



Funnels

with Warby Parker

Learn SQL from Scratch

Gregor Wienhold

07. August 2018

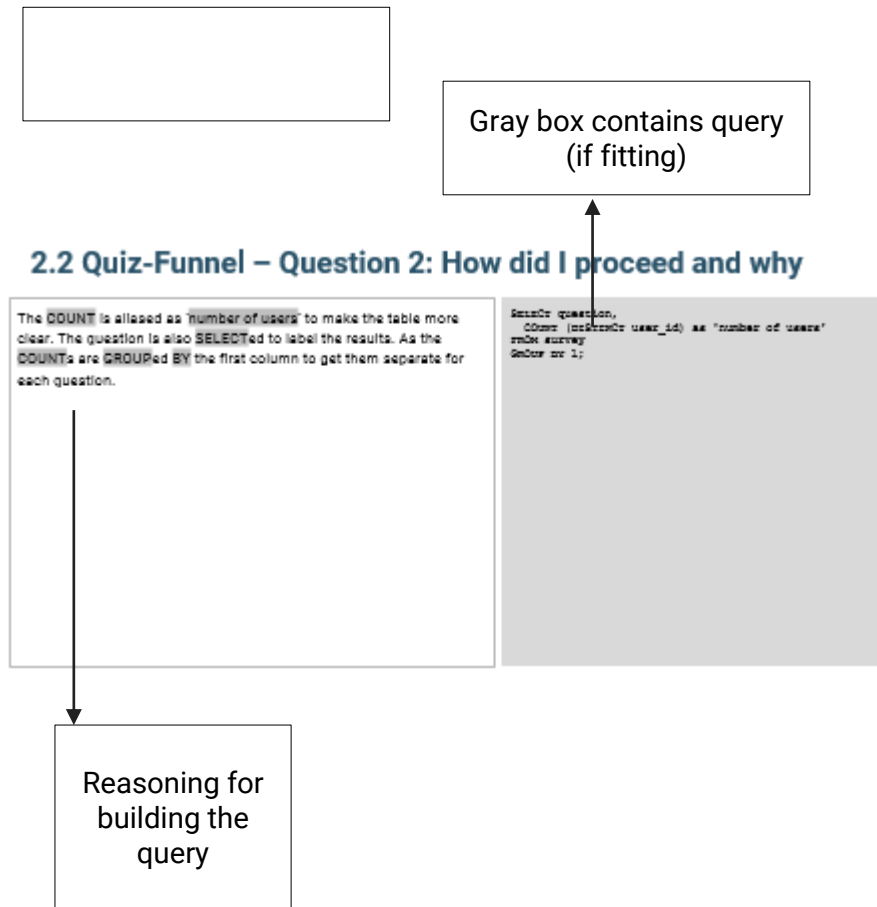
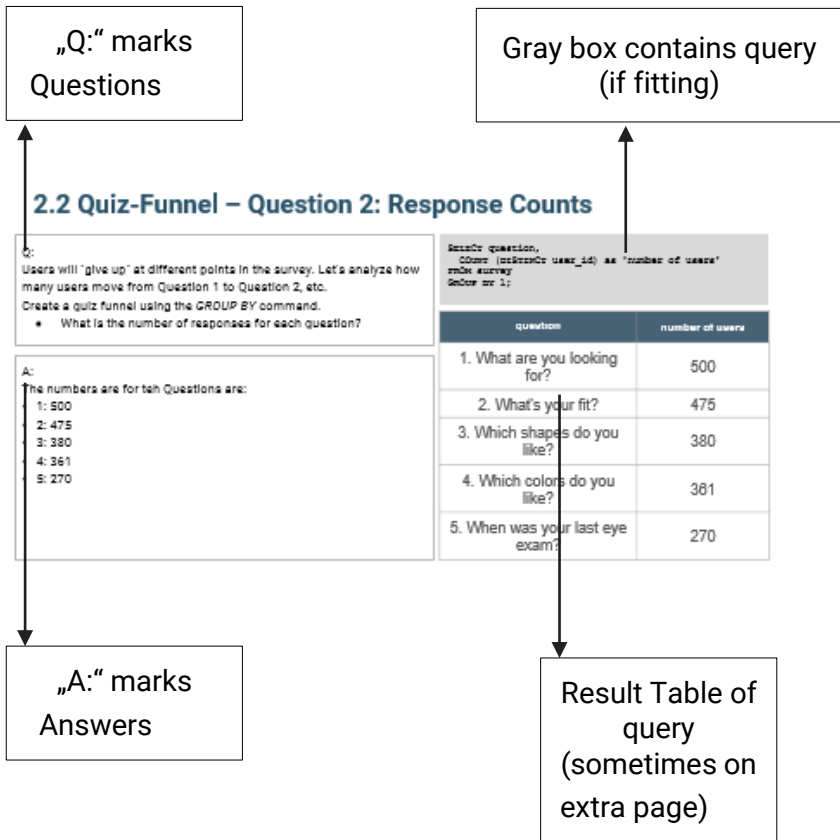
Table of Contents

