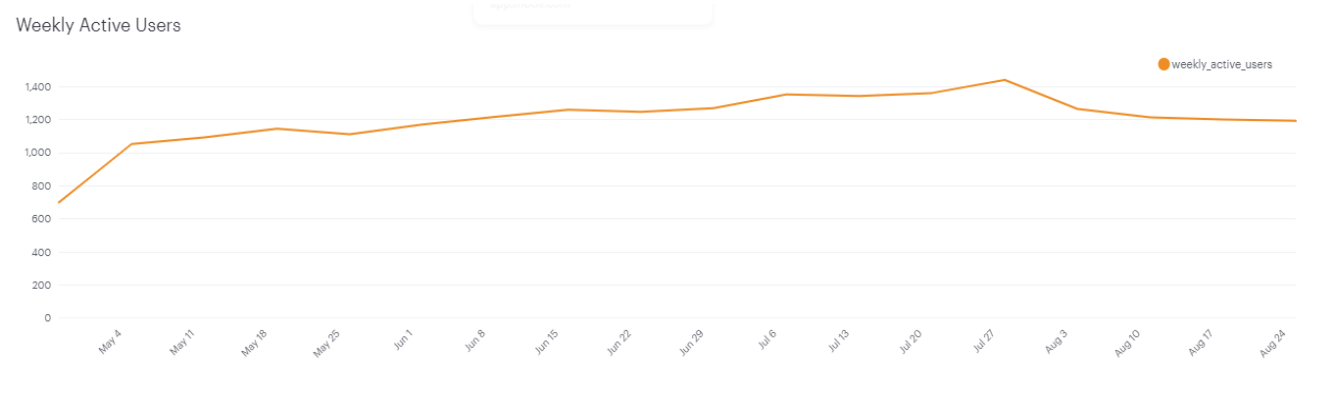
Investigating a Drop in User Engagement

The problem

You show up to work Tuesday morning, September 2, 2014. The head of the Product team walks over to your desk and asks you what you think about the latest activity on the user engagement dashboards. You fire them up, and something immediately jumps out:

```
SELECT DATE_TRUNC('week', e.occurred_at),
       COUNT(DISTINCT e.user_id) AS weekly_active_users
  FROM tutorial.yammer_events e
 WHERE e.event_type = 'engagement'
   AND e.event_name = 'login'
 GROUP BY 1
 ORDER BY 1
```



Any point in this chart can be interpreted as "the number of users who logged at least one engagement event during the week starting on that date."

You are responsible for determining what caused the dip at the end of the chart shown above and, if appropriate, recommending solutions for the problem.

Getting oriented

Before you even touch the data, come up with a list of possible causes for the dip in retention shown in the chart above. Make a list and determine the order in which you will check them. Make sure to note how you will test each hypothesis. Think carefully about the criteria you use to order them and write down the criteria as well.

Also, make sure you understand what the above chart shows and does not show.

- Holiday:
- Broken feature
- Broken tracking code:
- Traffic anomalies from bots
- Traffic shutdown to your site
- Bad data:
- Search crawler

Digging in

Once you have an ordered list of possible problems, it's time to investigate.

For this problem, you will need to use four tables. The tables names and column definitions are listed below—click a table name to view information about that *table.* Note: this data is fake and was generated for the purpose of this case study. It is similar in structure to Yammer's actual data, but for privacy and security reasons it is not real.

SELECT * FROM tutorial.yammer_users
SELECT * FROM tutorial.yammer_events
SELECT * FROM tutorial.yammer_emails

Table 1: Users

This table includes one row per user, with descriptive information about that user's account.

This table name in Mode is tutorial.yammer_users

user_id:	A unique ID per user. Can be joined to user_id in either of the other tables.
created_at:	The time the user was created (first signed up)
state:	The state of the user (active or pending)
activated_at:	The time the user was activated, if they are active
company_id:	The ID of the user's company
language:	The chosen language of the user

Table 2: Events

This table includes one row per event, where an event is an action that a user has taken on Yammer. These events include login events, messaging events, search events, events logged as users progress through a signup funnel, events around received emails.

