codecademy

Attribution Project

Learn SQL from Scratch – Capstone Option 3

Fabio Cerullo
June 5 Cohort

Table of Contents

1. Get familiar with CoolTShirts

1. – How many campaigns and sources are used?

To get familiar with the CoolTShirts project, first we need to determine the database that is being used on the web site.

Using the following table it is possible to identify each column and the purpose of the data, to later be used to resolve and answer the needs of the employer.

The flow of the users on the CoolTShirts site depends mainly on 2 factors, **utm_source** and **utm_campaign**.

The Source can be used to analyze the site that generated the traffic where the user is coming from, and in addition with the Campaign identifier, we can evaluate the specific commercial or promotion that was made to attract the user and trigger the interaction with the site.

page_visits	
A table describing each time a user visits the CoolTShirts website	
Column	**Description**
user_id	A unique identifier for each visitor to a page
timestamp	The time at which the visitor came to the page
page_name	The title of the section of the page that was visited
utm_source	Identifies which site sent the traffic (i.e., google, newsletter, or facebook_ad)
utm_medium	Identifies what type of link was used (i.e., cost-per-click or email)
utm_campaign	Identifies the specific ad or email blast (i.e., june-21-newsletter or memorial-day-sale)

1.A – How many campaigns and sources are used?

- Using the following query:

```
SELECT COUNT (DISTINCT utm_campaign) AS 'NUMBER OF CAMPAIGNS'
FROM page_visits;
```

to count the distinct number of campaigns of the database and display the result in one column, we can see that the total number is 8.

NUMBER OF CAMPAIGNS
8

- To count the distinct number of sources on the database we use the same query to select the **utm_source**,

```
SELECT COUNT (DISTINCT utm_source) AS 'NUMBER OF SOURCES'
FROM page_visits;
```

resulting a total of 6.

NUMBER OF SOURCES
6

- To determine the source of each individual campaign created, we select the distinct **utm_source** and **utm_campaign** to be displayed, noting that some sources can be displayed twice if the campaign was launched using the same source as tool.

```
SELECT DISTINCT utm_source AS 'SOURCES', utm_campaign AS 'CAMPAIGNS'
FROM page_visits;
```

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

1.B – What pages are on the Web Site?

- If we need to obtain the complete number of pages that make up the web site, using the following query we

  can see the result of every distinct value of **page_name** contained in **page_visits** database.

```
SELECT DISTINCT page_name AS 'PAGES'
FROM page_visits;
```

Getting the result of 4 individual pages, that consists of each step the user must take to complete the

purchase on the CoolTShirts site.

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

2. What is the user journey?

2.A – How many first touches each campaign is responsible?

To answer this question we need to break down the query in several parts to obtain the required solution.

- First, a clause is needed to store the result of the query to later be used. We can do this using a **WITH** command, stating **first_touch** as the selection of every user with their respective minimum timestamp grouped by **user_id** column.

```
WITH first_touch AS (
    SELECT user_id,
        MIN(timestamp) as first_touch_at
    FROM page_visits
    GROUP BY user_id),
```

- Selecting the **user_id** and **first_touch_at** (Minimum timestamp) from the clause that was created before including **utm_source** and **utm_campaign** from **page_visits** database, we can command an inner join of the rows with a match on **user_id** and **timestamp** to store the results as **ft_attr.**

```
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
)
```

2.A – How many first touches each campaign is responsible?

- After the clauses has been created, the selection of the source and campaign of **ft_attr**, and the count of the total values grouped by both, source and campaign figures, along with a descending order by the total number of first touches, prompts the result of the number of users who visited the web site for the first time on each campaign.

```sql
SELECT ft_attr.utm_source AS 'SOURCE',
       ft_attr.utm_campaign AS 'CAMPAIGN',
       COUNT(*) AS 'FIRTS TOUCH NUMBER OF USERS'
FROM ft_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

Including the source of that campaign.

```sql
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(*) AS 'LAST TOUCH NUMBER OF USERS'
FROM lt_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

SOURCE	CAMPAIGN	FIRTS TOUCH NUMBER OF USERS
medium	interview-with-cool-tshirts-founder	622
nytimes	getting-to-know-cool-tshirts	612
buzzfeed	ten-crazy-cool-tshirts-facts	576
google	cool-tshirts-search	169

2.B – How many last touches each campaign is responsible?

This question can be answered mimicking the behavior of the first touch query, but changing the values to obtain instead the last touch of the user on the site.