1. Topic & Layout
2. Questions, Answers Explanations

1.1 Topic: Warby Parker

[Warby Parker](#) is a transformative lifestyle brand with a lofty objective: to offer designer eyewear at a revolutionary price while leading the way for socially conscious businesses. Founded in 2010 and named after two characters in an early Jack Kerouac journal, Warby Parker believes in creative thinking, smart design, and doing good in the world — for every pair of eyeglasses and sunglasses sold, a pair is distributed to someone in need.

1.2 Layout



2.1 Quiz-Funnel – 1: Survey Table

Q:

To help users find their perfect frame, Warby Parker has a Style Quiz. The users' responses are stored in a table called *survey* . Select all columns from the first 10 rows.

- What columns does the table have?

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

A:

The column-names are *question*, *user_id* and *response* as shown in the next slide.

2.1 Quiz-Funnel – 1: Result

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	
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

2.2 Quiz-Funnel – 2: Response Counts

Q:

Users will "give up" at different points in the survey. Let's analyze how many users move from Question 1 to Question 2, etc.

Create a quiz funnel using the *GROUP BY* command.

- What is the number of responses for each question?

A:

The numbers are for the questions are:

- 1: 500
- 2: 475
- 3: 380
- 4: 361
- 5: 270

```
SELECT question,  
       COUNT (DISTINCT user_id) as 'number of users'  
FROM survey  
GROUP BY 1;
```

question	number of users
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

2.2 Quiz-Funnel – 2: How did I proceed and why

The `COUNT` is aliased as 'number of users' to make the table more clear. The question is also `SELECT`ed to label the results. As the `COUNT`s are `GROUP`ed `BY` the first column to get them separate for each question.

```
SELECT question,  
       COUNT (DISTINCT user_id) as 'number of users'  
FROM survey  
GROUP BY 1;
```


2.3 Quiz-Funnel – 3

Q:

Using a spreadsheet program like Excel or Google Sheets, calculate the percentage of users who answer each question.

- Which question(s) of the quiz have a lower completion rates?
- What do you think is the reason?

A:

The lowest completion rates can be found in questions 2 (80%) and 4 (75%). Those numbers are shown in the funnel-steps (here: questions) following those question. This is because customers need to complete the previous question to get to the next one.

For question 2 people might not exactly know how wide their face is in comparison to the average face.

For question 4 people might want to see more than one color-option or might not like any of the shown colors.

question	percentage of users
1. What are you looking for?	100%
2. What's your fit?	95%
3. Which shapes do you like?	80%
4. Which colors do you like?	95%
5. When was your last eye exam?	75%

2.4 Home Try-On Funnel – 4.1

Q:

Warby Parker's purchase funnel is: Take the Style Quiz → Home Try-On → Purchase the Perfect Pair of Glasses

During the Home Try-On stage, we will be conducting an A/B Test:

50% of the users will get 3 pairs to try on, 50% of the users will get 5 pairs to try on.

Let's find out whether or not users who get more pairs to try on at home will be more likely to make a purchase.

