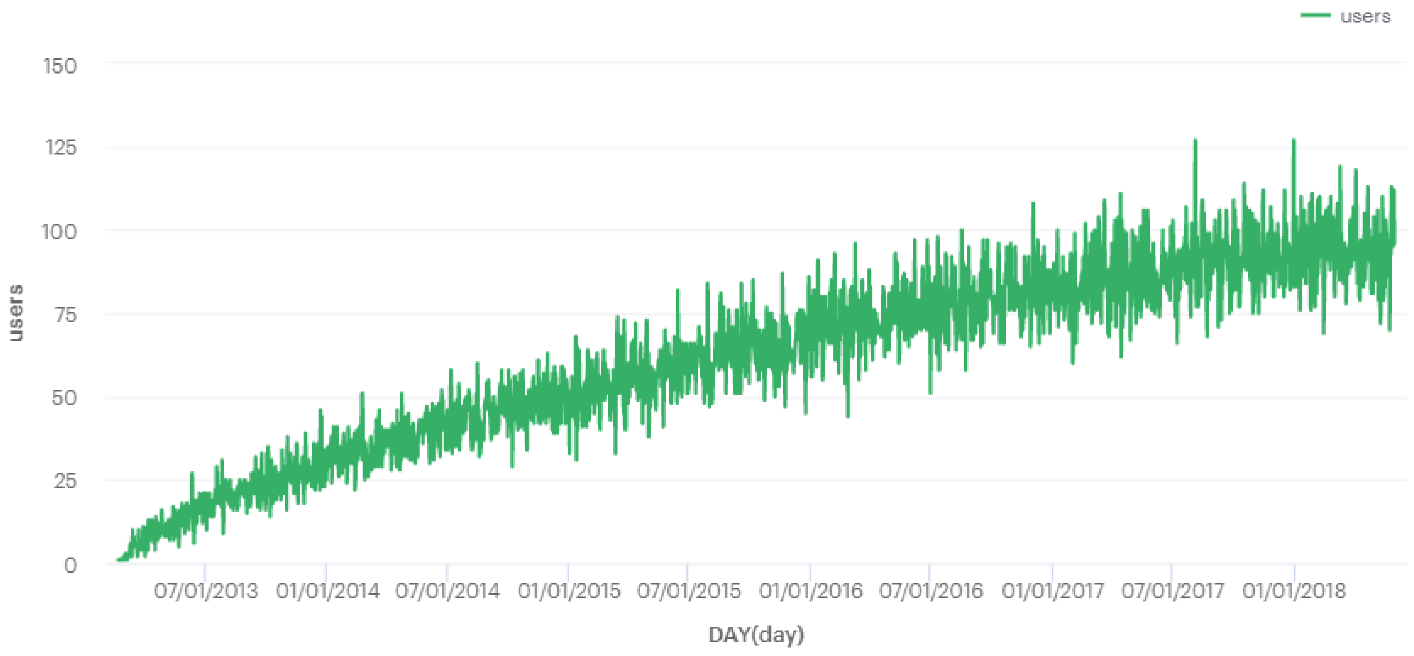
Counting Users

Exercise 1: We'll be using the users table to answer the question "How many new users are added each day?". Start by making sure you understand the columns in the table.

```
select * from dsv1069.users
```

Exercise 2: WIthout worrying about deleted user or merged users, count the number of users added each day.

