codecademy

Capstone Project Attribution Queries

Learn SQL from Scratch

Pim Boterman

31-01-2019

Table of Contents

Getting familiar with CoolTShirts

CoolTShirts, an innovative apparel shop, is running a bunch of marketing campaigns. In this section I discuss their campaigns, sources and utm codes.

Getting familiar with CoolTShirts

CoolTShirts has 8 different campaigns and 6 different sources. The 8 campaigns are visited by users that come from 6 different sources. For example, users that see organic or paid search results (campaign), come from the source Google. The difference between utm_campaign and utm_source is that the source is the place you get send from towards the campaign.

CoolTShirts has 4 distinct pages on their website:

- Landing_page
- Shopping_cart
- Checkout
- Purchase

The next slide shows the output of the query on the right.

```
-- Question 1; bulletpoint 1; Answer: 8

SELECT COUNT (DISTINCT utm_campaign)
FROM page_visits;

-- Question 1; bulletpoint 2; Answer: 6

SELECT COUNT (DISTINCT utm_source)
FROM page_visits;

-- Question 1; bulletpoint 3;

SELECT DISTINCT utm_campaign, utm_source
FROM page_visits;
```

Getting familiar with CoolTShirts

In the table on the right, the distinct utm_campaigns and utm_sources are shown. There are 8 distinct campaigns and 6 sources, where email and google are featured in two campaigns.

The website of CoolTShirts has 4 distinct pages, shown in the table below. You get these results by simply selecting the distinct page_name from the page_visits table.

```
SELECT DISTINCT page_name
FROM page_visits;
```

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

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

The user journey

The user journey of CoolTShirts is measured by the first- and last touches for each campaign and the number of purchases each user makes.

First touch attributions - results

In the table below, all first touch attributions, per utm_source and utm_campaign are shown. As you can see the utm_source ‘medium’ has led 622 users to the utm_campaign ‘interview-with-cool-thirts-founder’. These results were produced with the query that is shown on the next slide.

Source	Campaign	First_touch_at
medium	interview-with-cool-tshirts-founder	622
nytimes	getting-to-know-cool-tshirts	612
buzzfeed	ten-crazy-cool-tshirts-CoolTShirts	576
google	cool-tshirts-search	169

First touch attributions - query

The query on the right side of this slide was used to produce results of the question: How many first touches is each campaign responsible for?

As you can see in the query, I first created temporary tables first_touch an ft_attr that both select data from the page_visits table. The first_touch table uses the MIN(timestamp) clause to get the first touch on a website. The ft_attr table selects the timestamp from the first_touch table and combines it with the utm_source and utm_campaign from the page_visits table in order to check which user first touched the website via which source and campaign. The results are then grouped by utm_source and utm_campaign in order to clarify the results. The results are also ordered by the number of first touches (COUNT) to rank the sources and campaigns that were first touched (visited) by the users of CoolTShirts.

On the next slide, the last touches each campaign is responsible for, are presented.

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

Last touch attributions

The table shows the number of last touch attributions each campaign and source is responsible for. These results show that email (as a source) and the weekly newsletter (as a campaign) have the most last touches and the source 'google' and the campaign 'cool-tshirts-search' have the least last touches.

The query is similar to the first touch attributions query. The difference is the MAX(timestamp) clause that is selected in order to reproduce the last touch attributions. Also, the names of the (temporary) tables are changed from 'first' to 'last' and from 'ft' to 'lt', which creates a table with an overview of all last touches on the website. These results are also ordered by the number of last touches (COUNT).

Source	Campaign	COUNT(*)
email	weekly-newsletter	447
facebook	retargetting-ad	443
email	retargetting-campaign	245
nytimes	getting-to-know-cool-tshirts	232
buzzfeed	ten-crazy-cool-tshirts-faCoolTShirts	190
medium	interview-with-cool-tshirts-founder	184
google	paid-search	178
google	cool-tshirts-search	60

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

How many visitors make a purchase?

As shown in the query in this slide, I selected and counted the distinct number of user_id's that have been on the '4 - purchase' page in order to get the answer to the question: How many visitors make a purchase?

The answer to that question is **361 users** as shown in the little table on the right.

On the next slide, the number of last touches per utm_source and utm_campaign are presented. If you add up these numbers, you get to 361 last touches on the purchase page of CoolTShirts. For me this was a control mechanism to check whether I was presenting the right data and wrote the right query.

```
-- Q5; Answer: 361

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

COUNT (DISTINCT user_id)
361

Last touches on the purchase page

Source	Campaign	COUNT(*)
email	weekly-newsletter	115
facebook	retargetting-ad	113
email	retargetting-campaign	54
google	paid-search	52
buzzfeed	ten-crazy-cool-tshirts-faCoolTShirts	9
nytimes	getting-to-know-cool-tshirts	9
medium	interview-with-cool-tshirts-founder	7
google	cool-tshirts-search	2

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

Last touches on the purchase page

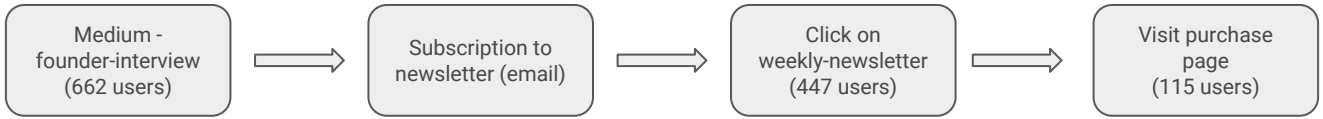
The query that was used to get the results from the previous slide is also similar to the first_touch and last_touch query that were used in this project. Two temporary tables were created in order to 1) simplify and clean up the query and 2) more easily retract data from different tables.

In this query, the MAX(timestamp) clause indicates that we're looking for the last touches a user has on a website. In this scenario, we wanted to know the last touches on the purchase page, because this allows us to check the best converting campaign and source. In this case, that is the weekly newsletter which comes from the email source, because this campaign led 115 users to the purchase page.

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

The user journey

Based on the output that the first_touch and last_touch queries produced, the most typical user journey can be traced back. The majority of users (662) come from the utm_source ‘medium’ and where the umt_campaign ‘interview-with-cool-tshirts-founder’ was published. After that, users subscribe to the weekly-newsletter of CoolTShirts. The last touch table shows that 447 users were last seen on the website by clicking on the link in the weekly-newsletter. Of these 447 users, 115 last visited the purchase page and thus made a purchase/converted.



Optimizing the campaign budget

An advice on how to optimize the campaign budget of CoolTShirts

Optimizing the campaign budget

I would advise CoolTShirts to re-invest in the 5 campaigns that are presented in the table on the right. The top 3 campaigns (weekly-newsletter, retagetting-ad and retargetting-campaign) are the campaigns with the most last touches and are therefore considered most succesful for CoolTShirts. The last two campaigns (getting-to-know-cool-tshirts and interview-with-cool-tshirts-founder) turned out to create the most first touches and therefore enabled CoolTShirts to retarget users via email and facebook.

Campaign	Source
weekly-newsletter	email
retargetting-ad	facebook
retargetting-campaign	email
getting-to-know-cool-tshirts	nytimes
interview-with-cool-tshirts-founder	medium