**Case Study 2: Investigating Metric Spike**

* **users**: Contains one row per user, with descriptive information about that user’s account.
* **events**: Contains one row per event, where an event is an action that a user has taken (e.g., login, messaging, search).
* **email\_events**: Contains events specific to the sending of emails.

**A.Weekly User Engagement:**

Measuring the activeness of users on a weekly basis.

SELECT

DATE\_TRUNC('week', created\_at) AS week\_start\_date,

COUNT(DISTINCT user\_id) AS active\_users\_count

FROM

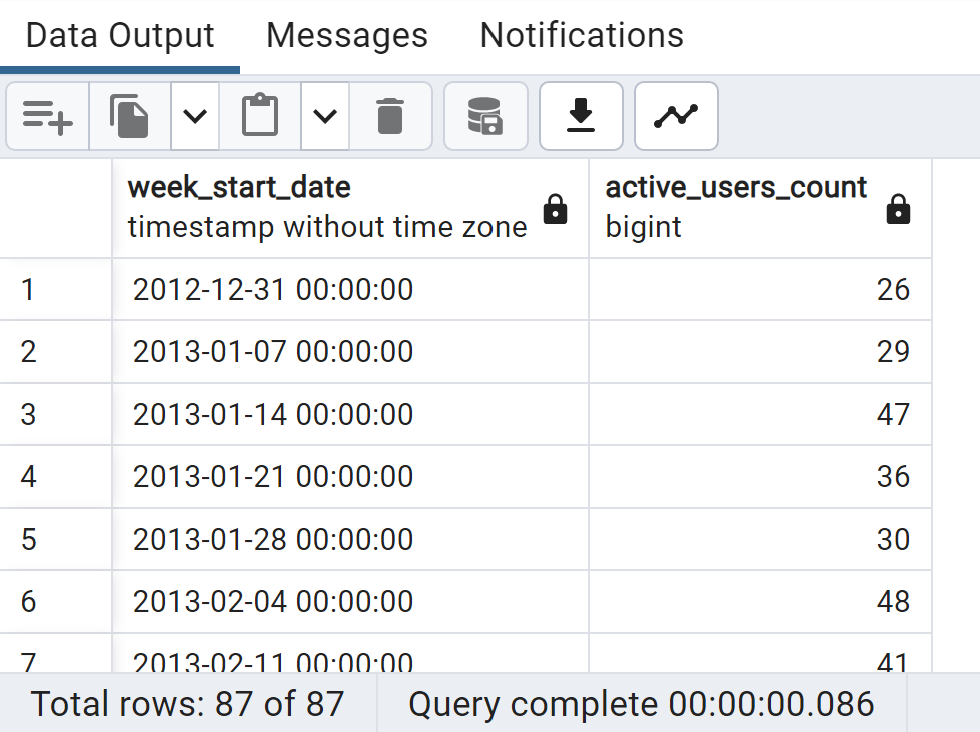
users

GROUP BY

week\_start\_date

ORDER BY

week\_start\_date;



**B.User Growth Analysis:**

Analyzing the growth of users over time for a product.