This table name in Mode is tutorial.yammer_events

user_id:	The ID of the user logging the event. Can be joined to user_id in either of the other tables.
occurred_at:	The time the event occurred.
event_type:	The general event type. There are two values in this dataset: "signup_flow", which refers to anything occuring during the process of a user's authentication, and "engagement", which refers to general product usage after the user has signed up for the first time.

event_name:	The specific action the user took. Possible values include: create_user: User is added to Yammer's database during signup process enter_email: User begins the signup process by entering her email address enter_info: User enters her name and personal information during signup process complete_signup: User completes the entire signup/authentication process home_page: User loads the home page like_message: User likes another user's message login: User logs into Yammer search_autocomplete: User selects a search result from the autocomplete list search_run: User runs a search query and is taken to the search results page search_click_result_X: User clicks search result X on the results page, where X is a number from 1 through 10. send_message: User posts a message view_inbox: User views messages in her inbox
location:	The country from which the event was logged (collected through IP address).
device:	The type of device used to log the event.

Table 3: Email Events

This table contains events specific to the sending of emails. It is similar in structure to the events table above.

This table name in Mode is tutorial.yammer_emails

user_id:	The ID of the user to whom the event relates. Can be joined to user_id in either of the other tables.
occurred_at:	The time the event occurred.
action:	The name of the event that occurred. "sent_weekly_digest" means that the user was delivered a digest email showing relevant conversations from the previous day. "email_open" means that the user opened the email. "email_clickthrough" means that the user clicked a link in the email.

Table 4: Rollup Periods

The final table is a lookup table that is used to create rolling time periods. Though you could use the INTERVAL() function, creating rolling time periods is often easiest with a table like this. You won't necessarily need to use this table in queries that you write, but the column descriptions are provided here so that you can understand the query that creates the chart shown above.

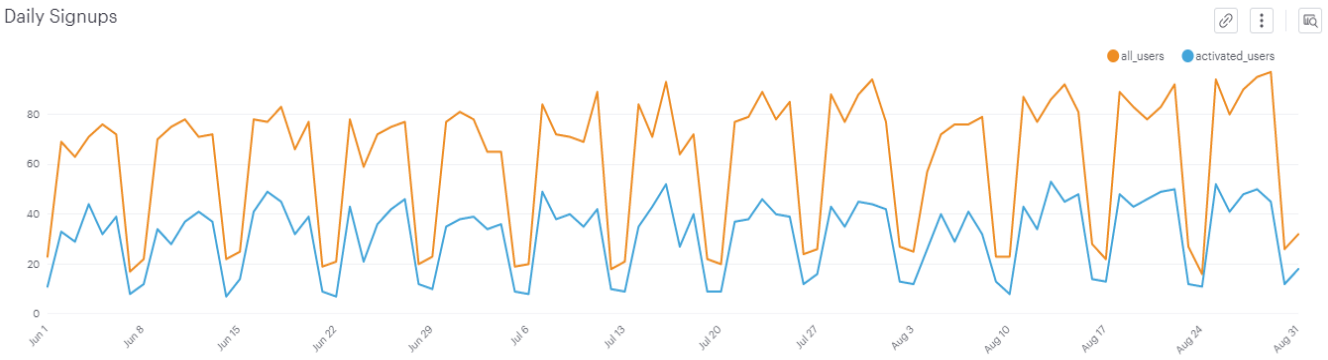
This table name in Mode is benn.dimension_rollup_periods

period_id:	This identifies the type of rollup period. The above dashboard uses period 1007, which is rolling 7-day periods.
time_id:	This is the identifier for any given data point — it's what you would put on a chart axis. If time_id is 2014-08-01, that means that is represents the rolling 7-day period leading up to 2014-08-01.
pst_start:	The start time of the period in PST. For 2014-08-01, you'll notice that this is 2014-07-25 — one week prior. Use this to join events to the table.
pst_end:	The start time of the period in PST. For 2014-08-01, the end time is 2014-08-01. You can see how this is used in conjunction with pst_start to join events to this table
utc_start:	The same as pst_start, but in UTC time.
pst_start:	The same as pst_end, but in UTC time.

