-- Step 1: Account metrics (basic metrics by grouping dimensions)

with account\_metrics AS (

SELECT

date, -- account creation date

sp.country,

a.send\_interval,

a.is\_verified,

a.is\_unsubscribed,

COUNT(DISTINCT a.id) AS account\_cnt -- number of created accounts

FROM `DA.account` a

JOIN `DA.account\_session` acs ON acs.account\_id = a.id

JOIN `DA.session\_params` sp ON sp.ga\_session\_id = acs.ga\_session\_id

JOIN `DA.session` s ON s.ga\_session\_id = acs.ga\_session\_id

GROUP BY date, sp.country, a.send\_interval, a.is\_verified, a.is\_unsubscribed

),

-- Step 2: Email metrics (basic metrics by grouping dimensions)

email\_metrics AS (

SELECT

DATE\_ADD(date, INTERVAL es.sent\_date DAY) AS date, -- email sent date = account creation date + sent offset

sp.country,

a.send\_interval,

a.is\_verified,

a.is\_unsubscribed,

COUNT(DISTINCT es.id\_message) AS sent\_msg, -- emails sent

COUNT(DISTINCT eo.id\_message) AS open\_msg, -- emails opened

COUNT(DISTINCT ev.id\_message) AS visit\_msg -- email link visits

FROM `DA.account\_session` acs

JOIN `DA.session` s ON s.ga\_session\_id = acs.ga\_session\_id

JOIN `DA.account` a ON a.id = acs.account\_id

JOIN `DA.session\_params` sp ON sp.ga\_session\_id = acs.ga\_session\_id

JOIN `DA.email\_sent` es ON es.id\_account = acs.account\_id

LEFT JOIN `DA.email\_open` eo ON eo.id\_message = es.id\_message

LEFT JOIN `DA.email\_visit` ev ON ev.id\_message = es.id\_message

GROUP BY date, sp.country, a.send\_interval, a.is\_verified, a.is\_unsubscribed

),

-- Step 3: Union both account and email metrics into one dataset

union\_all AS (

SELECT

date,

country,

send\_interval,

is\_verified,

is\_unsubscribed,

account\_cnt,

0 AS sent\_msg,

0 AS open\_msg,

0 AS visit\_msg

FROM account\_metrics

UNION ALL

SELECT

date,

country,

send\_interval,

is\_verified,

is\_unsubscribed,

0 AS account\_cnt,

sent\_msg,

open\_msg,

visit\_msg

FROM email\_metrics

),

-- Step 4: Aggregate totals by grouping fields (dimensions)

sum\_all AS (

SELECT

date,

country,

send\_interval,

is\_verified,

is\_unsubscribed,

SUM(account\_cnt) AS account\_cnt,

SUM(sent\_msg) AS sent\_msg,

SUM(open\_msg) AS open\_msg,

SUM(visit\_msg) AS visit\_msg

FROM union\_all

GROUP BY date, country, send\_interval, is\_verified, is\_unsubscribed

),

-- Step 5: Country-level totals for accounts

acc\_totals AS (

SELECT

country,

SUM(account\_cnt) AS total\_country\_account\_cnt

FROM sum\_all

GROUP BY country

),

-- Step 6: Country-level totals for emails

email\_totals AS (

SELECT

country,

SUM(sent\_msg) AS total\_country\_sent\_cnt

FROM sum\_all

GROUP BY country

),

-- Step 7: Country rankings

acc\_rank AS (

SELECT

country,

total\_country\_account\_cnt,

DENSE\_RANK() OVER (ORDER BY total\_country\_account\_cnt DESC) AS rank\_total\_country\_account\_cnt

FROM acc\_totals

),

email\_rank AS (

SELECT

country,

total\_country\_sent\_cnt,

DENSE\_RANK() OVER (ORDER BY total\_country\_sent\_cnt DESC) AS rank\_total\_country\_sent\_cnt

FROM email\_totals

)

-- Step 8: Final output

SELECT

sa.\*,

act.total\_country\_account\_cnt,

et.total\_country\_sent\_cnt,

ar.rank\_total\_country\_account\_cnt,

er.rank\_total\_country\_sent\_cnt

FROM sum\_all sa

JOIN acc\_rank ar ON sa.country = ar.country

JOIN email\_rank er ON sa.country = er.country

JOIN acc\_totals act ON sa.country = act.country

JOIN email\_totals et ON sa.country = et.country

WHERE ar.rank\_total\_country\_account\_cnt <= 10

OR er.rank\_total\_country\_sent\_cnt <= 10;

