Query 1

```
/* get a general idea of what each table contain
SELECT *
FROM tutorial.yammer_users
SELECT *
FROM  tutorial.yammer_events
SELECT *
FROM tutorial.yammer_emails
SELECT *
FROM benn.dimension_rollup_periods, probabily won't use this
*/

/*let's see all users vs new signups regardless of engagements*/
SELECT DATE_TRUNC('day', u.created_at) AS day,
       COUNT(DISTINCT u.user_id) AS daily_all_users,
       COUNT(DISTINCT CASE WHEN u.activated_at IS NOT NULL THEN u.user_id ELSE NULL
END) AS daily_activated_users
  FROM tutorial.yammer_users u
  WHERE u.created_at >= '2014-04-28' --Monday
   AND u.created_at < '2014-08-25'   --Another Monday, got the period right
 GROUP BY 1
 ORDER BY 1
```

Query 2

```
/*calculate user_age: engagement time - activated time
```

```
caculate each users age at each week period, and divide into categories,
for each time period, calculate the number for each category
*/

SELECT  DATE_TRUNC('week', joined_ue.occurred_at) AS week,
      COUNT (DISTINCT CASE WHEN joined_ue.user_age <= INTERVAL '1 week' THEN
joined_ue.user_id ELSE NULL END) AS age_1week,
      COUNT (DISTINCT CASE WHEN (joined_ue.user_age > INTERVAL '1 week'  AND
joined_ue.user_age <= INTERVAL '2 weeks') THEN joined_ue.user_id ELSE NULL END) AS
age_2weeks,
      COUNT (DISTINCT CASE WHEN (joined_ue.user_age > INTERVAL '2 weeks'  AND
joined_ue.user_age <= INTERVAL '3 weeks') THEN joined_ue.user_id ELSE NULL END) AS
age_3weeks,
      COUNT (DISTINCT CASE WHEN (joined_ue.user_age > INTERVAL '3 weeks'  AND
joined_ue.user_age <= INTERVAL '4 weeks') THEN joined_ue.user_id ELSE NULL END) AS
age_4weeks,
      COUNT (DISTINCT CASE WHEN (joined_ue.user_age > INTERVAL '4 weeks'  AND
joined_ue.user_age <= INTERVAL '5 weeks') THEN joined_ue.user_id ELSE NULL END) AS
age_5weeks,
      COUNT (DISTINCT CASE WHEN (joined_ue.user_age > INTERVAL '5 weeks'  AND
joined_ue.user_age <= INTERVAL '6 weeks') THEN joined_ue.user_id ELSE NULL END) AS
age_6weeks,
      COUNT (DISTINCT CASE WHEN (joined_ue.user_age > INTERVAL '6 weeks'  AND
joined_ue.user_age <= INTERVAL '7 weeks') THEN joined_ue.user_id ELSE NULL END) AS
age_7weeks,
      COUNT (DISTINCT CASE WHEN (joined_ue.user_age > INTERVAL '7 weeks'  AND
joined_ue.user_age <= INTERVAL '8 weeks') THEN joined_ue.user_id ELSE NULL END) AS
age_8weeks,
      COUNT (DISTINCT CASE WHEN (joined_ue.user_age > INTERVAL '8 weeks'  AND
joined_ue.user_age <= INTERVAL '9 weeks') THEN joined_ue.user_id ELSE NULL END) AS
age_9weeks,
      COUNT (DISTINCT CASE WHEN (joined_ue.user_age > INTERVAL '9 weeks'  AND
joined_ue.user_age <= INTERVAL '10 weeks') THEN joined_ue.user_id ELSE NULL END) AS
age_10weeks,
      COUNT (DISTINCT CASE WHEN (joined_ue.user_age > INTERVAL '10 weeks') THEN
joined_ue.user_id ELSE NULL END) AS age_morethan10weeks

FROM

(SELECT '2014-08-25'::timestamp - u.activated_at AS user_age,
      u.activated_at,
      e.occurred_at,
      u.user_id
FROM tutorial.yammer_users u

INNER JOIN tutorial.yammer_events e
ON u.user_id = e.user_id
AND   e.event_type = 'engagement'
AND   e.occurred_at >= '2014-04-28'
AND   e.occurred_at < '2014-08-25'
AND   u.activated_at is NOT NULL) joined_ue
GROUP BY 1
ORDER BY 1
```

