SQL Capstone Templates

You can use these templates in the capstone presentation for Learn SQL from Scratch!

Copy the ones you'd like to use and write your own content.

codecademy

Capstone Project

Learn SQL from Scratch

Jessica Dance

July 2018

Example Table of Contents

1. Usage Funnels with Warby Parker

1. Columns for Survey table

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

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

2 What is the number of responses for each question?

SELECT – columns ’questions’ and User_id to show the two colums
COUNT(DISTINCT User_id) – to count the number of users
GROUP BY – groups the number of users per the questions in the servey

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

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

3 Percentage of users who answer each question

Which question(s) of the quiz have a lower completion rates?

- *Question 5 has the lowest completion rate.*

What do you think is the reason?

- This could be due to many people not having done their eye exam, thus they are unable to answer the question.

Question	COUNT(DISTINCT user_id)	Percentage
1. What are you looking for?	500	100%
2. What's your fit?	475	95%
3. Which shapes do you like?	380	76%
4. Which colors do you like?	361	72%
5. When was your last eye exam?	270	54%

4 What are the column names?

Showing all the columns of the three tables:

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

```
SELECT *
FROM quiz
LIMIT 5;

SELECT *
FROM home_try_on
LIMIT 5;

SELECT *
FROM purchase
LIMIT 5;
```

user_id	style	fit	shape	color	
4e8118dc	Women's Styles	Medium	Rectangular	Tortoise	
291f1cca-	Women's Styles	Narrow	Round	Black	
75122300	Women's Styles	Wide	Rectangular	Two-Tone	
75bc6ebd	Women's Styles	Narrow	Square	Two-Tone	
ce965c4d	Women's Styles	Wide	Rectangular	Black	
user_id	**number_of_pairs**	**address**			
d8addd87	5 pairs	145 New York 9a			
f52b07c8-	5 pairs	383 Madison Ave			
8ba0d2d5	5 pairs	287 Pell St			
4e71850e	3 pairs	347 Madison Square N			
3bc8f97f-2	5 pairs	182 Cornelia St			
user_id	**product_id**	**style**	**model_name**	**color**	**price**
00a9dd17	8	Women's Styles	Lucy	Jet Black	150
00e15fe0-	7	Women's Styles	Lucy	Elderflower Crystal	150
017506f7-	4	Men's Styles	Dawes	Jet Black	150
0176bfb3-	10	Women's Styles	Eugene Narrow	Rosewood Tortoise	95
01fdf106-	8	Women's Styles	Lucy	Jet Black	150

5 New table layout

I started with with SELECT funtion, starting with q.user_id from the quiz table, h.user_id which I place in a CASE statement in order for the integers to return ‘True’ if the interger was not null/0 otherwise ‘false’ if that was the case, finally I ended the stament with END and renamed it is_home_try_on, if this is not done then the entire statement will be shown as the title.

The same is done with p.user_id in order to return the same results.
In FROM we start our LEFT JOIN. The JOINs will be added on the table ‘quiz’ with columns from home_try_on as follows q.user_id = h.user_id and from table purchase p.user_id = q.user_id thus pulling data from all three tables to create our new table.

This is the first step of building our funnel.

```
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 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;
```

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	NULL	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	NULL	FALSE
0143cb8b-bb81-4916-9750-ce956c9f9bd9	FALSE	NULL	FALSE
a4ccc1b3-cbb6-449c-b7a5-03af42c97433	TRUE	5 pairs	FALSE
b1dded76-cd60-4222-82cb-f6d464104298	TRUE	3 pairs	FALSE

6. Convertion aggregate across all rows

I put the first part of our funnel in a with clause to create a tempory table, which I named the new table browse,
From this table I drew from all tables:
Starting with user_id, which I put into a COUNT() function to calculate the number of users.
Then I encase the is_home_try_on in a SUM() funtion and start a statement to change the ‘true’ and ’false’ statements to 1 & 0 in order to calculate the number of ‘true’(s) and false
I put them in spereate tables so that is was easier to read.
The same was done for is_purchase to calculate the number of sales.
Then I grouped by number of pairs, which presented three colums.

From this you can see that in total there are 1000 user_ids in the system out of which 250 did not partipate in home try ons. This table shows that more people tried on 3 pairs that 5 pairs. However people who took home five pairs had a higher bying rate than people who took home three pairs

This could be due to people having tried on 5 pairs had more options to choose from thus the chances of a purchase were higher.

```
WITH browse AS
(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 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)SELECT
number_of_pairs,
SUM(
CASE WHEN  IS_HOME_TRY_ON = 'true’
THEN 1ELSE 0END) AS is_home_try_on_true,
SUM(
CASE WHEN  IS_HOME_TRY_ON = 'false’
THEN 1ELSE 0END) AS
is_home_try_on_false,
SUM(  CASE WHEN is_purchase = 'true’
THEN 1ELSE 0END) AS is_purchase_true,
SUM(  CASE WHEN is_purchase =
'false'THEN 1ELSE 0END) AS
is_purchase_false
FROM browse
GROUP BY number_of_pairs;
```

num_of_user_id	number_of_pairs	is_home_try_on_true	is_home_try_on_false	is_purchase_true	is_purchase_false
250		0	250	0	250
379	3 pairs	379	0	201	178
371	5 pairs	371	0	294	77

7. Rate of sales per style

As a business owner I need to which styles are most popular.