- We create a clause to store the result of the query, using a **WITH** command stating **last_touch** as the selection of every user with their respective maximum timestamp grouped by **user_id** column.

```
WITH last_touch AS (
    SELECT user_id,
        MAX(timestamp) as last_touch_at
    FROM page_visits
    GROUP BY user_id),
```

- Selecting the **user_id** and **last_touch_at** (Maximum timestamp) from the clause that was created before, adding **utm_source** and **utm_campaign** from **page_visits** database, an inner join of the rows with a match on **user_id** and **timestamp** can be created to store the results as **lt_attr.**

```
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
)
```

2.B – How many last touches each campaign is responsible?

- To obtain the number of each last touch of the users on every campaign.

  The source, campaign and the count of the total digits of **lt_attr**, grouped by source and campaign in descending order of that total values must be transcribed in the following query.

```sql
SELECT lt_attr.utm_source AS 'SOURCE',
       lt_attr.utm_campaign AS 'CAMPAIGN',
       COUNT(*) AS 'LAST TOUCH NUMBER OF USERS'
FROM lt_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

```sql
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(*) AS 'LAST TOUCH NUMBER OF USERS'
FROM lt_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

SOURCE	CAMPAIGN	LAST TOUCH NUMBER OF USERS
email	weekly-newsletter	447
facebook	retargetting-ad	443
email	retargetting-campaign	245
nytimes	getting-to-know-cool-tshirts	232
buzzfeed	ten-crazy-cool-tshirts-facts	190
medium	interview-with-cool-tshirts-founder	184
google	paid-search	178
google	cool-tshirts-search	60

2.C – How many visitors make a purchase?

To find the number of total users that entered the CoolTShirts site and made a purchase, we need to count the number of each distinct **user_id** on **page_visits** database that reached to the last **page_name** where the purchase takes place.

It can be done adding a **WHERE** conditional when the **page_name** value it's the same as '**4 – purchase**'.

Resulting 361 unique users who completed an order.

```sql
SELECT COUNT (DISTINCT user_id) AS 'USERS WHO PURCHASED'
FROM page_visits
WHERE page_name = '4 - purchase';
```

USERS WHO PURCHASED
361

2.D – How many last touches on purchase page each campaign is responsable for?

- To answer this requirement and find on every campaign the number of last touches by users who visited the web and made a purchase, we can modify our **last_touch** clause query with the following.

  Adding a **WHERE** conditional that filters all the rows that meet the condition of values where **page_name** is equal to '**4 – purchase**'.

```sql
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),
```

The result of the query gives us an insight of the total number of each source and campaign where an user completed an order successfully.

SOURCE	CAMPAIGN	USERS WHO PURCHASED
email	weekly-newsletter	115
facebook	retargetting-ad	113
email	retargetting-campaign	54
google	paid-search	52
buzzfeed	ten-crazy-cool-tshirts-facts	9
nytimes	getting-to-know-cool-tshirts	9
medium	interview-with-cool-tshirts-founder	7
google	cool-tshirts-search	2

```sql
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(*) AS 'USERS WHO PURCHASED'
FROM lt_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

2.E – What is the typical user journey?

To determine the typical user journey on the website, first we need to pick a number of users to study the behavior with the following process.

We select all the values from **page_visits** database and limit the results to 17 items to obtain a complete range for the **page_name** column.

```
SELECT *
FROM page_visits
LIMIT 17;
```

With this query we can see the actions of 5 users: 10006, 10030, 10045, 10048, 10069.

We can observe that not everyone completed an order, and when each user enters the site from the respective source, they check the items offered and adds them to the cart within 40 min to 3 hours on the same day of visit, an average of 2 hours and 20 min.

And once the user has selected the items they do not purchase on the same day, a retargetting campaign is needed to make the user come back again an complete the order. Of 5 users only 2 completed it.

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	buzzfeed
2 - shopping_cart	2018-01-25 23:05:02	10030	ten-crazy-cool-tshirts-facts	buzzfeed
3 - checkout	2018-01-28 13:26:02	10030	retargetting-campaign	email
4 - purchase	2018-01-28 13:38:02	10030	retargetting-campaign	email
1 - landing_page	2018-01-05 18:31:17	10045	getting-to-know-cool-tshirts	nytimes
2 - shopping_cart	2018-01-05 21:16:17	10045	getting-to-know-cool-tshirts	nytimes
3 - checkout	2018-01-09 03:05:17	10045	retargetting-ad	facebook
1 - landing_page	2018-01-16 04:17:46	10048	interview-with-cool-tshirts-founder	medium
2 - shopping_cart	2018-01-16 07:28:46	10048	interview-with-cool-tshirts-founder	medium
3 - checkout	2018-01-19 00:00:46	10048	retargetting-campaign	email
1 - landing_page	2018-01-02 23:14:01	10069	ten-crazy-cool-tshirts-facts	buzzfeed
2 - shopping_cart	2018-01-02 23:55:01	10069	ten-crazy-cool-tshirts-facts	buzzfeed
3 - checkout	2018-01-04 08:12:01	10069	retargetting-ad	facebook
4 - purchase	2018-01-04 08:13:01	10069	retargetting-ad	facebook

Then we filter by **page_name**