- Examine the first five rows of each table

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

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

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

-- query part shown on each slide is marked bold

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-a301-27ddd93b12e2	Women's Styles	Narrow	Square	Two-Tone
ce965c4d-7a2b-4db6-9847-601747fa7812	Women's Styles	Wide	Rectangular	Black

A: The column-names of the *quiz* table are: *user_id*, *style*, *fit*, *shape* and *color*

2.4 Home Try-On Funnel – 4.2

Q:

Warby Parker's purchase funnel is: Take the Style Quiz → Home Try-On → Purchase the Perfect Pair of Glasses

During the Home Try-On stage, we will be conducting an A/B Test:

50% of the users will get 3 pairs to try on, 50% of the users will get 5 pairs to try on.

Let's find out whether or not users who get more pairs to try on at home will be more likely to make a purchase.

- Examine the first five rows of each table

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

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

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

-- query part shown on each slide is marked bold

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

A: The column-names of the *home_try_on* table are: *user_id*, *number_of_pairs* and *address*

2.4 Home Try-On Funnel – 4.3

Q:

Warby Parker's purchase funnel is: Take the Style Quiz → Home Try-On → Purchase the Perfect Pair of Glasses

During the Home Try-On stage, we will be conducting an A/B Test:

50% of the users will get 3 pairs to try on, 50% of the users will get 5 pairs to try on.

Let's find out whether or not users who get more pairs to try on at home will be more likely to make a purchase.

- Examine the first five rows of each table

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

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

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

-- query part shown on each slide is marked bold

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

A: The column-names of the *purchase* table are: *user_id*, *product_id*, *style*, *model_name*, *color* and *price*

2.4 Home Try-On Funnel – 4.1-3: How did I proceed and why

To directly get all the answers, we can simply send the queries at once. Because they all are **LIMIT**ed to 5 rows, space shouldn't be a problem.

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

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

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

2.5 Home Try-On Funnel – 5

Q:
We'd like to create a new table (layout shown in question).
Each row will represent a single user from the browse table:

- If the user has any entries in *home_try_on*, then *is_home_try_on* will be 'True'
- *Number_of_pairs* comes from *home_try_on* table
- If the user has any entries in *is_purchase*, then *is_purchase* will be 'True'

Use a *LEFT JOIN* to combine the three tables, starting with the top of the funnel (browse) and ending with the bottom of the funnel (purchase).

Select only the first 10 rows from this table (otherwise, the query will run really slowly).

```
WITH 'all_data' AS(  
  SELECT *  
  FROM quiz  
  LEFT JOIN home_try_on  
    ON quiz.user_id = home_try_on.user_id  
  LEFT JOIN purchase  
    ON home_try_on.user_id = purchase.user_id)  
SELECT user_id,  
  CASE  
    WHEN address IS NOT NULL THEN 'True'  
    WHEN address IS NULL THEN 'False'  
    ELSE NULL  
  END AS 'is_home_try_on',  
  CASE  
    WHEN number_of_pairs = '3 pairs' THEN 3  
    WHEN number_of_pairs = '5 pairs' THEN 5  
    ELSE NULL  
  END AS 'number_of_pairs',  
  CASE  
    WHEN price IS NOT NULL THEN 'True'  
    WHEN price IS NULL THEN 'False'  
    ELSE NULL  
  END AS 'is_purchase'  
FROM 'all_data'  
LIMIT 10;
```

2.5 Home Try-On Funnel – 5: Result Table

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

2.5 Home Try-On Funnel – 5: How did I proceed and why

The new table shall get its data from the temporary table 'all_data'.

Then we start by **SELECT**ing user_id and open a **CASE** which checks if there is an address and fills True or False into the table accordingly. The same goes for checking for price and purchase*.

The CASE statement on 'number_of_pairs' is only slightly different.

They are all **ENDED** as the names that they shall have.

The output is **LIMIT**ed to 10 rows.

*As address and price are necessary for try-On and purchase: Address, Try-On and Price, Purchase **COUNT**s are the same. I checked this using the extra queries in the .sql file

```
WITH 'all_data' AS(
  SELECT *
  FROM quiz
  LEFT JOIN home_try_on
    ON quiz.user_id = home_try_on.user_id
  LEFT JOIN purchase
    ON home_try_on.user_id = purchase.user_id)
SELECT user_id,
  CASE
    WHEN address IS NOT NULL THEN 'True'
    WHEN address IS NULL THEN 'False'
    ELSE NULL
  END AS 'is_home_try_on',
  CASE
    WHEN number_of_pairs = '3 pairs' THEN 3
    WHEN number_of_pairs = '5 pairs' THEN 5
    ELSE NULL
  END AS 'number_of_pairs',
  CASE
    WHEN price IS NOT NULL THEN 'True'
    WHEN price IS NULL THEN 'False'
    ELSE NULL
  END AS 'is_purchase'
FROM 'all_data'
LIMIT 10;
```


2.6 Home Try-On Funnel –6.1: Conversion Rate End2End

Q:

- We can calculate end-to-end conversion rate by aggregating across all rows.

