Counting Users

Log onto Mode Analytics and from the home page, create a new report by clicking on the green plus sign button in the upper right-hand corner. Enter the starter code where provided for each exercise. You may want to create a new tab for each exercise.

Please use the discussion forum for any questions and/or comments you might have. Once you have tried the exercises, feel free to watch the solutions video. Good luck with your practice!

Note: *When querying a table, remember to prepend dsv1069, which is the schema, or folder that contains the course data.*

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.

Starter Code:

```
SELECT  * FROM dsv1069.users
```

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

Starter Code: (none)

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?

Starter Code:

```
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)
```

Exercise 4: Count the number of users deleted each day. Then count the number of users removed due to merging in a similar way.
Starter Code: (Use the result from #2 as a guide)

Exercise 5: Use the pieces you've built as subtables 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.

Starter Code:
(none)

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

Starter Code: (none)

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.

Starter Code:
SELECT * FROM dsv1069.dates_rollup