

SQL Queries: Investigating Metric Spike

Weekly User Engagement

Query 1: Measure the activeness of users on a weekly basis.

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

Query:

```
SELECT
    u.user_id,
    u.language,
    u.state,
    YEAR(e.occurred_at) AS year,
    WEEK(e.occurred_at) AS week_number,
    COUNT(e.event_type) AS event_count
FROM events e
JOIN users u ON e.user_id = u.user_id
GROUP BY u.user_id, u.language, u.state, YEAR(e.occurred_at), WEEK(e.occurred_at)
ORDER BY year DESC, week_number DESC, event_count DESC;
```

Output:

user_id	language	state	year	week_number	event_count
18885	japanese	active	2014	35	22
19044	english	active	2014	35	19
18048	english	active	2014	35	18
19040	spanish	active	2014	35	16
2789	spanish	active	2014	35	16
3633	russian	active	2014	35	16
3142	english	active	2014	35	15
18774	korean	active	2014	35	15
8950	spanish	active	2014	35	15
19047	english	active	2014	35	14
18795	english	active	2014	35	13
18653	english	active	2014	35	12
18781	english	active	2014	35	12
11928	german	active	2014	35	12
7677	english	active	2014	35	12
7839	english	active	2014	35	12
17971	portugese	active	2014	35	11
18746	english	active	2014	35	11
18754	japanese	active	2014	35	11
18773	japanese	active	2014	35	11
18797	english	active	2014	35	11
18806	english	active	2014	35	11
18741	french	active	2014	35	10
18761	indian	active	2014	35	10

User Growth Analysis

Query 2: Analyze the growth of users over time for a product.

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

Query:

```
WITH user_growth AS (  
    SELECT  
        YEAR(created_at) AS year,  
        WEEK(created_at) AS week_number,  
        COUNT(user_id) AS new_users  
    FROM users  
    GROUP BY year, week_number  
)  
SELECT  
    year, week_number,  
    new_users,  
    SUM(new_users) OVER (ORDER BY year, week_number) AS cumulative_users  
FROM user_growth  
ORDER BY year, week_number;
```

Output:

year	week_number	new_users	cumulative_users
2013	0	23	23
2013	1	30	53
2013	2	48	101
2013	3	36	137
2013	4	30	167
2013	5	48	215
2013	6	38	253
2013	7	42	295
2013	8	34	329
2013	9	43	372
2013	10	32	404
2013	11	31	435
2013	12	33	468
2013	13	39	507
2013	14	35	542
2013	15	43	585
2013	16	46	631
2013	17	49	680
2013	18	44	724
2013	19	57	781
2013	20	39	820
2013	21	49	869
2013	22	54	923

Weekly Retention Analysis

Query 3: Analyze the retention of users on a weekly basis after signing up for a product.

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

Query:

```
WITH user_signup AS (  
    -- Get signup week for each user  
    SELECT  
        user_id,  
        YEAR(created_at) AS signup_year,  
        WEEK(created_at) AS signup_week  
    FROM users  
),  
user_activity AS (  
    -- Get activity week for each user  
    SELECT  
        e.user_id,  
        YEAR(e.occurred_at) AS activity_year,  
        WEEK(e.occurred_at) AS activity_week  
    FROM events e  
),  
retention AS (  
    -- Join user signups with their activity weeks  
    SELECT  
        u.signup_year,  
        u.signup_week,  
        a.activity_year,  
        a.activity_week,  
        a.user_id,  
        (a.activity_year - u.signup_year) * 52 + (a.activity_week - u.signup_week) AS week_number  
    FROM user_signup u  
    JOIN user_activity a ON u.user_id = a.user_id  
)
```

```
)  
SELECT  
    signup_year,  
    signup_week,  
    week_number,  
    COUNT(DISTINCT user_id) AS retained_users  
FROM retention  
WHERE week_number >= 0 -- Only count retention for users after signup  
GROUP BY signup_year, signup_week, week_number  
ORDER BY signup_year DESC, signup_week DESC, week_number ASC;
```

