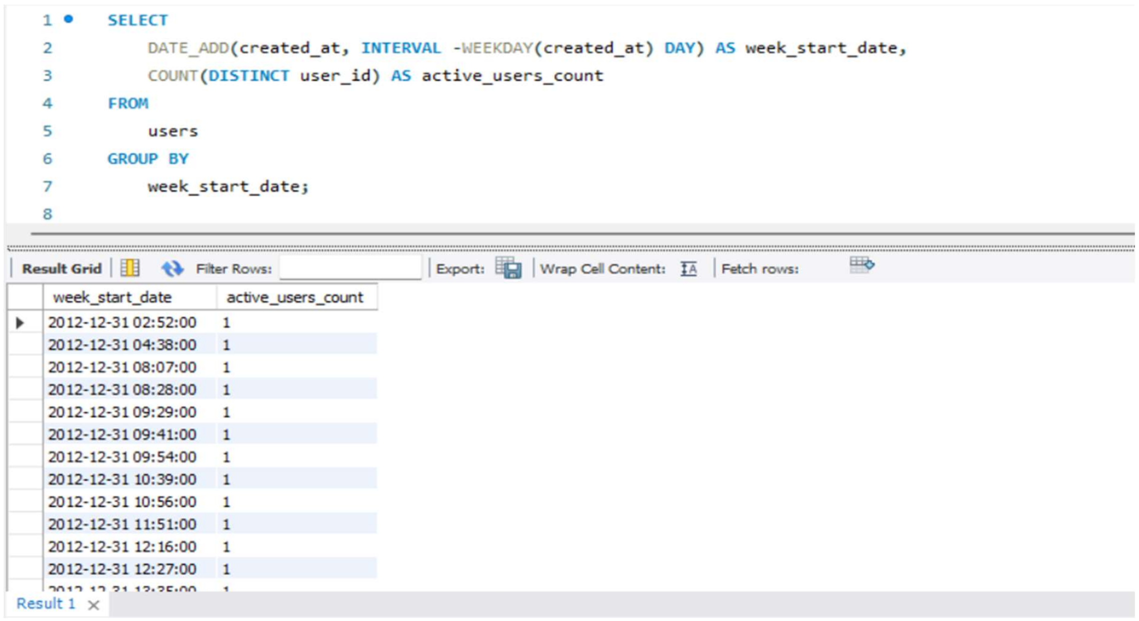
Case Study 2: Investigating Metric Spike

Weekly User Engagement:

Objective: Measure the activeness of users on a weekly basis.

Your Task: Write an SQL query to calculate the weekly user engagement.



```sql
SELECT
    DATE_ADD(created_at, INTERVAL -WEEKDAY(created_at) DAY) AS week_start_date,
    COUNT(DISTINCT user_id) AS active_users_count
FROM
    users
GROUP BY
    week_start_date;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

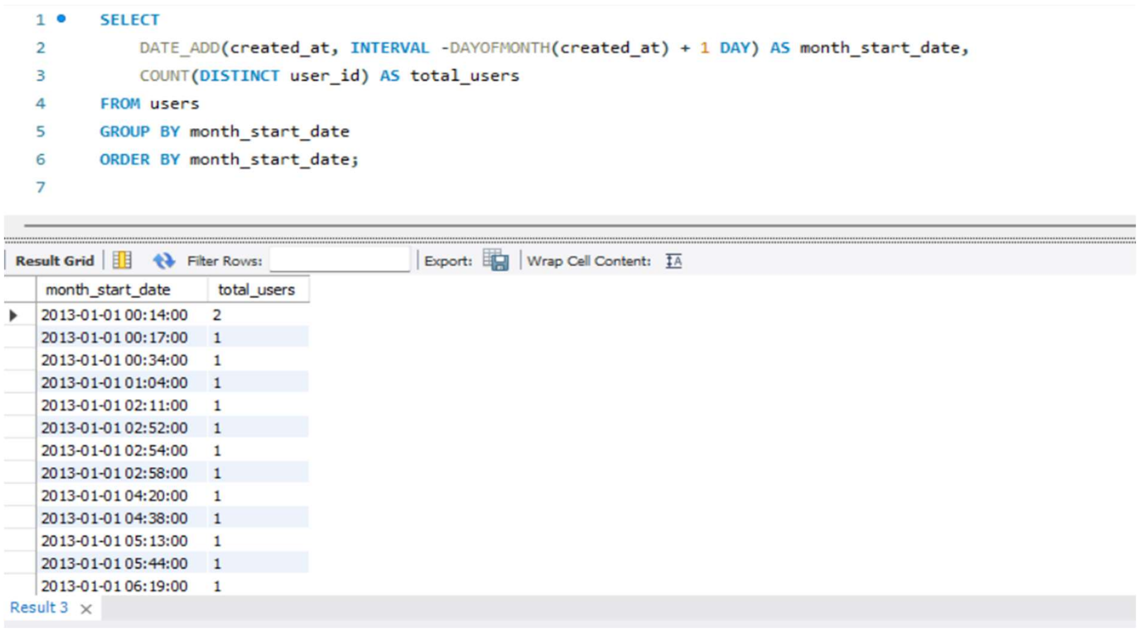
```
week_start_date      active_users_count
2012-12-31 02:52:00  1
2012-12-31 04:38:00  1
2012-12-31 08:07:00  1
2012-12-31 08:28:00  1
2012-12-31 09:29:00  1
2012-12-31 09:41:00  1
2012-12-31 09:54:00  1
2012-12-31 10:39:00  1
2012-12-31 10:56:00  1
2012-12-31 11:51:00  1
2012-12-31 12:16:00  1
2012-12-31 12:27:00  1
```

Result 1

User Growth Analysis:

Objective: Analyze the growth of users over time for a product.

Your Task: Write an SQL query to calculate the user growth for the product.



