codecademy

Capstone Project: CoolTShirts

Learn SQL from Scratch

Jacob Elbaum

June 12th, 2019

Table of Contents

1. Get familiar with CoolTShirts
2. What is the user journey?
3. Optimize the campaign budget

1. Get familiar with CoolTShirts

1.1 Page Visits Table at CoolTShirts

```
SELECT * FROM page_visits LIMIT 4;
```

page_name	timestamp	user_id	utm_campaign	utm_source
1 - landing_page	2018-01-24 03:12:16	10006	getting-to-know-cool-tshirts	nytimes
2 - shopping_cart	2018-01-24 04:04:16	10006	getting-to-know-cool-tshirts	nytimes
3 - checkout	2018-01-25 23:10:16	10006	weekly-newsletter	email
1 - landing_page	2018-01-25 20:32:02	10030	ten-crazy-cool-tshirts-facts	buzzfee

1.2 Distinct Campaigns & Sources

- To determine the # of distinct campaigns, we must perform a distinct count of the utm_campaign column. By doing so, we’re able to see the number of distinct campaigns that CoolTShirts have ran during the time frame.
- To determine the # of distinct sources, we must perform a distinct count of the utm_source column. By doing so, we’re able to see the number of distinct sources that CoolTShirts have used during the time frame.

COUNT(DISTINCT(utm_campaign))	COUNT(DISTINCT(utm_source))
8	6

```
SELECT COUNT(DISTINCT(utm_campaign)) FROM page_visits;

SELECT COUNT(DISTINCT(utm_source)) FROM page_visits;
```

1.3 Which Source is Used for Each Campaign?

- To determine which source is used for each campaign, we must think to ourselves, how can we define via a table for each specific campaign, which sources were used
- With that in mind, we know that to define each specific campaign by itself, we must use the DISTINCT function, but to make sure that we can still view the source of that distinct campaign, we have to include the utm_source column as well

```
SELECT DISTINCT(utm_campaign), utm_source FROM
page_visits;
```

utm_campaign	Utm_source
getting-to-know-cool-tshirts	nytimes
weekly-newsletter	email
ten-crazy-cool-tshirts-facts	buzzfeed
retargetting-campaign	email
retargetting-ad	facebook
interview-with-cool-tshirts-founder	medium
paid-search	google
cool-tshirts-search	google

1.4 What Pages Exist on the CoolTShirts Website?

- To determine this we have to first check what the schema shows for finding the pages. After a quick check, we can see that it’s defined by page_name
- If we run a query that pulls the distinct page name from the page visits table, we’re able to see the exact pages that exist in their website.

```
SELECT DISTINCT(page_name) FROM page_visits;
```

Page_name
1 - landing_page
2 - shopping_cart
3 - checkout
4 - purchase

2. Get familiar with CoolTShirts

2.1 How many first touches is each campaign responsible for?

```
WITH first_touch AS (SELECT user_id,
MIN(timestamp) as first_touch_at
FROM page_visits
GROUP BY user_id)
SELECT COUNT(first_touch_at),
ft.first_touch_at,
pv.utm_campaign
FROM first_touch ft
JOIN page_visits pv
ON ft.user_id = pv.user_id
AND ft.first_touch_at = pv.timestamp
GROUP BY pv.utm_campaign
ORDER BY 1 DESC;
```

- In order to figure this out, I had to take the existing first touch query, and make some adjustments
- I added a COUNT(first_touch_at) so that we could count the exact number of first touches for each campaign that appears in the dataset
- I added the “group by” which helped to count the exact number of first touches per campaign specifically as opposed to counting the number of first touches for other columns
- I added the “order by” just to make the table more visually appealing

COUNT(first_touch_at)	first_touch_at	utm_campaign
622	2018-01-13 23:30:09	interview-with-cool-tshirts-founder
612	2018-01-25 00:04:39	getting-to-know-cool-tshirts
576	2018-01-04 05:59:46	ten-crazy-cool-tshirts-facts
169	2018-01-13 13:20:49	cool-tshirts-search