A:

The conversion rate from people who took the quiz to buyers is 49.5%

```
--the result-table from question 5 is put in a WITH  
--statement AS 'structured_data'
```

```
'quiz_step' AS (  
  SELECT COUNT (user_id) AS 'quiz_participants'  
  FROM structured_data),  
'purchase_step' AS (  
  SELECT COUNT (is_purchase) as 'buyers'  
  FROM structured_data  
  WHERE is_purchase = 'True'),  
'step_counts' AS (  
  SELECT *  
  FROM quiz_step  
  CROSS JOIN purchase_step)  
SELECT 100.0*buyers/quiz_participants AS  
'conversion_rate_in_%'  
FROM step_counts;
```

conversion_rate_in_%
49.5

2.6 Home Try-On Funnel – 6.1: How did I proceed and why

The conversion rate can be found by dividing the number of buyers by the count of users who did the quiz.

As we have all the data in a structured way from the query in question 5 we can use it further by putting it in a **WITH** statement.

Then we need to count our quiz participants, which is everyone in this table. We can count them by counting **user_id**.

We also need to count the buyers. This can be done by counting the rows where **is_purchase** is true.

To directly see the result we put both of those counts in **WITH** statements, join those temporary tables (**CROSS JOIN** is ok as there is only 1 row in each table), store them as another temporary table and add a calculation with the latter. Doing the calculation directly in SQL is a more comfortable way than using e.g. Excel. The calculation needs a decimal number to avoid getting a rounded integer as result. To keep it legible and informative alias the output to what it is: the conversion rate in percent.

```
--the result-table from question 5 is put in a WITH  
--statement AS 'structured_data'
```

```
'quiz_step' AS (  
  SELECT COUNT (user_id) AS 'quiz_participants'  
  FROM structured_data),  
'purchase_step' AS (  
  SELECT COUNT (is_purchase) as 'buyers'  
  FROM structured_data  
  WHERE is_purchase = 'True'),  
'step_counts' AS (  
  SELECT *  
  FROM quiz_step  
  CROSS JOIN purchase_step)  
SELECT 100.0*buyers/quiz_participants AS  
'conversion_rate_in_%'  
FROM step_counts;
```

2.6 Home Try-On Funnel – 6.2.1: Quiz to Try-On Conversion

Q:

- We can compare conversion from *quiz*→*home_try_on* and *home_try_on*→*purchase*.

A:

The conversion rate from people who took the quiz to people who tried the glasses at home is 75 %.

```
--the result-table from question 5 is put in a WITH  
--statement AS 'structured_data'
```

```
'quiz_step' AS (  
  SELECT COUNT (user_id) AS 'quiz_participants'  
  FROM structured_data),  
'home_try_on_step' AS (  
  SELECT COUNT (is_home_try_on) as 'tried'  
  FROM structured_data  
  WHERE is_home_try_on = 'True'),  
'step_counts' AS (SELECT *  
  FROM quiz_step  
  CROSS JOIN home_try_on_step)  
SELECT 100.0*tried/quiz_participants AS  
'conversion_rate_quiz_to_try-on_in_%'  
FROM step_counts;
```

conversion_rate_quiz_to_try-on_in_%

75.0

2.6 Home Try-On Funnel – 6.2.1: How did I proceed and why

Here we can proceed like in question 6.1, but we need to change the count of buyers to the count of people who tried the glasses. The changed parts are marked **bold** in the query on the right.

```
--the result-table from question 5 is put in a WITH
--statement AS 'structured_data'

'quiz_step' AS (
  SELECT COUNT (user_id) AS 'quiz_participants'
  FROM structured_data),
'home_try_on_step' AS (
  SELECT COUNT (is_home_try_on) as 'tried'
  FROM structured_data
  WHERE is_home_try_on = 'True'),
'step_counts' AS (
  SELECT *
  FROM quiz_step
  CROSS JOIN home_try_on_step)
SELECT 100.0*tried/quiz_participants AS
'conversion_rate_quiz_to_try-on_in_%'
FROM step_counts;
```

2.6 Home Try-On Funnel – 6.2.2: Try-On to Buy Conversion

Q:

- We can compare conversion from *quiz*→*home_try_on* and *home_try_on*→*purchase*.