```sql
SELECT
    DATE_ADD(created_at, INTERVAL -DAYOFMONTH(created_at) + 1 DAY) AS month_start_date,
    COUNT(DISTINCT user_id) AS total_users
FROM users
GROUP BY month_start_date
ORDER BY month_start_date;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

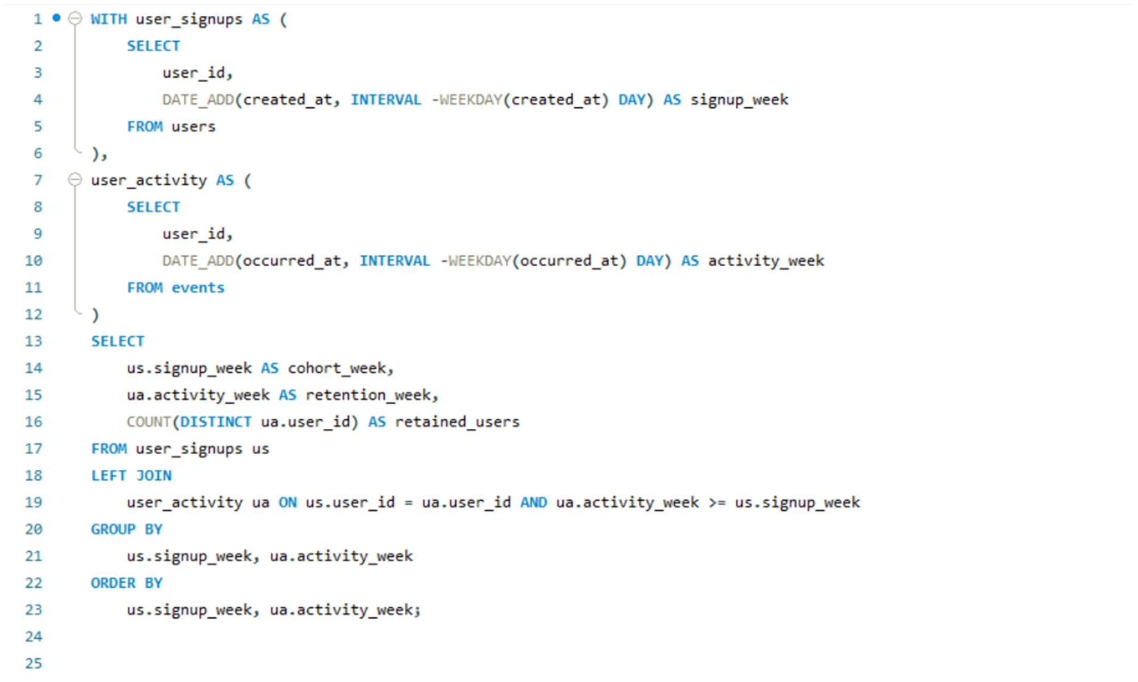
```
month_start_date     total_users
2013-01-01 00:14:00  2
2013-01-01 00:17:00  1
2013-01-01 00:34:00  1
2013-01-01 01:04:00  1
2013-01-01 02:11:00  1
2013-01-01 02:52:00  1
2013-01-01 02:54:00  1
2013-01-01 02:58:00  1
2013-01-01 04:20:00  1
2013-01-01 04:38:00  1
2013-01-01 05:13:00  1
2013-01-01 05:44:00  1
2013-01-01 06:19:00  1
```

