
YAMMER WAU

분석 체크

데잇걸즈 부서 6팀

CONTENTS

1. 문제 파악
 2. 동료의 쿼리 분석 (+ 테이블 파악)
 3. 제시된 해결 방법에 대한 비판
 4. 이외에 추가 아이디어 등 공유할 점
 5. 과제를 수행하면서 얻은 것, 느낀 점
-

문제 파악

❖ Yammer

동료들과 소통하기 위한 소셜 네트워크입니다.

개인은 문서, 업데이트 및 아이디어를 그룹으로 게시하여 공유합니다.

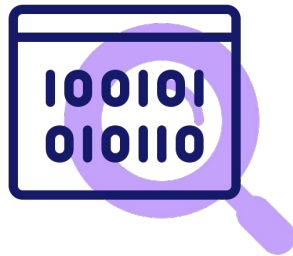
Yammer는 무제한으로 사용할 수 있지만 **Active Directory**와 같은 사용자 관리 시스템과의 통합을 포함하여 관리 제어에 액세스하려면 라이선스 요금을 지불해야 합니다.



사용자

1. 업무상 다양한 부서와 협업을 해야하는 직원들 :

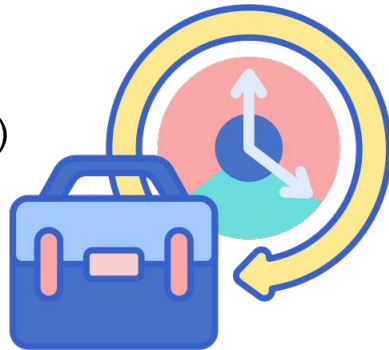
문서의 공유나 데이터의 공유가 활발히 진행되어야 합니다.



2. 데이터를 활용해야 하는 직무의 직원들

ex. 데이터 분석직무, 영업지원, 서비스기획, 인사팀(통과 교육을 제공)

3. 종합적으로 데이터 베이스를 관리할 수 있는 데이터 엔지니어 직원들



Yammer 서비스를 이용하는 사용자 참여 수가 감소한 원인을 분석

Weekly Active Users





문제상황 분석

“ 사용자 참여 감소 ” -> weekly 사용자 감소

→ **Weekly 사용자 정의**

해당 날짜(차트의 어느 지점)에 시작되는 주
中 하나 이상의 참여 이벤트를 기록한
사용자 수

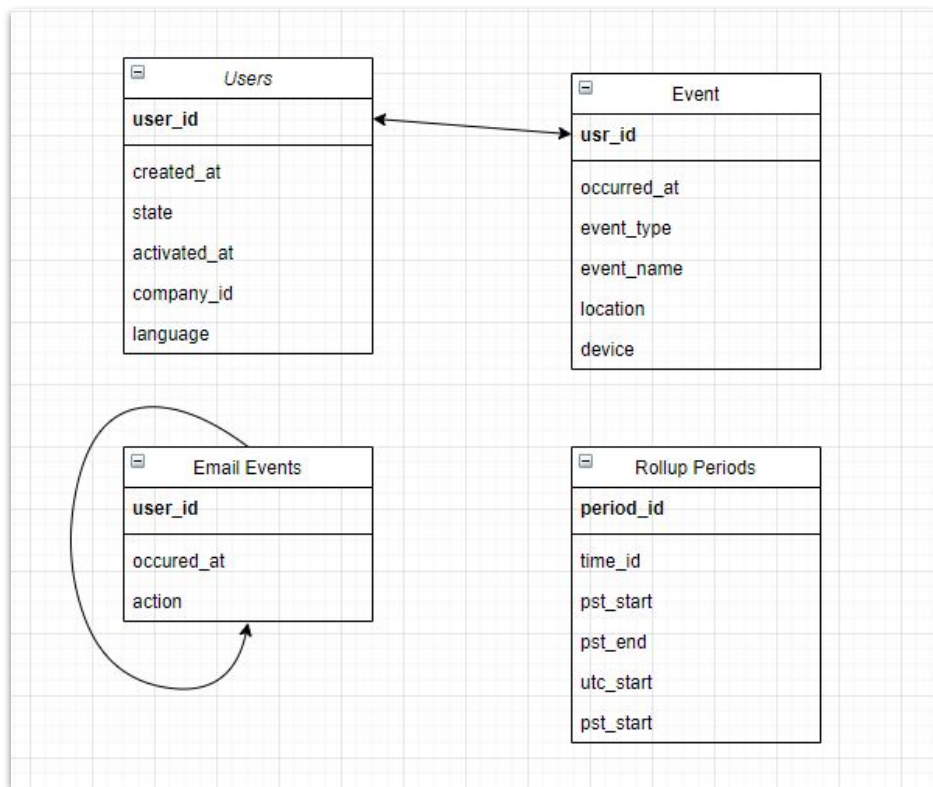
→ **참여 이벤트 정의**

Yammer에 로그인 한 유저

→ **제시한 5가지 질문**

- 1) 원래 가설에 대한 답변이 추가적인 질문으로 이어지는가?
- 2) 그렇다면 무엇이고 어떻게 test?
- 3) 데이터 만으로 답변할 수 없다면 어떻게 답변?
- 4) 참여 감소의 가장 유력한 원인은?
- 5) 원인이 있다면 어떻게 대응?

기존 TABLE



동료의 쿼리 분석_1번째 쿼리

```
SELECT DATE TRUNC('day',created_at) AS day,-- 1) 가입 날짜
COUNT(*) AS all_users,-- 2) 모든 사용자 수
COUNT(CASE WHEN activated_at IS NOT NULL THEN u.user_id ELSE NULL END) AS activated_users -- 3)
FROM tutorial.yammer users u
WHERE created_at >= '2014-06-01' -- 4) 6,7,8월만 확인
AND created_at < '2014-09-01'
GROUP BY 1
ORDER BY 1
```

DATE TRUNC : 시간, 주 또는 월 등 지정하는 시간 간격을 기준으로
타임스탬프 표현식

day 칼럼 vs 원래의 칼럼 차이

	day	created_at
1	2013-01-01 00:00:00	2013-01-01 20:59:39
2	2013-01-01 00:00:00	2013-01-01 13:07:46
3	2013-01-01 00:00:00	2013-01-01 10:59:05
4	2013-01-01 00:00:00	2013-01-01 18:40:36
5	2013-01-01 00:00:00	2013-01-01 14:37:51

3) 활성화 유저(activated_users)에 대한 정의:
activated_at (활성화 시점: yammer에 로그인 한
시점)에서 null 값이 없음

=> 즉, 로그인 한 유저!

동료의 쿼리 분석_2번째 쿼리_z테이블

```

SELECT DATE TRUNC('week', z.occurred_at) AS "week",
       AVG(z.age_at_event) AS "Average age during week",
       COUNT(DISTINCT CASE WHEN z.user_age > 70 THEN z.user_id ELSE NULL END) AS "10+ weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 70 AND z.user_age >= 63 THEN z.user_id ELSE NULL END) AS "9 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 63 AND z.user_age >= 56 THEN z.user_id ELSE NULL END) AS "8 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 56 AND z.user_age >= 49 THEN z.user_id ELSE NULL END) AS "7 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 49 AND z.user_age >= 42 THEN z.user_id ELSE NULL END) AS "6 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 42 AND z.user_age >= 35 THEN z.user_id ELSE NULL END) AS "5 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 35 AND z.user_age >= 28 THEN z.user_id ELSE NULL END) AS "4 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 28 AND z.user_age >= 21 THEN z.user_id ELSE NULL END) AS "3 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 21 AND z.user_age >= 14 THEN z.user_id ELSE NULL END) AS "2 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 14 AND z.user_age >= 7 THEN z.user_id ELSE NULL END) AS "1 week",
       COUNT(DISTINCT CASE WHEN z.user_age < 7 THEN z.user_id ELSE NULL END) AS "Less than a week"
FROM (
    SELECT e.occurred_at,
           u.user_id,
           DATE TRUNC('week', u.activated_at) AS activation_week,
           EXTRACT('day' FROM e.occurred_at - u.activated_at) AS age_at_event,
           EXTRACT('day' FROM '2014-09-01'::TIMESTAMP - u.activated_at) AS user_age
    FROM tutorial.yammer users u
    JOIN tutorial.yammer events e
      ON e.user_id = u.user_id
      AND e.event_type = 'engagement'
      AND e.event_name = 'login'
      AND e.occurred_at >= '2014-05-01'
      AND e.occurred_at < '2014-09-01'
    WHERE u.activated_at IS NOT NULL
) z

GROUP BY 1
ORDER BY 1
LIMIT 100
    
```

signup_flow 에 해당하는 event_name

event_type	
1	engagement
2	signup_flow

	event_type	event_name
1	signup_flow	complete_signup
2	signup_flow	create_user
3	signup_flow	enter_email
4	signup_flow	enter_info

