codecademy

Usage Funnel

Learn SQL from Scratch

Michael Clune

12/21/18

1. Quiz Funnel

Question 1 – Columns in Survey

Using the code on the right I was able to determine the columns in the table survey.

Question	User_id	Response

```
-- You can put your query here

SELECT *
FROM quiz
LIMIT 10;
```

Question 2 – Count of Responses

This code can be applied specifically to this table because the questions only “count” if they are answered therefore counting the questions answered provides the same results as searching for distinct user_id (as suggested in the hint)

The Results –
Question 1 – 500 responses
Q2 – 475
Q3 – 380
Q4 – 361
Q5 – 270

```
-- You can put your query here

select count (question), question
from survey
group by question;
```

Question 3 – Completion Rates

Completion Rates –
Q1 - 500/500 = 100%
Q2 - 475/500 = 95%
Q3 - 380/475 = 80%
Q4 - 361/380 = 95%
Q5 - 270/361 = 75%

Question 5 with 75% and question 3 at 80%

Question 5 because it's the end so people that made it that far just quit and question 3 because people felt they didn't need to answer that question to get an accurate result

These percentages were simple to calculate with a calculator so I did not use Excel or Sheets.

Lastly the overall participant total of 500 was not used everytime because this format shows the completion compared to only the users that answered the previous question – so while 75% of the people answered question 5 that had answered question 4, only a little more than 50% of the total participants actually answered the question. This is the difference between asking where participants gave up vs percentage of people that answered each question.

```
-- You can put your query here

select count (question), question
from survey
group by question;
```

Question 4 – Home Try-On Funnel

The code to the right produced 5 rows in each table allowing me to see the columns in use, limit 1 would accomplish the same.
The tables below are listed in the order of the script – Quiz, Home try on, Purchase.

Quiz

User_id	Style	Fit	Shape	Color	

Home Try-On

User_id	Number_of_Pairs	Address

Purchase

User_id	Product_id	Style	Model_name	Color	Price

```
-- You can put your query here

select * from quiz limit 5;
select * from home_try_on limit 5;
select * from purchase limit 5;
```

Question 5 – Left Join Table Creation

The code on the right produced the table as requested in the question – the number 1 respresenting True and 0 as False.

I created an example of one row of results in the table below

User_id	Is_home_try_on	Number_of_pairs	Is_purchase
User_id example	1	3 pairs	0

```
-- You can put your query here

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 q
LEFT JOIN home_try_on h
   ON q.user_id = h.user_id
LEFT JOIN purchase p
   ON p.user_id = q.user_id
LIMIT 10;
```

Question 6 – Actionable Insights

Actionable insights – First based on the questions in the survey and their percentage of answers – I would move the colors question in front of shapes and if the eye exam is important (ie cannot purchase without having had one) this question should arguably be first and include a note that purchases cannot be made without an active prescription which is obtained via an eye exam within the last year.

Lastly using the SQL script on the right I was able to count the number of purchases when sending out 3 pairs vs 5 pairs. The numbers 201 “3 Pairs” and 294 “5 pairs” suggests that sending 5 pairs does increase likelihood of purchase, however the number is a small enough difference – only 93 – out of 495 purchases made would actually show that they would probably save more money on shipping and returns if they only sent 3 pairs, while not losing too much business.

There should be a next step in their data as well that would show what percentage of people that asked for a home try on but did not make a purchase came back for a second home try on to see if perhaps they changed some answers to get different glasses via the quiz and try again or if they just gave up altogether after the first try on.

```
-- You can put your query here

SELECT count (h.number_of_pairs), q.user_id,
   h.user_id IS NOT NULL AS 'is_home_try_on',
   p.user_id IS NOT NULL AS 'is_purchase'
FROM quiz q
JOIN home_try_on h
   ON q.user_id = h.user_id
JOIN purchase p
   ON p.user_id = q.user_id
   where h.number_of_pairs = '3 pairs';

SELECT count (h.number_of_pairs), q.user_id,
   h.user_id IS NOT NULL AS 'is_home_try_on',
   p.user_id IS NOT NULL AS 'is_purchase'
FROM quiz q
JOIN home_try_on h
   ON q.user_id = h.user_id
JOIN purchase p
   ON p.user_id = q.user_id
   where h.number_of_pairs = '5 pairs';
```