A:

The conversion rate from people who tried the glasses at home to people who purchased them is 66 %.

```
--the result-table from question 5 is put in a WITH  
--statement AS 'structured_data'
```

```
'home_try_on_step' AS (  
  SELECT COUNT (is_home_try_on) as 'tried'  
  FROM structured_data  
  WHERE is_home_try_on = 'True'),  
'purchase_step' AS (  
  SELECT COUNT (is_purchase) as 'buyers'  
  FROM structured_data  
  WHERE is_purchase = 'True'),  
'step_counts' AS (SELECT *  
  FROM purchase_step  
  CROSS JOIN home_try_on_step)  
SELECT 100.0*buyers/tried AS  
'conversion_rate_try-on_to_purchase_in_%'  
FROM step_counts;
```

conversion_rate_try-on_to_purchase_in_%

66.0

2.6 Home Try-On Funnel – 6.2.2: How did I proceed and why

Again, we can proceed like in question 6.2.1, swapping ,`quiz_step`' for ,`purchase_step`' and modify the calculation accordingly. The changes to the query are marked **bold**.

```
--the result-table from question 5 is put in a WITH
--statement AS 'structured_data'

'home_try_on_step' AS (
  SELECT COUNT (is_home_try_on) as 'tried'
  FROM structured_data
  WHERE is_home_try_on = 'True'),
'purchase_step' AS (
  SELECT COUNT (is_purchase) as 'buyers'
  FROM structured_data
  WHERE is_purchase = 'True'),
'step_counts' AS (SELECT *
  FROM purchase_step
  CROSS JOIN home_try_on_step)
SELECT 100.0*buyers/tried AS
'conversion_rate_try-on_to_purchase_in_%'
FROM step_counts;
```

2.6 Home Try-On Funnel – 6.2.3: Comparing Conversion

- We can compare conversion from *quiz*→*home_try_on* and *home_try_on*→*purchase*.

```
--The query is too long to show here but is available  
--in the .sql file
```

As we can see the conversion rate from quiz to try-on is higher then the conversion rate from try on to purchase.

The reasons for this could e.g. be:

- While trying the frames at home is free (as noted on Warby Parker's website) the (potential) customers need to invest money to actually purchase the frames
- The standardized quiz might be more compelling than the non-standardized try-on at home.

Still, the conversion rates seem unrealistically high for an e-commerce use case.

conversion_rate_quiz_to_try-on_in_%	conversion_rate_try-on_to_purchase_in_%
75.0	66.0

2.6 Home Try-On Funnel – 6.2.3: How did I proceed and why

If we take all three temporary tables: `quiz_step`, `home_try_on_step`, `purchase_step` and calculate the conversion rates for the two funnel steps like before in 6.2.1 and 6.2.2. We take two distinct temporary tables for those calculations.

In the end we join the calculation tables into one, to show us the results juxtaposed.

```
--The query is too long to show here but is available  
--in the .sql file
```


2.6 Home Try-On Funnel - 6.3.1: Funnel A/B-Test

Q:

- We can calculate the difference in purchase rates between customers who had 3 *number_of_pairs* with ones who had 5.

```
--The query is too long to show here but is available  
-in the .sql file
```

A:

The conversion rates differ between the 3 and 5 frames setting. In the 1st step, from quiz to try-on, the conversion rate is slightly higher. An actual effect is seen in step 2. There the conversion rate goes up 26,22 percentage points. In relative terms, this is an improvement by 49,44 % and therefore nearly half.