Result 3

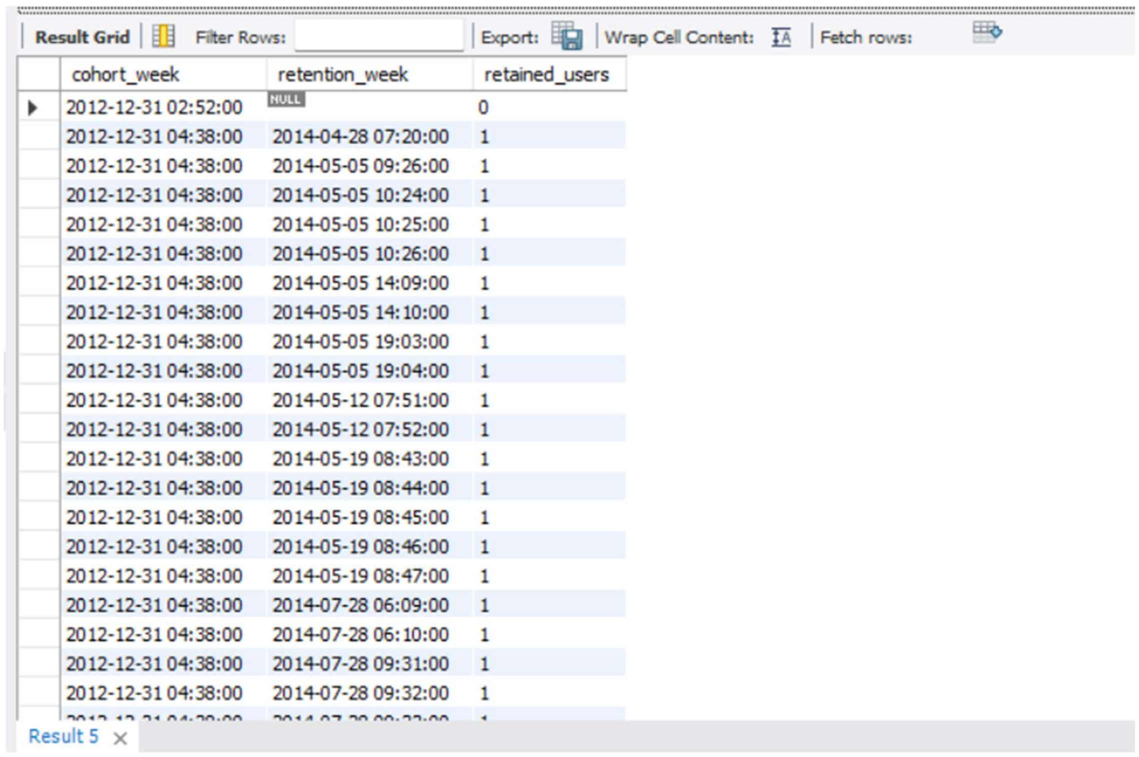
Weekly Retention Analysis:

Objective: Analyze the retention of users on a weekly basis after signing up for a product.

