-- Consolidated Athena Query for CV and Wavetronix Data Processing

WITH cv\_with\_sensor AS (

SELECT

cv.datapointid,

cv.journeyid,

cv.capturedtimestamp,

cv.latitude AS cv\_latitude,

cv.longitude AS cv\_longitude,

cv.route\_id,

cv.hour,

wl.device\_id AS sensor\_device\_id,

wl.latitude AS sensor\_latitude,

wl.longitude AS sensor\_longitude,

-- Calculate approximate distance for proximity matching

sqrt(power(cv.latitude - wl.latitude, 2) + power(cv.longitude - wl.longitude, 2)) AS distance

FROM

raghu\_cv\_data AS cv

JOIN

raghu\_wavetronix\_locations AS wl

ON

cv.route\_id = wl.route\_id

),

cv\_hourly\_count AS (

SELECT

sensor\_device\_id AS device\_id,

hour,

COUNT(DISTINCT journeyid) AS cv\_count

FROM

cv\_with\_sensor

WHERE distance < 0.1 -- Filter by distance threshold for nearest sensor

GROUP BY

sensor\_device\_id, hour

),

wavetronix\_hourly\_count AS (

SELECT

device\_id,

hour(cst\_time) AS hour,

SUM(lane\_count) AS total\_vehicle\_count

FROM

raghu\_wavetronix\_data

GROUP BY

device\_id, hour(cst\_time)

)

SELECT

cv.device\_id,

cv.hour,

cv.cv\_count,

wt.total\_vehicle\_count,

-- Calculate proportion of CVs if desired

IF(wt.total\_vehicle\_count > 0, CAST(cv.cv\_count AS DECIMAL(10, 3)) / CAST(wt.total\_vehicle\_count AS DECIMAL(10, 3)), 0) AS penetration\_rate

FROM

cv\_hourly\_count AS cv

JOIN

wavetronix\_hourly\_count AS wt

ON

cv.device\_id = wt.device\_id

AND cv.hour = wt.hour

ORDER BY

cv.device\_id, cv.hour;