While the number of frames is shown in the end of the quiz (see right screenshot), which doesn't seem to show much effect. The effect of actually being able to try more frames at home is much stronger.

no_of_frames	conversion_rate_quiz_to_try-on_in_%	conversion_rate_try-on_to_purchase_in_%
3	37.9	53.03
5	37.1	79.25

Pick 5 frames

Free trial (shipping both ways is on us)

Try them on at home

You've got 5 days to test 'em out :-)

Find a favorite?

Upload your prescription at checkout and we'll get cracking on your new pair (starting at \$95, including prescription lenses!)

Cool! Show me my results.

2.6 Home Try-On Funnel – 6.3.1: How did I proceed and why

To directly compare the conversion rates of the 3 & 5 frames cases, we need to discriminate between those cases for the funnel-steps Try-on and purchase. The quiz stage isn't different therefore we can use it the same way twice.

We take the 6.2.3 code and make two copies: one copy gets „3“ as a marker, the other one gets „5“.

To make it more legible we need to add another column which shows which conversion rates are from which case. This can be achieved by creating an extra column in the `1st_step_conversion` temporary tables. Which will be `LEFT JOIN`ed to their respective `2nd_step_conversions`. As there is only one row in the first `LEFT JOIN`s we don't need to say `ON` what shall be joined.

Because the column is named the same in cases 3 and 5, the subsequent `UNION` will stack them in the same column.

```
--The query is too long to show here but is available  
--in the .sql file
```

2.6 Home Try-On Funnel – 6.3.2: Volume of Funnel Steps

Q:

To achieve a complete impression of the two funnels, I am interested in:

- How many people are in each step of the funnels?

```
--The query is too long to show here but is available  
--in the .sql file
```

A:

There are 1000 participants in the quiz in both cases as the quiz stage doesn't differ. In the try-on stage we have 379 testers in the 3 frames case and 371 in the 5 frames case. Of those 201 bought, when they got 3 frames and 294 bought, when they got 5 frames. As already used in the queries for the 6.2 questions, the relationship between number of people in the funnel steps and the conversion rates are:

$$\text{conversion rate } \textit{step } n \text{ to } \textit{step } n+1 = \frac{\text{number of People in } \textit{step } n+1}{\text{number of people in } \textit{step } n}$$

As we have the information on the step-volumes and the conversion rates available, they are visualized in 6.3.3.

no_of_pairs	quiz_participants	testers	buyers
3	1000	379	201
5	1000	371	294

2.6 Home Try-On Funnel – 6.3.2: How did I proceed and why

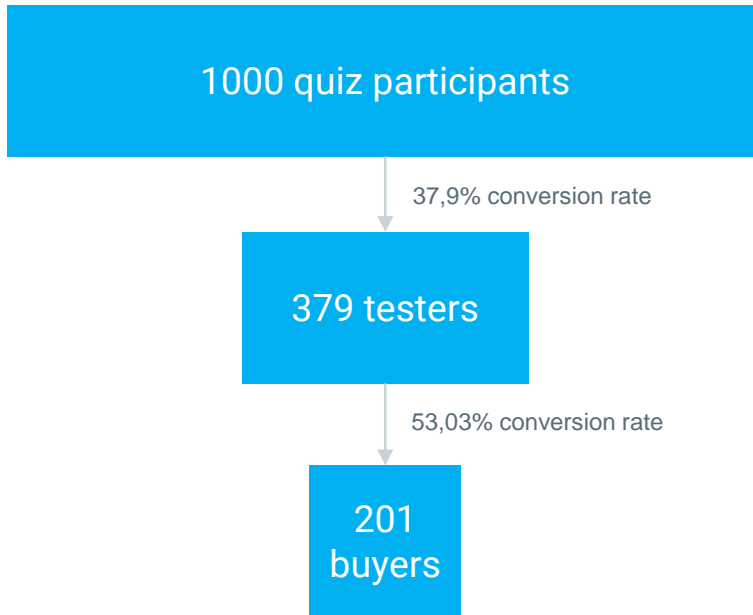
We use the `,structured_data'` temporary table. We then use temporary tables for the single funnel steps in the 3 and 5 cases. They also get a labeling column `,no_of_pairs'` with the respective amount of glasses. Those are then `UNIONed` and `JOINED` using more temporary tables. The labeling column also is used to `LEFT JOIN ON`.

