**Activity 2**

create table gps\_data (

datapoint\_id varchar ,

journey\_id varchar ,

latitude float,

longitude float,

month int,

day int,

hour int,

primary key (datapoint\_id)

);

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

from 'D:\gps\_data.csv'

delimiter ','

csv header;

select \* from gps\_data

where month = 10

select \* from gps\_data

where month = 10 and day = 3 and hour = 4

SELECT COUNT(DISTINCT journey\_id)

FROM gps\_data

where month = 10 and day = 3 and hour >= 4

AND month = 10 and day = 3 and hour < 5

SELECT \*

FROM gps\_data

WHERE month = 10 and day >= 1 and day <= 3

AND hour >= 3 and hour <= 6

SELECT journey\_id

FROM gps\_data

WHERE journey\_id LIKE '%abc%';

SELECT COUNT(\*)

FROM gps\_data

WHERE datapoint\_id LIKE '9%';

SELECT \*

FROM gps\_data

WHERE journey\_id LIKE '%cc';

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

**Activity 3**

create table vehicle\_data (

datapoint\_id varchar ,

geohash varchar ,

speed float ,

make varchar ,

model varchar ,

route\_id varchar ,

segment\_start\_measure float,

primary key (datapoint\_id)

);

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

from 'D:\vehicle\_data.csv'

delimiter ','

csv header;

SELECT MAX(speed) AS max\_speed

FROM vehicle\_data;

SELECT

AVG(speed) AS avg\_speed,

MIN(speed) AS min\_speed,

MAX(speed) AS max\_speed,

STDDEV(speed) AS stddev\_speed

FROM vehicle\_data

WHERE make = 'CHEVROLET';

SELECT

route\_id || '\_' || segment\_start\_measure AS seg\_id

FROM vehicle\_data;

-- Step 1: Add the new column to the table

ALTER TABLE vehicle\_data

ADD COLUMN seg\_id TEXT;

-- Step 2: Update the table to populate seg\_id

UPDATE vehicle\_data

SET seg\_id = route\_id || '\_' || segment\_start\_measure;

-- Step 3 (Optional): Verify the new column values

SELECT route\_id, segment\_start\_measure, seg\_id

FROM vehicle\_data;

SELECT

seg\_id,

PERCENTILE\_CONT(0.85) WITHIN GROUP (ORDER BY speed) AS percentile\_85\_speed

FROM

vehicle\_data

GROUP BY

seg\_id;

SELECT

model,

PERCENTILE\_CONT(0.95) WITHIN GROUP (ORDER BY speed) AS percentile\_95\_speed

FROM

vehicle\_data

GROUP BY

model;

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

**Activity 4**

SELECT

vehicle\_data.\*,

gps\_data.\*

FROM

vehicle\_data

JOIN

gps\_data

ON

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

-- Step 1: Join tables and select specific columns

WITH joined\_data AS (

SELECT

vehicle\_data.datapoint\_id,

gps\_data.journey\_id,

vehicle\_data.speed,

vehicle\_data.seg\_id

FROM

vehicle\_data

JOIN

gps\_data

ON

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

)

-- Step 2: Count unique journey\_ids that came to stop at least once

SELECT

COUNT(DISTINCT journey\_id) AS stopped\_journey\_count

FROM

joined\_data

WHERE

speed = 0;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated A screenshot of a computer program

Description automatically generated

**Activity 5.**

-- Step 1: Join tables and select specific columns

WITH joined\_data AS (

SELECT

vehicle\_data.datapoint\_id,

gps\_data.journey\_id,

vehicle\_data.speed,

vehicle\_data.seg\_id

FROM

vehicle\_data

JOIN

gps\_data

ON

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

)

-- Step 2: Count unique journey\_ids that came to stop at least once

SELECT

make,

model,

COUNT(DISTINCT journey\_id) AS unique\_journey\_count

FROM

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

FROM vehicle\_data

JOIN gps\_data ON vehicle\_data.datapoint\_id = gps\_data.datapoint\_id) AS joined\_data

GROUP BY

make, model

ORDER BY

make DESC, model DESC;

3.

SELECT

journey\_id,

MAX(speed) AS max\_speed

FROM

vehicle\_data

JOIN

gps\_data

ON

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

GROUP BY

journey\_id

ORDER BY

max\_speed DESC

LIMIT 1;

-- Using a subquery to dynamically find the journey with the highest speed

SELECT

gps\_data.\*

FROM

gps\_data

JOIN

(SELECT journey\_id

FROM vehicle\_data

JOIN gps\_data

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

GROUP BY journey\_id

ORDER BY MAX(speed) DESC

LIMIT 1) AS highest\_speed\_journey

ON

gps\_data.journey\_id = highest\_speed\_journey.journey\_id;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated