Investigating a Drop in User Engagement - SQL Script

Checking Growth:

Using the *yammer_users* table:

```
SELECT
                    DATE_TRUNC('day', created_at) AS days,
                    COUNT(*) AS users,
                    COUNT(CASE
                      WHEN activated_at IS NOT NULL THEN user_id ELSE NULL
                      END)
                      AS active_users
FROM tutorial.yammer_users
GROUP BY 1
ORDER BY 1
```

Checking Users engagement:

Using the *yammer_users* table and join it with *yammer_events*:

```
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
```

Checking Devices Engagement:

Using the *yammer_events table*:

```
SELECT
  DATE_TRUNC('week', categorized_devices.occurred_at) AS "week",
  COUNT(CASE WHEN device_category = 'desktop' THEN 1 END) AS desktop_count,
  COUNT(CASE WHEN device_category = 'phone' THEN 1 END) AS phone_count,
  COUNT(CASE WHEN device_category = 'tablet' THEN 1 END) AS tablet_count
FROM (
  SELECT
    occurred_at,
    CASE
      WHEN device LIKE '%desktop%' OR device LIKE '%Macbook%' OR device LIKE '%chromebook%'
      OR device LIKE '%surface%' OR device LIKE '%thinkpad%' OR device LIKE '%notebook%'
      THEN 'desktop'
```

```
   WHEN device LIKE '%phone%' OR device LIKE '%iphone%' OR device LIKE '%htc one%'
   OR device LIKE '%lumia%' OR device LIKE '%galaxy%' THEN 'phone'
   WHEN device LIKE '%tablet%' OR device LIKE '%ipad%' OR device LIKE '%nexus%'
   OR device LIKE '%kindle%' THEN 'tablet'
   ELSE 'other' -- In case there's a device that doesn't fit into these categories
  END AS device_category
 FROM tutorial.yammer_events
) AS categorized_devices
GROUP BY 1
ORDER BY 1;
```

Checking Email Engagement:

Using the *yammer_emails* table:

```
SELECT
 DATE_TRUNC('week', categorized_actions.occurred_at) AS "week",
 COUNT(CASE WHEN action_category = 'email open' THEN 1 END) AS email_open,
 COUNT(CASE WHEN action_category = 'email clickthrough' THEN 1 END) AS email_clickthrough,
 COUNT(CASE WHEN action_category = 'weekly digest' THEN 1 END) AS sent_weekly_digest,
 COUNT(CASE WHEN action_category = 'sent_reengagement_email' THEN 1 END) AS
sent_reengagement_email
FROM (
 SELECT
  occurred_at,
  CASE
   WHEN action LIKE 'email_open' THEN 'email open'
   WHEN action LIKE 'email_clickthrough' THEN 'email clickthrough'
   WHEN action LIKE 'sent_weekly_digest' THEN 'weekly digest'
   WHEN action LIKE 'sent_reengagement_email' THEN 'sent_reengagement_email'
   ELSE 'other' -- In case there's a device that doesn't fit into these categories
  END AS action_category
 FROM tutorial.yammer_emails
) AS categorized_actions
GROUP BY 1
ORDER BY 1;
```

Checking Email KPIs:

Using the *yammer_emails* table:

```
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
 LIMIT 30
```