**CE 5900 – 16   
Lecture 7 – In-class Activity**

**By Revanth Mamidala**

**Activity 1: Create table and load data from gps\_data, vehicle\_data**

**Script:**

-- First, create the tables

CREATE TABLE gps\_data (

datapoint\_id VARCHAR PRIMARY KEY,

journey\_id VARCHAR(50),

captured\_timestamp TIMESTAMP,

latitude NUMERIC(10, 7),

longitude NUMERIC(10, 7),

month INTEGER,

day INTEGER,

hour INTEGER

);

CREATE TABLE vehicle\_data (

datapoint\_id VARCHAR PRIMARY KEY,

geohash VARCHAR(12),

speed NUMERIC(5, 2),

make VARCHAR(50),

model VARCHAR(50),

route\_id VARCHAR(50),

segment\_start\_measure NUMERIC(5, 2)

);

-- Now, use the COPY command to import data into gps\_data

COPY gps\_data(datapoint\_id, journey\_id, latitude, longitude, month, day, hour)

--FROM 'C:\\Users\\revan\\Desktop\\Data\\gps\_data.csv'

FROM 'D:\\gps\_data.csv'

DELIMITER ','

CSV HEADER;

-- COPY command for vehicle\_data

COPY vehicle\_data(datapoint\_id, geohash, speed, make, model, route\_id, segment\_start\_measure)

FROM 'D:\\vehicle\_data.csv'

DELIMITER ','

CSV HEADER;

Output:

A screenshot of a computer

Description automatically generated

A close-up of a computer screen

Description automatically generated

**Activity 2: Filtering**

1. Select datapoints from October 3, 4AM

**Script:**

SELECT \*

FROM gps\_data

WHERE month = 10 AND day = 3 AND hour = 4;

SELECT COUNT(\*)

FROM gps\_data;

SELECT COUNT(DISTINCT journey\_id)

FROM gps\_data

WHERE month = 10 AND day = 3 AND hour = 4;

**Output:**A screenshot of a computer

Description automatically generated

1. Count how many rows in gps\_data table (Ans: 244614)

**Script and Output:**

A screenshot of a computer

Description automatically generated

1. Find how many unique journey\_ids in the data filtered as per (1) (Ans: 7)

A screenshot of a computer

Description automatically generated

1. Filter data on Oct 1st and 3rd and hour between 3AM and 6AM (101116 rows)

Solution:

If we select only days Oct 1 and 3:

A screenshot of a computer

Description automatically generated

If we select days between Oct 1 and 3:

