Create a Test Metric

Exercise 1:

-- Using the table from Exercise 4.3 and compute a metric that measures
-- Whether a user created an order after their test assignment
-- Requirements: Even if a user had zero orders, we should have a row that counts their number of orders as zero
-- If the user is not in the experiment, they should not be included

```sql
select assignments.user_id,
       assignments.test_id,
       assignments.test_assignment,
       max(case
               when orders.created_at > assignments.event_time then 1
               else 0
           end) as order_binary
from
  ( select event_id,
           event_time,
           user_id,
           max(case
                   when parameter_name = 'test_id' then cast(parameter_value as int)
                   else null
               end) as test_id,
           max(case
                   when parameter_name = 'test_assignment' then cast(parameter_value as int)
                   else null
               end) as test_assignment
   from dsv1069.events
   where event_name = 'test_assignment'
   group by event_id,
            event_time,
            user_id
   order by event_id) assignments
left outer join dsv1069.orders
  on assignments.user_id = orders.user_id
group by assignments.user_id,
         assignments.test_id,
         assignments.test_assignment limit 5
```

Exercise 2:

--Using the table from the previous exercise, add the following metrics
--1) the number of orders/invoices
--2) the number of items/line-items ordered
--3) the total revenue from the order after treatment

```sql
select assignments.user_id,
       assignments.test_id,
       assignments.test_assignment,
       count(distinct case
                          when orders.created_at > assignments.event_time then orders.invoice_id
                          else null
                      end) as invoices,
       count(distinct case
                          when orders.created_at > assignments.event_time then orders.line_item_id
                          else null
                      end) as line_items,
       coalesce(sum(case
                        when orders.created_at > assignments.event_time then orders.price
                        else 0
                    end), 0) as total_revenue
from
  ( select event_id,
           event_time,
           user_id,
           max(case
                   when parameter_name = 'test_id' then cast(parameter_value as int)
                   else null
               end) as test_id,
           max(case
                   when parameter_name = 'test_assignment' then cast(parameter_value as int)
                   else null
               end) as test_assignment
   from dsv1069.events
   where event_name = 'test_assignment'
   group by event_id,
            event_time,
            user_id
   order by event_id) as assignments
left outer join dsv1069.orders
  on assignments.user_id = orders.user_id
group by assignments.user_id,
         assignments.test_id,
         assignments.test_assignment
```

Mode Report Link: https://app.mode.com/sum14/reports/8d8894297b27