

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

```
1. select assignments.user_id,
2.        assignments.test_id,
3.        assignments.test_assignment,
4.        max(case
5.            when orders.created_at > assignments.event_time then 1
6.            else 0
7.        end) as order_binary
8. from
9.     ( select event_id,
10.            event_time,
11.            user_id,
12.            max(case
13.                when parameter_name = 'test_id' then cast(parameter_value as int)
14.                else null
15.            end) as test_id,
16.            max(case
17.                when parameter_name = 'test_assignment' then cast(parameter_value as int)
18.                else null
19.            end) as test_assignment
20.     from dsv1069.events
21.     where event_name = 'test_assignment'
22.     group by event_id,
23.            event_time,
24.            user_id
25.     order by event_id) assignments
26. left outer join dsv1069.orders
27. on assignments.user_id = orders.user_id
28. group by assignments.user_id,
29.        assignments.test_id,
30.        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

```
1. select assignments.user_id,
2.        assignments.test_id,
3.        assignments.test_assignment,
4.        count(distinct case
5.            when orders.created_at > assignments.event_time then orders.invoice_id
6.            else null
7.        end) as invoices,
8.        count(distinct case
9.            when orders.created_at > assignments.event_time then orders.line_item_id
10.           else null
11.        end) as line_items,
12.        coalesce(sum(case
13.            when orders.created_at > assignments.event_time then orders.price
14.            else 0
15.        end), 0) as total_revenue
16. from
17.     ( select event_id,
18.            event_time,
```

```
19.         user_id,
20.         max(case
21.             when parameter_name = 'test_id' then cast(parameter_value as int)
22.             else null
23.         end) as test_id,
24.         max(case
25.             when parameter_name = 'test_assignment' then cast(parameter_value as int)
26.             else null
27.         end) as test_assignment
28. from dsv1069.events
29. where event_name = 'test_assignment'
30. group by event_id,
31.          event_time,
32.          user_id
33. order by event_id) as assignments
34. left outer join dsv1069.orders
35. on assignments.user_id = orders.user_id
36. group by assignments.user_id,
37.          assignments.test_id,
38.          assignments.test_assignment
```

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