codecademy

# Capstone: Usage Funnels

Learn SQL from Scratch

Kyle Beacham

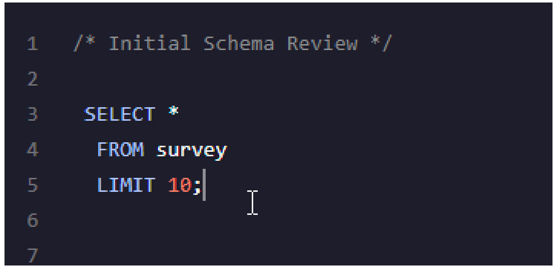
11/03/2018

## Table of Contents

# 1.0 - Basic Select Statements

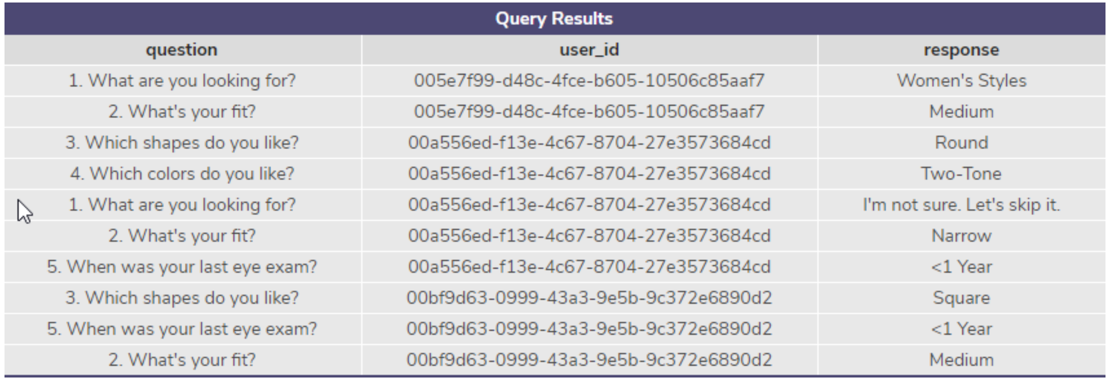
To start reviewing the data that actually appears in the tables, you can use a basic select statement to see the results.

- Take note of the formatting of data in the columns
- Take note of what might be important data points



```
/* Initial Schema Review */

 SELECT *
  FROM survey
  LIMIT 10;
```



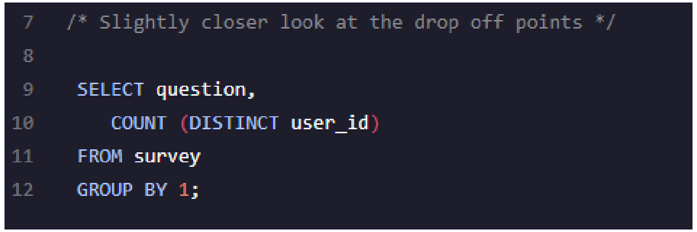
Query Results

| 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 |

# 1.1 - Basic Select Statements

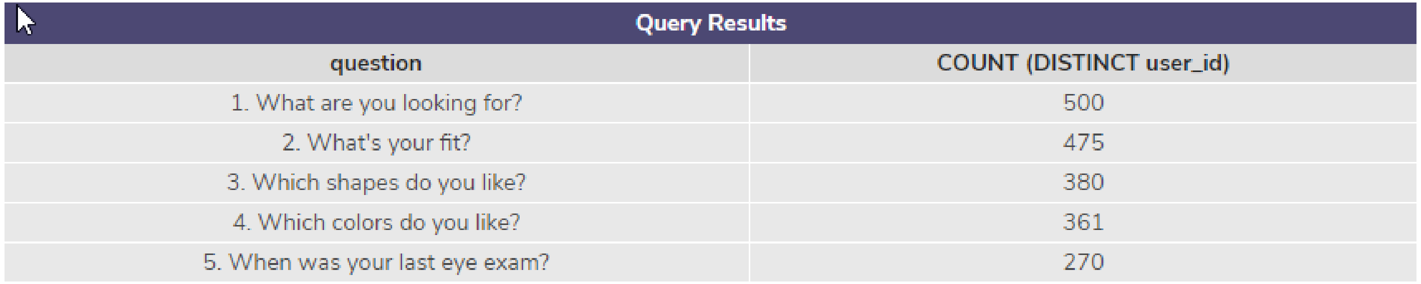
Once you know what data is in the table, you can start to analyze it