	event_name
1	complete_signup
2	create_user
3	enter_email
4	enter_info
5	home_page
6	like_message
7	login
8	search_autocomplete
9	search_click_result_1
10	search_click_result_10
11	search_click_result_2
12	search_click_result_3
13	search_click_result_4
14	search_click_result_5
15	search_click_result_6
16	search_click_result_7
17	search_click_result_8
18	search_click_result_9
19	search_run
20	send_message
21	view_inbox

z 테이블

FROM 문	User 테이블과 events 테이블 left 조인 합치는 기준은 <u>user id</u> 1) AND: 이벤트 타입은 engagement 2) AND: engagement들 중에서 로그인만 가져오고 3)&4) AND: 가입날짜는 5,6,7,8월 가입한 사람들 만 가져온다.
WHERE 문	활성화 시점이 비어 있지 않은 것 -> 로그인 안 한 사람들은 제외
SELECT문	이벤트가 일어난 시점, 유저 아이디, 활성화 주, 이벤트가 일어난 날짜-활성화 시점, 9월 1일 이전까지의 활성화 시점 날짜(<u>user age</u>)를 추출했다.

```
COUNT(DISTINCT CASE WHEN z.user_age < 14 AND z.user_age >= 7 THEN z.user_id ELSE NULL END) AS "1 week",
COUNT(DISTINCT CASE WHEN z.user_age < 7 THEN z.user_id ELSE NULL END) AS "Less than a week"
FROM (
  SELECT e.occurred_at,
         u.user_id,
         DATE_TRUNC('week', u.activated_at) AS activation_week,
         EXTRACT('day' FROM e.occurred_at - u.activated_at) AS age_at_event,
         EXTRACT('day' FROM '2014-09-01'::TIMESTAMP - u.activated_at) AS user_age
  FROM tutorial.yammer users u
  JOIN tutorial.yammer events e
    ON e.user_id = u.user_id
   AND e.event_type = 'engagement'
   AND e.event_name = 'login'
   AND e.occurred_at >= '2014-05-01'
   AND e.occurred_at < '2014-09-01'
 WHERE u.activated_at IS NOT NULL
) z

GROUP BY 1
ORDER BY 1
LIMIT 100
```

z 테이블

	occurred_at	user_id	activation_week	age_at_event	user_age
1	2014-07-01 15:24:47	4	2012-12-31 00:00:00	546	607
2	2014-07-01 09:31:23	4	2012-12-31 00:00:00	545	607
3	2014-07-07 07:46:04	4	2012-12-31 00:00:00	551	607
4	2014-06-30 15:18:47	4	2012-12-31 00:00:00	545	607
5	2014-06-25 11:03:57	4	2012-12-31 00:00:00	539	607
6	2014-06-24 09:31:33	4	2012-12-31 00:00:00	538	607
7	2014-06-19 15:07:00	4	2012-12-31 00:00:00	534	607
8	2014-06-10 06:02:28	4	2012-12-31 00:00:00	524	607
9	2014-06-05 15:20:16	4	2012-12-31 00:00:00	520	607
10	2014-05-30 10:21:18	4	2012-12-31 00:00:00	513	607
11	2014-05-28 11:21:02	4	2012-12-31 00:00:00	511	607
12	2014-05-27 15:09:09	4	2012-12-31 00:00:00	511	607
13	2014-05-20 09:31:30	4	2012-12-31 00:00:00	503	607
14	2014-05-31 15:58:46	4	2012-12-31 00:00:00	515	607
15	2014-05-24 11:39:53	4	2012-12-31 00:00:00	507	607

동료의 쿼리 분석_2번째 쿼리

```
SELECT DATE_TRUNC('week', z.occurred_at) AS "week",
       AVG(z.age_at_event) AS "Average age during week",
       COUNT(DISTINCT CASE WHEN z.user_age > 70 THEN z.user_id ELSE NULL END) AS "10+ weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 70 AND z.user_age >= 63 THEN z.user_id ELSE NULL END) AS "9 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 63 AND z.user_age >= 56 THEN z.user_id ELSE NULL END) AS "8 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 56 AND z.user_age >= 49 THEN z.user_id ELSE NULL END) AS "7 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 49 AND z.user_age >= 42 THEN z.user_id ELSE NULL END) AS "6 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 42 AND z.user_age >= 35 THEN z.user_id ELSE NULL END) AS "5 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 35 AND z.user_age >= 28 THEN z.user_id ELSE NULL END) AS "4 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 28 AND z.user_age >= 21 THEN z.user_id ELSE NULL END) AS "3 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 21 AND z.user_age >= 14 THEN z.user_id ELSE NULL END) AS "2 weeks",
       COUNT(DISTINCT CASE WHEN z.user_age < 14 AND z.user_age >= 7 THEN z.user_id ELSE NULL END) AS "1 week",
       COUNT(DISTINCT CASE WHEN z.user_age < 7 THEN z.user_id ELSE NULL END) AS "Less than a week"
FROM (
    SELECT e.occurred_at,
           u.user_id,
           DATE_TRUNC('week', u.activated_at) AS activation_week,
           EXTRACT('day' FROM e.occurred_at - u.activated_at) AS age_at_event,
           EXTRACT('day' FROM '2014-09-01'::TIMESTAMP - u.activated_at) AS user_age
    FROM tutorial.yammer_users u
    JOIN tutorial.yammer_events e
      ON e.user_id = u.user_id
     AND e.event_type = 'engagement'
     AND e.event_name = 'login'
     AND e.occurred_at >= '2014-05-01'
     AND e.occurred_at < '2014-09-01'
    WHERE u.activated_at IS NOT NULL
) z

GROUP BY 1
ORDER BY 1
LIMIT 100
```

코호트 분석: user_age를 주별로 나누어 활성화 시간 보기

동료의 쿼리 분석_3번째 쿼리

```
SELECT DATE TRUNC('week', occurred at) AS week,  
COUNT(DISTINCT e.user_id) AS weekly_active_users,  
COUNT(DISTINCT CASE WHEN e.device IN ('macbook pro', 'lenovo  
thinkpad', 'macbook air', 'dell inspiron notebook',  
    'asus chromebook', 'dell inspiron desktop', 'acer aspire  
notebook', 'hp pavilion desktop', 'acer aspire desktop', 'mac mini')  
    THEN e.user_id ELSE NULL END) AS computer,  
COUNT(DISTINCT CASE WHEN e.device IN ('iphone 5', 'samsung  
galaxy s4', 'nexus 5', 'iphone 5s', 'iphone 4s', 'nokia lumia 635',  
    'htc one', 'samsung galaxy note', 'amazon fire phone') THEN  
e.user_id ELSE NULL END) AS phone,  
COUNT(DISTINCT CASE WHEN e.device IN ('ipad air', 'nexus  
7', 'ipad mini', 'nexus 10', 'kindle fire', 'windows surface',  
    'samsung galaxy tablet') THEN e.user_id ELSE NULL END) AS  
tablet  
FROM tutorial.yammer_events e  
WHERE e.event_type = 'engagement'  
AND e.event_name = 'login'  
GROUP BY 1  
ORDER BY 1  
LIMIT 100
```

활성화된 주, 각 기기로 로그인한 것 중에서 주별 활성화
유저(weekly_active_user)의 수 확인

	device	기기 종류
1	acer aspire desktop	
2	acer aspire notebook	
3	amazon fire phone	
4	asus chromebook	
5	dell inspiron desktop	
6	dell inspiron notebook	
7	hp pavilion desktop	
8	htc one	
9	ipad air	
10	ipad mini	
11	iphone 4s	
12	iphone 5	
13	iphone 5s	
14	kindle fire	
15	lenovo thinkpad	
16	macbook air	
17	macbook pro	
18	mac mini	
19	nexus 10	
20	nexus 5	
21	nexus 7	
22	nokia lumia 635	
23	samsung galaxy tablet	
24	samsung galaxy note	
25	samsung galaxy s4	
26	windows surface	

동료의 쿼리 분석_4번째 쿼리

```
SELECT DATE TRUNC('week', occurred_at) AS week,  
       COUNT(CASE WHEN e.action =  
'sent weekly digest' THEN e.user_id ELSE NULL END) AS  
weekly_emails,  
       COUNT(CASE WHEN e.action =  
'sent reengagement email' THEN e.user_id ELSE NULL  
END) AS reengagement_emails,  
       COUNT(CASE WHEN e.action = 'email_open' THEN  
e.user_id ELSE NULL END) AS email_opens,  
       COUNT(CASE WHEN e.action =  
'email clickthrough' THEN e.user_id ELSE NULL END) AS  
email_clickthroughs  
FROM tutorial.yammer_emails e  
GROUP BY 1  
ORDER BY 1
```

위에서 언급한 액션들이 일어난
week
-> 날짜는 week의 시작 날짜

원래의 email 테이블의 action 컬럼 종류

	action
1	email_open
2	email_clickthrough
3	sent_weekly_digest
4	sent_reengagement_email

	week	weekly_emails	reengagement_emails	email_opens	email_clickthroughs
1	2014-04-28 00:00:00	908	98	332	187
2	2014-05-05 00:00:00	2602	164	919	434
3	2014-05-12 00:00:00	2665	175	971	479
4	2014-05-19 00:00:00	2733	179	995	498
5	2014-05-26 00:00:00	2822	179	1026	453
6	2014-06-02 00:00:00	2911	199	993	492
7	2014-06-09 00:00:00	3003	190	1070	533

