



Attribution Queries

# CoolTShirts

Analyze Data with SQL

Attribution Queries

Alex Ricciardi

Date: 04/10/2020

# CoolTShirts

CoolTShirts, an innovative apparel shop, is running a bunch of marketing campaigns. In this project, you'll be helping them answer these questions about their campaigns:

1 Get familiar with the company.

How many campaigns and sources does CoolTShirts use and how are they related? Be sure to explain the difference between `utm_campaign` and `utm_source`.

What pages are on their website?

2. What is the user journey?

- How many first touches is each campaign responsible for?
- How many last touches is each campaign responsible for?
- How many visitors make a purchase?
- How many last touches on the purchase page is each campaign responsible for?

What is the typical user journey?

3. Optimize the campaign budget.

CoolTShirts can re-invest in 5 campaigns. Which should they pick and why?

CoolTShirts Project Database Schema:

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_campaign	Identifies the specific ad or email blast (i.e., june-21-newsletter or memorial-day-sale)

Database Schema		
page_visits		5692 rows
page_name		TEXT
timestamp		TEXT
user_id		INTEGER
utm_campaign		TEXT
utm_source		TEXT

## Project Tasks:

### Get familiar with CoolTShirts

1. How many campaigns and sources does CoolTShirts use? Which source is used for each campaign?
2. What pages are on the CoolTShirts website?

### What is the user journey?

3. How many first touches is each campaign responsible for?
4. How many last touches is each campaign responsible for?
5. How many visitors make a purchase?
6. How many last touches on the purchase page is each campaign responsible for?
7. CoolTShirts can re-invest in 5 campaigns. Given your findings in the project, which should they pick and why?number of segments?

# 1. Project Task:

## How many campaigns and sources

# 1.1 Description of the task

How many campaigns and sources does CoolTShirts use?

Which source is used for each campaign?

Use three queries:

- one for the number of distinct campaigns,
- one for the number of distinct sources,
- one to find how they are related.

# 1.2 How many campaigns and sources does CoolTShirts use?

CoolTShirts uses 8 campaigns:

- getting-to-know-cool-tshirts
- weekly-newsletter
- ten-crazy-cool-tshirts-facts
- retargetting-campaign
- retargetting-ad
- interview-with-cool-tshirts-founder
- paid-search
- cool-tshirts-search

CoolTShirts uses 6 sources:

- nytime
- email
- buzzfeed
- facebook
- medium
- google

Queries	Query code
utm_campaign	<pre>SELECT DISTINCT utm_campaign FROM page_visits;</pre>
getting-to-know-cool-tshirts	
weekly-newsletter	<pre>SELECT DISTINCT utm_source FROM page_visits;</pre>
ten-crazy-cool-tshirts-facts	
retargetting-campaign	
retargetting-ad	
interview-with-cool-tshirts-founder	
paid-search	
cool-tshirts-search	
utm_source	
nytimes	
email	
buzzfeed	
facebook	
medium	
google	

## 1.3 Which source is used for each campaign?

CoolTShirts sources and campaigns connection:

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

Query code

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

Queries

```
SELECT
  DISTINCT utm_source,
  utm_campaign
FROM page_visits
ORDER BY 1;
```



## 2. Project Task:

### The page\_name column

## 2.1 Description of the task

What pages are on the CoolTShirts website?

Find the distinct values of the page\_name column.

## 2.2 What pages are on the CoolTShirts website?

Values of the page\_name column :

- 1 - landing\_page
- 2 - shopping\_cart
- 3 - checkout
- 4 - purchase

The page\_name column is a marketing Funnel from landing page to purchase.

Query code

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

Queries

page\_name

1 - landing\_page

2 - shopping\_cart

3 - checkout

4 - purchase

### 3. Project Task:

How many first touches is each campaign responsible for?

## 3.1 Description of the task

How many first touches is each campaign responsible for?

Group by campaign and count the number of first touches for each.

## 3.2 How many first touches is each campaign responsible for?

Number of first touches for each campaign:

- interview-with-cool-tshirts-founder 622
- getting-to-know-cool-tshirts 612
- ten-crazy-cool-tshirts-facts 576
- cool-tshirts-search 169

I created a temporary `first_touch` table storing the `user_id` and `first_touch_at`, and a second temporary `ft_attr` join table of the temporary `first_touch` table and the `page_visits` table, to query the number of first touches for each campaign.