Your Task: Write an SQL query to calculate the weekly retention of users based on their sign-up cohort.



```sql
WITH user_signups AS (
    SELECT
        user_id,
        DATE_ADD(created_at, INTERVAL -WEEKDAY(created_at) DAY) AS signup_week
    FROM users
),
user_activity AS (
    SELECT
        user_id,
        DATE_ADD(occurred_at, INTERVAL -WEEKDAY(occurred_at) DAY) 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;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

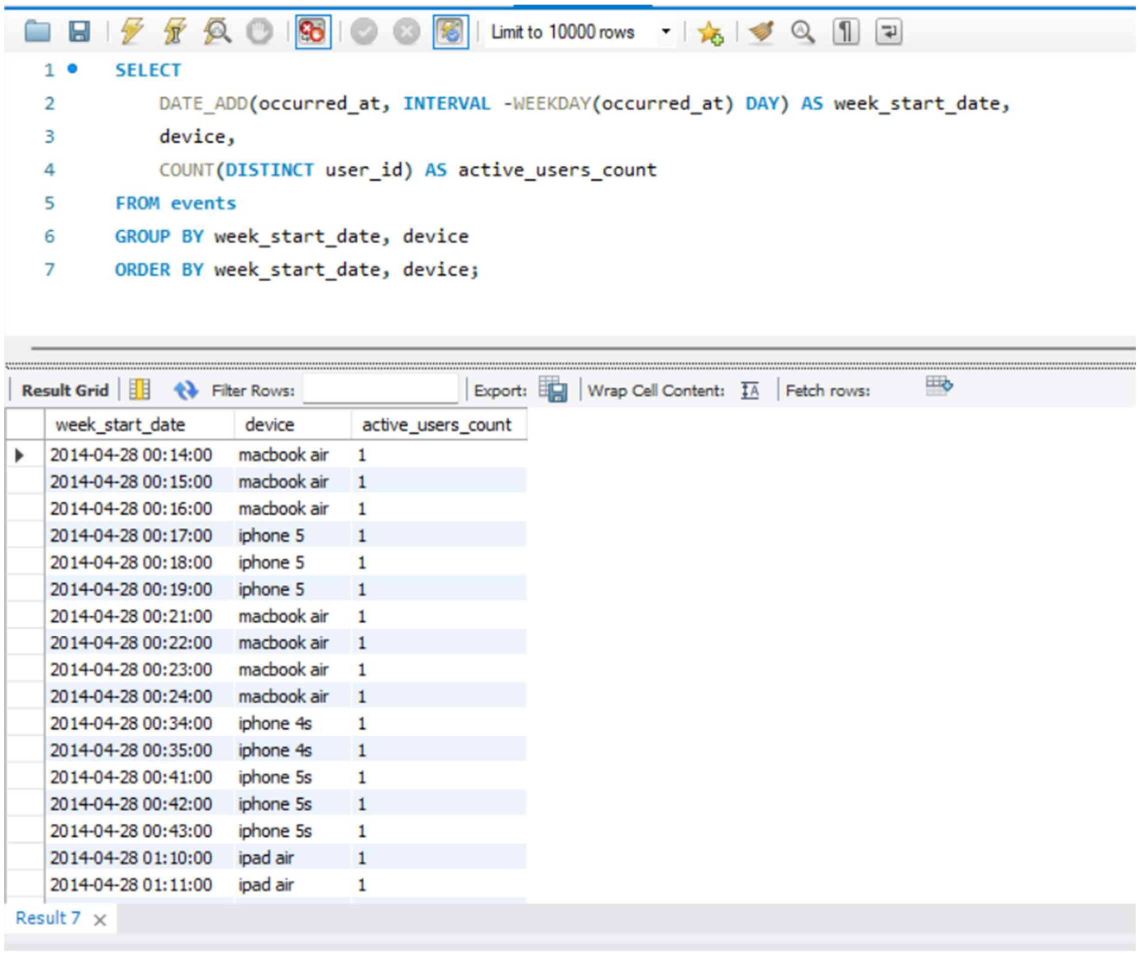
```
cohort_week          retention_week        retained_users
2012-12-31 02:52:00  NULL                  0
2012-12-31 04:38:00  2014-04-28 07:20:00   1
2012-12-31 04:38:00  2014-05-05 09:26:00   1
2012-12-31 04:38:00  2014-05-05 10:24:00   1
2012-12-31 04:38:00  2014-05-05 10:25:00   1
2012-12-31 04:38:00  2014-05-05 10:26:00   1
2012-12-31 04:38:00  2014-05-05 14:09:00   1
2012-12-31 04:38:00  2014-05-05 14:10:00   1
2012-12-31 04:38:00  2014-05-05 19:03:00   1
2012-12-31 04:38:00  2014-05-05 19:04:00   1
2012-12-31 04:38:00  2014-05-12 07:51:00   1
2012-12-31 04:38:00  2014-05-12 07:52:00   1
2012-12-31 04:38:00  2014-05-19 08:43:00   1
2012-12-31 04:38:00  2014-05-19 08:44:00   1
2012-12-31 04:38:00  2014-05-19 08:45:00   1
2012-12-31 04:38:00  2014-05-19 08:46:00   1
2012-12-31 04:38:00  2014-05-19 08:47:00   1
2012-12-31 04:38:00  2014-07-28 06:09:00   1
2012-12-31 04:38:00  2014-07-28 06:10:00   1
2012-12-31 04:38:00  2014-07-28 09:31:00   1
2012-12-31 04:38:00  2014-07-28 09:32:00   1
```

Result 5 ×