In this way the temporary tables can be rearranged later on for different purposes.

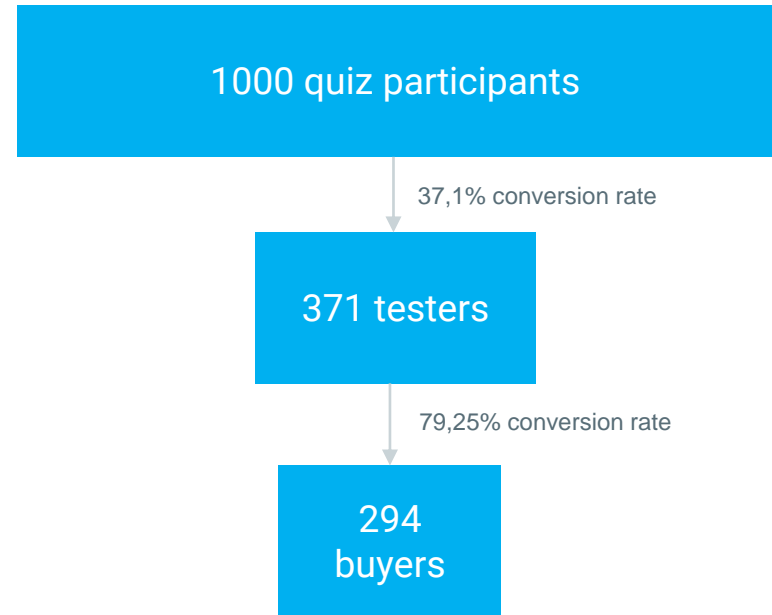
```
--The query is too long to show here but is available  
-in the .sql file
```

2.6 Home Try-On Funnel – 6.3.3: Two Funnels

Funnel for 3 Frames



Funnel for 5 Frames



2.6 Home Try-On Funnel – 6.4: A different Way

Q:

- What else can we do with those data?

A:

If we do not want to solve the sorting of data with the combination of temporary tables. We can also use a combination of **GROUP BY** and **WHERE**. This is shown in the Query on the right. Here we have the buyers **GROUPed** by the number of pairs they received for the home try-on.

number_of_pairs	buyers
3	201
5	294

```
--the result-table from question 5 is put in a WITH  
--statement AS 'structured_data'
```

```
SELECT number_of_pairs,  
       COUNT (is_purchase) as 'buyers'  
FROM structured_data  
WHERE is_purchase = 'True' AND  
      (number_of_pairs = 3 OR  
       number_of_pairs = 5)  
GROUP BY number_of_pairs;
```

2.6 Home Try-On Funnel – 6.4: How did I proceed and why

As the `structured_data` table contains all the information we need, we directly start to `SELECT` the data for `,number_of_pairs'` and `COUNT` the buyers by using `,is_purchase'`.

To only `COUNT` the rows where an actual purchase was made and where people were sent frames, we use `WHERE`, here we combine `AND` and `OR`. The `OR` has to be set in parantheses, to combine `AND` and `OR` in the right order. Then we `GROUP` those data by the `,number_of_pairs'`.

```
--the result-table from question 5 is put in a WITH  
--statement AS 'structured_data'
```

```
SELECT number_of_pairs,  
       COUNT (is_purchase) as 'buyers'  
FROM structured_data  
WHERE is_purchase = 'True' AND  
      (number_of_pairs = 3 OR  
       number_of_pairs = 5)  
GROUP BY number_of_pairs;
```

2.6 Quiz-Funnel – 6.5: Quiz answers

Q:
We can also use the original tables to calculate things like:

- The most common results of the style *quiz*.