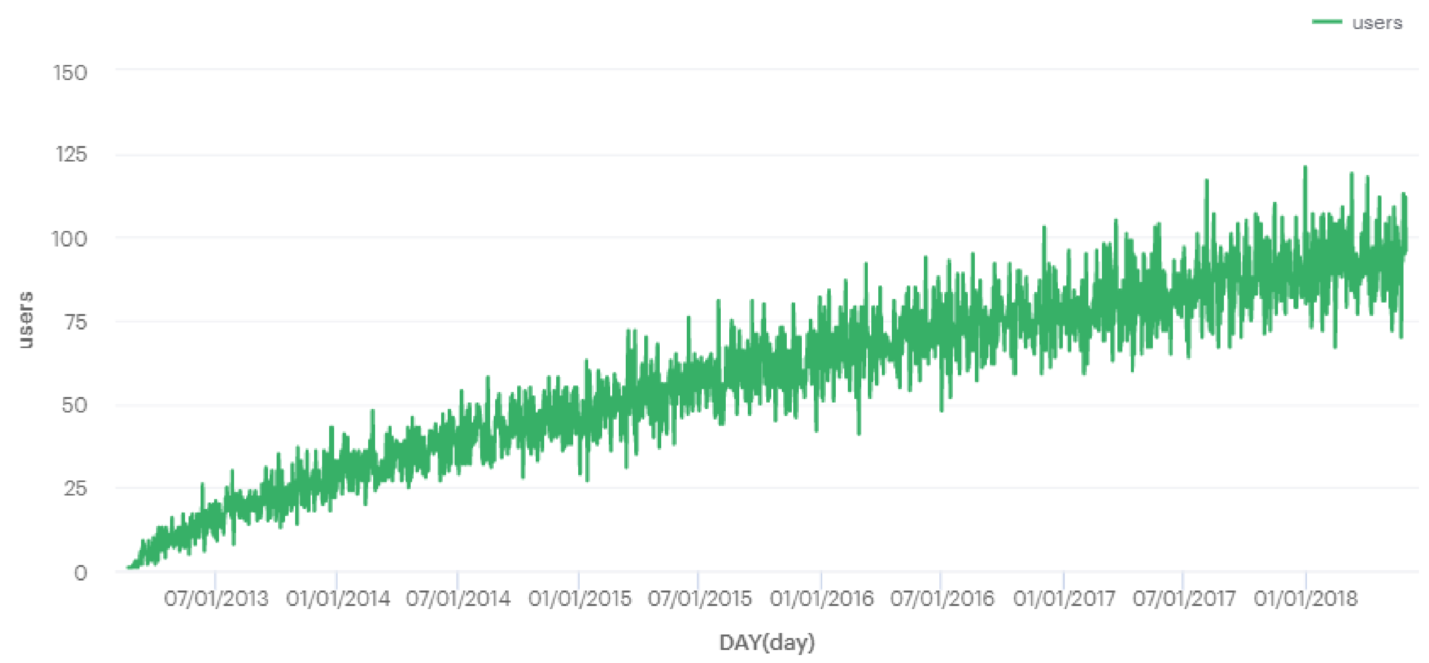
```
select
  date(created_at) as day,
  count(*)         as users
from
  dsv1069.users
group by
  date(created_at)
```



Exercise 3: Consider the following query. Is this the right way to count merged or deleted users? If all of our users were deleted tomorrow what would the result look like?