동료의 쿼리 분석_4번째 쿼리

```
SELECT DATE TRUNC('week', occurred_at) AS week,  
       COUNT(CASE WHEN e.action = [redacted]  
'sent weekly digest' THEN e.user_id ELSE NULL END) AS  
weekly_emails,  
       COUNT(CASE WHEN e.action =  
'sent reengagement_email' THEN e.user_id ELSE NULL  
END) AS reengagement_emails,  
       COUNT(CASE WHEN e.action = 'email_open' THEN  
e.user_id ELSE NULL END) AS email_opens,  
       COUNT(CASE WHEN e.action =  
'email_clickthrough' THEN e.user_id ELSE NULL END) AS  
email_clickthroughs  
FROM tutorial.yammer_emails e  
GROUP BY 1  
ORDER BY 1
```

weekly 요약 이메일을 받은
사용자 수를 의미 -> 안받은
사람은 제외!

	week	weekly_emails	reengagement_emails	email_opens	email_clickthroughs
1	2014-04-28 00:00:00	908	98	332	187
2	2014-05-05 00:00:00	2602	164	919	434
3	2014-05-12 00:00:00	2665	175	971	479
4	2014-05-19 00:00:00	2733	179	995	498
5	2014-05-26 00:00:00	2822	179	1026	453
6	2014-06-02 00:00:00	2911	199	993	492
7	2014-06-09 00:00:00	3003	190	1070	533

원래의 email 테이블의 action 컬럼 종류

	action
1	email_open
2	email_clickthrough
3	sent weekly digest
4	sent_reengagement_email

user가 전날의 관련 대화를
보여주는 weekly 요약 전자 메일을
전달받았다.

동료의 쿼리 분석_4번째 쿼리

```
SELECT DATE TRUNC('week', occurred_at) AS week,  
       COUNT(CASE WHEN e.action =  
'sent weekly digest' THEN e.user_id ELSE NULL END) AS  
weekly_emails,  
       COUNT(CASE WHEN e.action =  
'sent reengagement email' THEN e.user_id ELSE NULL  
END) AS reengagement_emails,  
       COUNT(CASE WHEN e.action = 'email_open' THEN  
e.user_id ELSE NULL END) AS email_opens,  
       COUNT(CASE WHEN e.action =  
'email clickthrough' THEN e.user_id ELSE NULL END) AS  
email_clickthroughs  
FROM tutorial.yammer_emails e  
GROUP BY 1  
ORDER BY 1
```

재참여 이메일을 받은 사용자 수를 의미
안받은 사람은 제외!

원래의 email 테이블의 action 컬럼 종류

	action
1	email_open
2	email_clickthrough
3	sent_weekly_digest
4	sent_reengagement_email

user가 재참여를 독려하는 이메일을
전달받았다.

	week	weekly_emails	reengagement_emails	email_opens	email_clickthroughs
1	2014-04-28 00:00:00	908	98	332	187
2	2014-05-05 00:00:00	2602	164	919	434
3	2014-05-12 00:00:00	2665	175	971	479
4	2014-05-19 00:00:00	2733	179	995	498
5	2014-05-26 00:00:00	2822	179	1026	453
6	2014-06-02 00:00:00	2911	199	993	492
7	2014-06-09 00:00:00	3003	190	1070	533

동료의 쿼리 분석_4번째 쿼리

```
SELECT DATE TRUNC('week', occurred_at) AS week,  
       COUNT(CASE WHEN e.action =  
'sent weekly digest' THEN e.user_id ELSE NULL END) AS  
weekly_emails,  
       COUNT(CASE WHEN e.action =  
'sent reengagement email' THEN e.user_id ELSE NULL  
END) AS reengagement_emails,  
       COUNT(CASE WHEN e.action = 'email_open' THEN  
e.user_id ELSE NULL END) AS email_opens,  
       COUNT(CASE WHEN e.action =  
'email clickthrough' THEN e.user_id ELSE NULL END) AS  
email_clickthroughs  
FROM tutorial.yammer_emails e  
GROUP BY 1  
ORDER BY 1
```

이메일을 오픈한 사용자 수를
의미
-> 오픈 하지 않은 사람은
제외해서 count!

원래의 email 테이블의 action 컬럼 종류

	action
1	email_open
2	email_clickthrough
3	sent_weekly_digest
4	sent_reengagement_email

user가 이메일을 오픈한 액션

	week	weekly_emails	reengagement_emails	email_opens	email_clickthroughs
1	2014-04-28 00:00:00	908	98	332	187
2	2014-05-05 00:00:00	2602	164	919	434
3	2014-05-12 00:00:00	2665	175	971	479
4	2014-05-19 00:00:00	2733	179	995	498
5	2014-05-26 00:00:00	2822	179	1026	453
6	2014-06-02 00:00:00	2911	199	993	492
7	2014-06-09 00:00:00	3003	190	1070	533

동료의 쿼리 분석_4번째 쿼리

```
SELECT DATE_TRUNC('week', occurred_at) AS week,  
       COUNT(CASE WHEN e.action =  
'sent weekly digest' THEN e.user_id ELSE NULL END) AS  
weekly_emails,  
       COUNT(CASE WHEN e.action =  
'sent reengagement_email' THEN e.user_id ELSE NULL  
END) AS reengagement_emails,  
       COUNT(CASE WHEN e.action = 'email_open' THEN  
e.user_id ELSE NULL END) AS email_opens,  
       COUNT(CASE WHEN e.action =  
'email_clickthrough' THEN e.user_id ELSE NULL END) AS  
email_clickthroughs  
FROM tutorial.yammer_emails e  
GROUP BY 1
```

이메일안에 포함된 링크를 클릭한
사용자 수를 의미
-> 클릭 하지 않은 사람은
제외해서 count!

		week	weekly_emails	reengagement_emails	email_opens	email_clickthroughs
1		2014-04-28 00:00:00	908	98	332	187
2		2014-05-05 00:00:00	2602	164	919	434
3		2014-05-12 00:00:00	2665	175	971	479
4		2014-05-19 00:00:00	2733	179	995	498
5		2014-05-26 00:00:00	2822	179	1026	453
6		2014-06-02 00:00:00	2911	199	993	492
7		2014-06-09 00:00:00	3003	190	1070	533

원래의 email 테이블의 action 컬럼 종류

	action
1	email_open
2	email_clickthrough
3	sent_weekly_digest
4	sent_reengagement_email

user가 이메일안에 있는 링크를
클릭한 액션

동료의 쿼리 분석_5번째 쿼리_a 테이블

```
SELECT week,
       weekly_opens/CASE WHEN weekly_emails = 0 THEN 1 ELSE weekly_emails END::FLOAT AS weekly_open_rate,
       weekly_ctr/CASE WHEN weekly_opens = 0 THEN 1 ELSE weekly_opens END::FLOAT AS weekly_ctr,
       retain_opens/CASE WHEN retain_emails = 0 THEN 1 ELSE retain_emails END::FLOAT AS retain_open_rate,
       retain_ctr/CASE WHEN retain_opens = 0 THEN 1 ELSE retain_opens END::FLOAT AS retain_ctr
FROM (
SELECT DATE_TRUNC('week',e1.occurred_at) AS week,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e1.user_id ELSE NULL END) AS weekly_emails,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e2.user_id ELSE NULL END) AS weekly_opens,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e3.user_id ELSE NULL END) AS weekly_ctr,
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e1.user_id ELSE NULL END) AS retain_emails,
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e2.user_id ELSE NULL END) AS retain_opens,
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e3.user_id ELSE NULL END) AS retain_ctr
FROM tutorial.yammer_emails e1
LEFT JOIN tutorial.yammer_emails e2
      ON e2.occurred_at >= e1.occurred_at
      AND e2.occurred_at < e1.occurred_at + INTERVAL '5 MINUTE'
      AND e2.user_id = e1.user_id
      AND e2.action = 'email_open'
LEFT JOIN tutorial.yammer_emails e3
      ON e3.occurred_at >= e2.occurred_at
      AND e3.occurred_at < e2.occurred_at + INTERVAL '5 MINUTE'
      AND e3.user_id = e2.user_id
      AND e3.action = 'email_clickthrough'
WHERE e1.occurred_at >= '2014-06-01'
      AND e1.occurred_at < '2014-09-01'
      AND e1.action IN ('sent_weekly_digest','sent_reengagement_email')
GROUP BY 1
) a
ORDER BY 1
```

a 테이블

```
SELECT DATE_TRUNC('week', e1.occurred_at) AS week,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e1.user_id ELSE NULL END) AS weekly_emails, -- 1)
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e2.user_id ELSE NULL END) AS weekly_opens, -- 2)
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e3.user_id ELSE NULL END) AS weekly_ctr, -- 3)
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e1.user_id ELSE NULL END) AS retain_emails, -- 4)
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e2.user_id ELSE NULL END) AS retain_opens, -- 5)
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e3.user_id ELSE NULL END) AS retain_ctr --6)
FROM tutorial.yammer_emails e1
LEFT JOIN tutorial.yammer_emails e2
  ON e2.occurred_at >= e1.occurred_at
 AND e2.occurred_at < e1.occurred_at + INTERVAL '5 MINUTE'
 AND e2.user_id = e1.user_id
 AND e2.action = 'email_open'
LEFT JOIN tutorial.yammer_emails e3
  ON e3.occurred_at >= e2.occurred_at
 AND e3.occurred_at < e2.occurred_at + INTERVAL '5 MINUTE'
 AND e3.user_id = e2.user_id
 AND e3.action = 'email_clickthrough'
WHERE e1.occurred_at >= '2014-06-01'
 AND e1.occurred_at < '2014-09-01'
 AND e1.action IN ('sent_weekly_digest', 'sent_reengagement_email')
GROUP BY 1
```

