



Warby Parker - Funnels

Learn SQL from Scratch

Toby Fry

12/12/2018

Warby Parker – Examining Funnels

Contents Page

1. The Quiz Funnel

1.1 Determining Columns

1.2 Producing The Funnel

1.3 Funnel Analysis

2. The Home Try-On Funnel

2.1 Determining Columns

2.2 Combining The Tables

2.3 Raw Funnel Output

2.4 Revised Funnel Output

2.5 Funnel Analysis – Overall Users

2.6 Funnel Analysis – Three Pairs or Five?

2.7 Additional Insights

1. The Quiz Funnel

1.1 – Determining Columns

Before analysis can be done on the survey funnel, initially we must get a feel for the table. This short query allows one to fetch the first 10 records for all columns in the “Survey” table, which gives the below output.

This data is not in itself useful, but does show that the three columns in this table are ‘question’, ‘user_id’ and ‘response’.
With this information, we can begin to examine the funnel.

-- Question 1: Determining the columns

```
SELECT *  
FROM survey  
LIMIT 10;
```

Query Results		
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.
2. What's your fit?	00a556ed-f13e-4c67-8704-27e3573684cd	Narrow
5. When was your last eye exam?	00a556ed-f13e-4c67-8704-27e3573684cd	<1 Year
3. Which shapes do you like?	00bf9d63-0999-43a3-9e5b-9c372e6890d2	Square
5. When was your last eye exam?	00bf9d63-0999-43a3-9e5b-9c372e6890d2	<1 Year
2. What's your fit?	00bf9d63-0999-43a3-9e5b-9c372e6890d2	Medium

1.2 – Producing the Funnel

To count the users completing each step, we need to know the question and the number of responses to each. This can be accomplished by **SELECTING** the question and performing a **COUNT** on all columns, then using a **GROUP BY** command to provide a count for each user that responded. This produces the table on the left.

In order to make this data more presentable, adding a simple **AS** to each **SELECT** produces a more readily understandable table, shown on the right.

We can see that the number of users completing decreases with each question, it appears that the biggest drops were on questions 3 and 5.

```
-- Question 2 - Finding the funnel

SELECT question, COUNT(*)
FROM survey
GROUP BY question;

-- Question 2a - Improving readability

SELECT
    question AS 'Question',
    COUNT(*) AS 'Responses'
FROM survey
GROUP BY question;
```

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

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

1.3 – Funnel Analysis

Plugging this output into Microsoft Excel, we can quickly and easily determine the percentage of users that passed from each question to the next.

This allows one to spot which questions have the lowest completion rates, in this case questions 3 and 5 have the lowest rate by a considerable margin.

In the case of question 3 this may be due to the unusual nature of the question, as in most cases the shapes of different types of glasses are not easy to distinguish between.

In the case of question 5, however, this is likely to be because many customers will not be able to remember, or will not have the details of their last eye exam to hand.

Question	Responses	Percentage Completion Rate
1. What are you looking for?	500	100.00%
2. What's your fit?	475	95.00%
3. Which shapes do you like?	380	80.00%
4. Which colours do you like?	361	95.00%
5. When was your last eye exam?	270	74.79%

2. The Home Try-On Funnel

2.1 – Determining Columns

Using a query similar to step 1.1 for the Quiz Funnel, we can determine the column names for all three tables for our scrutiny of the Home Try-On Funnel.

The below screen-dump shows the output of this query, from this we can see that the primary key for all three of these tables is the `user_id` column, which we can use for joining tables together to provide a useful funnel analysis.

```
SELECT *
FROM quiz
LIMIT 5;

SELECT *
FROM home_try_on
LIMIT 5;

SELECT *
FROM purchase
LIMIT 5;
```

Query Results					
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
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-acc6-49a7bb46c586		3 pairs		347 Madison Square N	
3bc8f97f-2336-4dab-bd86-e391609dab97		5 pairs		182 Cornelia St	
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

2.2 – Combining the Tables

As mentioned in the previous slide, the tables must now be combined on their primary keys to produce a set of data that can be manipulated into providing valuable insights.

To do this we need all `user_ids`, whether they chose a home try-on, how many pairs of glasses they tried, and whether or not this led to a purchase.

We can identify all unique `user_ids` by using `SELECT DISTINCT`, though because we're using multiple tables we must specify `quiz.user_id` to make sure that we're getting every user from the very beginning of the funnel.

Because `LEFT JOIN` will produce null values for any missing `user_ids` in the subsequent tables, we must specify `IS NOT NULL` when selecting the columns for the `home_try_on` and `purchase` tables, leading to an output of 1 for a user that underwent the try-on or purchase steps, or a 0 if they didn't.

We cannot do this on the `number_of_pairs` column, however, as this will specify 1 or 0 based on whether there is an entry, which is not useful to us as we need to know whether they had three or five pairs, so some `NULL` values will appear in the output.

```
-- Question 5 - Combining the tables

SELECT DISTINCT quiz.user_id,
                home_try_on.user_id IS NOT NULL AS
'is_home_try_on',
                home_try_on.number_of_pairs IS NOT NULL,
                purchase.user_id IS NOT NULL AS
'is_purchase'

FROM quiz

LEFT JOIN home_try_on
        ON quiz.user_id = home_try_on.user_id

LEFT JOIN purchase
        ON quiz.user_id = purchase.user_id

LIMIT 10;
```

2.2 - Combining the Tables

Now we have specified the columns we need, the next step is to finish constructing the query by joining the tables together.