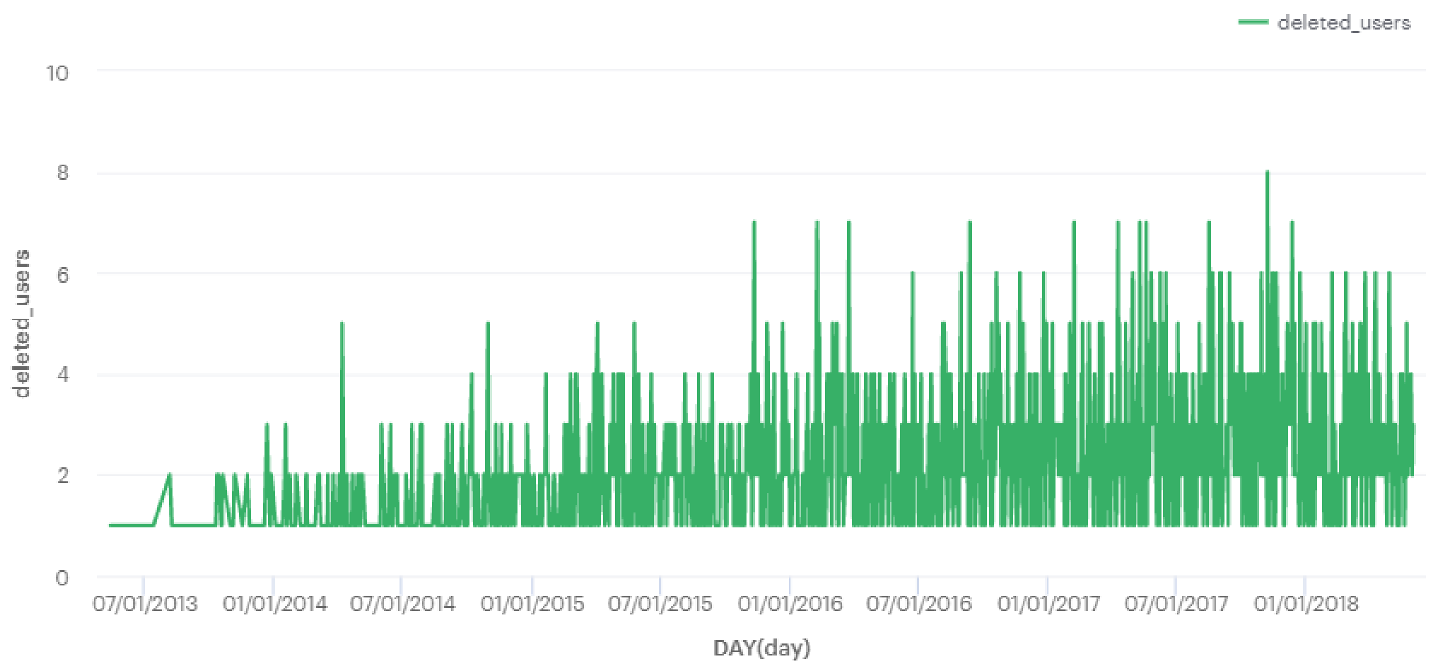
```
select
  date(created_at) as day,
  count(*)         as users
from
  dsv1069.users
where
  deleted_at is null
and
  (id <> parent_user_id or parent_user_id is null)
group by
  date(created_at)
```

Yes.

