--

-- Return all active users who signed up before Jan 1, 2024.

--

SELECT \* FROM USERS WHERE signup\_date < '2024-01-01';

--

-- List distinct cities from which users signed up using a free trial.

--

SELECT DISTINCT u.city

FROM USERS u

JOIN SUBSCRIPTIONS s ON s.user\_id = u.user\_id

WHERE s.subscription\_type = 'trial';

--

-- Find all users who signed up in the same month as the campaign named 'Summer Sale'.

--

SELECT u.\* FROM USERS u

WHERE MONTH(u.signup\_date) = (

SELECT MONTH(c.start\_date)

FROM CAMPAIGN c

WHERE c.campaign\_name = 'Summer Sale'

);

--

-- Retrieve the top 10 most recently signed-up users.

--

SELECT u.\* FROM USERS u

ORDER BY u.signup\_date DESC

LIMIT 10;

--

-- Join users and notifications to find all users who received a push notification in March 2024.

--

SELECT DISTINCT u.\* FROM USERS u

INNER JOIN NOTIFICATIONS n ON n.user\_id = u.user\_id

INNER JOIN CAMPAIGN c ON c.campaign\_id = n.campaign\_id

WHERE MONTH(n.sent\_at) = 3 AND YEAR(n.sent\_at) = 2024

AND c.campaign\_type = 'push';

--

-- Which users received more than one campaign in April 2024?

--

SELECT u.\*

FROM USERS u

INNER JOIN NOTIFICATIONS n ON n.user\_id = u.user\_id

INNER JOIN CAMPAIGN c ON c.campaign\_id = n.campaign\_id

WHERE MONTH(n.sent\_at) = 4 AND YEAR(n.sent\_at) = 2024

GROUP BY u.user\_id

HAVING COUNT(DISTINCT n.campaign\_id);

--

-- Join users, notifications, and campaigns to find users who received both an email and a push campaign.

--

SELECT u.\*

FROM USERS u

INNER JOIN NOTIFICATIONS n ON n.user\_id = u.user\_id

INNER JOIN CAMPAIGN c ON c.campaign\_id = n.campaign\_id

WHERE c.campaign\_type IN ('email', 'push')

GROUP BY u.user\_id

HAVING COUNT(DISTINCT c.campaign\_type) = 2;

--

-- Find all users who were notified but never opened any notification.

--

SELECT u.\*

FROM USERS u

INNER JOIN NOTIFICATIONS n ON n.user\_id = u.user\_id

WHERE n.opened\_at IS NULL;

--

-- Count the number of users per campaign who opened their notification.

--

SELECT c.campaign\_id, COUNT(DISTINCT n.user\_id) AS user\_count

FROM CAMPAIGN c

INNER JOIN NOTIFICATIONS n ON c.campaign\_id = n.campaign\_id

WHERE n.opened\_at IS NOT NULL

GROUP BY c.campaign\_id;

--

-- What is the average number of notifications per user?

--

SELECT AVG(notification\_count) AS average\_notifications\_per\_user

FROM (

SELECT n.user\_id, COUNT(n.notification\_id) AS notification\_count

FROM NOTIFICATIONS n

GROUP BY n.user\_id

) AS user\_notifications;

--

-- Calculate the percentage of users who opened at least one notification.

--

SELECT (COUNT(DISTINCT n.user\_id) \* 100.0 / COUNT(DISTINCT u.user\_id)) AS percentage\_users\_opened

FROM USERS u

LEFT JOIN NOTIFICATIONS n ON u.user\_id = n.user\_id AND n.opened\_at IS NOT NULL;

--

-- Count the number of cities with more than 50 active users.

--

SELECT COUNT(DISTINCT u.city) AS city\_count, c.city

FROM USERS u

LEFT JOIN SUBSCRIPTION s ON s.user\_id = u.user\_id

WHERE s.status = 'active'

GROUP BY c.city

HAVING city\_count > 50;

--

-- Use a CTE to return each user's total number of notifications received and sort by the highest count.

--

WITH UserNotificationCounts AS (

SELECT

n.user\_id,

COUNT(n.notification\_id) AS total\_notifications

FROM

NOTIFICATIONS n

GROUP BY

n.user\_id

)

SELECT

u.user\_id,

unc.total\_notifications

FROM

USERS u

LEFT JOIN

UserNotificationCounts unc ON u.user\_id = unc.user\_id

ORDER BY

total\_notifications DESC;

--

-- Use a window function to get each user's most recent notification date.

--

WITH RankedNotifications AS (

SELECT

n.user\_id,

n.sent\_at,

ROW\_NUMBER() OVER (PARTITION BY n.user\_id ORDER BY n.sent\_at DESC) AS rn

FROM

NOTIFICATIONS n

)

SELECT

u.user\_id,

rn.sent\_at AS most\_recent\_notification\_date

FROM

USERS u

LEFT JOIN

RankedNotifications rn ON u.user\_id = rn.user\_id

WHERE

rn.rn = 1;