2.2 How many last touches is each campaign responsible for?

```
WITH last_touch AS(
SELECT user_id,
MAX(timestamp) as last_touch_at
FROM page_visits
GROUP BY user_id)
SELECT COUNT(last_touch_at),
lt.last_touch_at,
pv.utm_campaign
FROM last_touch lt
JOIN page_visits pv
ON lt.user_id = pv.user_id
AND lt.last_touch_at = pv.timestamp
GROUP BY pv.utm_campaign
ORDER BY 1 DESC;
```

- In order to figure this out, I had to take the existing first touch query, and make some adjustments to spin it to last touch
- I added a COUNT(last_touch_at) so that we could count the exact number of last touches for each campaign that appears in the dataset
- I added the “group by” which helped to count the exact number of last touches per campaign specifically as opposed to counting the number of last touches for other columns
- I added the “order by” just to make the table more visually appealing

COUNT(last_touch_at)	last_touch_at	utm_campaign
447	2018-01-26 06:18:39	weekly-newsletter
443	2018-01-24 05:26:09	retargetting-ad
245	2018-01-16 11:35:09	retargetting-campaign
232	2018-01-15 04:55:43	getting-to-know-cool-tshirts
190	2018-01-04 05:59:47	ten-crazy-cool-tshirts-facts
184	2018-01-02 07:40:34	interview-with-cool-tshirts-founder
178	2018-01-10 04:58:48	paid-search
60	2018-01-18 21:36:32	cool-tshirts-search

3. Optimize The Campaign Budget

3.1 How many visitors make a purchase?

- To determine this, I first re-checked the columns of the page_visits table by performing this query: SELECT * FROM page_visits LIMIT 10;
- Once I saw the columns, I saw that to come up with the solution, I had to just perform a distinct count (distinct because one visitor can make multiple purchases) and add a ‘where’ clause to define the name of the page (this case the purchase) page where the user would be counted.

```
SELECT COUNT(DISTINCT(user_id))
FROM page_visits
WHERE page_name = '4 - purchase’;
```

COUNT(DISTINCT(user_id))
361

3.2 How many last touches on the *purchase page* is each campaign responsible for?

- To determine this, I had to revisit the last touch temporary table that I created 2 questions ago.
- Once I saw the table, I realized that the table already showed me how many last touches each campaign was responsible for, but was not showing me how many last touches lead to users arriving on the campaign’s purchase page.
- To do this, I just had to add a basic ‘where’ clause with ‘4 – purchase page’

```
WITH last_touch AS(
SELECT user_id,MAX(timestamp) as last_touch_at
FROM page_visits
GROUP BY user_id)
SELECT COUNT(last_touch_at),lt.last_touch_at,pv.utm_campaign,
lt.user_id, pv.utm_source
FROM last_touch lt
JOIN page_visits pv
ON lt.user_id = pv.user_id
AND lt.last_touch_at = pv.timestamp
WHERE pv.page_name = '4 - purchase’
GROUP BY pv.utm_campaign
ORDER BY 1 DESC;
```

COUNT(last_touch_at)	last_touch_at	utm_campaign	user_id	utm_source
114	2018-01-26 06:18:39	weekly-newsletter	99933	email
112	2018-01-06 09:41:19	retargetting-ad	99897	facebook
53	2018-01-24 09:00:58	retargetting-campaign	99285	email
52	2018-01-19 16:37:58	paid-search	94567	google
9	2018-01-16 15:15:29	getting-to-know-cool-tshirts	92172	nytimes
9	2018-01-15 04:17:36	ten-crazy-cool-tshirts-facts	98651	buzzfeed
7	2018-01-10 18:20:21	interview-with-cool-tshirts-founder	83547	medium
2	2018-01-18 00:25:00	cool-tshirts-search	95650	google