Because we want to keep the rows that are present in earlier steps of the funnel, we must use a LEFT JOIN to maintain the user_ids that did not progress through all the steps.

We specify 'quiz' as the FROM, as this is our initial table to which we will LEFT JOIN the others.

With the user_id as the primary key of all three tables, we can use the user_id as the point at which the 'purchase' and 'home_try_on' tables will be joined.

Lastly, to prevent the query taking too long to run, we LIMIT the number of results to 10.

```
-- Question 5 - Combining the tables

SELECT DISTINCT quiz.user_id,
                home_try_on.user_id IS NOT NULL AS
'is_home_try_on',
                home_try_on.number_of_pairs IS NOT NULL,
                purchase.user_id IS NOT NULL AS
'is_purchase'

FROM quiz

LEFT JOIN home_try_on
        ON quiz.user_id = home_try_on.user_id

LEFT JOIN purchase
        ON quiz.user_id = purchase.user_id

LIMIT 10;
```

2.3 – Raw Funnel Output

When executed, the constructed code gives the following output.

For each user_id, the second and fourth columns output a value of 1 where the user has progressed to that stage. For each user that took the home try-on offer, it also specifies whether the user was offered three pairs or five pairs for their home try-on.

Now that we have the basic structure of the query, let’s alter it so that a value of 1 in the is_home_try_on and is_purchase columns gives an answer of ‘True’ or ‘False’. We’ll do this using CASE statements.

Query Results			
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	∅	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	∅	0
0143cb8b-bb81-4916-9750-ce956c9f9bd9	0	∅	0
a4ccc1b3-cbb6-449c-b7a5-03af42c97433	1	5 pairs	0
b1dded76-cd60-4222-82cb-f6d464104298	1	3 pairs	0

2.3 – Raw Funnel Output – True or false

By taking the code from the previous output, we can alter the SELECT values for columns 2 and 4 by instead using a case statement.

Due to the nature of LEFT JOIN, the user_id column on the 'home_try_on' and 'purchase' tables will contain a NULL value for any user that has not progressed to that stage.

Therefore, all we must establish for each column is whether or not the primary key from the LEFT JOINED tables has become NULL.

For each one, a CASE argument that WHEN the primary key is not a NULL value returns the output 'True', in all other cases it will return the answer 'False'. We can use identical structure to do the same for both of the extra tables.

Specifying the 'True' output as the selected choice will ensure that any errors that can occur will produce a 'False' output for the purposes of analysis.

```
-- Question 5 - Combining the tables

SELECT DISTINCT quiz.user_id,

CASE
    WHEN home_try_on.user_id IS NOT NULL THEN 'True'

    ELSE 'False'
END AS 'is_home_try_on',

home_try_on.number_of_pairs,

CASE
    WHEN purchase.user_id IS NOT NULL THEN
'True'

    ELSE 'False'
END AS 'is_purchase'

FROM quiz

LEFT JOIN home_try_on
    ON quiz.user_id = home_try_on.user_id

LEFT JOIN purchase
    ON quiz.user_id = purchase.user_id

LIMIT 10;
```

2.4 – Revised Funnel Output

Executing the code from the last slide gives the following output. Now that the data is in a more easily interpretable state, we can begin to alter the code to produce information that can help us understand the nature of the funnel and the habits of Warby Parker’s userbase.

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

2.5 – Funnel Analysis – Overall Users

The first thing we will do to simplify our queries is to encase all of the code written so far into a “With” statement renamed as the easily memorable ‘funnel’.

By removing the ELSE statements from columns 2 and 4 on our main query, this will return all ‘False’ statements to NULL values. This way, we can simply count the values for columns 1, 2 and 4 to return the total number of users who progressed to each stage, giving the below output.

Using a calculator, we can easily work out the stage-to-stage percentage retention.

This is interesting, but let’s examine how the number of pairs each user took home reflects on whether they purchased.

Took Quiz	Tried On	Purchased
1000	750	495

Took Quiz	Tried On	Purchased
100%	75%	66%

```
WITH 'funnel' AS (  
  
  --Basic Query  
  
  SELECT DISTINCT quiz.user_id,  
    CASE  
      WHEN home_try_on.user_id IS NOT NULL THEN  
      'True'  
    END AS 'is_home_try_on',  
    home_try_on.number_of_pairs,  
    CASE  
      WHEN purchase.user_id IS NOT NULL THEN  
      'True'  
    END AS 'is_purchase'  
  
  FROM quiz  
  
  LEFT JOIN home_try_on  
    ON quiz.user_id = home_try_on.user_id  
  
  LEFT JOIN purchase  
    ON quiz.user_id = purchase.user_id)  
  
  --Basic Query End  
  
  SELECT COUNT (*) as 'Took Quiz',  
    COUNT(is_home_try_on) AS 'Tried On',  
    COUNT(is_purchase) AS 'Purchased'  
  FROM funnel;
```

2.6 – Comparing Funnels – Three Pairs or Five?