Output:

signup_year	signup_week	week_number	retained_users
2014	35	0	18
2014	34	0	259
2014	34	1	43
2014	33	0	261
2014	33	1	187
2014	33	2	8
2014	32	0	245
2014	32	1	174
2014	32	2	81
2014	32	3	8
2014	31	0	193
2014	31	1	136
2014	31	2	69
2014	31	3	52
2014	31	4	1
2014	30	0	238
2014	30	1	171
2014	30	2	94
2014	30	3	65
2014	30	4	43
2014	30	5	3
2014	29	0	221
2014	29	1	160
2014	29	2	81
2014	29	3	53
2014	29	4	39
2014	29	5	33
2014	29	6	1
2014	28	0	215

Weekly Engagement Per Device

Query 4: Measure the activeness of users on a weekly basis per device.

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

Query:

```
SELECT
    YEAR(occurred_at) AS year,
    WEEK(occurred_at) AS week_number,
    device,
    COUNT(event_type) AS event_count
FROM events
GROUP BY year, week_number, device
ORDER BY year DESC, week_number DESC, event_count DESC;
```

Output:

year	week_number	device	event_count
2014	35	lenovo thinkpad	126
2014	35	macbook pro	124
2014	35	dell inspiron notebook	69
2014	35	macbook air	66
2014	35	iphone 4s	58
2014	35	asus chromebook	38
2014	35	nexus 5	35
2014	35	kindle fire	32
2014	35	windows surface	31
2014	35	acer aspire notebook	29
2014	35	samsung galaxy s4	29
2014	35	mac mini	25
2014	35	ipad mini	22
2014	35	iphone 5s	22
2014	35	htc one	19
2014	35	nexus 7	17
2014	35	nexus 10	15
2014	35	hp pavilion desktop	10
2014	35	iphone 5	9
2014	35	nokia lumia 635	8
2014	35	acer aspire desktop	7
2014	35	samsung galaxy note	6
2014	35	dell inspiron desktop	5
2014	34	macbook pro	3182
2014	34	lenovo thinkpad	1945
2014	34	macbook air	1472
2014	34	dell inspiron notebook	1100
2014	34	iphone 5	982
2014	34	samsung galaxy s4	896

Email Engagement Analysis

Query 5: Analyze how users are engaging with the email service.

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

Query:

```
SELECT
    DATE(occurred_at) AS engagement_date,
    action,
    COUNT(user_id) AS action_count,
    COUNT(DISTINCT user_id) AS unique_users
FROM email_events
GROUP BY engagement_date, action
ORDER BY engagement_date DESC, action;
```

Output:

engagement_date	action	action_count	unique_users
2014-08-31	email_clickthrough	38	38
2014-08-31	email_open	41	41
2014-08-31	sent_reengagement_email	48	48
2014-08-30	email_clickthrough	34	34
2014-08-30	email_open	38	38
2014-08-30	sent_reengagement_email	41	41
2014-08-29	email_clickthrough	110	110
2014-08-29	email_open	276	276
2014-08-29	sent_reengagement_email	52	52
2014-08-29	sent_weekly_digest	735	735
2014-08-28	email_clickthrough	57	57
2014-08-28	email_open	247	247
2014-08-28	sent_reengagement_email	11	11
2014-08-28	sent_weekly_digest	758	758
2014-08-27	email_clickthrough	56	56
2014-08-27	email_open	229	229
2014-08-27	sent_reengagement_email	12	12
2014-08-27	sent_weekly_digest	719	719
2014-08-26	email_clickthrough	84	84
2014-08-26	email_open	299	299
2014-08-26	sent_reengagement_email	50	50
2014-08-26	sent_weekly_digest	746	746
2014-08-25	email_clickthrough	114	114
2014-08-25	email_open	403	403
2014-08-25	sent_reengagement_email	49	49