- Using the formatting and the important data points that we observed in the first select statement, we are able to start to review the details of the data
- Using the COUNT and GROUP BY functions lets you start to view where users are dropping off during their survey and allows you to see who finished the survey.



```sql
/* Slightly closer look at the drop off points */

SELECT question,
   COUNT (DISTINCT user_id)
FROM survey
GROUP BY 1;
```



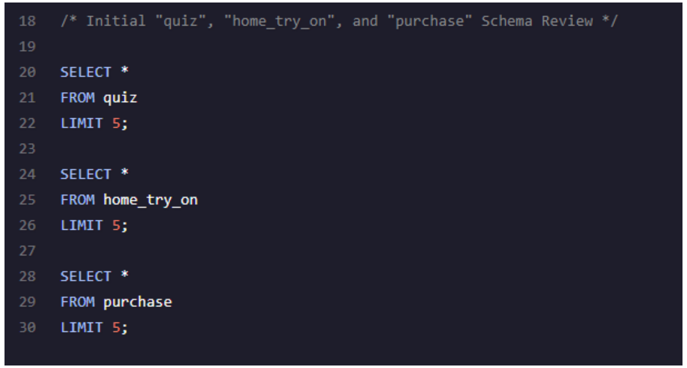
Query Results

| question | COUNT (DISTINCT user_id) |
|---|---|
| 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 |

## 1.2 - Basic Select Statements

The same principal of reviewing the data should take place even when there are multiple tables involved.

- Take note of the columns that match between tables
- Take note of the important pieces of data we may want to analyze from each of the tables

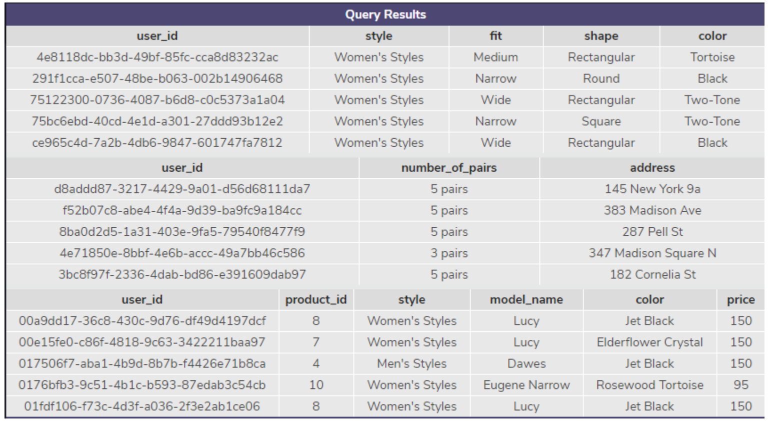


```sql
/* Initial "quiz", "home_try_on", and "purchase" Schema Review */

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

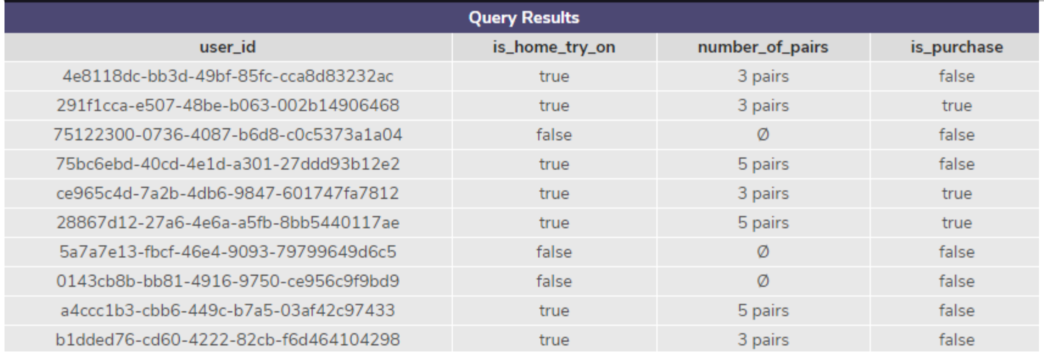
| user_id | number_of_pairs | address |
|---|---|---|
| d8addd87-3217-4429-9a01-d56d68111da7 | 5 pairs | 145 New York 9a |
| f52b07c8-abe4-4f4a-9d39-ba9fc9a184cc | 5 pairs | 383 Madison Ave |
| 8ba0d2d5-1a31-403e-9fa5-79540f8477f9 | 5 pairs | 287 Pell St |
| 4e71850e-8bbf-4e6b-accc-49a7bb46c586 | 3 pairs | 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 |

## 2.0 – Cases, Aliases, Joins

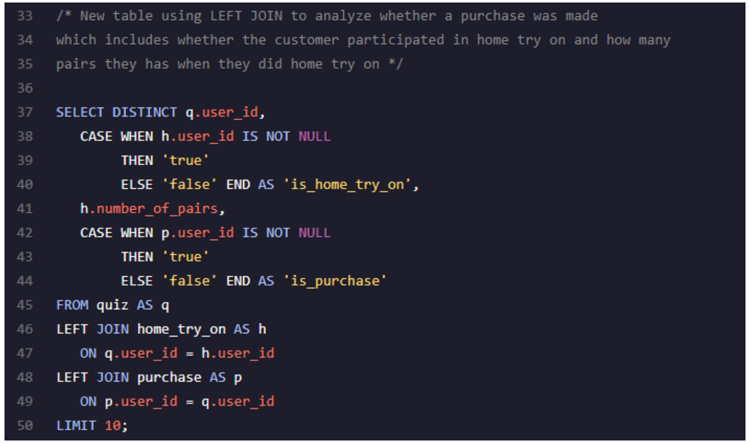
Once you are familiar with the tables you need to utilize more in-depth functions to review that data.

- Cases allow you to take action based on the specific results to the statement. As you can see here, we wanted to apply true and false values to a column to make it easier to read.
- Aliases allow you to reference tables easier. This makes the code easier to read and quicker to write
- Joins allow you to combine the data between the tables. Combined with aliases you can write detailed efficient code



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 | Ø | 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 | Ø | false |
| 0143cb8b-bb81-4916-9750-ce956c9f9bd9 | false | Ø | false |
| a4ccc1b3-cbb6-449c-b7a5-03af42c97433 | true | 5 pairs | false |
| b1dded76-cd60-4222-82cb-f6d464104298 | true | 3 pairs | false |



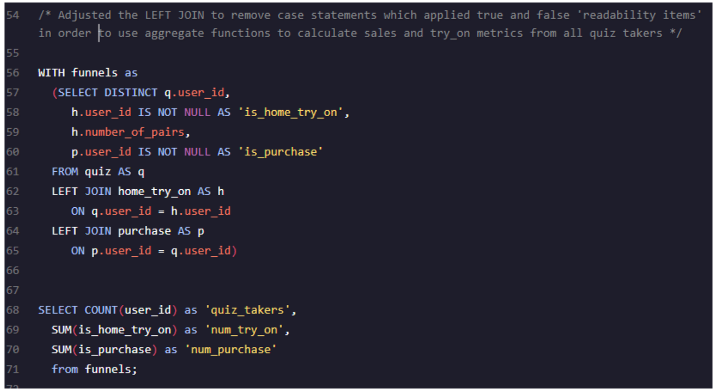
```
/* New table using LEFT JOIN to analyze whether a purchase was made
which includes whether the customer participated in home try on and how many
pairs they has when they did home try on */

