Data Wrangling, Analysis and AB Testing with SQL

Final Assignment: AB Testing

ASSIGNMENT TASK

We are running an experiment at an item-level, which means all users who visit will see the same page, but the layout of different item pages may differ.

1. DATA QUALITY CHECK

Compare the final_assignments_qa table to the assignment events we captured for user_level_testing. Write an answer to the following question: Does this table have everything you need to compute metrics like 30-day view-binary?

```
SELECT * FROM dsv1069.final_assignments_qa
```

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	2810	1	0	0	1	0	0

This table only shows the first 1,000 rows.

Answer: No, the final_assignments_qa table does not have everything needed to compute metrics like the 30-day view-binary. Specifically, we also need the datetime information to accurately track when the assignment events occurred. Without timestamps, it's not possible to measure activity within a 30-day window.

2. REFORMAT THE DATA

Write a query and table creation statement to make final_assignments_qa look like the final_assignments table. If you discovered something missing in part 1, you may fill in the value with a place holder of the appropriate data type.

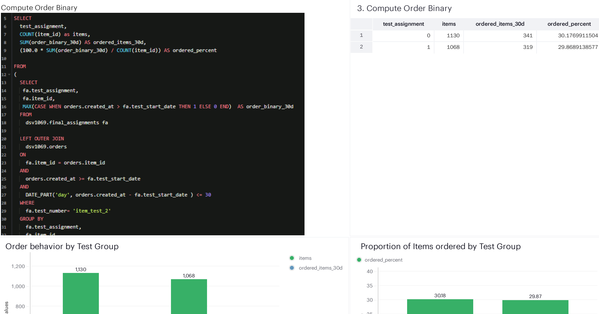
```
SELECT item_id,
       test_a AS test_assignment,
       'test_a' AS test_number,
       CAST('2025-01-17 00:00:00' AS timestamp) AS test_start_date
FROM dsv1069.final_assignments_qa
UNION ALL
SELECT item_id,
       test_b AS test_assignment,
       'test_b' AS test_number,
       CAST('2025-01-17 00:00:00' AS timestamp) AS test_start_date
FROM dsv1069.final_assignments_qa
UNION ALL
SELECT item_id,
       test_c AS test_assignment,
       'test_c' AS test_number,
       CAST('2025-01-17 00:00:00' AS timestamp) AS test_start_date
FROM dsv1069.final_assignments_qa
UNION ALL
```

```
SELECT item_id,
       test_d AS test_assignment,
       'test_d' AS test_number,
       CAST('2025-01-17 00:00:00' AS timestamp) AS test_start_date
FROM dsv1069.final_assignments_qa
UNION ALL
SELECT item_id,
       test_e AS test_assignment,
       'test_e' AS test_number,
       CAST('2025-01-17 00:00:00' AS timestamp) AS test_start_date
FROM dsv1069.final_assignments_qa
UNION ALL
SELECT item_id,
       test_f AS test_assignment,
       'test_f' AS test_number,
       CAST('2025-01-17 00:00:00' AS timestamp) AS test_start_date
FROM dsv1069.final_assignments_qa
ORDER BY test_number
```

2. Reformat the Data

	item_id	test_assignment	test_number	test_start_date
1	2512	1	test_a	2025-01-17 00:00:00
2	482	0	test_a	2025-01-17 00:00:00
3	2446	0	test_a	2025-01-17 00:00:00
4	1312	0	test_a	2025-01-17 00:00:00
5	2810	1	test_a	2025-01-17 00:00:00

This table only shows the first 1,000 rows.



3. COMPUTE ORDER BINARY

Use the final_assignments table to calculate the order binary for the 30 day window after the test assignment for item_test_2 (You may include the day the test started)

Compute 30 day Order Binary

```
SELECT
  fa.test_assignment,
  fa.item_id,
  MAX(CASE WHEN orders.created_at > fa.test_start_date THEN 1 ELSE 0 END)  AS order_binary_30d
FROM
  dsv1069.final_assignments fa

LEFT OUTER JOIN
  dsv1069.orders
ON
  fa.item_id = orders.item_id
AND
  orders.created_at >= fa.test_start_date
AND
  DATE_PART('day', orders.created_at - fa.test_start_date ) <= 30
WHERE
  fa.test_number= 'item_test_2'
GROUP BY
  fa.test_assignment,
  fa.item_id
```

Compute 30 day Order Binary

test_assignment	item_id	order_binary_30d
SUM: 1068	COUNT DISTINCTS: 2198	SUM: 660
AVG: 0.49		AVG: 0.30
MIN: 0		MIN: 0
MAX: 1		MAX: 1
0	0	1
1	3313	0
0	2123	1
1	1687	0
0	96	0
1	869	1
0	2454	0
0	3475	1
1	2810	1

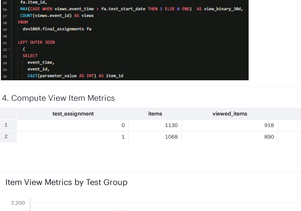
Page 1 of 22 — Showing rows 1-100 of 2,198