- 1) weekly이메일 받은 user수
- 2) weekly이메일 열어본 user 수
- 3) weekly이메일에 포함된 링크 클릭한 user수
- 4) 재참여 이메일 받은 user수
- 5) 재참여 이메일 열어본 user 수
- 6) 재참여 이메일에 포함된 링크 클릭한 user수

	week	1) weekly_emails	2) weekly_opens	3) weekly_ctr	4) retain_emails	5) retain_opens	6) retain_ctr
1	2014-05-26 00:00:00	0	0	0	37	31	28
2	2014-06-02 00:00:00	2911	820	342	199	173	150
3	2014-06-09 00:00:00	3003	898	377	190	172	156
4	2014-06-16 00:00:00	3105	951	375	234	210	188
5	2014-06-23 00:00:00	3207	927	372	187	163	152
6	2014-06-30 00:00:00	3302	967	379	222	201	180
7	2014-07-07 00:00:00	3399	1038	451	214	192	171
8	2014-07-14 00:00:00	3499	1057	416	226	203	191
9	2014-07-21 00:00:00	3592	1031	419	206	180	165
10	2014-07-28 00:00:00	3706	1180	446	230	206	187

a 테이블

```
SELECT DATE_TRUNC('week', e1.occurred_at) AS week,
COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e1.user_id ELSE NULL END) AS weekly_emails,
COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e2.user_id ELSE NULL END) AS weekly_opens,
COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e3.user_id ELSE NULL END) AS weekly_ctr,
COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e1.user_id ELSE NULL END) AS retain_emails,
COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e2.user_id ELSE NULL END) AS retain_opens,
COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e3.user_id ELSE NULL END) AS retain_ctr
FROM tutorial.yammer_emails e1
LEFT JOIN tutorial.yammer_emails e2
  ON e2.occurred_at >= e1.occurred_at
 AND e2.occurred_at < e1.occurred_at + INTERVAL '5 MINUTE'
 AND e2.user_id = e1.user_id
 AND e2.action = 'email_open'
LEFT JOIN tutorial.yammer_emails e3
  ON e3.occurred_at >= e2.occurred_at
 AND e3.occurred_at < e2.occurred_at + INTERVAL '5 MINUTE'
 AND e3.user_id = e2.user_id
 AND e3.action = 'email_clickthrough'
WHERE e1.occurred_at >= '2014-06-01'
 AND e1.occurred_at < '2014-09-01'
 AND e1.action IN ('sent_weekly_digest', 'sent_reengagement_email')
GROUP BY 1
```

email 테이블 x 3개 활용 -> e1, e2, e3

5분 안에 열어야 email open으로 간주

5분 안에 클릭해야 email clickthrough로 간주

1	user_id_duplicate_column_name_1	occurred_at_duplicate_column_name_1	action_duplicate_column_name_1	user_type	user_id	occurred_at	action	user_type
90376	18806	2014-08-31 19:40	email_clickthrough	3				
90377	18809	2014-08-31 22:59	sent_reengagement_email	3	18809	2014-08-31 23:00	email_open	3
90378	18809	2014-08-31 23:00	email_open	3	18809	2014-08-31 23:00	email_open	3

WHERE 문: action은 weekly요약 이메일(sent_weekly_digest), 재참여 독려 이메일(sent_reengagement_email) 모두 포함
기간은 6,7,8월 안에 것만 보겠다.

동료의 쿼리 분석_5번째 쿼리

```
SELECT week,  
       weekly_opens/CASE WHEN weekly_emails = 0 THEN 1 ELSE weekly_emails END::FLOAT AS weekly_open_rate,  
       weekly_ctr/CASE WHEN weekly_opens = 0 THEN 1 ELSE weekly_opens END::FLOAT AS weekly_ctr,  
       retain_opens/CASE WHEN retain_emails = 0 THEN 1 ELSE retain_emails END::FLOAT AS retain_open_rate,  
       retain_ctr/CASE WHEN retain_opens = 0 THEN 1 ELSE retain_opens END::FLOAT AS retain_ctr  
FROM (  
SELECT DATE_TRUNC('week',e1.occurred_at) AS week,  
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e1.user_id ELSE NULL END) AS weekly_emails,  
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e2.user_id ELSE NULL END) AS weekly_opens,  
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e3.user_id ELSE NULL END) AS weekly_ctr,  
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e1.user_id ELSE NULL END) AS retain_emails,  
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e2.user_id ELSE NULL END) AS retain_opens,  
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e3.user_id ELSE NULL END) AS retain_ctr  
FROM tutorial.yammer_emails e1  
LEFT JOIN tutorial.yammer_emails e2  
  ON e2.occurred_at >= e1.occurred_at  
 AND e2.occurred_at < e1.occurred_at + INTERVAL '5 MINUTE'  
 AND e2.user_id = e1.user_id  
 AND e2.action = 'email_open'  
LEFT JOIN tutorial.yammer_emails e3  
  ON e3.occurred_at >= e2.occurred_at  
 AND e3.occurred_at < e2.occurred_at + INTERVAL '5 MINUTE'  
 AND e3.user_id = e2.user_id  
 AND e3.action = 'email_clickthrough'  
WHERE e1.occurred_at >= '2014-06-01'  
 AND e1.occurred_at < '2014-09-01'  
 AND e1.action IN ('sent_weekly_digest','sent_reengagement_email')  
GROUP BY 1  
       ) a  
ORDER BY 1
```

위클리 이메일 오픈율 (weekly_open_rate)

위클리 이메일 5분 안에 열어본 유저 수 분자

위클리 이메일을 받은 유저 수
(받지 않았을 때는 분모가 1) 분모

동료의 쿼리 분석_5번째 쿼리

```
SELECT week,  
       weekly_opens/CASE WHEN weekly_emails = 0 THEN 1 ELSE weekly_emails END::FLOAT AS weekly_open_rate,  
       weekly_ctr/CASE WHEN weekly_opens = 0 THEN 1 ELSE weekly_opens END::FLOAT AS weekly_ctr,  
       retain_opens/CASE WHEN retain_emails = 0 THEN 1 ELSE retain_emails END::FLOAT AS retain_open_rate,  
       retain_ctr/CASE WHEN retain_opens = 0 THEN 1 ELSE retain_opens END::FLOAT AS retain_ctr  
FROM (  
SELECT DATE_TRUNC('week',e1.occurred_at) AS week,  
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e1.user_id ELSE NULL END) AS weekly_emails,  
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e2.user_id ELSE NULL END) AS weekly_opens,  
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e3.user_id ELSE NULL END) AS weekly_ctr,  
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e1.user_id ELSE NULL END) AS retain_emails,  
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e2.user_id ELSE NULL END) AS retain_opens,  
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e3.user_id ELSE NULL END) AS retain_ctr  
FROM tutorial.yammer_emails e1  
LEFT JOIN tutorial.yammer_emails e2  
  ON e2.occurred_at >= e1.occurred_at  
 AND e2.occurred_at < e1.occurred_at + INTERVAL '5 MINUTE'  
 AND e2.user_id = e1.user_id  
 AND e2.action = 'email_open'  
LEFT JOIN tutorial.yammer_emails e3  
  ON e3.occurred_at >= e2.occurred_at  
 AND e3.occurred_at < e2.occurred_at + INTERVAL '5 MINUTE'  
 AND e3.user_id = e2.user_id  
 AND e3.action = 'email_clickthrough'  
WHERE e1.occurred_at >= '2014-06-01'  
 AND e1.occurred_at < '2014-09-01'  
 AND e1.action IN ('sent_weekly_digest','sent_reengagement_email')  
GROUP BY 1  
       ) a  
ORDER BY 1
```

위클리 이메일 클릭률 (weekly_ctr)