1. One of the easiest things to check is growth, both because it's easy to measure and because most companies (Yammer included) track this closely already. In this case, you have to make it yourself, though. You'll notice that nothing has really changed about the growth rate—it continues to be high during the week, low on weekends:

```
SELECT DATE_TRUNC('day',created_at) AS day,
```

```
       COUNT(*) AS all_users,
       COUNT(CASE WHEN activated_at IS NOT NULL THEN u.user_id ELSE NULL END)
AS activated_users
  FROM tutorial.yammer_users u
 WHERE created_at >= '2014-06-01'
   AND created_at < '2014-09-01'
 GROUP BY 1
 ORDER BY 1
```

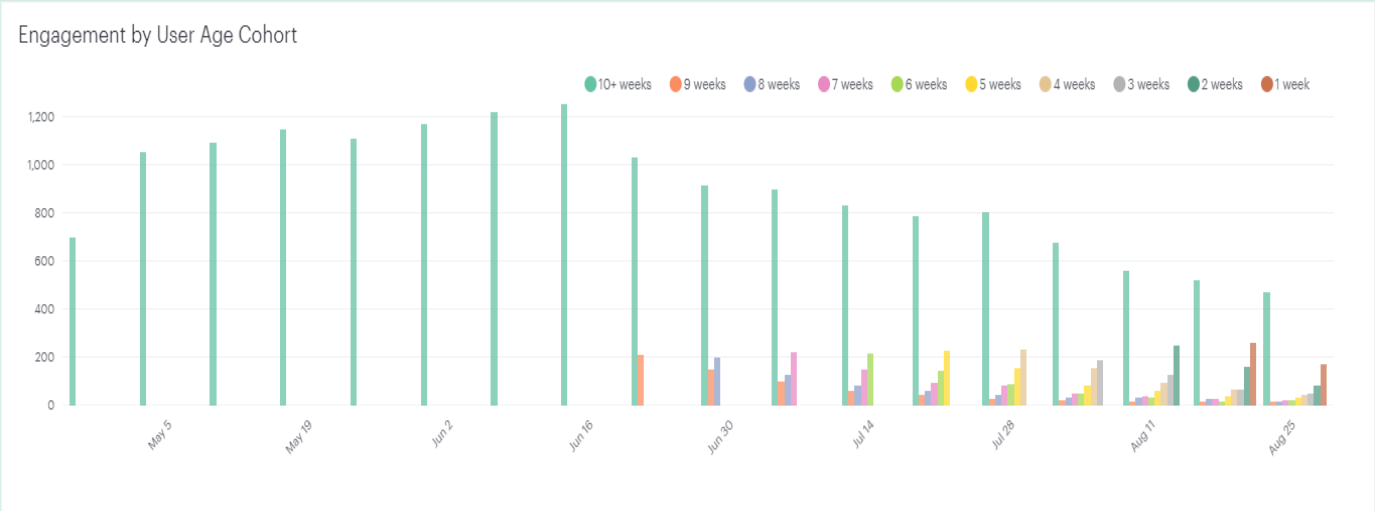


2. Since growth is normal, it's possible that the dip in engagement is coming from existing users as opposed to new ones. One of the most effective ways to look at this is to cohort users based on when they signed up for the product. This chart shows a decrease in engagement among users who signed up more than 10 weeks prior:

```
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",
```