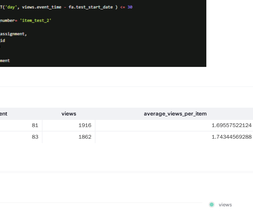
Compute Order Binary

```
SELECT
  test_assignment,
  COUNT(item_id) as items,
  SUM(order_binary_30d) AS ordered_items_30d,
  (100.0 * SUM(order_binary_30d) / COUNT(item_id)) AS ordered_percent

FROM
(
  SELECT
    fa.test_assignment,
    fa.item_id,
    MAX(CASE WHEN orders.created_at > fa.test_start_date THEN 1 ELSE 0 END)  AS order_binary_30d
  FROM
    dsv1069.final_assignments fa

  LEFT OUTER JOIN
    dsv1069.orders
  ON
    fa.item_id = orders.item_id
  AND
    orders.created_at >= fa.test_start_date
  AND
    DATE_PART('day', orders.created_at - fa.test_start_date ) <= 30
  WHERE
    fa.test_number= 'item_test_2'
  GROUP BY
    fa.test_assignment,
    fa.item_id
)
```

3. Compute Order Binary

	test_assignment	items	ordered_items_30d	ordered_percent
1	0	1130	341	30.1769911504
2	1	1068	319	29.8689138577





4. COMPUTE VIEW BINARY

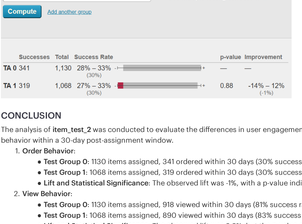
Use the final_assignments table to calculate the view binary, and average views for the 30 day window after the test assignment for item_test_2. (You may include the day the test started)

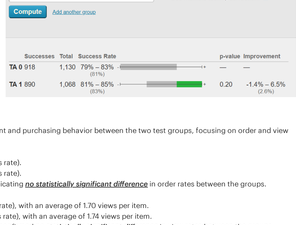
```
SELECT
test_assignment,
COUNT(item_id) AS items,
SUM(view_binary_30d) AS viewed_items,
CAST(SUM(view_binary_30d)/COUNT(item_id) AS FLOAT) AS viewed_percent,
SUM(views) AS views,
SUM(views)/COUNT(item_id) AS average_views_per_item
FROM
(
SELECT
  fa.test_assignment,
  fa.item_id,
  MAX(CASE WHEN views.event_time > fa.test_start_date THEN 1 ELSE 0 END)  AS view_binary_30d,
  COUNT(views.event_id) AS views
FROM
  dsv1069.final_assignments fa

LEFT OUTER JOIN
  (
  SELECT
    event_time,
    event_id,
    CAST(parameter_value AS INT) AS item_id
```

```
  FROM
    dsv1069.events
  WHERE
    event_name = 'view_item'
  AND
    parameter_name = 'item_id'
  ) views
ON
  fa.item_id = views.item_id
AND
  views.event_time >= fa.test_start_date
AND
  DATE_PART('day', views.event_time - fa.test_start_date ) <= 30
WHERE
  fa.test_number= 'item_test_2'
GROUP BY
  fa.test_assignment,
  fa.item_id
) item_level
GROUP BY
test_assignment
```

4. Compute View Item Metrics

	test_assignment	items	viewed_items	viewed_percent	views	average_views_per_item
1	0	1130	918	81	1916	1.69557522124
2	1	1068	890	83	1862	1.74344569288





5. COMPUTE LIFT AND P-VALUE

Use the https://thumbtack.github.io/abba/demo/abba.html to compute the lifts in metrics and the p-values for the binary metrics (30 day order binary and 30 day view binary) using a interval 95% confidence.

30 DAY ORDER BINARY LIFTS AND P-VALUES

Label	Number of successes	Number of trials	
TA 0	341	1130	Remove
TA 1	319	1068	Remove

Interval confidence level: 0.95 — Use multiple testing correction: ☑

Compute — Add another group

	Successes	Total	Success Rate	p-value	Improvement
TA 0	341	1,130	28% – 33% (30%)	—	—
TA 1	319	1,068	27% – 33% (30%)	0.88	-14% – 12% (-1%)

30 DAY VIEW BINARY LIFT AND P-VALUES

Label	Number of successes	Number of trials	
TA 0	918	1130	Remove
TA 1	890	1068	Remove

Interval confidence level: 0.95 — Use multiple testing correction: ☑

Compute — Add another group

	Successes	Total	Success Rate	p-value	Improvement
TA 0	918	1,130	79% – 83% (81%)	—	—
TA 1	890	1,068	81% – 85% (83%)	0.20	-1.4% – 6.5% (2.6%)

CONCLUSION

The analysis of **item_test_2** was conducted to evaluate the differences in user engagement and purchasing behavior between the two test groups, focusing on order and view behavior within a 30-day post-assignment window.

1. **Order Behavior**:
   - **Test Group 0**: 1130 items assigned, 341 ordered within 30 days (30% success rate).
   - **Test Group 1**: 1068 items assigned, 319 ordered within 30 days (30% success rate).
   - **Lift and Statistical Significance**: The observed lift was -1%, with a p-value indicating *no statistically significance* in order rates between the groups.
2. **View Behavior**:
   - **Test Group 0**: 1130 items assigned, 918 viewed within 30 days (81% success rate), with an average of 1.70 views per item.
   - **Test Group 1**: 1068 items assigned, 890 viewed within 30 days (83% success rate), with an average of 1.74 views per item.
   - **Lift and Statistical Significance**: The observed lift was 2.6%, but the p-value confirmed *no statistically significant difference* in view rates between the groups.

KEY INSIGHTS:

- Both test groups exhibited similar performance across the analyzed metrics.
- The lack of statistically significant differences in order and view rates suggests that the test intervention for **item_test_2** did not yield a measurable impact on user behavior within the 30-day evaluation window.

RECOMMENDATIONS:

- Consider additional iterations of the test with alternative strategies or interventions to achieve measurable effects.
- Analyze secondary metrics or extend the observation window to explore potential longer-term impacts.