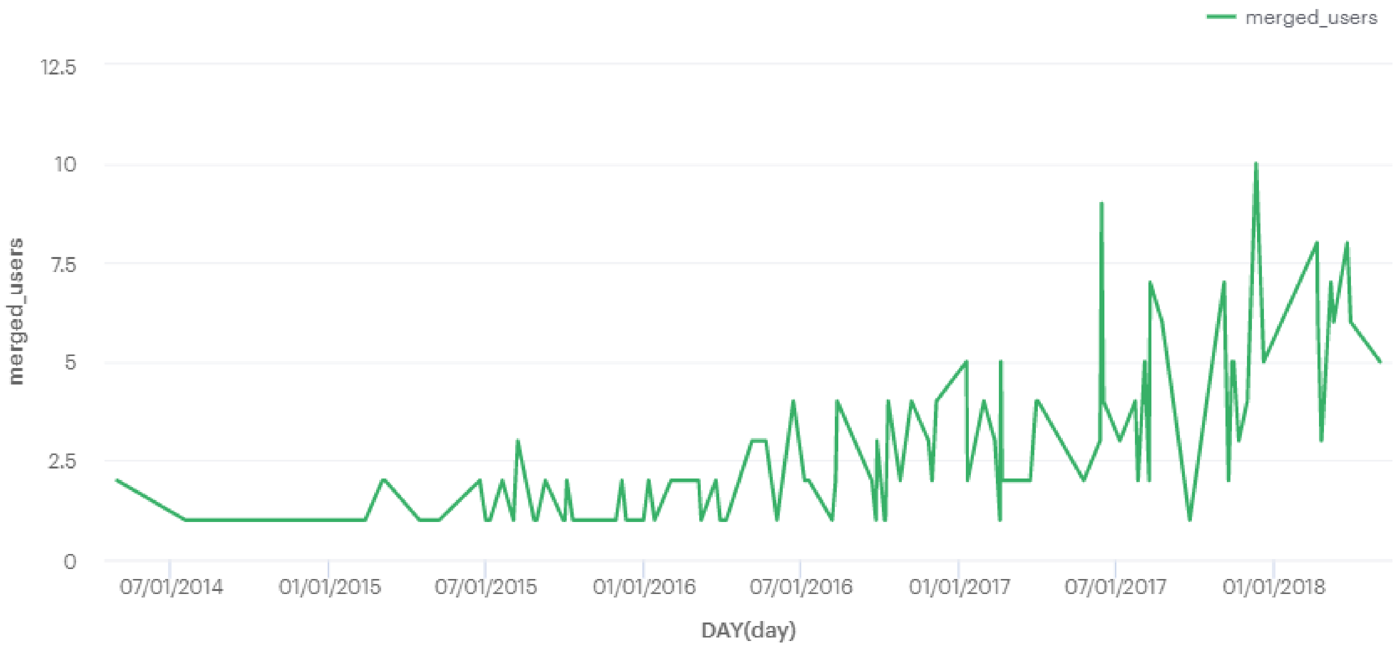


Exercise 4: Count the number of users deleted each day. Then count the number of users removed due to merging in a similar way.

```
select
  date(deleted_at) as day,
  count(*)         as deleted_users
from
  dsv1069.users
where
  deleted_at is not null
group by
  date(deleted_at)
```



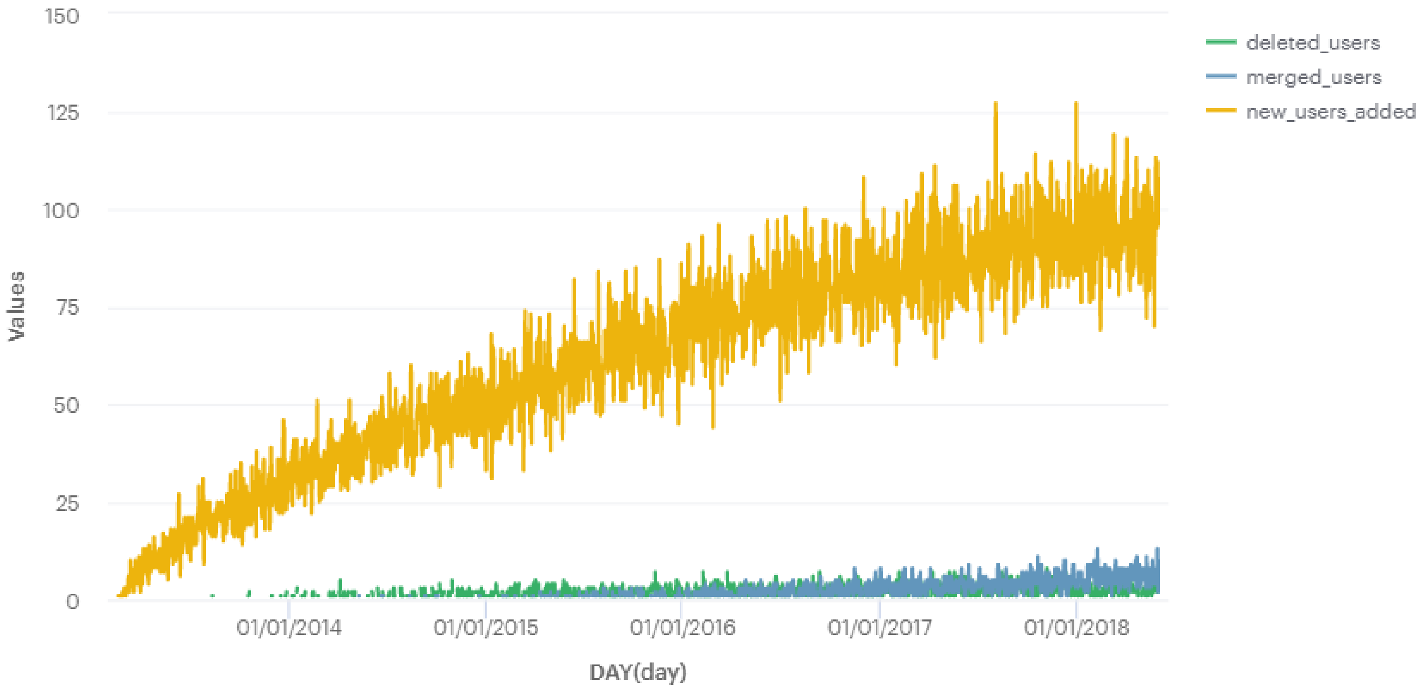
```sql
select
    date(merged_at) as day,
    count(*)          as merged_users
  from
    dsv1069.users
  where
    id <> parent_user_id
  and
    parent_user_id is not null
  group by
    date(merged_at)
```