SELECT DISTINCT q.user_id,
   CASE WHEN h.user_id IS NOT NULL
        THEN 'true'
        ELSE 'false' END AS 'is_home_try_on',
   h.number_of_pairs,
   CASE WHEN p.user_id IS NOT NULL
        THEN 'true'
        ELSE 'false' END 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 p.user_id = q.user_id
LIMIT 10;
```

## 3.0 – Temporary Tables & Aggregate Functions

We were able to take the SELECT statement from the previous exercise and utilize that as a table & a resource to pursue further analytics on the overall data.

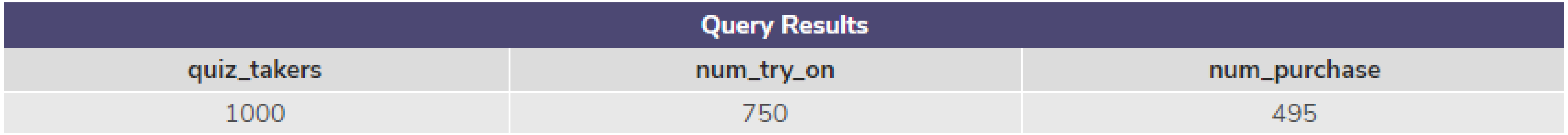
- By using WITH you can create that temporary table
- Using aliases, again, allows quick reference
- You can run new queries (including aggregate functions) on the data that we built into the temporary tables.



