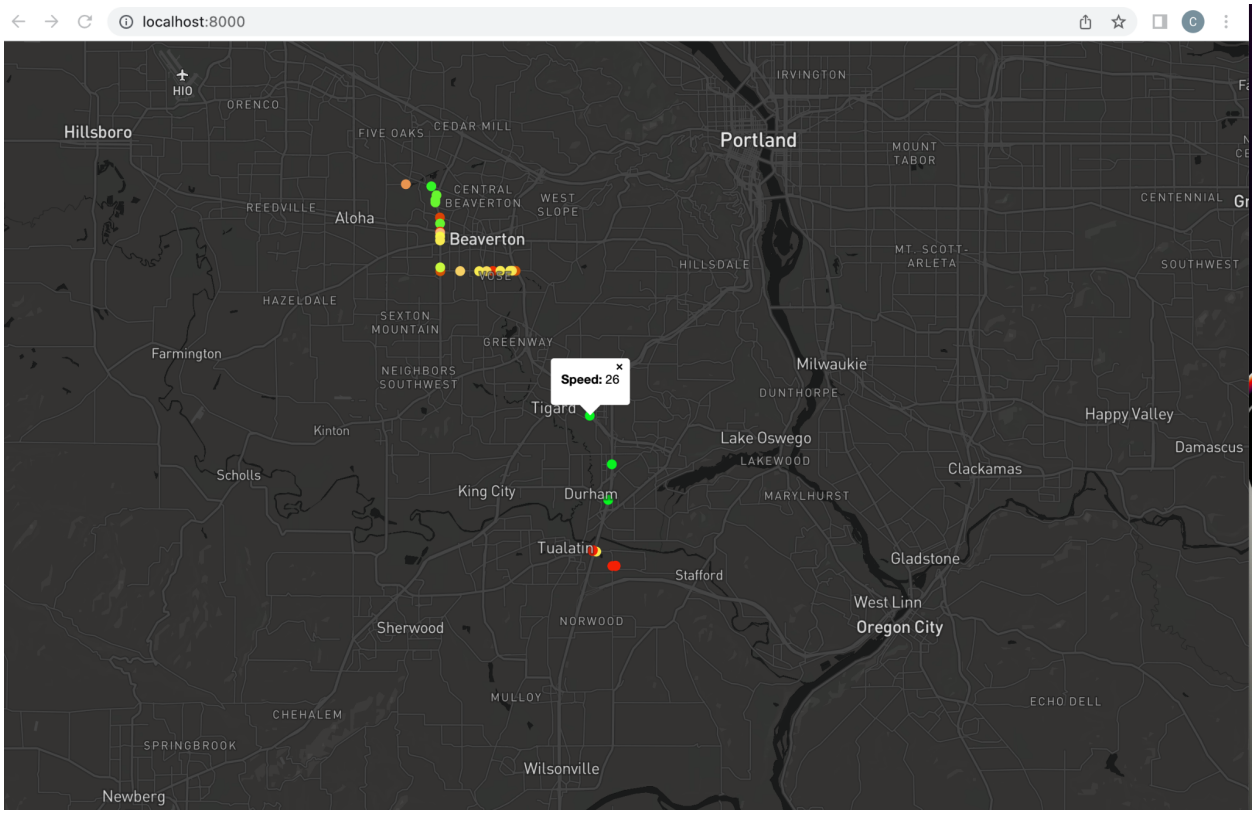
Answer:

We wanted to visualize the route of the trip which has the highest number of instances where the speed has crossed 90% of the max_speed. So, the trip has a lot of indications of speeding

To get the breadcrumb data we used the following query:

with trip_max as (with max_speed as (select max(speed) as mspeed from breadcrumb) select trip_id, count(*) as ct from breadcrumb, max_speed where speed > mspeed * 90/100 and speed <= mspeed group by trip_id order by ct desc limit 1) select b.longitude, b.latitude, b.speed from breadcrumb b, trip_max where b.trip_id = trip_max.trip_id;

*PGPASSWORD=password psql -h localhost -p 5432 -U postgres -d postgres -A -F $'\t' -c "with trip_max as (with max_speed as (select max(speed) as mspeed from breadcrumb) select trip_id, count(*) as ct from breadcrumb, max_speed where speed > mspeed * 90/100 and speed <= mspeed group by trip_id order by ct desc limit 1) select b.longitude, b.latitude, b.speed from breadcrumb b, trip_max where b.trip_id = trip_max.trip_id;" -o output9.tsv*



Your Code

Provide a reference to the repository where you store your code. If you are keeping it private then share it with Bruce and Mina.
https://github.com/Chitramvanan/DataEngineering_Project/tree/main/Project%203