Exercise 5: Use the pieces you've built as sub-tables and create a table that has a column for the date, the number of users created, the number of users deleted and the number of users merged that day.

```sql
SELECT
  new.day,
  new.new_users_added,
  deleted.deleted_users,
  merged.merged_users
FROM
  (SELECT
    date(created_at) AS day,
    COUNT(*)         AS new_users_added
  FROM
    dsv1069.users
  GROUP BY
    date(created_at)
  ) new
LEFT JOIN
  (SELECT
    date(deleted_at) AS day,
    COUNT(*)         AS deleted_users
  FROM
    dsv1069.users
  WHERE
```

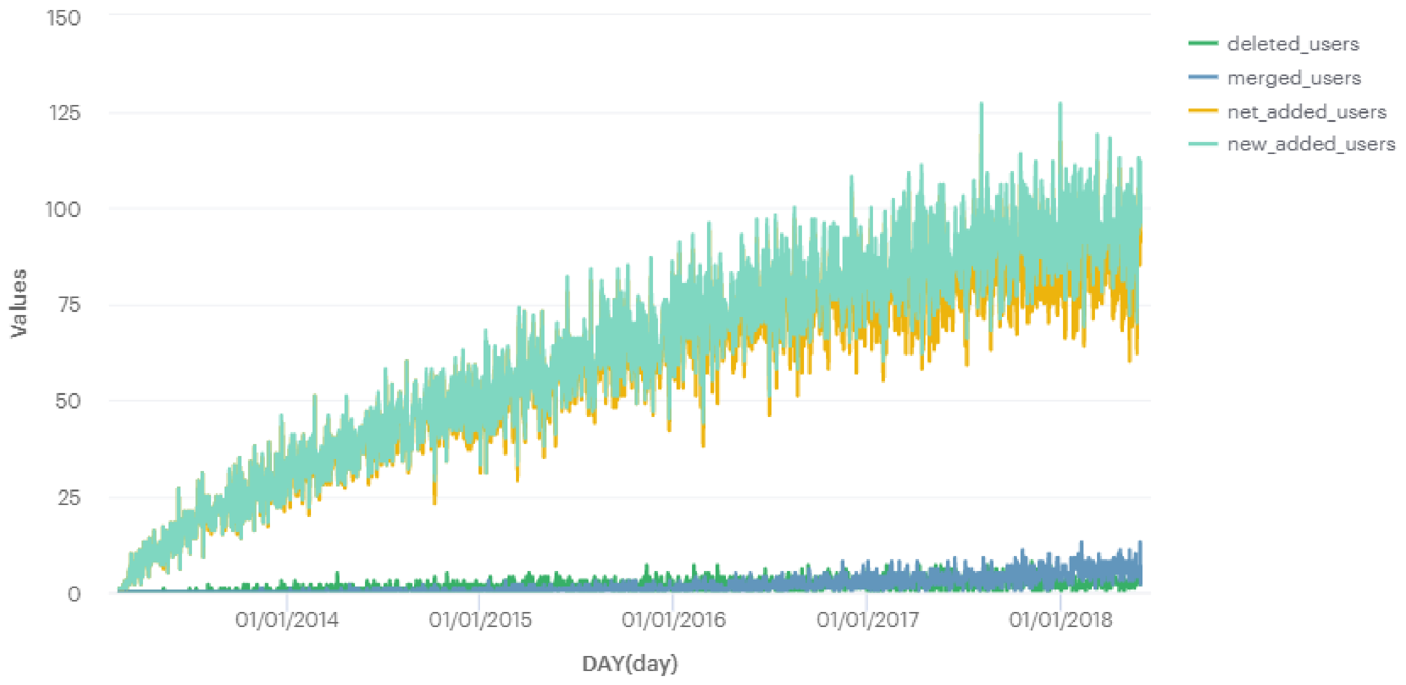
```
    deleted_at IS NOT NULL
  GROUP BY
    date(deleted_at)
  ) deleted
ON deleted.day = new.day
LEFT JOIN
  (SELECT
    date(merged_at) AS day,
    COUNT(*)         AS merged_users
  FROM
    dsv1069.users
  WHERE
    id <> parent_user_id
  AND
    parent_user_id IS NOT NULL
  GROUP BY
    date(merged_at)
  ) merged
ON merged.day = new.day
```



