



# WARBY PARKER Usage Funnels

Analyze Data with SQL

Ahmad Bassel

11.10.2024

# Example Table of Contents

1. Quiz Funnel
2. Home Try-On Funnel
3. Purchase Funnel

**Note:** This project is a collaboration with Warby Parker's Data Science team (thank you!) and uses fictional data.



*Codecademy's very own Nick Duckwiler (left) and Ryan Tuck from Warby Parker (right) adding finishing touches to the project.*

# 1. Quiz Funnel

# 1.1 Survey

To help users find their perfect frame, Warby Parker has a Style Quiz (see next section). The users' responses are stored in a table called survey.

```
SELECT * FROM survey LIMIT 5;
```

question	user_id	response
1. What are you looking for?	005e7f99-d48c-4fce-b605-10506c85aaf7	Women's Styles
2. What's your fit?	005e7f99-d48c-4fce-b605-10506c85aaf7	Medium
3. Which shapes do you like?	00a556ed-f13e-4c67-8704-27e3573684cd	Round
4. Which colors do you like?	00a556ed-f13e-4c67-8704-27e3573684cd	Two-Tone
1. What are you looking for?	00a556ed-f13e-4c67-8704-27e3573684cd	I'm not sure. Let's skip it.

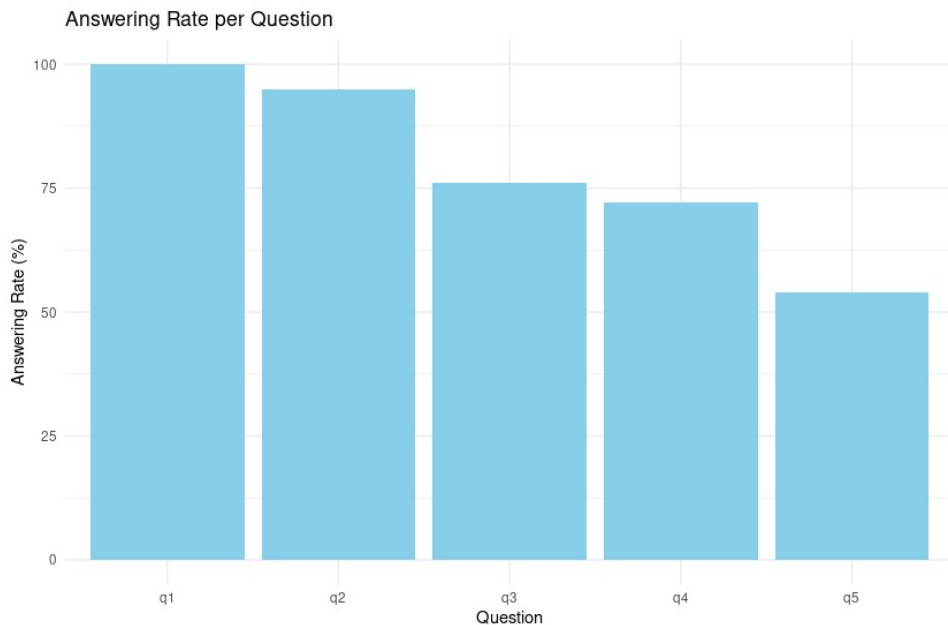
survey	
name	type
question	TEXT
user_id	TEXT
response	TEXT
Rows: 1986	

## 1.2 Users Count

The count of the users who participated in the survey

```
SELECT question,  
COUNT(DISTINCT user_id) AS  
distinct_user_countFROM  
surveyGROUP BY  
questionORDER BY question;
```

question	distinct_user_count
1. What are you looking for?	500
2. What's your fit?	475
3. Which shapes do you like?	380
4. Which colors do you like?	361
5. When was your last eye exam?	270



## 2. Home Try-On Funnel

## 2.1 Quiz

To help users find their perfect frame, Warby Parker has a Style Quiz that has the following questions:

“What are you looking for?”

“What’s your fit?”

“Which shapes do you like?”

“Which colors do you like?”

“When was your last eye exam?”

Quiz data is saved in the quiz table.

```
SELECT * FROM quiz LIMIT 5;
```

user_id	style	fit	shape	color
4e8118dc-bb3d-49bf-85fc-cca8d83232ac	Women's Styles	Medium	Rectangular	Tortoise
291f1cca-e507-48be-b063-002b14906468	Women's Styles	Narrow	Round	Black
75122300-0736-4087-b6d8-c0c5373a1a04	Women's Styles	Wide	Rectangular	Two-Tone
75bc6ebd-40cd-4e1d-a301-27dd93b12e2	Women's Styles	Narrow	Square	Two-Tone
ce965c4d-7a2b-4db6-9847-601747fa7812	Women's Styles	Wide	Rectangular	Black

quiz	
name	type
user_id	TEXT
style	TEXT
fit	TEXT
shape	TEXT
color	TEXT
Rows: 1000	