By slightly altering the code we have just used, we can look at how the funnel changes based on how many pairs people took home to try.

To make writing this code easier, column 3 has been given the name ‘Pairs’ with an AS statement. Then we can include ‘Pairs’ in the SELECT section of our query.

If we GROUP BY ‘Pairs’, then the results will be separated into those who had three and those who had five pairs, and the number of each who went on to make a purchase.

Using a calculator, 53% of customers purchased after trying three pairs. However, 79% of customers purchased after trying five. This shows that Warby Parker can increase the number of customers who purchase by allowing all customers to try five pairs instead of three.

Pairs	Took Quiz	Tried On	Purchased
∅	250	0	0
3 pairs	379	379	201
5 pairs	371	371	294

```
WITH 'funnel' AS (
  SELECT DISTINCT quiz.user_id,

                CASE
                    WHEN home_try_on.user_id IS NOT
NULL THEN 'True'
                END AS 'is_home_try_on',

                home_try_on.number_of_pairs AS 'Pairs',

                CASE
                    WHEN purchase.user_id IS NOT NULL
THEN 'True'
                END AS 'is_purchase'

  FROM quiz

  LEFT JOIN home_try_on
        ON quiz.user_id = home_try_on.user_id

  LEFT JOIN purchase
        ON quiz.user_id = purchase.user_id)
--Query Below
SELECT pairs,
COUNT(*) AS 'Took Quiz'
COUNT(is_home_try_on) AS 'Tried On',
COUNT(is_purchase) AS 'Purchased'
from funnel
GROUP BY pairs;
```

2.7 – Additional Insights

Now that the funnels have been examined, let's play around with some of the other information in the tables to try and glean some valuable insight into Warby Parkers' customer preferences.

In order to establish what the most popular responses to the quiz were, we can use many simple queries at once to produce a list showing the most common responses to all of the questions.

```
SELECT quiz.style AS 'Style', COUNT(*)  
FROM quiz  
GROUP BY 1  
ORDER BY 2 desc;
```

```
SELECT quiz.fit AS 'Fit', COUNT(*)  
FROM quiz  
GROUP BY 1  
ORDER BY 2 desc;
```

```
SELECT quiz.shape AS 'Shape', COUNT(*)  
FROM quiz  
GROUP BY 1  
ORDER BY 2 desc;
```

```
SELECT quiz.color AS 'Colour', COUNT(*)  
FROM quiz  
GROUP BY 1  
ORDER BY 2 desc;
```


2.7 – Additional Insights

Running multiple queries presents this easy-to-examine set of results. We can see that women's and men's styles are roughly equal, that narrow-fit glasses are the most common size, that angular-shaped glasses are more popular than round ones, and that black and tortoise-coloured glasses are also very popular.

Does this interest translate into actual sales, however? Let's do the same code again but this time examine the purchases.

Query Results	
Style	COUNT(*)
Women's Styles	469
Men's Styles	432
I'm not sure. Let's skip it.	99
Fit	COUNT(*)
Narrow	408
Medium	305
Wide	198
I'm not sure. Let's skip it.	89
Shape	COUNT(*)
Rectangular	397
Square	326
Round	180
No Preference	97
Colour	COUNT(*)
Tortoise	292
Black	280
Crystal	210
Neutral	114
Two-Tone	104

2.7 – Additional Insights

To examine customers' purchasing habits as compared to how they answered in the quiz, let's run a similar query on the two analogous columns in the 'Purchase' table, as seen to the right.

The two things in the table that correspond most closely to the quiz are the colour and style of the purchased products, so we'll examine those alongside the export from the last query.

We will group in both cases by the different styles and colours, in descending order by count.

```
SELECT purchase.style AS 'Style', COUNT(*)  
FROM purchase  
GROUP BY 1  
ORDER BY 2 desc;
```

```
SELECT purchase.color AS 'Colour', COUNT(*)  
FROM purchase  
GROUP BY 1  
ORDER BY 2 desc;
```

2.7 – Additional Insights

The output from this query shows that the slight prevalence of women’s styles to men’s styles continues. From this we can deduce that the questions asked in the quiz section of the funnel lost roughly equal numbers of both men and women.

We can also tell that although jet black is the single most popular colour, the number of people who answered ‘Black’ as their favourite colour in the quiz did not go on to purchase jet black glasses. However, the large number of customers who answered ‘Tortoise’ are reflected in the many different types of tortoise finish that were purchased.

Further examination might be needed to determine why so few people went on to purchase black glasses, perhaps Warby Parker needs to expand their range of black glasses?

Query Results	
Style	COUNT(*)
Women's Styles	252
Men's Styles	243
Fit	COUNT(*)
Jet Black	86
Driftwood Fade	63
Rosewood Tortoise	62
Rose Crystal	54
Layered Tortoise Matte	52
Pearled Tortoise	50
Elderflower Crystal	44
Sea Glass Gray	43
Endangered Tortoise	41