Exercise 6: Refine your query from #5 to have informative column names and so that null columns return 0.

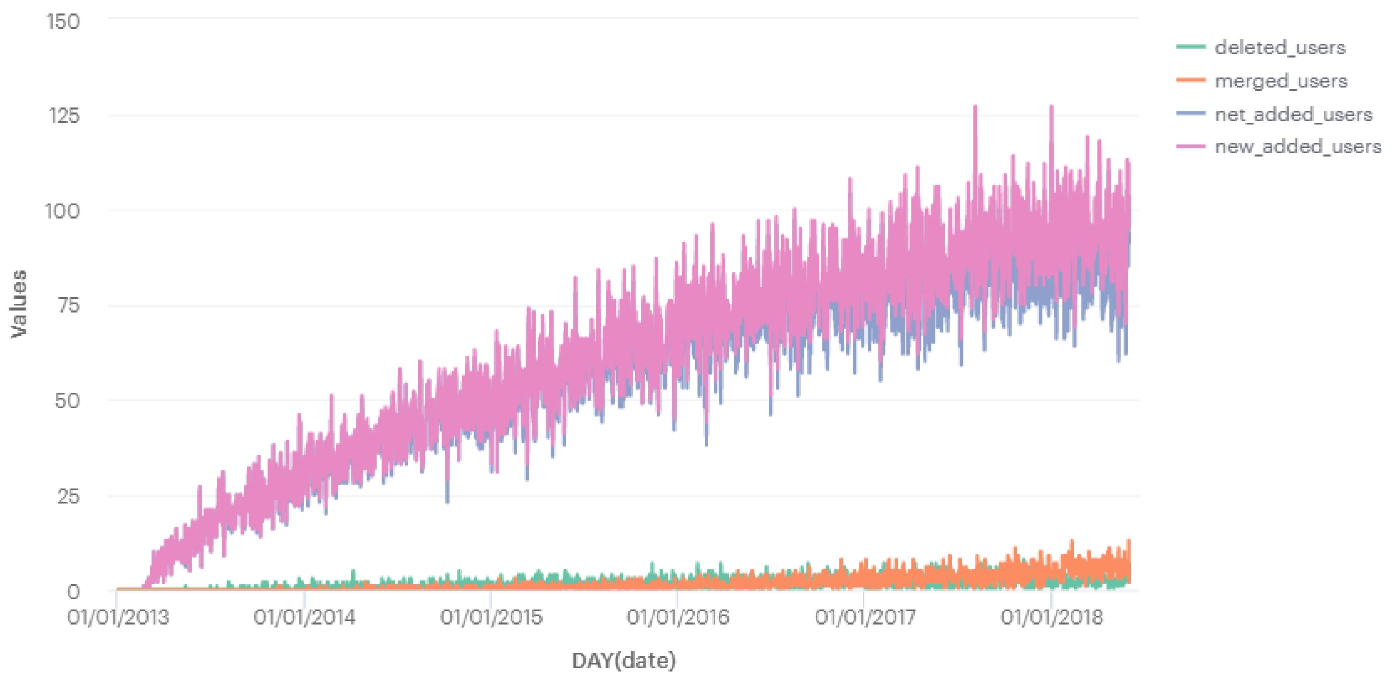
```
SELECT
  new.day,
  new.new_added_users,
  COALESCE(deleted.deleted_users,0) AS deleted_users,
  COALESCE(merged.merged_users,0)   AS merged_users,
  (new.new_added_users - COALESCE(deleted.deleted_users,0)- COALESCE(merged.merged_users,0))
    AS net_added_users
FROM
  (SELECT
    date(created_at) AS day,
    COUNT(*)         AS new_added_users
  FROM
    dsv1069.users
  GROUP BY
```

```
    date(created_at)
  ) new
LEFT OUTER JOIN
  (SELECT
    date(deleted_at) AS day,
    COUNT(*)         AS deleted_users
  FROM
    dsv1069.users
  WHERE
    deleted_at IS NOT NULL
  GROUP BY
    date(deleted_at)
  ) deleted
ON deleted.day = new.day
LEFT OUTER JOIN
  (SELECT
    date(merged_at) AS day,
    COUNT(*)        AS merged_users
  FROM
    dsv1069.users
  WHERE
    merged_at IS NOT NULL
  AND
    id <> parent_user_id
  GROUP BY
    date(merged_at)
  ) merged
ON
  merged.day = new.day
```



