

## 1. Data Quality Check

```
1  --We are running an experiment at an item-level, which means all users
2  --who visit will see the same page, but the layout of different item
3  --pages may differ. Compare this table to the assignment events we
4  --captured for user_level_testing.
5
6  --Does this table have everything you need to compute metrics like
7  --30-day view-binary?
8  -- No, this table does not have all of the data required to
9  -- compute a 30-day view-binary. Specifically, we will need to
10 -- know the dates of the tests and the orders so we can create
11 -- a 30-day window.
12
13 SELECT
14     *
15 FROM
16     dsv1069.final_assignments_qa;
17
```

## 2. Reformat the Data

```
1  --Reformat the final_assignments_qa to look like the final_assignments
2  --table, filling in any missing values with a placeholder of the
3  --appropriate data type.
4
5  SELECT
6      item_id,
7      test_a AS test_assignment,
8      (CASE
9          WHEN test_a is NOT NULL
10         THEN 'test_a'
11         ELSE NULL
12     END) AS test_number,
13     (CASE
14         WHEN test_a is NOT NULL
15         THEN '2025-01-01 00:00:00'
16         ELSE NULL
17     END) AS test_start_date
18 FROM dsv1069.final_assignments_qa
19
20 UNION
21
22 SELECT
23     item_id,
24     test_b AS test_assignment,
25     (CASE
26         WHEN test_b is NOT NULL
27         THEN 'test_b'
28         ELSE NULL
29     END) AS test_number,
30     (CASE
31         WHEN test_b is NOT NULL
32         THEN '2025-01-01 00:00:00'
33         ELSE NULL
34     END) AS test_start_date
35 FROM dsv1069.final_assignments_qa
36
37 UNION
38
39 SELECT
40     item_id,
41     test_c AS test_assignment,
42     (CASE
43         WHEN test_c is NOT NULL
44         THEN 'test_c'
45         ELSE NULL
```

```

46     END) AS test_number,
47     (CASE
48         WHEN test_c IS NOT NULL
49         THEN '2025-01-01 00:00:00'
50         ELSE NULL
51     END) AS test_start_date
52 FROM dsv1069.final_assignments_qa
53
54 UNION
55
56 SELECT
57     item_id,
58     test_d AS test_assignment,
59     (CASE
60         WHEN test_d IS NOT NULL
61         THEN 'test_d'
62         ELSE NULL
63     END) AS test_number,
64     (CASE
65         WHEN test_d IS NOT NULL
66         THEN '2025-01-01 00:00:00'
67         ELSE NULL
68     END) AS test_start_date
69 FROM dsv1069.final_assignments_qa
70
71 UNION
72
73 SELECT
74     item_id,
75     test_e AS test_assignment,
76     (CASE
77         WHEN test_e IS NOT NULL
78         THEN 'test_e'
79         ELSE NULL
80     END) AS test_number,
81     (CASE
82         WHEN test_e IS NOT NULL
83         THEN '2025-01-01 00:00:00'
84         ELSE NULL
85     END) AS test_start_date
86 FROM dsv1069.final_assignments_qa
87
88 UNION
89
90 SELECT
91     item_id,

```

```
92 test_f AS test_assignment,  
93 (CASE  
94     WHEN test_f is NOT NULL  
95     THEN 'test_f'  
96     ELSE NULL  
97 END) AS test_number,  
98 (CASE  
99     WHEN test_f is NOT NULL  
100    THEN '2025-01-01 00:00:00'  
101    ELSE NULL  
102 END) AS test_start_date  
103 FROM dsv1069.final_assignments_qa;  
104
```

### 3. Compute Order Binary

```
1  -- Use this table to compute order_binary for
2  -- the 30 day window after the test_start_date
3  -- for the test named item_test_2
4
5  SELECT
6      test_assignment,
7      SUM(order_binary) AS ordered_items,
8      COUNT(DISTINCT item_id) AS number_of_items
9  FROM
10     (SELECT
11         item_test_2.item_id,
12         item_test_2.test_assignment,
13         item_test_2.test_number,
14         item_test_2.test_start_date,
15         item_test_2.created_at,
16         MAX(CASE
17             WHEN (created_at > test_start_date
18                 AND DATE_PART('day', created_at - test_start_date) <= 30)
19             THEN 1
20             ELSE 0
21         END) AS order_binary
22     FROM
23         (
24             SELECT
25                 final_assignments.*,
26                 DATE(orders.created_at) AS created_at
27             FROM
28                 dsv1069.final_assignments AS final_assignments
29             LEFT JOIN
30                 dsv1069.orders AS orders
31             ON
32                 final_assignments.item_id = orders.item_id
33             WHERE
34                 test_number = 'item_test_2'
35         ) AS item_test_2
36     GROUP BY
37         item_test_2.item_id,
38         item_test_2.test_assignment,
39         item_test_2.test_number,
40         item_test_2.test_start_date,
41         item_test_2.created_at) AS order_binary
42 GROUP BY
43     test_assignment;
```