```
SELECT user_id AS 'USER', page_name AS 'LAST PAGE VISITED'
FROM page_visits
GROUP BY 1
LIMIT 5;
```

USER	LAST PAGE VISITED
10006	3 - checkout
10030	4 - purchase
10045	3 - checkout
10048	3 - checkout
10069	4 - purchase

3. Optimize the campaign budget

3. – Which campaigns should they re-invest in?

For this final question, we need to create a budgeting plan for an effective use of the campaigns, required to help achieve a greater customer acquisition and a sustainable growth of CoolTShirts website.

Using the data gathered with the queries created before, we can understand the efficacy of each campaign along side with the source and the average flow of the user when entering the website.

CoolTShirts executives stated that only 5 campaigns could be renewed, with that goal in mind we can proceed to provide a solution.

3. – Which campaigns should they re-invest in?

Looking to the First Touch results that we obtained before, we can see that of the 8 campaigns available, 4 were used to engage the first interaction of the users to meet the site.

SOURCE	CAMPAIGN	FIRTS TOUCH NUMBER OF USERS
medium	interview-with-cool-tshirts-founder	622
nytimes	getting-to-know-cool-tshirts	612
buzzfeed	ten-crazy-cool-tshirts-facts	576
google	cool-tshirts-search	169

Of the total of users on the database (1.979 unique id's), 1.810 users got to know the site from campaigns that were conceived with content creation, as interviews and web articles. Only 169 entered the site directly from a Google search itself.

This 4 campaigns, if we include Google searches as an expense, are the base factors for the influx of new users, therefore should be prioritized.

3. – Which campaigns should they re-invest in?

If we consider the values collected when the last touch queries were used, we can determine which of the remaining campaigns are useful for the site.

Here we can see that even though some of the campaigns are responsible for a large number of last touches, only a few helped to achieve a monetary gain.

SOURCE	CAMPAIGN	USERS WHO PURCHASED
email	weekly-newsletter	115
facebook	retargetting-ad	113
email	retargetting-campaign	54
google	paid-search	52
buzzfeed	ten-crazy-cool-tshirts-facts	9
nytimes	getting-to-know-cool-tshirts	9
medium	interview-with-cool-tshirts-founder	7
google	cool-tshirts-search	2

SOURCE	CAMPAIGN	LAST TOUCH NUMBER OF USERS
email	weekly-newsletter	447
facebook	retargetting-ad	443
email	retargetting-campaign	245
nytimes	getting-to-know-cool-tshirts	232
buzzfeed	ten-crazy-cool-tshirts-facts	190
medium	interview-with-cool-tshirts-founder	184
google	paid-search	178
google	cool-tshirts-search	60

As we found on the user journey section 2-E of this document, the average user needed an extra encouragement to complete the payment of the items added to the cart, since it took a couple of days after visiting the site for the user to remember the product and checking out.

The email campaigns as **weekly-newsletter**, **retargeting-campaign**, and Facebook **retargeting-campaign** are the most effective to captivate the final payment of the user.

3. – Which campaigns should they re-invest in?

As a marketing solution, the best campaigns to get the brand of CoolTShirst popular are:

- Interview with CoolTShirts founder (Medium)
- Getting to know CoolTShirts (NYtimes)
- Ten crazy CoolTShirts facts (Buzzfeed)
- CoolTShirts searches on Google (If we consider it an expense instead of Word of Mouth advertising)

And as retargetting solution:

- Weekly Newsletter (Email)
- Retargetting Campaing (Facebook)

Depending of the cost of each campaign and the consideration of the CoolTShirts team, since only 5 can be used, this would be the best options to increase the revenue and maintain a constant clientele.