Exercise 7: What if there were days where no users were created, but some users were deleted or merged. Does the previous query still work? No, it doesn't. Use the dates_rollup as a backbone for this query, so that we won't miss any dates.

```sql
SELECT
  dates_rollup.date,
  new.new_added_users,
  COALESCE(deleted.deleted_users,0) AS deleted_users,
  COALESCE(merged.merged_users,0)   AS merged_users,
  (new.new_added_users - COALESCE(deleted.deleted_users,0)- COALESCE(merged.merged_users,0))
    AS net_added_users
FROM
  dsv1069.dates_rollup
LEFT OUTER JOIN
  (SELECT
    date(created_at) AS day,
    COUNT(*)         AS new_added_users
  FROM
    dsv1069.users
  GROUP BY
    date(created_at)
  ) new
ON
  new.day = date(dates_rollup.date)
LEFT OUTER JOIN
  (SELECT
    date(deleted_at) AS day,
    COUNT(*)         AS deleted_users
  FROM
    dsv1069.users
  WHERE
    deleted_at IS NOT NULL
  GROUP BY
    date(deleted_at)
  ) deleted
ON deleted.day = date(dates_rollup.date)
LEFT OUTER JOIN
  (SELECT
    date(merged_at) AS day,
    COUNT(*)         AS merged_users
  FROM
    dsv1069.users
  WHERE
    merged_at IS NOT NULL
  AND
    id <> parent_user_id
  GROUP BY
    date(merged_at)
  ) merged
ON
 merged.day = date(dates_rollup.date)
```


150
125
100
75
50
25
0
Values
01/01/2013
01/01/2014
01/01/2015
01/01/2016
01/01/2017
01/01/2018
DAY(date)
deleted_users
merged_users
net_added_users
new_added_users