Query 3

```
/*let's see any difference across different login devices*/
/*-- first check how many devices are contained in the device column
```

```
SELECT DISTINCT device
FROM tutorial.yammer_events
--gives me 26 sets of devices
*/
SELECT DATE_TRUNC('week', joined_ue.occurred_at) AS week,
      COUNT(DISTINCT joined_ue.user_id) AS all_users,
      COUNT(DISTINCT CASE WHEN joined_ue.device IN ('macbook pro', 'acer aspire
notebook','acer aspire desktop', 'lenovo thinkpad','mac mini','dell inspiron desktop', 'dell
inspiron notebook','windows surface','macbook air', 'asus chromebook', 'hp pavilion
desktop' )
       THEN joined_ue.user_id ELSE NULL END) AS computer_users,
      COUNT(DISTINCT CASE WHEN joined_ue.device IN ('kindle fire', 'ipad mini','nexus
7','nexus 10','samsumg galaxy tablet','nexus 5','ipad air')
       THEN joined_ue.user_id ELSE NULL END) AS tablet_users,
      COUNT(DISTINCT CASE WHEN joined_ue.device IN ('iphone 5s','samsung galaxy
note','nokia lumia 635','amazon fire phone','iphone 4s','htc one','iphone 5','samsung galaxy
s4')
        THEN joined_ue.user_id ELSE NULL END) AS phone_users

FROM
(SELECT e.device,
       e.occurred_at,
       u.user_id
FROM tutorial.yammer_users u
INNER JOIN tutorial.yammer_events e
ON u.user_id = e.user_id
AND   e.event_type = 'engagement'
AND   e.occurred_at >= '2014-04-28'
AND   e.occurred_at < '2014-08-25'
AND   u.activated_at is NOT NULL) joined_ue
GROUP BY 1
ORDER BY 1
```

Query 4

```
/*now check the email interactions */
/*--first look at the dataset
SELECT DISTINCT action
```

```
FROM tutorial.yammer_emails
--gives 4 categories of actions
*/
SELECT DATE_TRUNC('week', occurred_at) AS week,
    COUNT(DISTINCT CASE WHEN action IN
('sent_weekly_digest','sent_reengagement_email') THEN user_id ELSE NULL END) AS
sent_total,
    COUNT(DISTINCT CASE WHEN action IN ('email_open','email_clickthrough') THEN
user_id ELSE NULL END) AS reponded_total,
    COUNT(DISTINCT CASE WHEN action = 'sent_weekly_digest' THEN user_id ELSE NULL
END) AS sent_digest,
    COUNT(DISTINCT CASE WHEN action = 'sent_reengagement_email' THEN user_id ELSE
NULL END) AS sent_reengagement,
    COUNT(DISTINCT CASE WHEN action = 'email_open' THEN user_id ELSE NULL END) AS
email_open,
    COUNT(DISTINCT CASE WHEN action = 'email_clickthrough' THEN user_id ELSE NULL
END) AS email_clickthrough

FROM tutorial.yammer_emails
WHERE occurred_at >= '2014-04-28'
AND   occurred_at < '2014-08-25'
GROUP BY 1
ORDER BY 1
```

Query 5

```
/*investigate the rate of response
from weekly digest emails and reengagement emails separately
*/
/*SELECT *
FROM tutorial.yammer_emails
-- after initially checking the dataset, we could find most of the emails are opened
within 1 minute, and clickedthrough within another minute */

SELECT allem.week,
      allem.digest_open/(CASE WHEN allem.sent_digest != 0 THEN allem.sent_digest ELSE
NULL END)::float AS digest_open_rate,
      allem.digest_click/(CASE WHEN allem.sent_digest != 0 THEN allem.sent_digest ELSE
NULL END)::float AS digest_click_rate,
      allem.reengagement_open/(CASE WHEN allem.sent_reengagement != 0 THEN
allem.sent_reengagement ELSE NULL END)::float AS reengagement_open_rate,
      allem.reengagement_click/(CASE WHEN allem.sent_reengagement != 0 THEN
allem.sent_reengagement ELSE NULL END)::float AS reengagement_click_rate
FROM(
SELECT DATE_TRUNC('week',em1.occurred_at) AS week,
       COUNT(CASE WHEN em1.action = 'sent_weekly_digest' THEN em1.user_id ELSE NUll
END) as sent_digest,
       COUNT(CASE WHEN em1.action = 'sent_weekly_digest' THEN em2.user_id ELSE NUll
END) as digest_open,
       COUNT(CASE WHEN em1.action = 'sent_weekly_digest' THEN em3.user_id ELSE NUll
END) as digest_click,
       COUNT(CASE WHEN em1.action = 'sent_reengagement_email' THEN em1.user_id ELSE
NUll END) as sent_reengagement,
       COUNT(CASE WHEN em1.action = 'sent_reengagement_email' THEN em2.user_id ELSE
NUll END) as reengagement_open,
       COUNT(CASE WHEN em1.action = 'sent_reengagement_email' THEN em3.user_id ELSE
NUll END) as reengagement_click
FROM tutorial.yammer_emails em1
LEFT JOIN tutorial.yammer_emails em2  --keep all emails sent
     ON  em2.user_id = em1.user_id
     AND em2.action ='email_open'
     AND em2.occurred_at > em1.occurred_at
     AND em2.occurred_at <= em1.occurred_at + INTERVAL '1 min'
LEFT JOIN tutorial.yammer_emails em3
      ON em3.user_id = em2.user_id
     AND em3.action = 'email_clickthrough'
     AND em3.occurred_at > em2.occurred_at
     AND em3.occurred_at <= em2.occurred_at + INTERVAL '1 min'
WHERE em1.occurred_at >= '2014-04-28'
  AND em1.occurred_at < '2014-08-25'
  AND em1.action in ('sent_weekly_digest','sent_reengagement_email')
  GROUP BY 1
     ) allem
ORDER BY 1
```