```sql
/* Adjusted the LEFT JOIN to remove case statements which applied true and false 'readability items'
in order to use aggregate functions to calculate sales and try_on metrics from all quiz takers */

WITH funnels as
  (SELECT DISTINCT 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 p.user_id = q.user_id)


SELECT COUNT(user_id) as 'quiz_takers',
  SUM(is_home_try_on) as 'num_try_on',
  SUM(is_purchase) as 'num_purchase'
  from funnels;
```

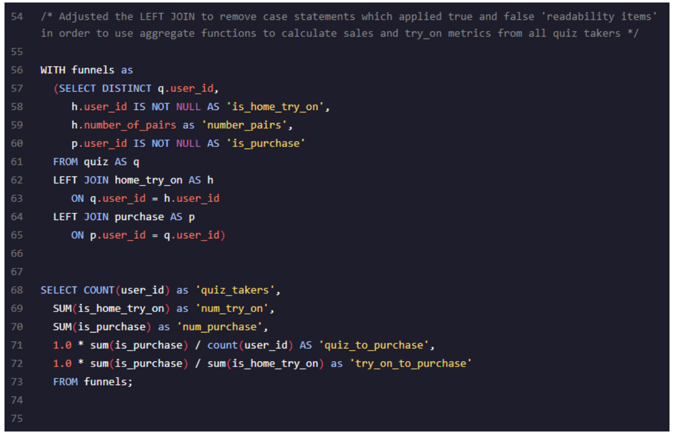


| Query Results | | |
|---|---|---|
| quiz_takers | num_try_on | num_purchase |
| 1000 | 750 | 495 |

## 3.1 – Temporary Tables & Aggregate Functions

By expanding upon the aggregate functions we can get a better picture of the usage funnels.

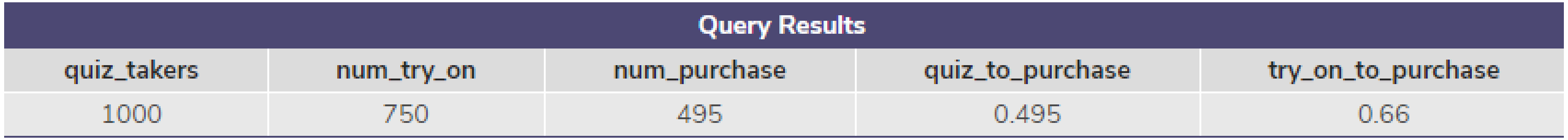
- As you can see, we were able to not only view the total number of users, but which ones made it to try on, which made it to purchase.
- Along with the raw numbers of how many made it, we were able to see the percentages for who made it through to various stages.



```sql
/* Adjusted the LEFT JOIN to remove case statements which applied true and false 'readability items'
in order to use aggregate functions to calculate sales and try_on metrics from all quiz takers */

WITH funnels as
  (SELECT DISTINCT q.user_id,
     h.user_id IS NOT NULL AS 'is_home_try_on',
     h.number_of_pairs as 'number_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 p.user_id = q.user_id)


