Activity 0: Create a folder with your name

A screenshot of a computer

Description automatically generated

Activity 1: Creating tables

A screenshot of a computer

Description automatically generated

Activity 2: GeoSQL

SELECT ST\_POINT(-93.610065, 41.948637) AS point1

SELECT ST\_DISTANCE(ST\_POINT(-93.610065, 41.948637), ST\_POINT(-93.610229, 41.997339)) AS distance\_between\_points

WITH points AS (

SELECT

ST\_POINT(-93.610065, 41.948637) AS point1,

ST\_POINT(-93.610229, 41.997339) AS point2,

ST\_POINT(-93.619738, 41.972017) AS point3

)

SELECT

ST\_DISTANCE(ST\_LINESTRING(array[point1, point2]), point3) AS distance\_from\_point3

FROM points

WITH points AS (

SELECT

ST\_POINT(-93.610065, 41.948637) AS point1,

ST\_POINT(-93.610229, 41.997339) AS point2,

ST\_POINT(-93.619738, 41.972017) AS point3

)

SELECT

ST\_DISTANCE(ST\_LINESTRING(array[point1, point2]), point3) \* 364567 AS distance\_from\_point3\_feet

FROM points

SELECT

ST\_POINT(longitude, latitude) AS point\_geometry,

latitude,

longitude

FROM sam\_gps\_data

WITH buffer\_area AS (

SELECT ST\_BUFFER(ST\_POINT(-94.295980, 41.502954), 0.05) AS buffer

)

SELECT

g.\*,

ST\_POINT(g.longitude, g.latitude) AS point\_geometry

FROM

sam\_gps\_data AS g,

buffer\_area

WHERE

ST\_CONTAINS(buffer, ST\_POINT(g.longitude, g.latitude))

A screenshot of a computer

Description automatically generated

Activity 3: Data conflation through spatial join

WITH line\_segments AS (

SELECT

route\_id,

ST\_LINESTRING(ARRAY[

ST\_POINT(segment\_start\_longitude, segment\_start\_latitude),

ST\_POINT(segment\_end\_longitude, segment\_end\_latitude)

]) AS route\_line

FROM sam\_lrn\_data

WHERE NOT (segment\_start\_latitude = segment\_end\_latitude AND segment\_start\_longitude = segment\_end\_longitude) -- Exclude duplicate points

),

point\_to\_line\_distances AS (

SELECT

g.datapointid,

g.journeyid,

g.latitude,

g.longitude,

l.route\_id,

ST\_DISTANCE(ST\_POINT(g.longitude, g.latitude), l.route\_line) AS distance\_to\_route

FROM (

SELECT \*

FROM sam\_gps\_data

WHERE hour = 1 AND day = 2 -- Filter for test run

) AS g

CROSS JOIN line\_segments AS l

)

SELECT

datapointid,

journeyid,

latitude,

longitude,

route\_id,

distance\_to\_route

FROM (

SELECT

datapointid,

journeyid,

latitude,

longitude,

route\_id,

distance\_to\_route,

ROW\_NUMBER() OVER (PARTITION BY datapointid ORDER BY distance\_to\_route) AS row\_num

FROM point\_to\_line\_distances

) AS ranked\_distances

WHERE row\_num = 1

A screenshot of a computer

Description automatically generated