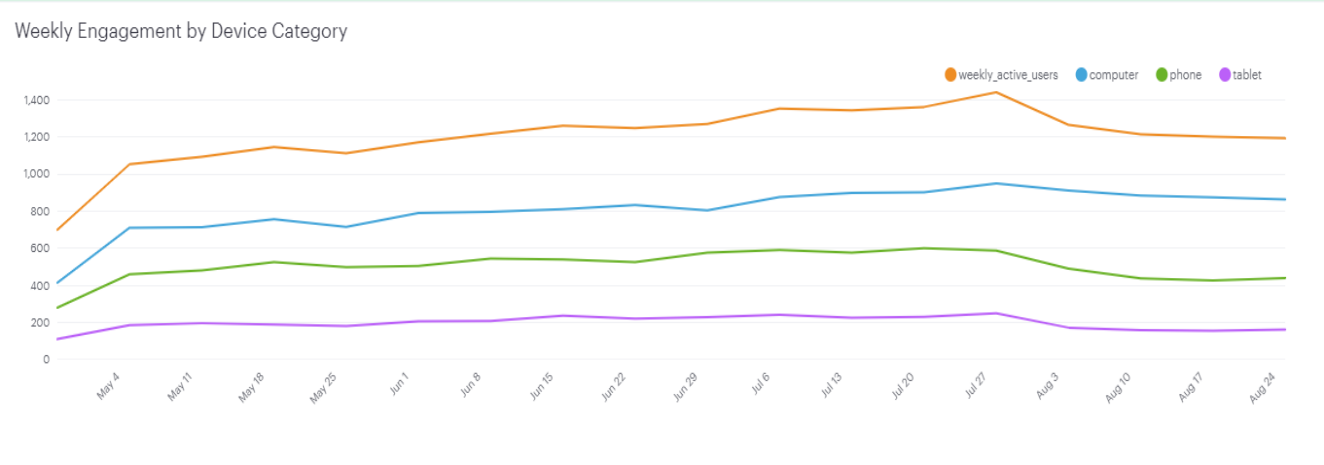
```
      COUNT(DISTINCT CASE WHEN z.user_age < 35 AND z.user_age >= 28 THEN z.user_id
ELSE NULL END) AS "4 weeks",
      COUNT(DISTINCT CASE WHEN z.user_age < 28 AND z.user_age >= 21 THEN z.user_id
ELSE NULL END) AS "3 weeks",
      COUNT(DISTINCT CASE WHEN z.user_age < 21 AND z.user_age >= 14 THEN 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
```



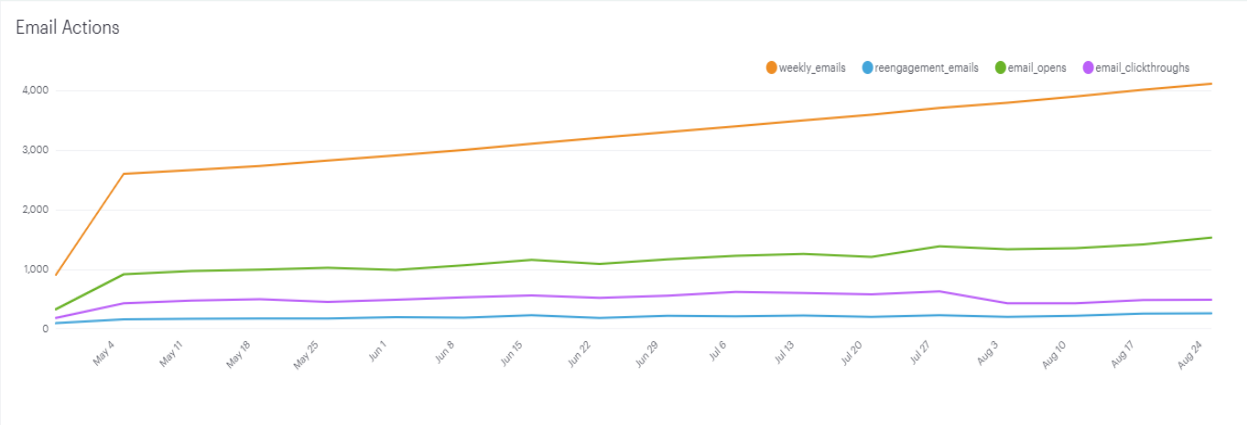
The understanding that the problem is localized to older users leads us to believe that the problem probably isn't related to a one-time spike from marketing traffic or something that is affecting new traffic to the site like being blocked or changing rank on search engines. Now let's take a look at various device types to see if the problem is localized to any particular product:

```
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',
       'samsumg 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
```



If you filter the above chart down to phones (double-click the dot next to "phone" in the legend), you will see that there's a pretty steep drop in phone engagement rates. So it's likely that there's a problem with the mobile app related to long-time user retention. At this point, you're in a good position to ask around and see if anything changed recently with the mobile app to try to figure out the problem. You might also think about what causes people to engage with the product. The purpose of the digest email mentioned above is to bring users back into the product. Since we know this problem relates to the retention of long-time users, it's worth checking out whether the email has something to do with it:

```
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
```



Follow through

After investigation, it appears that the problem has to do with mobile use and digest emails. The intended action here should be clear: notify the head of product (who approached you in the first place) that the problem is localized in these areas and that it's worth checking to make sure something isn't broken or poorly implemented. It's not clear from the data *exactly* what the problem is or how it should be solved, but the above work can save other teams a lot of time in figuring out where to look.