SELECT

DATE\_TRUNC('month', created\_at) AS month\_start\_date,

COUNT(DISTINCT user\_id) AS total\_users

FROM

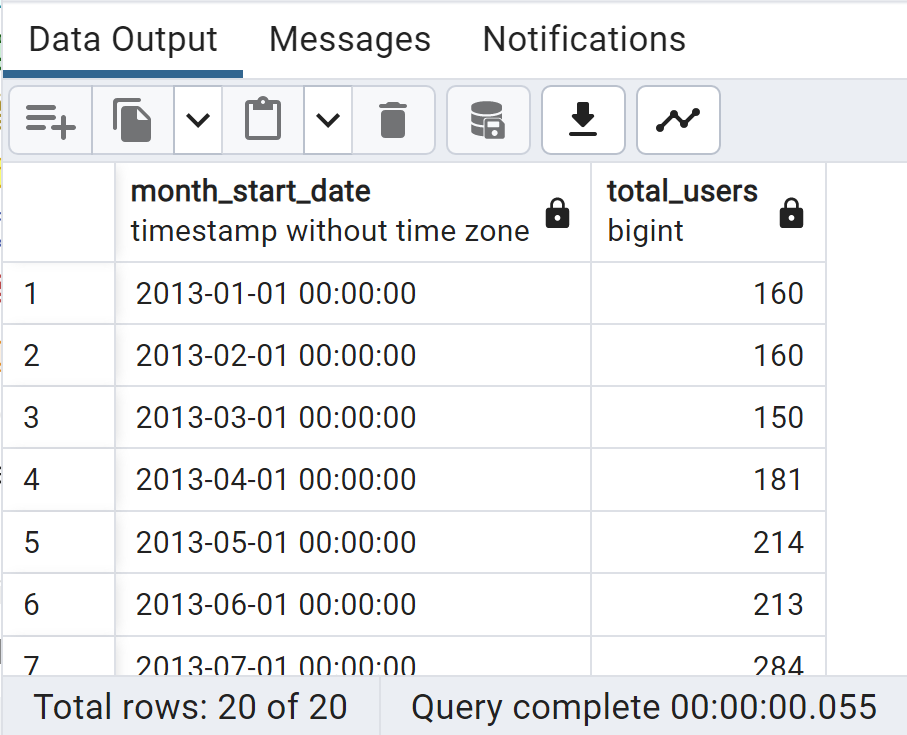
users

GROUP BY

month\_start\_date

ORDER BY

month\_start\_date;



**C.Weekly Retention Analysis:**

Analyzing the retention of users on a weekly basis after signing up for a product.

WITH user\_signups AS (

SELECT

user\_id,

DATE\_TRUNC('week', created\_at) AS signup\_week

FROM

users

),

user\_activity AS (

SELECT

user\_id,

DATE\_TRUNC('week', occurred\_at) AS activity\_week

FROM

events

)

SELECT

us.signup\_week AS cohort\_week,

ua.activity\_week AS retention\_week,

COUNT(DISTINCT ua.user\_id) AS retained\_users

FROM

user\_signups us

LEFT JOIN

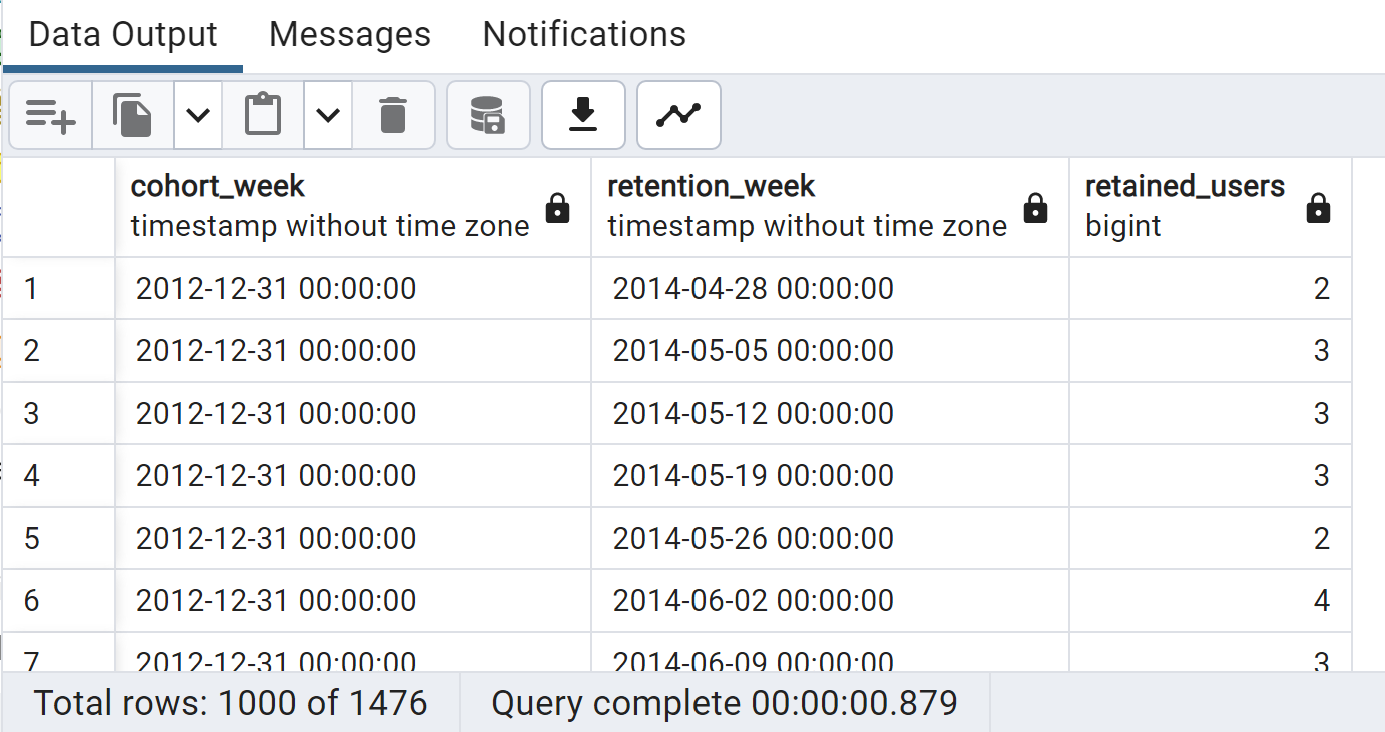
user\_activity ua ON us.user\_id = ua.user\_id AND ua.activity\_week >= us.signup\_week