위클리 이메일 5분안에 클릭한 유저 수 **분자**

위클리 이메일을 5분안에 열어본 유저 수
(열지 않았을 때는 분모가 1) **분모**

동료의 쿼리 분석_5번째 쿼리

```
SELECT week,
       weekly_opens/CASE WHEN weekly_emails = 0 THEN 1 ELSE weekly_emails END::FLOAT AS weekly_open_rate,
       weekly_ctr/CASE WHEN weekly_opens = 0 THEN 1 ELSE weekly_opens END::FLOAT AS weekly_ctr,
       retain_opens/CASE WHEN retain_emails = 0 THEN 1 ELSE retain_emails END::FLOAT AS retain_open_rate,
       retain_ctr/CASE WHEN retain_opens = 0 THEN 1 ELSE retain_opens END::FLOAT AS retain_ctr
FROM (
SELECT DATE_TRUNC('week',e1.occurred_at) AS week,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e1.user_id ELSE NULL END) AS weekly_emails,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e2.user_id ELSE NULL END) AS weekly_opens,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e3.user_id ELSE NULL END) AS weekly_ctr,
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e1.user_id ELSE NULL END) AS retain_emails,
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e2.user_id ELSE NULL END) AS retain_opens,
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e3.user_id ELSE NULL END) AS retain_ctr
FROM tutorial.yammer_emails e1
LEFT JOIN tutorial.yammer_emails e2
  ON e2.occurred_at >= e1.occurred_at
 AND e2.occurred_at < e1.occurred_at + INTERVAL '5 MINUTE'
 AND e2.user_id = e1.user_id
 AND e2.action = 'email_open'
LEFT JOIN tutorial.yammer_emails e3
  ON e3.occurred_at >= e2.occurred_at
 AND e3.occurred_at < e2.occurred_at + INTERVAL '5 MINUTE'
 AND e3.user_id = e2.user_id
 AND e3.action = 'email_clickthrough'
WHERE e1.occurred_at >= '2014-06-01'
 AND e1.occurred_at < '2014-09-01'
 AND e1.action IN ('sent_weekly_digest','sent_reengagement_email')
GROUP BY 1
       ) a
ORDER BY 1
```

리텐션 이메일 오픈율 (retain_open_rate)

재참여 이메일 5분 안에 열어본 유저 수 분자

재참여 이메일 받은 유저 수
(받지 않았을 때는 분모가 1)

분모

동료의 쿼리 분석_5번째 쿼리

```
SELECT week,
       weekly_opens/CASE WHEN weekly_emails = 0 THEN 1 ELSE weekly_emails END::FLOAT AS weekly_open_rate,
       weekly_ctr/CASE WHEN weekly_opens = 0 THEN 1 ELSE weekly_opens END::FLOAT AS weekly_ctr,
       retain_opens/CASE WHEN retain_emails = 0 THEN 1 ELSE retain_emails END::FLOAT AS retain_open_rate,
       retain_ctr/CASE WHEN retain_opens = 0 THEN 1 ELSE retain_opens END::FLOAT AS retain_ctr
FROM (
SELECT DATE_TRUNC('week',e1.occurred_at) AS week,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e1.user_id ELSE NULL END) AS weekly_emails,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e2.user_id ELSE NULL END) AS weekly_opens,
       COUNT(CASE WHEN e1.action = 'sent_weekly_digest' THEN e3.user_id ELSE NULL END) AS weekly_ctr,
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e1.user_id ELSE NULL END) AS retain_emails,
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e2.user_id ELSE NULL END) AS retain_opens,
       COUNT(CASE WHEN e1.action = 'sent_reengagement_email' THEN e3.user_id ELSE NULL END) AS retain_ctr
FROM tutorial.yammer_emails e1
LEFT JOIN tutorial.yammer_emails e2
  ON e2.occurred_at >= e1.occurred_at
 AND e2.occurred_at < e1.occurred_at + INTERVAL '5 MINUTE'
 AND e2.user_id = e1.user_id
 AND e2.action = 'email_open'
LEFT JOIN tutorial.yammer_emails e3
  ON e3.occurred_at >= e2.occurred_at
 AND e3.occurred_at < e2.occurred_at + INTERVAL '5 MINUTE'
 AND e3.user_id = e2.user_id
 AND e3.action = 'email_clickthrough'
WHERE e1.occurred_at >= '2014-06-01'
 AND e1.occurred_at < '2014-09-01'
 AND e1.action IN ('sent_weekly_digest','sent_reengagement_email')
GROUP BY 1
       ) a
ORDER BY 1
```

리텐션 이메일 클릭률 (retain_ctr)

재참여 이메일 5분 안에 클릭한 유저 수 분자

재참여 이메일 5분 안에 열어본 유저 수 분모
(열지 않았을 때는 분모가 1)

1번째 쿼리 : Daily Signups

동료의 분석: 야머의 성장을 확인할 수 있으며, 성장률은 주중에는 계속 높지만 주말에는 낮다.

Daily Signups



1번째 쿼리 : Daily Signups

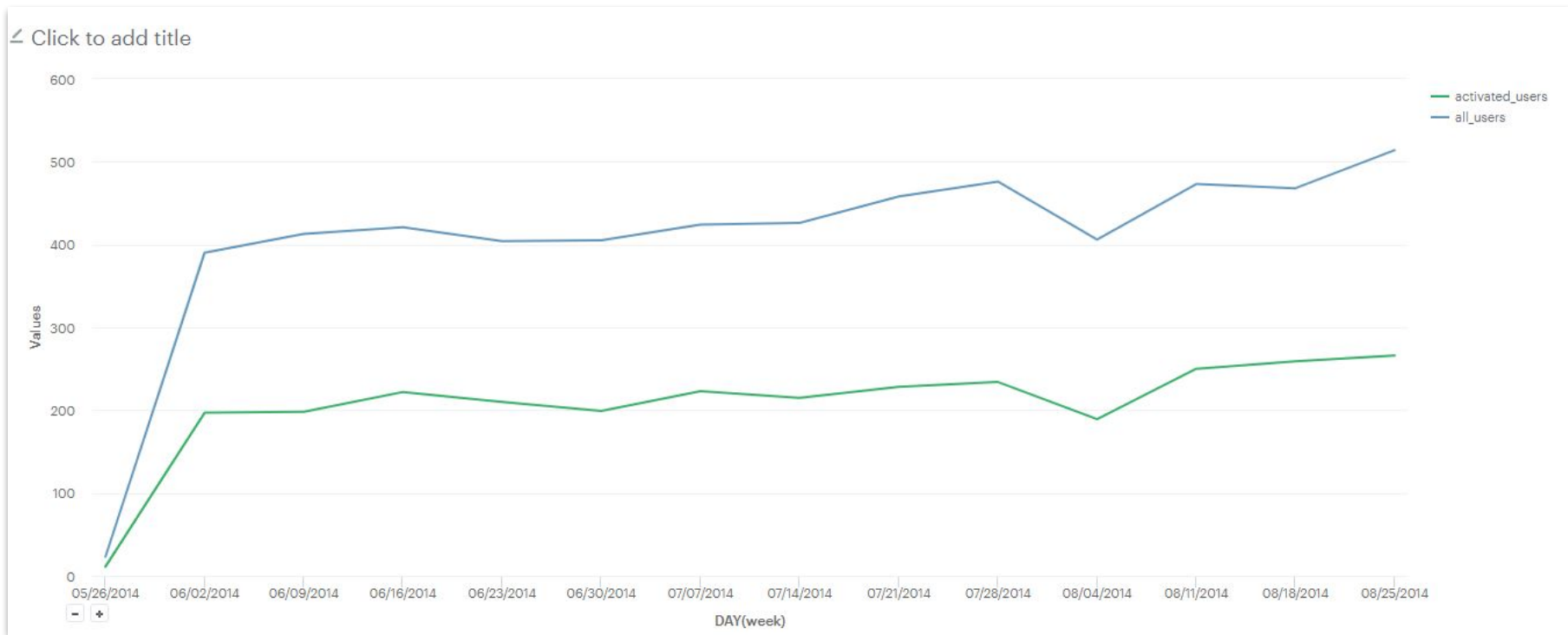
날짜	요일	All Users	All Users 변화율	Activated Users	Activated Users 변화율
2014-07-21	월	77	285.00%	37	311.11%
2014-07-22	화	79	2.60%	38	2.70%
2014-07-23	수	89	12.66%	46	21.05%
2014-07-24	목	78	-12.36%	40	-13.04%
2014-07-25	금	85	8.97%	39	-2.50%
2014-07-26	토	24	-71.76%	12	-69.23%
2014-07-27	일	26	8.33%	16	33.33%
2014-07-28	월	88	238.46%	43	168.75%
2014-07-29	화	77	-12.50%	35	-18.60%
2014-07-30	수	88	14.29%	45	28.57%
2014-07-31	목	94	6.82%	44	-2.22%
2014-08-01	금	77	-18.09%	42	-4.55%
2014-08-02	토	27	-64.94%	13	-69.05%
2014-08-03	일	25	-7.41%	12	-7.69%
2014-08-04	월	57	128.00%	26	116.67%
2014-08-05	화	72	26.32%	40	53.85%
2014-08-06	수	76	5.56%	29	-27.50%
2014-08-07	목	76	0.00%	41	41.38%
2014-08-08	금	79	3.95%	32	-21.95%
2014-08-09	토	23	-70.89%	13	-59.38%
2014-08-10	일	23	0.00%	8	-38.46%
2014-08-11	월	87	278.26%	43	437.50%
2014-08-12	화	77	-11.49%	34	-20.93%
2014-08-13	수	86	11.69%	53	55.88%
2014-08-14	목	92	6.98%	45	-15.09%
2014-08-15	금	81	-11.96%	48	6.67%
2014-08-16	토	28	-65.43%	14	-70.83%
2014-08-17	일	22	-21.43%	13	-7.14%

우리의 분석: 테이블의 데이터 수치로
확인해보면 동료의 분석처럼 요일별
신규가입자의 추이에 큰 변화는 없음.

