Warby Parker Capstone

Learn SQL from Scratch

Christopher Salonia

2/25/19

Question 1. What columns does the table have? Select the first ten rows.

The different columns that are within the **survey** table are the following:

question

user_id

response

Using the **SELECT *** means querying ALL columns that appear within the survey table. It was also requested to limit to the first 10 entries of the table (**the LIMIT <enter # of rows> function**) and ending with a semi-colon to close the query

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

Question 2. *What is the number of responses for each question within the table?*

To get this response, you first SELECT the **question** column along with the **DISTINCT COUNT** of user_id that answered each question within the survey table

Grouping by the **question** column gives an accurate count of users traveling through the funnel. Without the **Group by**, not all questions would be captured in the query. I have also added an alias (**AS <renamed column of choice>)** to rename the column to ‘responses’

```
 SELECT question,
   COUNT(DISTINCT user_id) AS ‘responses’
 FROM survey
GROUP BY question;
```

question	responses
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

Question 3. *Which question(s) of the quiz have a lower completion rates and why?*

The lowest completion rate is Question 5 with **270** (74% completion rate) responses from **361** responses in Question 4 followed by Question 3 with **380** responses (80% completion rate) from **475** in Question 2. In my review, Questions 1 through 4 are opinion based questions while the 5th question is based off an event date in the past that the user may not remember off-hand.

NOTE: Spreadsheet software was used to calculate the completion rates

Question	Responses	Completion Rates
1. What are you looking for?	500	100%
2. What’s your Fit?	475	95%
3. Which Shapes do you like?	380	80%
4. Which colors do you like?	361	95%
5. When was your last eye exam?	270	74%

```
 SELECT question,
   COUNT(DISTINCT user_id) AS ‘responses’
 FROM survey
GROUP BY question;


EXCEL FORMULAS USED TO ANSWER QUESTION 3.
```

Question 4. What are the column names for the 3 different tables?

The **quiz** table has the following 5 columns using the **SELECT *** clause:
user_id, style, fit, shape, color

The **home_try_on** table has the following 3 columns using the **SELECT *** clause:
user_id, number_of_pairs, address

The **purchase** table has the following 6 columns using the **SELECT *** clause:
user_id, product_id, style, model_name, color, price

The 3 tables are **separately** queried by SELECT * from **<table name>** while limiting 5 entries (as requested) for the Analyst's review of the data therein. **Limiting** the number of outputs allows for the ability to quickly review without summoning the whole table which could run slowly depending on how much data is in the table.

```
SELECT *
FROM quiz
LIMIT 5;

SELECT *
FROM home_try_on
LIMIT 5;

SELECT *
FROM purchase
LIMIT 5;
```

Question 5. Use a Left Join to combine the 3 tables (starting from the top of the funnel)

We must first **SELECT** the **DISTINCT** user_ids from the table that is the top of the funnel (Quiz table). We then query the user_id from home_try_on table (traveling to the middle of the funnel). **TRUE** returns ‘1’ while **FALSE** returns ‘0’. TRUE means a Try-on kit was ordered with the similar rule reflecting the is_purchase column (‘1’ is a purchase and ‘0’ is not). I have aliased each table as the first letter of the name to make it easier to query (ex. **LEFT JOIN** home_try_on AS **‘h’**). Blanks within a row are **NULL** values. **Number_of_pairs** are the number of pairs for each user_id listed in the query

```
 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 h.user_id = q.user_id
LEFT JOIN purchase AS 'p'
             ON p.user_id = h.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		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

Question 6. What Kind of Insights can we draw?

I created a subquery using the **WITH <insert subquery table name of choice> AS** clause that takes you through the funnel from quiz to purchase. Using the query from the previous question except adding a **LIMIT** of **100** entries to expand the sample size, I can see that from the top of the funnel (quiz table) to the bottom of the funnel (purchase table), the overall conversion rate is 41% while the conversion comparison for each step is the following:

From the **quiz** to **home_try_on** tables, it's a 68% conversion

From the **home_try_on** to **purchase** tables, it's 60% conversion

Numbers can sometimes be deceiving! If I hadn't expanded the sample size to 100 users, the **home_try_on** to **purchase** rate was listed as 42% for the first 10 users which wouldn't give a good representation of the conversion in the last step (~18% in differential variance between 100 users and 10 users!).

num_browse	num_try_on	num_purchase	browse_to_home_try_on	home_to_purchase
100	68	41	0.68	0.602941176470588

num_browse	num_try_on	num_purchase	browse_to_home_try_on	home_to_purchase
10	7	3	0.7	0.428571428571429

```
 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 h.user_id = q.user_id
LEFT JOIN purchase AS 'p'
              ON p.user_id = h.user_id
  LIMIT 100)
SELECT COUNT(*) AS 'num_browse',
   SUM(is_home_try_on) AS 'num_try_on',
   SUM(is_purchase) AS 'num_purchase',
   1.0 * SUM(is_home_try_on) / COUNT(user_id) AS
'browse_to_home_try_on',
   1.0 * SUM(is_purchase) / SUM(is_home_try_on) AS
'home_to_purchase'
FROM funnels;
```