A screenshot of a computer

Description automatically generated

1. Select all journey\_ids contain “abc” from whole gps\_data table (find 8 of them)

A screenshot of a computer

Description automatically generated

1. Count how many datapoint\_ids start with “9” (Ans: 15256)

A screenshot of a computer

Description automatically generated

1. Select all rows where journey\_id ends with “cc” (66)

A screenshot of a computer

Description automatically generated

8. Select all columns, also prepare a new column that combines datapoint\_id and day with a “-”

symbol

Ex(“sajgdfaj1334254kgjhsa-3”) and name it dpid\_day

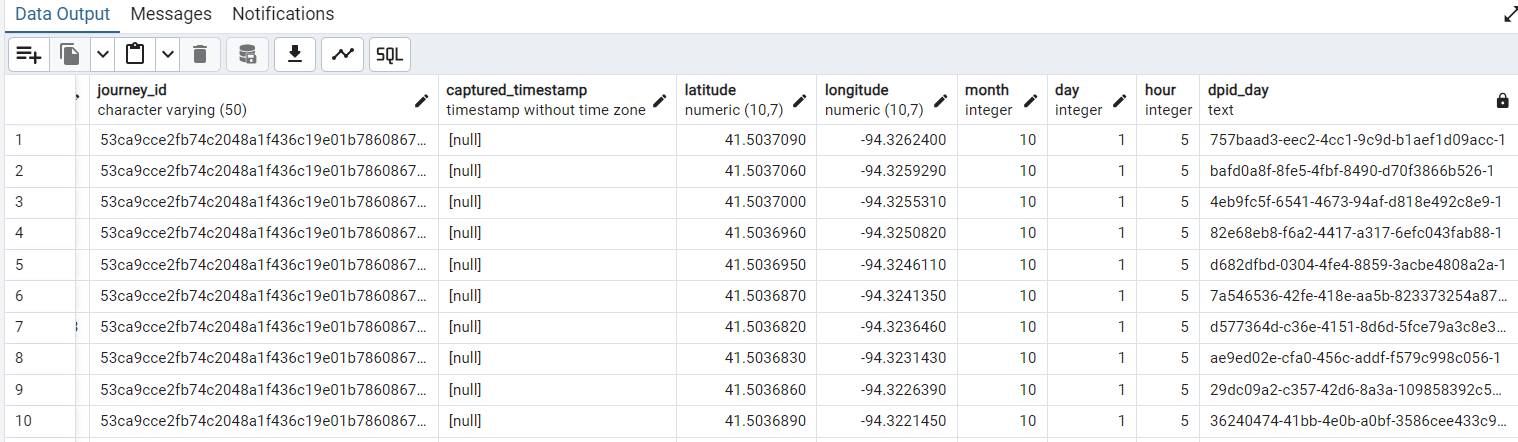
**Script:**

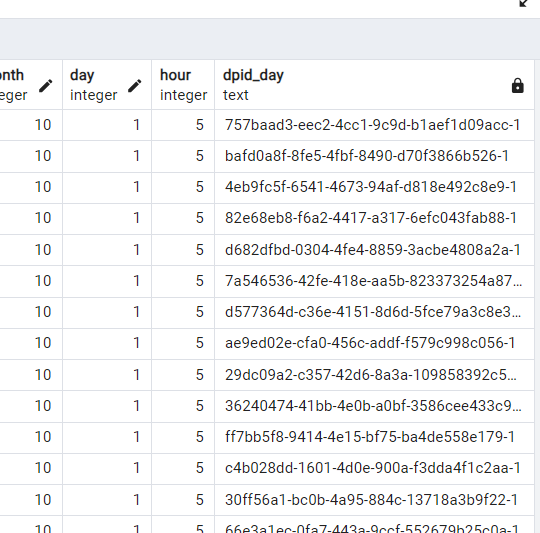
SELECT \*,

CONCAT(datapoint\_id, '-', day) AS dpid\_day

FROM gps\_data;

**Output:**

****



**Activity – 3: Aggregations**

1. Find maximum speed from table vehicle\_data (Ans: 162.43)

A screenshot of a computer

Description automatically generated

1. Find average, minimum, maximum and standard deviations of speeds for “CHEVROLET” make

A screenshot of a computer

Description automatically generated

1. Create seg\_id by combining “route\_id” and “segment\_start\_measure”

**Script:**

SELECT \*,

CONCAT(route\_id, '-', segment\_start\_measure) AS seg\_id

FROM vehicle\_data;

**Output:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Add this seg\_id to existing vehicle\_data table and verify it • Alter\_table and update

**Script:**

-- Alter the Table to Add the seg\_id Column

ALTER TABLE vehicle\_data

ADD COLUMN seg\_id VARCHAR(100);

-- Update the Table to Populate seg\_id

UPDATE vehicle\_data

SET seg\_id = CONCAT(route\_id, '-', segment\_start\_measure);

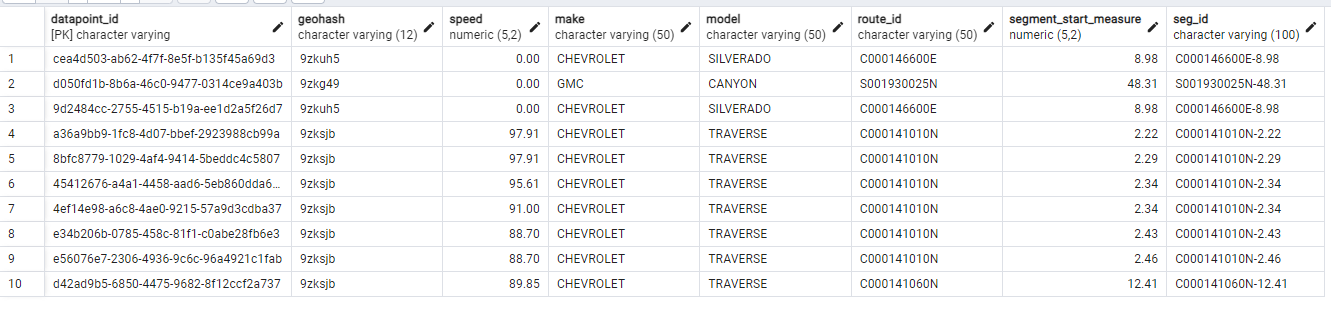
-- Verify the seg\_id Column

SELECT \*

FROM vehicle\_data

LIMIT 10; -- Use LIMIT to view the first 10 rows for verification

**Output:**



A screenshot of a computer

Description automatically generated

1. Find 85th percentiles speeds for each segment\_id

**Script:**

SELECT seg\_id,

PERCENTILE\_CONT(0.85) WITHIN GROUP (ORDER BY speed) AS speed\_85th\_percentile

FROM vehicle\_data

GROUP BY seg\_id;

**Output:**

**A screenshot of a data output

Description automatically generated**

1. Find the highest speeds for each model? (Compare 95th percentile speeds for each model)

**Script:**

SELECT model,

PERCENTILE\_CONT(0.95) WITHIN GROUP (ORDER BY speed) AS speed\_95th\_percentile

FROM vehicle\_data

GROUP BY model;