분석 과정에서 시행착오: daily active user로
수치를 쪼개서 확인해보려 했지만, 요일별
특성(주중 & 주말)이 강해 분석이 어려웠음.
→ 요일별 집중 마케팅

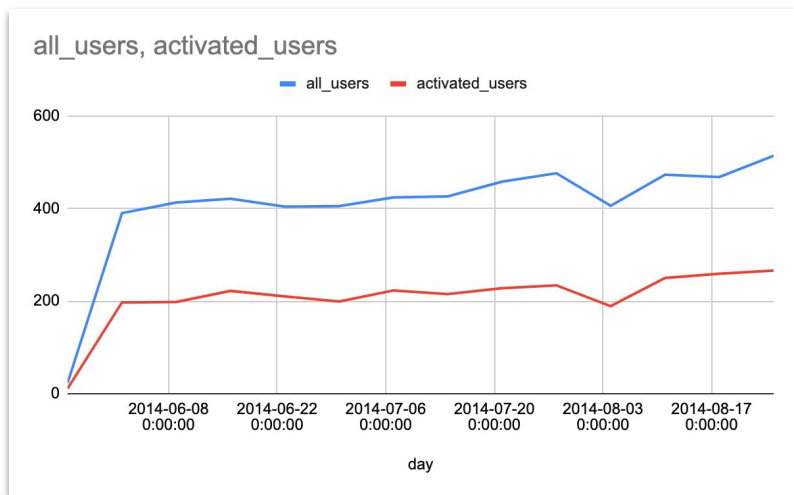
1번째 쿼리 : Daily Signups

우리의 분석: 주 별로 보았을 때, 8월 1째 주 (8월 4일 시작)에서 신규가입자 및 신규 가입자 중 활성화 유저 모두 감소했다.



1번째 쿼리 : Daily Signups

우리의 분석: 8월 4일~8월 10일에 해당하는 기간동안 고객사 여름 holiday가 해당 기간 신규 가입자 감소 원인이 될 수 있을 것이라고 판단. 단, 이 가설에는 회사 휴일 시즌이 모두 다르기 때문에 애매할 수 있다는 문제점 존재



week	all_users	activated_users
2014-05-26 0:00	23	11
2014-06-02 0:00	390	197
2014-06-09 0:00	413	198
2014-06-16 0:00	421	222
2014-06-23 0:00	404	210
2014-06-30 0:00	405	199
2014-07-07 0:00	424	223
2014-07-14 0:00	426	215
2014-07-21 0:00	458	228
2014-07-28 0:00	476	234
2014-08-04 0:00	406	189
2014-08-11 0:00	473	250
2014-08-18 0:00	468	259
2014-08-25 0:00	514	266

2번째 쿼리 : Engagement by User Age Cohort

동료의 분석: 성장이 정상적이라면 기존 유저들의 활동은 감소하고, 새로운 유저의 활동은 늘어남.

이를 확인할 수 있는 방안 → 코호트 분석



2번째 쿼리 : Engagement by User Age Cohort

우리의 분석

- 표의 수치에 근거하여 8월4일, 8월 11일에 10+weeks users가 급 감소함을 확인 → 기존 유저 활동 감소 즉, 이 두 날짜가 포함된 기간에 이탈을 야기하는 이벤트가 있었을 것이라고 판단.

week	Average age during week	Average age 변화량	10+ weeks	10+ weeks 변화량
2014-04-28	124.0072389	-	701	-
2014-05-05	124.3816908	0.30%	1054	50.36%
2014-05-12	131.9386442	6.08%	1094	3.80%
2014-05-19	132.3266284	0.29%	1147	4.84%
2014-05-26	132.3453634	0.01%	1113	-2.96%
2014-06-02	131.8311091	-0.39%	1173	5.39%
2014-06-09	131.0425824	-0.60%	1219	3.92%
2014-06-16	136.4805654	4.15%	1255	2.95%
2014-06-23	136.2789056	-0.15%	1034	-17.61%
2014-06-30	136.4192975	0.10%	917	-11.32%
2014-07-07	135.8887505	-0.39%	899	-1.96%
2014-07-14	143.4488157	5.56%	832	-7.45%
2014-07-21	141.70278	-1.22%	791	-4.93%
2014-07-28	144.0786604	1.68%	805	1.77%
2014-08-04	140.732238	-2.32%	678	-15.78%
2014-08-11	125.9943101	-10.47%	562	-17.11%
2014-08-18	128.0217181	1.61%	522	-7.12%
2014-08-25	128.2698104	0.19%	474	-9.20%

2번째 쿼리 : Engagement by User Age Cohort

우리의 분석 : event_name을 login으로 설정, email_open 등 더 구체화된 활성화 유저 설정 가능하다고 생각했지만, engagement event 중 중복되지 않고 unique한 값을 가지는 것은 login 이기 때문에 동료가 event_name을 login으로 정한 것이라고 판단.

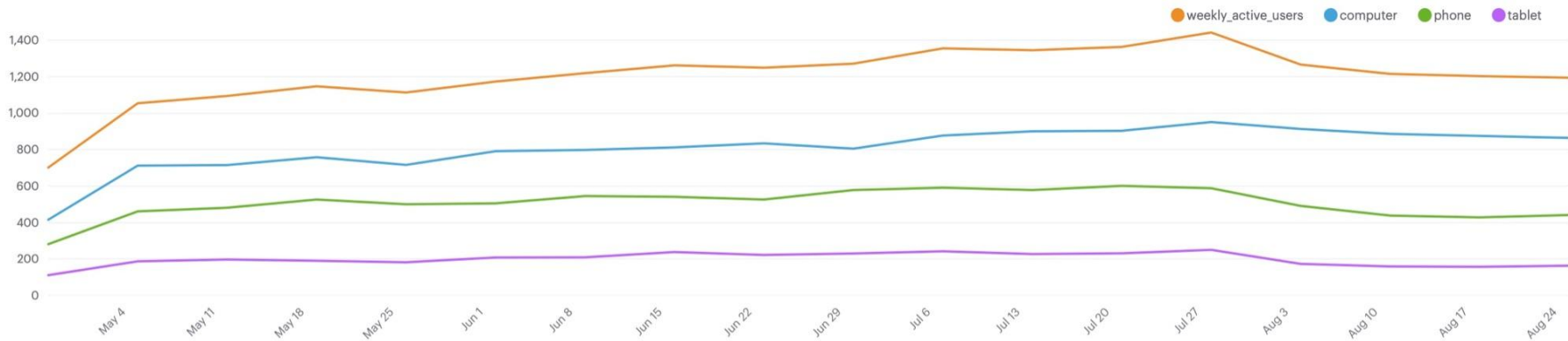
```
SELECT e.occurred_at,
       u.user_id,
       DATE_TRUNC('week', u.activated_at) AS activation_week,
       EXTRACT('day' FROM e.occurred_at - u.activated_at) AS age_at_event,
       EXTRACT('day' FROM '2014-09-01'::TIMESTAMP - u.activated_at) AS user_age
FROM tutorial.yammer_users u
JOIN tutorial.yammer_events e
  ON e.user_id = u.user_id
 AND e.event_type = 'engagement'
 AND e.event_name = 'login'
 AND e.occurred_at >= '2014-05-01'
 AND e.occurred_at < '2014-09-01'
WHERE u.activated_at IS NOT NULL
```

	event_name
1	complete_signup
2	create_user
3	enter_email
4	enter_info
5	home_page
6	like_message
7	login
8	search_autocomplete
9	search_click_result_1
10	search_click_result_10
11	search_click_result_2
12	search_click_result_3
13	search_click_result_4
14	search_click_result_5
15	search_click_result_6
16	search_click_result_7
17	search_click_result_8
18	search_click_result_9
19	search_run
20	send_message
21	view_inbox

3번째 쿼리 : Weekly Engagement by Device Category

동료의 분석 : 1,2번 쿼리를 기반으로 신규가입자가 아닌, 기존 오래된 진성 유저의 이탈 문제로 판단 -> 특정 디바이스에 한정되어 트래픽이 감소하는지 알아보는 것으로 분석 방향 변경 -> **Phone** 사용자 그래프에서 급감하므로 **모바일 앱과 관련된 문제**가 있었을 것으로 보인다고 제언.