## 2.2 Home Try-On

Home try-on stage data are saved to the table home\_try\_on.

```
SELECT * FROM home_try_on LIMIT 5;
```

user_id	number_of_pairs	address
d8add87-3217-4429-9a01-d56d68111da7	5 pairs	145 New York 9a
f52b07c8-abe4-4f4a-9d39-ba9fc9a184cc	5 pairs	383 Madison Ave
8ba0d2d5-1a31-403e-9fa5-79540f8477f9	5 pairs	287 Pell St
4e71850e-8bbf-4e6b-accb-49a7bb46c586	3 pairs	347 Madison Square N
3bc8f97f-2336-4dab-bd86-e391609dab97	5 pairs	182 Cornelia St

home_try_on	
name	type
user_id	TEXT
number_of_pairs	TEXT
address	TEXT
Rows: 750	

## 2.3 Purchase

Purchase data saved to purchase.

```
SELECT * FROM purchase LIMIT 5;
```

user_id	product_id	style	model_name	color	price
00a9dd17-36c8-430c-9d76-df49d4197dcf	8	Women's Styles	Lucy	Jet Black	150
00e15fe0-c86f-4818-9c63-3422211baa97	7	Women's Styles	Lucy	Elderflower Crystal	150
017506f7-aba1-4b9d-8b7b-f4426e71b8ca	4	Men's Styles	Dawes	Jet Black	150
0176bfb3-9c51-4b1c-b593-87edab3c54cb	10	Women's Styles	Eugene Narrow	Rosewood Tortoise	95
01fdf106-f73c-4d3f-a036-2f3e2ab1ce06	8	Women's Styles	Lucy	Jet Black	150

purchase	
name	type
user_id	TEXT
product_id	INTEGER
style	TEXT
model_name	TEXT
color	TEXT
price	INTEGER
Rows: 495	

# 3. Purchase Funnel

## 3.1 Home Try-On to Purchase

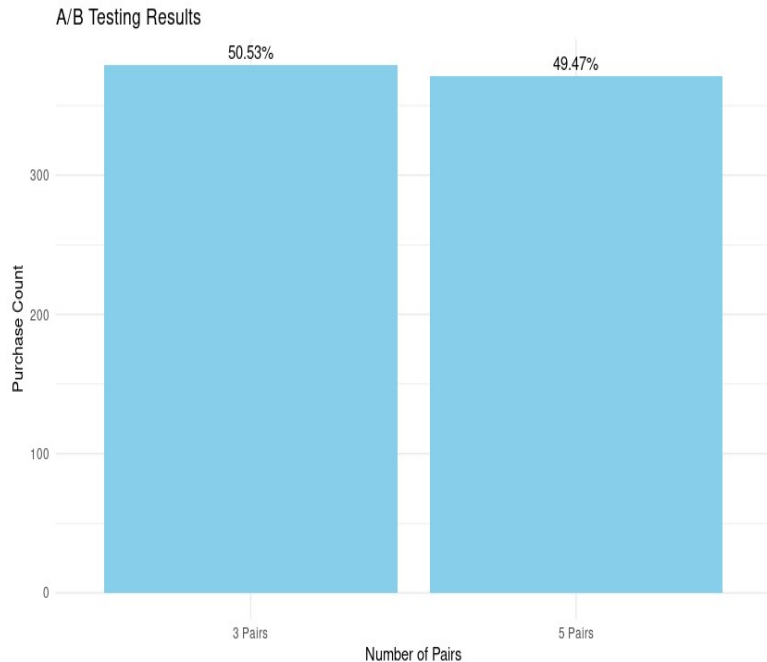
```
SELECT q.user_id,  
       h.user_id IS NOT NULL AS 'is_home_try_on',  
       h.number_of_pairs, p.user_id IS NOT NULL AS 'is_purchase'  
FROM quiz AS 'q'  
LEFT JOIN home_try_on AS 'h'  
      ON q.user_id = h.user_id  
LEFT JOIN purchase AS 'p'  
      ON h.user_id = p.user_id  
LIMIT 10;
```

user_id	is_home_try_on	number_of_pairs	is_purchase
4e8118dc-bb3d-49bf-85fc-cca8d83232ac	1	3 pairs	0
291f1cca-e507-48be-b063-002b14906468	1	3 pairs	1
75122300-0736-4087-b6d8-c0c5373a1a04	0	Ø	0
75bc6ebd-40cd-4e1d-a301-27ddd93b12e2	1	5 pairs	0
ce965c4d-7a2b-4db6-9847-601747fa7812	1	3 pairs	1
28867d12-27a6-4e6a-a5fb-8bb5440117ae	1	5 pairs	1
5a7a7e13-fbcf-46e4-9093-79799649d6c5	0	Ø	0
0143cb8b-bb81-4916-9750-ce956c9f9bd9	0	Ø	0
a4ccc1b3-cbb6-449c-b7a5-03af42c97433	1	5 pairs	0
b1dded76-cd60-4222-82cb-f6d464104298	1	3 pairs	0

