-- ■ Marketing Analytics:

-- ● Analyse marketing campaign performance by channel (spend, conversions, and conversion rate).

SELECT channel,

SUM(spend) AS total\_spend,

SUM(conversions) AS total\_conversions,

(SUM(conversions) / SUM(clicks)) \* 100 AS conversion\_rate

FROM `appdataanalysis`.`app\_data`.`marketing\_events`

GROUP BY channel;

-- ● Calculate the Cost Per Acquisition (CPA) for each campaign.

SELECT campaign\_id,

round(spend / conversions,3) AS cpa

FROM `appanalysis-447013`.`app\_data`.`marketing\_events`

WHERE conversions > 0;

-- -- ● Identify which channels are most cost-effective in terms of conversions.

SELECT channel,

round(SUM(conversions) / SUM(spend),4) AS cost\_effectiveness

FROM `appanalysis-447013`.`app\_data`.`marketing\_events`

WHERE spend > 0

GROUP BY channel

ORDER BY cost\_effectiveness DESC;

-- ■ Product Analytics:

-- ● Analyse feature adoption rates (how many users engaged with specific features).

SELECT feature\_name,

COUNT(DISTINCT user\_id) AS adoption\_rate

FROM `appdataanalysis`.`app\_data`.`app\_events`

GROUP BY feature\_name

ORDER BY 2 DESC;

-- ● Calculate daily or weekly active users (DAU/WAU).

-- DAU:

SELECT event\_date,

COUNT(DISTINCT user\_id) AS daily\_active\_users

FROM `appdataanalysis`.`app\_data`.`app\_events`

GROUP BY event\_date

ORDER BY 2 DESC;

-- WAU:

SELECT DATE\_TRUNC(event\_date, WEEK) AS week\_start,

COUNT(DISTINCT user\_id) AS weekly\_active\_users

FROM `appdataanalysis`.`app\_data`.`app\_events`

GROUP BY week\_start;

-- -- ● Perform cohort analysis to track user retention over time.

WITH cohorts AS (

SELECT user\_id, MIN(event\_date) AS cohort\_date

FROM `appdataanalysis`.`app\_data`.`app\_events`

GROUP BY user\_id

),

cohort\_retention AS (

SELECT  cohorts.cohort\_date,

COUNT(DISTINCT cohorts.user\_id) AS cohort\_size,

COUNT(DISTINCT app\_events.user\_id) AS retained\_users

FROM `cohorts`

LEFT JOIN `appdataanalysis`.`app\_data`.`app\_events` AS app\_events

ON cohorts.user\_id = app\_events.user\_id

AND DATE(app\_events.event\_date) = DATE\_ADD(cohorts.cohort\_date, INTERVAL 1 DAY)

GROUP BY cohorts.cohort\_date

)

SELECT

cohort\_retention.cohort\_date,

cohort\_retention.cohort\_size,

cohort\_retention.retained\_users,

(cohort\_retention.retained\_users / cohort\_retention.cohort\_size) \* 100 AS retention\_rate

FROM cohort\_retention;

-- ● Analyse average session duration per feature and user location.

SELECT feature\_name, location,

ROUND(AVG(session\_duration),2) AS avg\_session\_duration

FROM `appdataanalysis`.`app\_data`.`app\_events`

LEFT JOIN `appdataanalysis`.`app\_data`.`user\_data`

ON app\_events.user\_id = user\_data.user\_id

GROUP BY feature\_name, location;