Weekly Engagement by Device Category

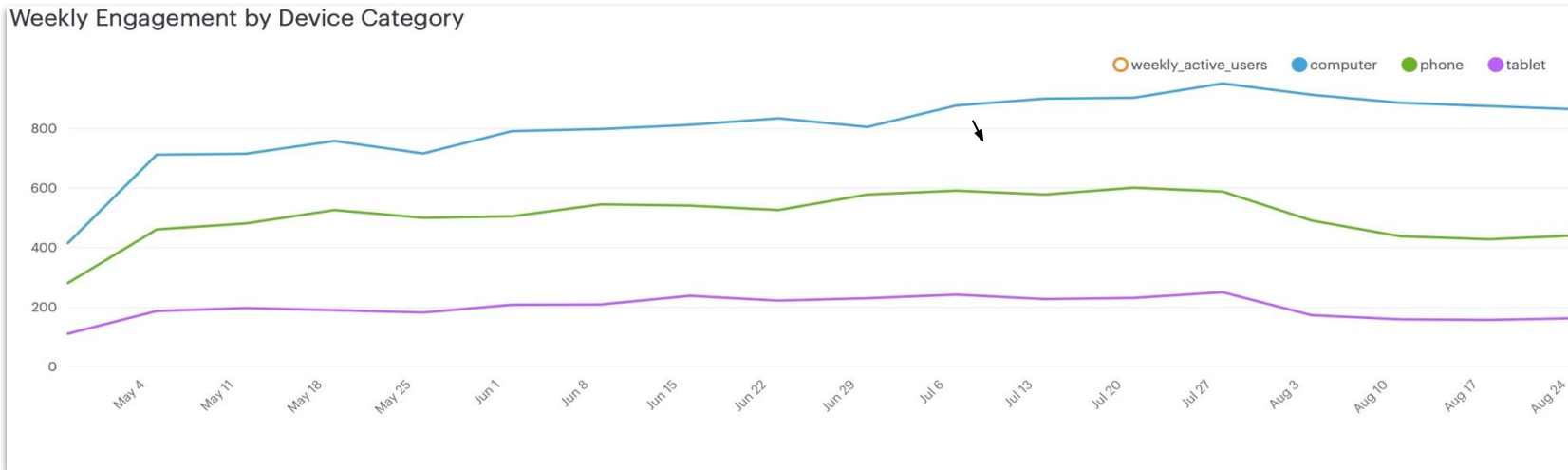


3번째 쿼리 : Weekly Engagement by Device Category

우리의 분석

(동료의 분석 그래프에 근거) 전반적으로 computer 사용자의 수가 많은걸로 보아 Yammer user는 computer 사용을 선호

-> 사무실과 같은 장소에서 업무용으로 많이 사용됨.



3번째 쿼리 : Weekly Engagement by Device Category

우리의 분석

(아래 동료의 분석 데이터 테이블에 근거) 디바이스 유형에서 “computer 사용률은 유지되었지만 **phone**, **tablet**의 사용률은 줄었다.”

week	WAU	WAU 변화량	tablet 사용자수	tablet 사용자수 변화량	phone 사용자수	phone 사용자수 변화량	computer 사용자수	computer 사용자수 변화량
2014-04-28	701	-	111	-	281	-	415	-
2014-05-05	1054	50.36%	187	68.47%	461	64.06%	712	71.57%
2014-05-12	1094	3.80%	197	5.35%	481	4.34%	715	0.42%
2014-05-19	1147	4.84%	190	-3.55%	526	9.36%	758	6.01%
2014-05-26	1113	-2.96%	182	-4.21%	500	-4.94%	716	-5.54%
2014-06-02	1173	5.39%	208	14.29%	505	1.00%	791	10.47%
2014-06-09	1219	3.92%	209	0.48%	545	7.92%	798	0.88%
2014-06-16	1262	3.53%	238	13.88%	541	-0.73%	812	1.75%
2014-06-23	1249	-1.03%	222	-6.72%	526	-2.77%	834	2.71%
2014-06-30	1271	1.76%	230	3.60%	578	9.89%	805	-3.48%
2014-07-07	1355	6.61%	242	5.22%	591	2.25%	877	8.94%
2014-07-14	1345	-0.74%	227	-6.20%	578	-2.20%	900	2.62%
2014-07-21	1363	1.34%	231	1.76%	601	3.98%	903	0.33%
2014-07-28	1442	5.80%	250	8.23%	588	-2.16%	951	5.32%
2014-08-04	1266	-12.21%	173	-30.80%	491	-16.50%	913	-4.00%
2014-08-11	1215	-4.03%	159	-8.09%	438	-10.79%	886	-2.96%
2014-08-18	1203	-0.99%	157	-1.26%	428	-2.28%	875	-1.24%
2014-08-25	1194	-0.75%	163	3.82%	441	3.04%	864	-1.26%

3번째 쿼리 : Weekly Engagement by Device Category

가설1.

Yammer is a social network for communicating with coworkers. Individuals share documents, updates, and ideas by posting them in groups. Yammer is free to use indefinitely, but companies must pay license fees if they want access to administrative controls, including integration with user management systems like ActiveDirectory.

computer = 대부분 회사 사용자 (pay) / **phone & tablet** = 대부분 개인 사용자 (free)

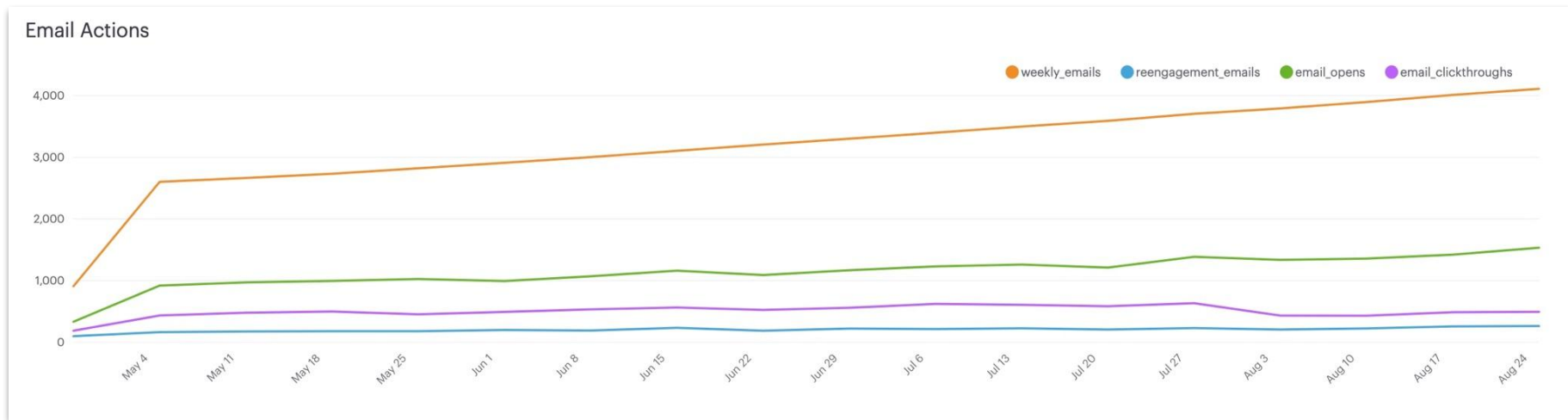
→ **computer**를 주로 사용하는 회사(pay)사용자를 늘리기 위한 이벤트가 있었으며, 그 이벤트가 **phone**과 **tablet**을 주로 사용하는 개인사용자를 이탈 시키는 원인이 되었을거라 추측.