## 3.2 A/B Testing

```
WITH ab_query AS (  
    SELECT q.user_id,  
    h.user_id IS NOT NULL AS 'is_home_try_on',  
    CASE  
        h.number_of_pairs WHEN h.number_of_pairs IS NULL  
        THEN 0  
        ELSE h.number_of_pairs  
    END AS 'number_of_pairs',  
    p.user_id IS NOT NULL AS 'is_purchase'  
FROM quiz AS 'q'  
LEFT JOIN home_try_on AS 'h' ON q.user_id = h.user_id  
LEFT JOIN purchase AS 'p' ON h.user_id = p.user_id  
)  
SELECT number_of_pairs,  
    COUNT(is_purchase) AS 'purchase_count'  
FROM ab_query  
WHERE number_of_pairs > 0  
GROUP BY number_of_pairs;
```

number_of_pairs	purchase_count
3 pairs	379
5 pairs	371



# Attachments

# Project Rubric

## Warby Parker's Marketing Funnels: Project Rubric

Criteria	Levels			
	4 (Exceeds Expectations)	3 (Meets Expectations)	2 (Approaches Expectations)	1 (Does Not Meet Expectations)
<b>Report</b> Are conclusions clear and supported by evidence in all eight answers?	<b>Every answer</b> includes text and query results (as supporting evidence). Related queries are uploaded in a separate .sql file. <b>Note:</b> this does not apply to final question, where conclusions are derived from previous questions.  Conclusions are clearly stated and based on evidence.	<b>One answer</b> is missing text, query results (as supporting evidence), or the related queries.  Conclusions are <b>not</b> well explained or supported by evidence.	<b>2 - 3 answers</b> are missing text, query results (as supporting evidence), or the related queries.  Conclusions are <b>not</b> well explained or supported by evidence.	<b>4 or more answers</b> are missing text, query results (as supporting evidence), or the related queries.  Conclusions are <b>not</b> well explained or supported by evidence.
<b>Query Accuracy</b> Do queries do what they were supposed to do?	All queries accurately retrieve the data intended (as stated or implied by comments or the presentation).	<b>90% - 99%</b> of queries are accurate.	<b>75% - 89%</b> of queries are accurate.	<b>&lt; 75%</b> of queries are accurate.
<b>Query Formatting</b>	Queries are formatted according to the <a href="#">linked style guide</a> .	Queries adhere to at least <b>7 rules</b> of the style guide.	Queries adhere to at least <b>5 rules</b> of the style guide.	Queries adhere to <b>4 or fewer rules</b> of the style guide.
<b>Understanding underlying concepts and terminology</b>	Report demonstrates <b>clear understanding</b> of funnels..  Terms <b>are used</b> correctly throughout the report.	Report demonstrates a <b>partial understanding</b> of funnels.  Terms <b>are used</b> correctly throughout the report.	Report demonstrates an <b>insufficient understanding</b> of funnels.  Terms <b>are used</b> correctly throughout the report.	Report demonstrates an <b>insufficient understanding</b> of funnels.  Terms <b>are not used</b> correctly throughout the report.

# SQL Style Guide for Project

- Use snake case, e.g. `page_visits`, `first_touches`
- Aliases are used when they improve the readability of the query and results. Aliases should relate in some way to the object or expression they are aliasing
- GROUP BY and ORDER BY use column numbers or aliases when they refer to aggregate or calculated columns, like COUNT() and AVG().
- Always use uppercase for the reserved keywords like SELECT and WHERE
- Each column name gets its own line in SELECT clauses
- There are no extraneous columns, i.e. columns that are not asked for and/or not relevant to the topic
- DISTINCT does not use parentheses
- WITH clauses are separated by commas, rather than nested
- Spaces should be used to left-align queries, subqueries, and joins, respectively

```
WITH sub_query_1 AS (  
    SELECT DISTINCT t1.column1,  
                   t2.column2  
    FROM table1 t1  
    JOIN table2 t2  
        ON t1.columnA = t2.columnA  
    WHERE t1.columnB = 'foobar'),  
sub_query_2 AS (  
    SELECT t3.column3,  
           SUM(t4.column4) AS aggregate4  
    FROM table4 t4  
    LEFT JOIN table5 t5  
        ON t4.columnC = t5.columnC  
    GROUP BY 1  
    HAVING SUM(t4.column4) > 5)  
SELECT sql.column1,  
       sq2.aggregate4  
FROM sub_query_1 sql  
JOIN sub_query_2 sq2  
    ON sql.column1 = sq2.column3  
WHERE sq2.column3 > 5  
ORDER BY 1, 2 DESC  
LIMIT 20;
```