SELECT COUNT(user_id) as 'quiz_takers',
  SUM(is_home_try_on) as 'num_try_on',
  SUM(is_purchase) as 'num_purchase',
  1.0 * sum(is_purchase) / count(user_id) AS 'quiz_to_purchase',
  1.0 * sum(is_purchase) / sum(is_home_try_on) as 'try_on_to_purchase'
  FROM funnels;
```



Query Results

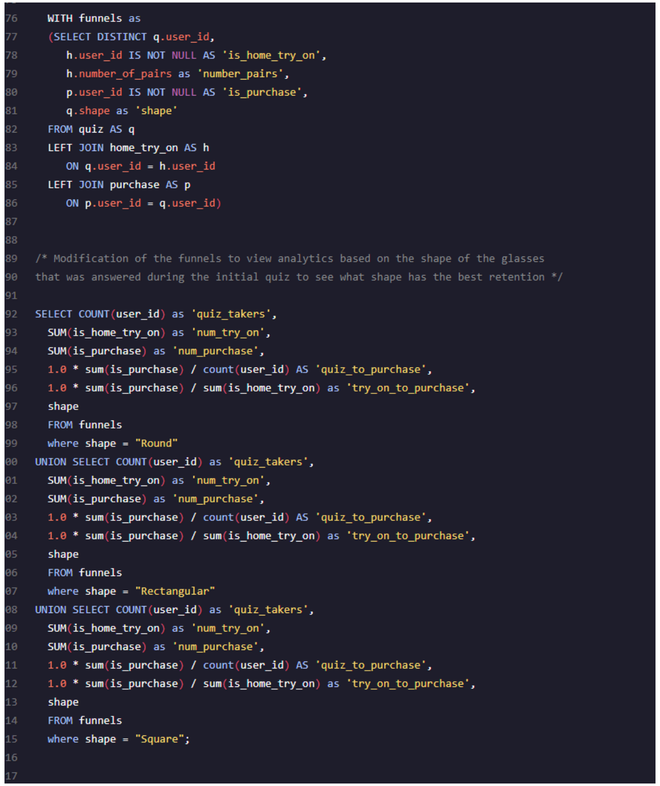
| quiz_takers | num_try_on | num_purchase | quiz_to_purchase | try_on_to_purchase |
|---|---|---|---|---|
| 1000 | 750 | 495 | 0.495 | 0.66 |

## 4.0 – Further Analytics

There is virtually no limit to what you can do with the power of SQL in a database environment. As long as the data is available you can analyze it.

Below, we wanted to consider which users made it from the quiz, to try on, and to purchase based on the answer to the "What shape do you prefer" question during the quiz.

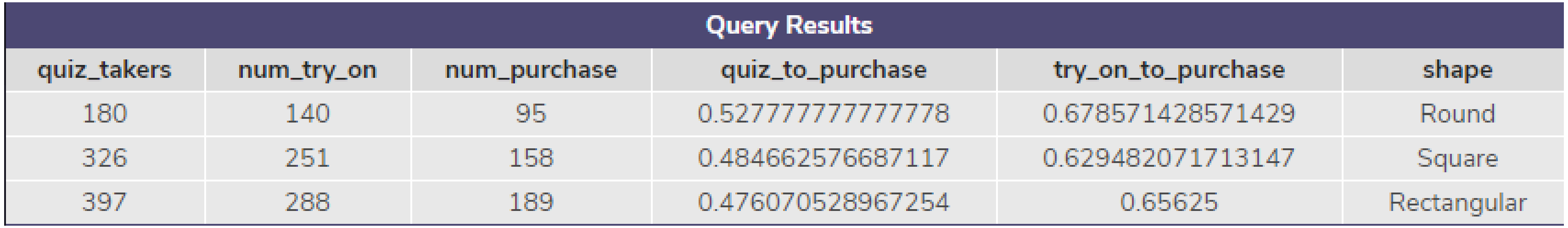
If we had even more data and more time, we would also be able to analyze which of these users actually bought glasses which corresponded to various questions during the quiz and which they tried on.



```sql
  WITH funnels as
  (SELECT DISTINCT q.user_id,
     h.user_id IS NOT NULL AS 'is_home_try_on',
     h.number_of_pairs as 'number_pairs',
     p.user_id IS NOT NULL AS 'is_purchase',
     q.shape as 'shape'
  FROM quiz AS q
  LEFT JOIN home_try_on AS h
     ON q.user_id = h.user_id
  LEFT JOIN purchase AS p
     ON p.user_id = q.user_id)


/* Modification of the funnels to view analytics based on the shape of the glasses
that was answered during the initial quiz to see what shape has the best retention */

SELECT COUNT(user_id) as 'quiz_takers',
  SUM(is_home_try_on) as 'num_try_on',
  SUM(is_purchase) as 'num_purchase',
  1.0 * sum(is_purchase) / count(user_id) AS 'quiz_to_purchase',
  1.0 * sum(is_purchase) / sum(is_home_try_on) as 'try_on_to_purchase',
  shape
  FROM funnels
  where shape = "Round"
UNION SELECT COUNT(user_id) as 'quiz_takers',
  SUM(is_home_try_on) as 'num_try_on',
  SUM(is_purchase) as 'num_purchase',
  1.0 * sum(is_purchase) / count(user_id) AS 'quiz_to_purchase',
  1.0 * sum(is_purchase) / sum(is_home_try_on) as 'try_on_to_purchase',
  shape
  FROM funnels
  where shape = "Rectangular"
UNION SELECT COUNT(user_id) as 'quiz_takers',
  SUM(is_home_try_on) as 'num_try_on',
  SUM(is_purchase) as 'num_purchase',
  1.0 * sum(is_purchase) / count(user_id) AS 'quiz_to_purchase',
  1.0 * sum(is_purchase) / sum(is_home_try_on) as 'try_on_to_purchase',
  shape
  FROM funnels
  where shape = "Square";
```



Query Results

| quiz_takers | num_try_on | num_purchase | quiz_to_purchase | try_on_to_purchase | shape |
|---|---|---|---|---|---|
| 180 | 140 | 95 | 0.527777777777778 | 0.678571428571429 | Round |
| 326 | 251 | 158 | 0.484662576687117 | 0.629482071713147 | Square |
| 397 | 288 | 189 | 0.476070528967254 | 0.65625 | Rectangular |