가설 2. Application의 기술적 문제로 인한 개인 사용자 이탈하는 원인이 생겼지 않았을까 하는 추측. (동료의 분석과 어느정도 일치)

가설 3. Application의 UI 변경으로 인한 개인 사용자 이탈하는 원인이 생겼지 않았을까 하는 추측. (동료의 분석과 어느정도 일치)

4번째 쿼리 : Email Actions

동료의 분석: 장기 유저가 빠져나간 현상에 대해서, weekly_digest_email가 관련 있는지 확인이 필요하다



4번째 쿼리 : Email Actions

동료의 분석: 장기 사용자 보존과 관련하여 `weekly_digest_email` 에 문제가 있을 것이라고 방향 설정하고 분석

week	weekly emails	reengagement emails	email opens	email opens 변화량	email clickthroughs	email clickthroughs 변화량
2014-04-28	908	98	332	-	187	-
2014-05-05	2602	164	919	176.81%	434	132.09%
2014-05-12	2665	175	971	5.66%	479	10.37%
2014-05-19	2733	179	995	2.47%	498	3.97%
2014-05-26	2822	179	1026	3.12%	453	-9.04%
2014-06-02	2911	199	993	-3.22%	492	8.61%
2014-06-09	3003	190	1070	7.75%	533	8.33%
2014-06-16	3105	234	1161	8.50%	563	5.63%
2014-06-23	3207	187	1090	-6.12%	524	-6.93%
2014-06-30	3302	222	1168	7.16%	559	6.68%
2014-07-07	3399	214	1230	5.31%	622	11.27%
2014-07-14	3499	226	1260	2.44%	607	-2.41%
2014-07-21	3592	206	1211	-3.89%	584	-3.79%
2014-07-28	3706	230	1386	14.45%	633	8.39%
2014-08-04	3793	206	1336	-3.61%	432	-31.75%
2014-08-11	3897	224	1357	1.57%	430	-0.46%
2014-08-18	4012	257	1421	4.72%	487	13.26%
2014-08-25	4111	263	1533	7.88%	493	1.23%

4번째 쿼리 : Email Actions

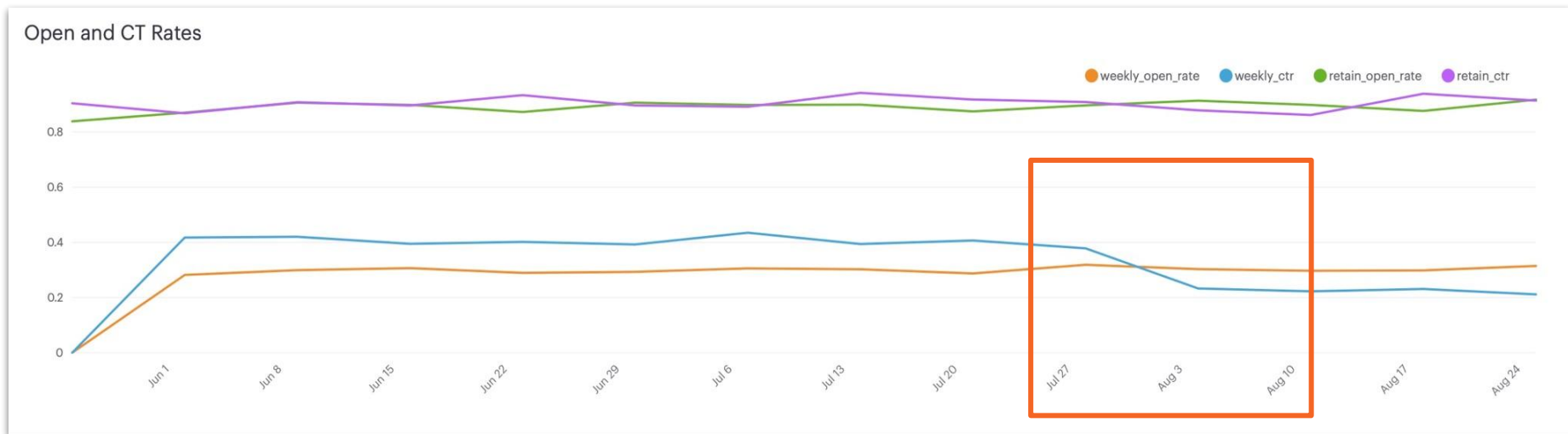
우리의 분석

- 메일(위클리, 리인게이지먼트) 발송률, 메일 오픈률은 꾸준히 유지
- 8월 4일부터 메일 클릭률이 56%→32%로 급감소하며, 이후 감소세 유지
- 7월부터 8월까지 메일 오픈은 6.5% 증가했지만 클릭률은 27% 감소함
- 따라서 메일의 콘텐츠가 사용자가 야머로 유입시킬만큼의 매력적인 콘텐츠가 아니었을 것이라 추측
- 장기 사용자를 다시 유입시킬만한 콘텐츠 필요
- 동료가 분석한 위클리 다이제스트 메일에 문제가 있음을 확인하기 위해 메일의 종류가 무엇이었는데(digest or reengagement)에 대한 정보 필요

week	open_ratio	clickthrough_ratio
2014-04-28	36.56	56.33
2014-05-05	35.32	47.23
2014-05-12	36.44	49.33
2014-05-19	36.41	50.05
2014-05-26	36.36	44.15
2014-06-02	34.11	49.55
2014-06-09	35.63	49.81
2014-06-16	37.39	48.49
2014-06-23	33.99	48.07
2014-06-30	35.37	47.86
2014-07-07	36.19	50.57
2014-07-14	36.01	48.17
2014-07-21	33.71	48.22
2014-07-28	37.40	45.67
2014-08-04	35.22	32.34
2014-08-11	34.82	31.69
2014-08-18	35.42	34.27
2014-08-25	37.29	32.16

5번째 쿼리 : Open and CT Rates

동료의 분석: weekly digest e-mail 의 clickthrough가 낮음. weekly digest e-mail에 문제가 있음.



5번째 쿼리 : Open and CT Rates

우리의 분석

* retain : reengagement e-mail과 관련된 수치 / weekly: weekly digest e-mail과 관련된 수치

- retain ctr은 변화가 크지 않지만, weekly ctr 수치는 8월 4일 급 감소함(-3.21%)

- reengagement e-mail은 dip과 무관하며, weekly digest e-mail이 dip과 연관이 있다. => 장기 유저가 빠져나간 현상과 관련 있음

week	weekly open rate	weekly open rate 변화율	weekly CTR	weekly CTR 변화율	retain open rate	retain open rate 변화율	retain CTR	retain CTR 변화율
2014-06-02	0.28	-	0.42	-	0.87	-	0.87	-
2014-06-09	0.30	6.16%	0.42	-6.07%	0.91	4.13%	0.91	4.60%
2014-06-16	0.31	2.42%	0.39	1.77%	0.90	-0.86%	0.90	-1.29%
2014-06-23	0.29	-5.62%	0.40	-2.33%	0.87	-2.87%	0.93	4.16%
2014-06-30	0.29	1.31%	0.39	10.86%	0.91	3.87%	0.90	-3.97%
2014-07-07	0.31	4.28%	0.43	-9.42%	0.90	-0.91%	0.89	-0.55%
2014-07-14	0.30	-1.08%	0.39	3.26%	0.90	0.12%	0.94	5.64%
2014-07-21	0.29	-4.99%	0.41	-7.00%	0.87	-2.72%	0.92	-2.57%
2014-07-28	0.32	10.93%	0.38	-38.47%	0.90	2.50%	0.91	-0.97%
2014-08-04	0.30	-4.94%	0.23	-4.41%	0.91	1.89%	0.88	-3.32%
2014-08-11	0.30	-1.99%	0.22	3.80%	0.90	-1.68%	0.86	-1.93%
2014-08-18	0.30	0.49%	0.23	-8.44%	0.88	-2.43%	0.94	8.96%
2014-08-25	0.31	5.43%	0.21	-100.00%	0.92	4.67%	0.91	-2.66%

추가 아이디어 및 공유하고 싶은 내용

1. **Activation Rate**를 분석에서 **day** 기준으로 수치 비교, 이외의 분석에서는 **week** 기준으로 수치 비교
→ 기준을 통일한다면 더 수월한 수치 비교 가능
2. 더욱 디테일한 분석을 위해서는 테이블을 활용한 더 세세한 분석 결과도 함께 시각화했다면
데이터 관련 직무가
아닌 팀원들도 수월한 이해 가능
3. **email_open** 에 카운트하는 기준 로그인 후 **5분**으로 설정
→ 예외 가능성, 시간 설정에 대한 고민이 필요 (한 주 간의 내용을 정리해서 보내주는 **weekly digest** 메일)

과제 수행 후기

김다은

현직자 업무를 간접 체험
해봤는데, 실제로 테이블이
더 많으면 상당히 막막할 것
같다. 팀원들과 소통하는
것이 큰 도움이 된다는 것을
느꼈다.

유빈

데이터 분석 주제가 마치
실제 회사 업무와 유사해
굉장히 의미있었습니다! 다른
분들의 인사이트를 듣고
나누는 과정을 통해 분석
시각을 넓힐 수 있는 기회를
얻었습니다.

이정인

이번 프로젝트를 통해 다른
분의 SQL 코드를 보고 분석
및 비평을 해보는 신선한
경험을 할 수 있었습니다!
그리고 팀원들이랑 같이
하니까 막막했던것이 좀 더
빨리 수월하게 해결된 것
같아 협업의 중요성을 또다시
느껴봤습니다.

과제 수행 후기

조은정

- 실제 회사에서는
가설설정을 하고 이 것을
SQL짜서 어떻게 인사이트를
도출해내는지 과정을 엮은 것
같았다. 두루뭉실 한게
한발짝 와닿는 느낌!

홍은선

전문적인 데이터 분석가는 제
성향과 맞지 않을 수
있겠다라고 느꼈지만,
전체적인 분석과 비평 과정이
재미있었다! SQL을 간단하게
이해하고 이용하는 스킬이
요구되는 파생된 직무에
도전해보고 싶다는 생각을
했습니다.

DAITGIRLS 화이팅!

감사합니다.

6조 발표였습니다.