GROUP BY

us.signup\_week, ua.activity\_week

ORDER BY

us.signup\_week, ua.activity\_week;



**D.Weekly Engagement Per Device:**

Measuring the activeness of users on a weekly basis per device.

SELECT

DATE\_TRUNC('week', e.occurred\_at) AS week\_start\_date,

e.device,

COUNT(DISTINCT e.user\_id) AS active\_users\_count

FROM

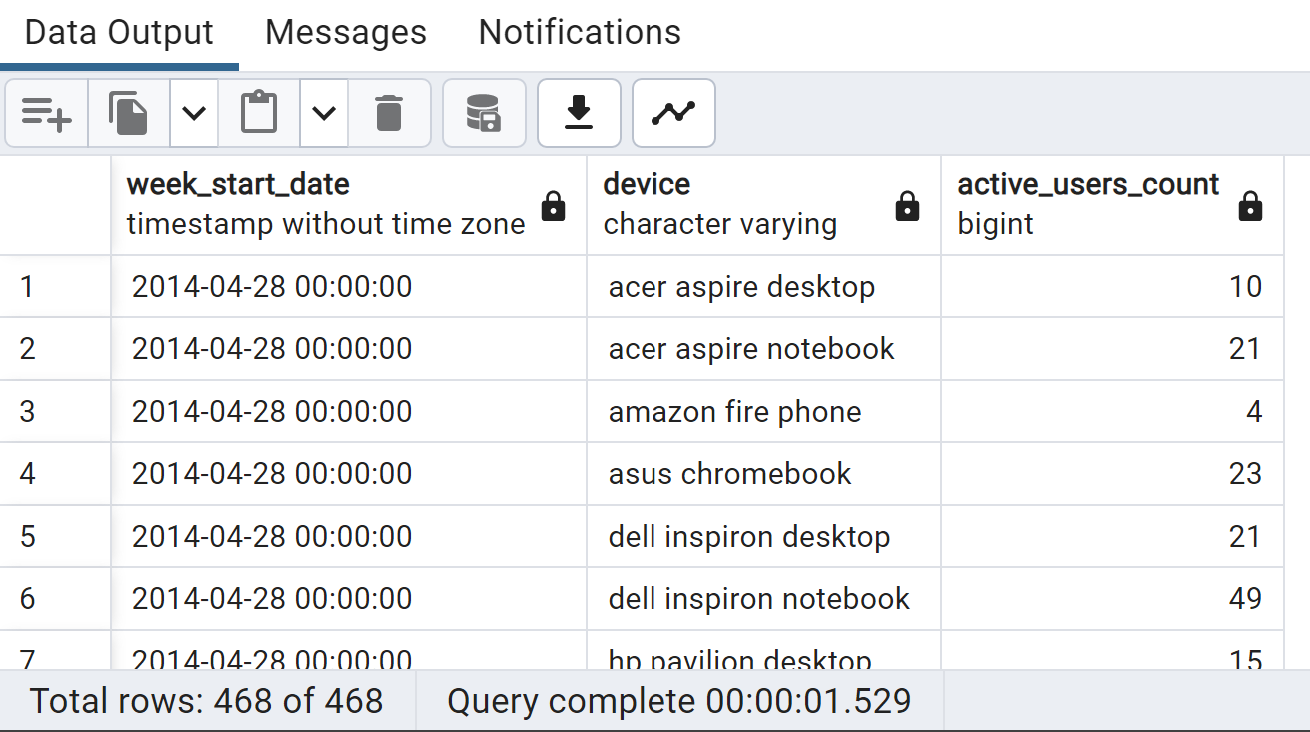
events e

GROUP BY

week\_start\_date, e.device

ORDER BY

week\_start\_date, e.device;



**E.Email Engagement Analysis:**

Analyzing how users are engaging with the email service.

SELECT

action,

COUNT(DISTINCT user\_id) AS unique\_users\_count,

COUNT(\*) AS total\_actions\_count

FROM

email\_events

GROUP BY

action

ORDER BY

action;