In our browse table I have added q.style to pull the data replating the specific purchase in this category.
In my first column I placed ‘style and changed the name to ‘style_types’
Second column is COUNT() funtion to calculate the number of per these catagories.
Then again using SUM() & CASE to calculate the the number of purchases this time only focusing on purchases that are true.
Finally calculation the convertion rate by dividing the number of items per style against the total number of purchases and then grouping them by those specific styles.

Style_types	Num_ Style	Purchased	sale_rate
I'm not sure. Let's skip it.	99	0	
Men's Styles	432	243	1.77777777777778
Women's Styles	469	252	1.86111111111111

```
WITH browse AS
(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,
q.style,
CASE
WHEN
p.user_id IS NOT NULL
THEN 'true'
ELSE 'false'  END 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)
SELECT style AS 'Style_types',
count(style) AS ’num_Style',
SUM(CASE WHEN is_purchase = 'true’
THEN 1ELSE 0END) AS 'Purchased',
1.0 * COUNT(style) / SUM(CASE
WHENis_purchase = 'true'THEN 1ELSE 0END)
AS 'sale_rate'FROM browse
GROUP BY style;
```

8. Quiz to Home_try_on Conversion

Calculating the conversion from quiz to home_try_on,
Starting off by using a COUNT() funtion to calculate user_id as in our first part of the funnel the user_id was pulled from the quiz table, thus calculating the number of participants.
Them calculating the number of home_try_ons by again using the the SUM() and CASE statements.
And finally putting them together to get our conversion rate by dividing home_try_on by quiz_participants.

This shows you the from the number of people who took part in the quiz, the number of people who took items home to try on.

This will help the owners see that a good 75% of the participants did try on glasses thus we can assume the quiz did help the customers and helped push them further to a sale.

quiz_participants	Home_try_on	quiz/home_try_on_rate
1000	750	1.33333333333333

```
WITH browse AS (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 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)
SELECT count(user_id) AS
'quiz_participants',
SUM(CASE
WHEN is_home_try_on = 'true' THEN 1
ELSE 0
END) AS 'Home_try_on',1.0 *
COUNT(user_id) / SUM(CASE
WHEN is_home_try_on = 'true'
THEN 1     ELSE 0     END) AS
'quiz/home_try_on_rate'
FROM browse
LIMIT 10;
```

9. Home_try_on to Purchase Conversion

Now that we have calculated the convertion from quiz to home_try_on we can calculate the home_try_on to purchase.

Using the same formula as the last table but instead of quiz we use home_try_on and instead of home_try_on we use purchase.

Out of the 750 individuals that took home products to try on, only 495 products were purchased, which is just over 50%.

This will help the owners estimate their sales rate by the number of people that take product home to try on.

home_try_on	Purchased	home_try_on/purchase_rate
750	495	1.51515151515152

```
WITH browse AS
(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 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)
SELECT SUM(CASE
WHEN is_home_try_on = 'true'
THEN 1  ELSE 0
END) AS 'Home_try_on',
SUM(CASE
WHEN is_purchase = 'true'
THEN 1  ELSE 0
END) AS 'Purchased', 1.0 * SUM(CASE
WHEN is_home_try_on = 'true' THEN 1
ELSE 0  END) / SUM(CASE
WHEN is_purchase  = 'true'
THEN 1 ELSE 0
END) AS 'home_try_on/purchase_rate'
FROM browse
LIMIT 10;
```

10. Selling rate of the most expensive items

product_id	price	num_of_purchases
3	150	63
4	150	44
7	150	44
8	150	42
1	95	52
2	95	43
6	95	50
9	95	54
10	95	62
5	50	41
NULL	NULL	0

I thought this might be interesting especially when I shop online I always look at the most expensive and the least expensive items and compare them.

for this table I added product _id and price into our browse table.

I grouped this table by the product_id, as an owner I would like to see the number of sales per price range.

Ordering by the price in a descending order we can see that our highest purchases are of product_item 3 which turns out to be one of the higher priced items.

However not all of the most expensive items sold as well.

```
WITH browse AS(
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,
p.product_id,
p.price,
CASE WHEN p.user_id IS NOT NULL
THEN 'true' ELSE 'false'
END 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)
SELECT product_id,
price,
SUM( CASE
WHEN is_purchase = 'true'
THEN 1 ELSE 0 END)
AS 'num_of_purchases'
FROM browse
GROUP BY product_id
ORDER BY price desc;
```

11. Most purchased items

is_purchase	price	product_id
63	150	3
62	95	10
54	95	9
52	95	1
50	95	6
44	150	4
44	150	7
43	95	2
42	150	8
41	50	5
0	NULL	NULL

this was very similar to the previous table how ever ordering the table by is_purchase has pulled through varying results.

We see in this table that again one of the most expensive items had the most sales.
We see that the middle priced items have a higher selling rate and the cheapest item has the lowest selling rate.

This will help the owner sift out the items that have a poor selling rate and increase their stock on their best sellers.

```
WITH browse AS(
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,
p.product_id,
p.price,
CASE WHEN p.user_id IS NOT NULL
THEN 'true' ELSE 'false'
END 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)
SELECT
SUM( CASE WHEN is_purchase = 'true'
THEN 1 ELSE 0 END)
AS 'is_purchase',
price,
product_id
FROM browse
GROUP BY product_id
ORDER BY is_purchase DESC;
```

END.