**Output:**

|  |  |
| --- | --- |
| **model** | **speed\_95th\_percentile** |
| ACADIA | 126.71 |
| ATS | 118.65 |
| AVEO\_SONIC | 130.17 |
| BLAZER | 126.71 |
| CAMARO | 127.87 |
| CANYON | 80.64 |
| COLORADO | 129.02 |
| CORVETTE\_STINGRAY | 124.41 |
| CRUZE | 129.02 |
| CTS | 127.87 |
| ENCLAVE | 123.26 |
| ENCORE | 136.4474999999998 |
| ENVISION | 131.32 |
| EQUINOX | 126.71 |
| ESCALADE | 130.17 |
| ESCALADE\_ESV | 23.5575 |
| IMPALA | 125.56 |
| LACROSSE | 133.63 |
| MALIBU | 131.32 |
| SIERRA | 126.71 |
| SIERRA\_HD | 130.17 |
| SILVERADO | 127.87 |
| SILVERADO\_HD | 125.56 |
| SPARK | 131.32 |
| SRX | 118.65 |
| SUBURBAN | 130.17 |
| TAHOE | 126.71 |
| TERRAIN | 124.41 |
| TRAVERSE | 129.02 |
| TRAX\_TRACKER\_TRAILBLAZER\_B | 127.87 |
| VERANO | 127.87 |
| XT4 | 125.56 |
| YUKON | 131.32 |
| YUKON\_XL | 127.87 |

**Activity 4: Joins**

1. Join the “gps\_data” and “vehicle\_data” tables using “JOIN” on datapoint\_id

**Script:**

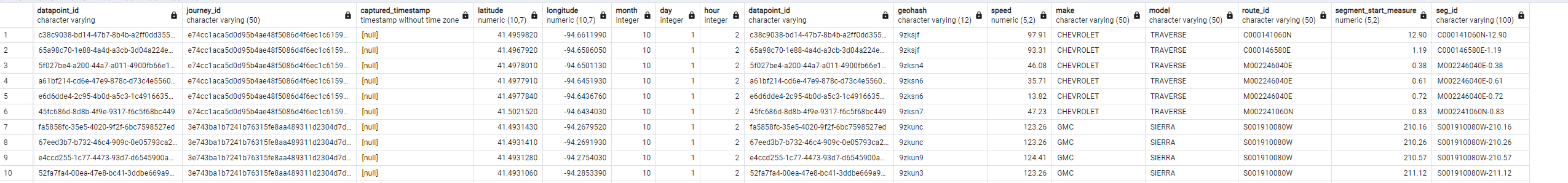
SELECT \*

FROM gps\_data

JOIN vehicle\_data

ON gps\_data.datapoint\_id = vehicle\_data.datapoint\_id;

**Output:**



1. Select datapoint\_id, journey\_id, speed, seg\_id from the above join

**Script:**

SELECT gps\_data.datapoint\_id,

gps\_data.journey\_id,

vehicle\_data.speed,

vehicle\_data.seg\_id

FROM gps\_data

JOIN vehicle\_data

ON gps\_data.datapoint\_id = vehicle\_data.datapoint\_id;

**Output:**

A screenshot of a computer code

Description automatically generated

1. Count the unique journey\_ids that came to stop at least once (Speed = 0)

**Output:**

A screenshot of a computer program

Description automatically generated

**Activity 5: Nested queries**

1. Filter points in “gps\_data” with model = GMC from “vehicle\_data”
   1. Use join

**Script:**

SELECT gps\_data.\*

FROM gps\_data

JOIN vehicle\_data

ON gps\_data.datapoint\_id = vehicle\_data.datapoint\_id

WHERE vehicle\_data.make = 'GMC';

**Output:**

**A screenshot of a computer

Description automatically generated**

* 1. Use nested query

**Script:**SELECT \*  
FROM gps\_data  
WHERE datapoint\_id IN (  
 SELECT datapoint\_id  
 FROM vehicle\_data  
 WHERE make = 'GMC'  
);

**Output:**

A screenshot of a computer

Description automatically generated

1. Count number of unique journeys in each car make and model using nested query order by make and model in descending order for each

**Script:**

SELECT make, model, COUNT(DISTINCT journey\_id) AS unique\_journey\_count

FROM (

SELECT gps\_data.journey\_id, vehicle\_data.make, vehicle\_data.model

FROM gps\_data

JOIN vehicle\_data

ON gps\_data.datapoint\_id = vehicle\_data.datapoint\_id

) AS combined\_data

GROUP BY make, model

ORDER BY make DESC, model DESC;

**Output:**

A screenshot of a computer

Description automatically generated

1. Filter the gps data of the journey with highest speed from the whole dataset
   1. You should find the jounrney\_id with max speed from joining tables (JID: a37cf85fe887cb5b54fd828664ace413938a6e5d)

A screenshot of a computer

Description automatically generated

* 1. Then filter the corresponding journey data from gps\_data (Ans: 343 rows)

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Activity 6:**

1. Find distance between points below:
   * Point 1: 41.948637, -93.610065, Point 2: 41.997339, -93.610229

A screenshot of a computer

Description automatically generated

1. Create a line string from above points and calculate distance from Point 3 : (41.972017, 93.619738)

A screenshot of a computer program

Description automatically generated

1. Make point geometry from latitude and longitude in gps\_data table

A screenshot of a computer code

Description automatically generated

1. Create a buffer around point: 3 for 3 miles and filter all datapoints from gps\_data within the buffer

**Script:**

According to the script result, the nearest point to the given point: 3 is at 54 miles distance.

A screenshot of a computer program

Description automatically generated

The details of the point are as below:

A screenshot of a computer

Description automatically generated