#### 4. Compute View Item Metrics

```
1  -- Use the final_assignments table to calculate the view binary, and
2  -- average views for the 30 day window after the test assignment for
3  -- item_test_2. (You may include the day the test started)
4
5  SELECT
6      test_assignment,
7      SUM(binary_view) AS viewed_items,
8      COUNT(item_id) AS items_assignment,
9      SUM(views) AS total_views,
10     SUM(views)/COUNT(item_id) AS average_views
11 FROM
12 (
13     SELECT
14         final_assignments.test_assignment,
15         final_assignments.item_id,
16         MAX(
17             CASE
18                 WHEN views.event_time > final_assignments.test_start_date
19                 THEN 1
20                 ELSE 0
21             END) AS binary_view,
22         COUNT(views.event_id) AS views
23     FROM
24         dsv1069.final_assignments
25     LEFT JOIN
26     (
27         SELECT
28             event_time,
29             event_id,
30             CAST(parameter_value AS INT) AS item_id
31         FROM
32             dsv1069.events
33         WHERE
34             event_name = 'view_item'
35         AND
36             parameter_name = 'item_id'
37     ) views
38     ON
39         final_assignments.item_id = views.item_id
40     AND
41         views.event_time >= final_assignments.test_start_date
42     AND
43         DATE_PART('day', views.event_time - final_assignments.test_start_date ) <= 30
```

```
44 WHERE
45     final_assignments.test_number = 'item_test_2'
46 GROUP BY
47     final_assignments.item_id,
48     final_assignments.test_assignment
49 ) view_metrics
50
51 GROUP BY
52     view_metrics.test_assignment;
```



# FINAL ASSIGNMENT

## 1. DATA QUALITY CHECK

	item_id	test_a	test_b	test_c	test_d	test_e	test_f
1	2512	1	0	1	1	0	1
2	482	0	1	1	1	0	0
3	2446	0	1	1	0	1	0
4	1312	0	0	0	0	0	1
5	3556	1	1	0	1	0	0
6	131	0	0	0	0	1	1
7	1178	1	0	1	0	1	1
8	110	0	1	1	1	1	0
9	47	0	0	1	0	1	1
10	1696	0	0	1	1	1	1
11	3196	0	0	0	1	0	1
12	1578	1	0	0	1	0	1

Does this table have everything you need to compute metrics like 30-day view-binary?

No, this table does not have all of the data required to compute a 30-day view-binary. Specifically, we will need to know the dates of the tests and the orders so we can create a 30-day window.



## 2. REFORMAT THE DATA

	item_id	test_assignment	test_number	test_start_date
1	3824	1	test_f	2025-01-01 00:00:00
2	2098	0	test_b	2025-01-01 00:00:00
3	556	0	test_d	2025-01-01 00:00:00
4	3033	1	test_f	2025-01-01 00:00:00
5	445	1	test_a	2025-01-01 00:00:00
6	3332	0	test_a	2025-01-01 00:00:00
7	805	1	test_c	2025-01-01 00:00:00
8	610	1	test_e	2025-01-01 00:00:00
9	3504	0	test_a	2025-01-01 00:00:00
10	1281	0	test_c	2025-01-01 00:00:00
11	3706	0	test_b	2025-01-01 00:00:00
12	259	1	test_e	2025-01-01 00:00:00

### 3. COMPUTE ORDER BINARY

	test_assignment	ordered_items	number_of_items
1	0	386	1130
2	1	363	1068

#### LIFT AND P-VALUE FOR ORDER METRICS:

Condition Success Rate

Control 31% - 37% (34%)

Treatment 31% - 37% (34%)

Improvement: -12% - 11% (-0.5%)

p-value: 0.93

When comparing orders in each condition, there is no observed difference. Further, with a p-value far above 0.05, we are unable to say whether there is any difference.

#### 4. COMPUTE VIEW ITEM METRICS

	test_assignment	viewed_items	items_assignment	total_views	average_views
1	0	918	1130	1916	1.69557522124
2	1	890	1068	1862	1.74344569288

#### LIFT AND P-VALUE FOR VIEW METRICS:

Condition Success Rate

Control 79% - 83% (81%)

Treatment 81% - 85% (83%)

Improvement: -1.4% - 6.5% (2.6%)

p-value: 0.20

For views, the treatment condition does seem to have increased the success rate slightly. However, with a p-value of 0.2, we are unable to reject the null hypothesis and conclude that the treatment condition contributes to a higher view rate.