A:
The most common results of the quiz can be found when showing rankings of the answers for each question.
The most common answers are

- What are you looking for? Women's Styles.
- What's your fit? Narrow.
- Which shapes do you like? Rectangular.
- Which colors do you like? Tortoise.

```
SELECT style,  
       COUNT (style) AS 'number_of_answers'  
FROM quiz  
GROUP BY style  
ORDER BY COUNT (style) DESC;
```

```
SELECT fit,  
       COUNT (fit) AS 'number_of_answers'  
FROM quiz  
GROUP BY fit  
ORDER BY COUNT (fit) DESC;
```

```
SELECT shape,  
       COUNT (shape) AS 'number_of_answers'  
FROM quiz  
GROUP BY shape  
ORDER BY COUNT (shape) DESC;
```

```
SELECT color,  
       COUNT (color) AS 'number_of_answers'  
FROM quiz  
GROUP BY color  
ORDER BY COUNT (color) DESC;
```


2.6 Quiz-Funnel – 6.5: Tables

shape	number_of_answers
Rectangular	397
Square	326
Round	180
No Preference	97
color	number_of_answers
Tortoise	292
Black	280
Crystal	210
Neutral	114
Two-Tone	104

style	number_of_answers
Women's Styles	469
Men's Styles	432
I'm not sure. Let's skip it.	99
fit	number_of_answers
Narrow	408
Medium	305
Wide	198
I'm not sure. Let's skip it.	89

2.6 Quiz-Funnel – 6.5: Quiz answers

We can **COUNT** the distinct answers by **GROUPING** them. To make the results clear we can also **ORDER** them **DESCending**, so the most common answers are on the top.

We do this for every table and separate the queries, so we have all the answers on one page.

```
SELECT style,
       COUNT (style) AS 'number_of_answers'
FROM quiz
GROUP BY style
ORDER BY COUNT (style) DESC;
```

```
SELECT fit,
       COUNT (fit) AS 'number_of_answers'
FROM quiz
GROUP BY fit
ORDER BY COUNT (fit) DESC;
```

```
SELECT shape,
       COUNT (shape) AS 'number_of_answers'
FROM quiz
GROUP BY shape
ORDER BY COUNT (shape) DESC;
```

```
SELECT color,
       COUNT (color) AS 'number_of_answers'
FROM quiz
GROUP BY color
ORDER BY COUNT (color) DESC;
```

2.6 Purchase - 6.6: Purchase Types

Q:

The most common types of *purchase* made:

- Which model is the most sold?

A:

The most sold model is called „Eugene Narrow“.

With this information we could base the next Warby Parker marketing campaign on this model, as it could create the most pull.

From the same table we can also see that the model „Monocle“ is the least sold, maybe we can offer a discount on this one to sell more.

```
SELECT model_name,  
       COUNT(model_name) AS 'units_sold'  
FROM purchase  
GROUP BY model_name  
ORDER BY COUNT (model_name) DESC;
```

model_name	units_sold
Eugene Narrow	116
Dawes	107
Brady	95
Lucy	86
Olive	50
Monocle	41

2.6 Purchase – 6.6: How did I proceed and why

For identification of each **COUNT**, the **model_name** is also **SELECT**ed. The **COUNT**s column is aliased as **'units_sold'** for a clear labeling. To **COUNT** the different **model_names**, they are **GROUP**ed and **ORDER**ed **DESC**ending for a clear presentation of the most sold.

```
SELECT model_name,  
       COUNT(model_name) AS 'units_sold'  
FROM purchase  
GROUP BY model_name  
ORDER BY COUNT (model_name) DESC;
```

2.6 Purchase – 6.7: Revenue

Q:

- Which model brings the most revenue?

A:

As the models have different prices it's also valuable to know which one makes the most revenue.

As we can see it is not the most sold model „Eugene Narrow“ from before but the „Dawes“ model.

```
SELECT model_name,  
       SUM (price) AS 'revenue'  
FROM purchase  
GROUP BY model_name  
ORDER BY 2 DESC;
```

model_name	revenue
Dawes	16050
Lucy	12900
Eugene Narrow	11020
Brady	9025
Olive	4750
Monocle	2050

2.6 Quiz-Funnel – Query 6.6: How did I proceed and why

We **SELECT** the **model_name** to have labeling column in the end and **SUM (price)** as the sum of all sold items' prices is the revenue.
To get the **SUM** by model we **GROUP** them by **model_name**.
To make it more clear, the table is **ORDER**ed by it's second column and is shown **DESC**ending.

```
SELECT model_name,  
       SUM (price) AS 'revenue'  
FROM purchase  
GROUP BY model_name  
ORDER BY 2 DESC;
```