Weekly Engagement Per Device:

Objective: Measure the activeness of users on a weekly basis per device.

Your Task: Write an SQL query to calculate the weekly engagement per device.



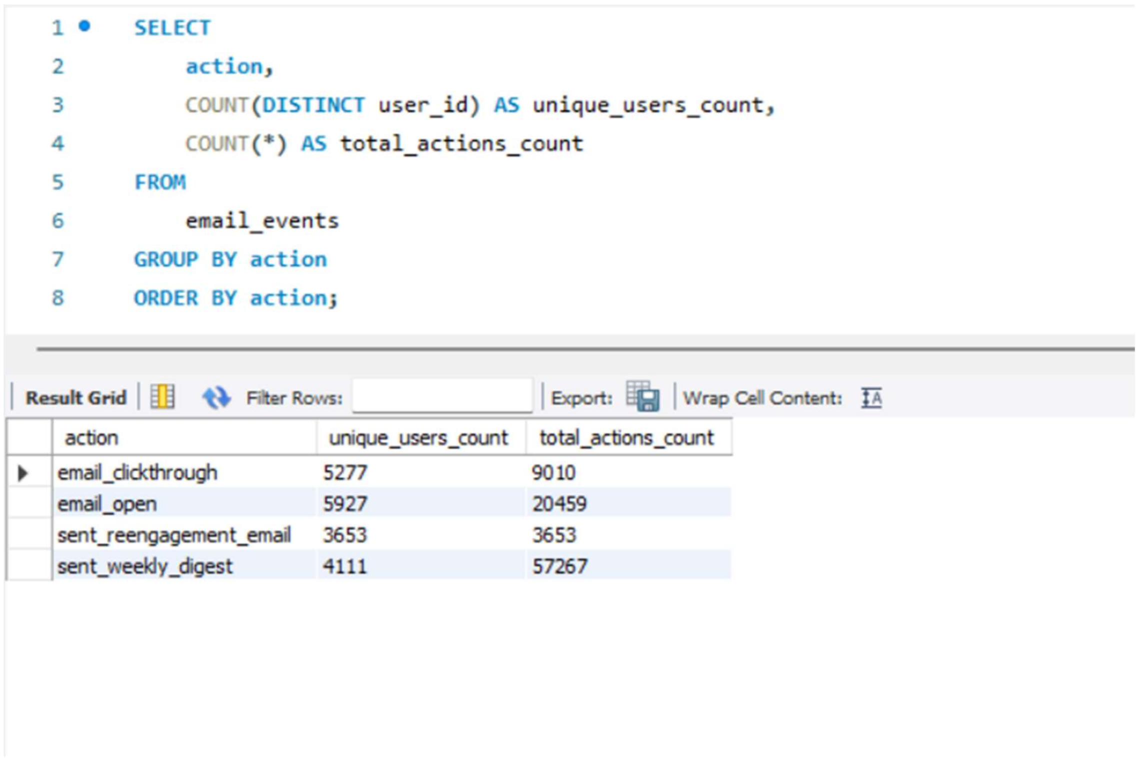
```sql
SELECT
    DATE_ADD(occurred_at, INTERVAL -WEEKDAY(occurred_at) DAY) AS week_start_date,
    device,
    COUNT(DISTINCT user_id) AS active_users_count
FROM events
GROUP BY week_start_date, device
ORDER BY week_start_date, device;
```

week_start_date	device	active_users_count
2014-04-28 00:14:00	macbook air	1
2014-04-28 00:15:00	macbook air	1
2014-04-28 00:16:00	macbook air	1
2014-04-28 00:17:00	iphone 5	1
2014-04-28 00:18:00	iphone 5	1
2014-04-28 00:19:00	iphone 5	1
2014-04-28 00:21:00	macbook air	1
2014-04-28 00:22:00	macbook air	1
2014-04-28 00:23:00	macbook air	1
2014-04-28 00:24:00	macbook air	1
2014-04-28 00:34:00	iphone 4s	1
2014-04-28 00:35:00	iphone 4s	1
2014-04-28 00:41:00	iphone 5s	1
2014-04-28 00:42:00	iphone 5s	1
2014-04-28 00:43:00	iphone 5s	1
2014-04-28 01:10:00	ipad air	1
2014-04-28 01:11:00	ipad air	1

Email Engagement Analysis:

Objective: Analyze how users are engaging with the email service.

Your Task: Write an SQL query to calculate the email engagement metrics.



```sql
SELECT
    action,
    COUNT(DISTINCT user_id) AS unique_users_count,
    COUNT(*) AS total_actions_count
FROM
    email_events
GROUP BY action
ORDER BY action;
```

action	unique_users_count	total_actions_count
email_clickthrough	5277	9010
email_open	5927	20459
sent_reengagement_email	3653	3653
sent_weekly_digest	4111	57267