Queries

utm_campaign	utm_source	number_of_first_touches
interview-with-cool-tshirts-founder	medium	622
getting-to-know-cool-tshirts	nytimes	612
ten-crazy-cool-tshirts-facts	buzzfeed	576
cool-tshirts-search	google	169

Query code

```
----- first_touch table
WITH first_touch AS (
  SELECT
    user_id,
    MIN(timestamp) AS first_touch_at
  FROM page_visits
  GROUP BY user_id
),
----- ft_attr table
ft_attr AS (
  SELECT
    ft.user_id,
    ft.first_touch_at,
    pv.utm_source,
    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
)
----- query
SELECT
  utm_campaign,
  utm_source,
  COUNT(*) AS number_of_first_touches
FROM ft_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

## 4. Project Task:

Create a temporary table `cross_join`

## 4.1 Description of the task

How many last touches is each campaign responsible for?

Group by campaign and count the number of last touches for each.



## 4.2 Group by campaign and count the number of last touches for each.

Number of last touches for each campaign:

• weekly-newsletter	447	• ten-crazy-cool-tshirts-facts	190
• retargeting-ad	443	• interview-with-cool-tshirts-founder	184
• retargeting-campaign	245	• paid-search	178
• getting-to-know-cool-tshirts	232	• cool-tshirts-search	60

I created a temporary `last_touch` table storing the `user_id` and `last_touch_at`, and a second temporary `lt_attr` join table of the temporary `last_touch` table and the `page_visits` table, to query the number of last touches for each campaign.

Queries

utm_campaign	utm_source	number_of_last_touches
weekly-newsletter	email	447
retargeting-ad	facebook	443
retargeting-campaign	email	245
getting-to-know-cool-tshirts	nytimes	232
ten-crazy-cool-tshirts-facts	buzzfeed	190
interview-with-cool-tshirts-founder	medium	184
paid-search	google	178
cool-tshirts-search	google	60

Query code

```
----- last_touch table
WITH last_touch AS (
  SELECT
    user_id,
    MAX(timestamp) AS last_touch_at
  FROM page_visits
  GROUP BY user_id
),
----- lt_attr table
lt_attr AS (
  SELECT
    lt.user_id,
    lt.last_touch_at,
    pv.utm_source,
    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
)
----- query
SELECT
  utm_campaign,
  utm_source,
  COUNT(*) AS number_of_last_touches
FROM lt_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

## 5. Project Task:

Create a temporary table status

## 5.1 Description of the task

How many visitors make a purchase?

Count the distinct users who visited the page named 4 - purchase.

## 5.2 How many visitors make a purchase?

361 visitors made a purchase.

Query output

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

Query code

number_of_purchases
---------------------

361
-----

## 6. Project Task:

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

## 6.1 Description of the task

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

This query will look similar to your last-touch query, but with an additional WHERE clause.

## 6.2 How many last touches on the purchase page is each campaign responsible for?

Query code

```
----- last_touch table
WITH last_touch AS (
  SELECT
    user_id,
    MAX(timestamp) AS last_touch_at
  FROM page_visits
  GROUP BY user_id
),
----- lt_attr_purchases table
lt_attr_purchases AS (
  SELECT
    lt.user_id,
    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
  WHERE pv.page_name = '4 - purchase'
)
----- query
SELECT
  utm_campaign,
  COUNT(*) AS number_of_last_touch_purchases
FROM lt_attr_purchases
GROUP BY 1
ORDER BY 2 DESC;
```

Number of last touch purchases for each campaign:

- weekly-newsletter 114
- retargeting-ad 112
- retargeting-campaign 53
- paid-search 52
- getting-to-know-cool-tshirts 9
- ten-crazy-cool-tshirts-facts 9
- interview-with-cool-tshirts-founder 7
- cool-tshirts-search 2

Query output

utm_campaign	number_of_last_touch_purchases
weekly-newsletter	114
retargeting-ad	112
retargeting-campaign	53
paid-search	52
getting-to-know-cool-tshirts	9
ten-crazy-cool-tshirts-facts	9
interview-with-cool-tshirts-founder	7
cool-tshirts-search	2

**7. Project Task:**  
**CoolTShirts can re-invest in 5**  
**campaigns.**



## 7.1 Description of the task

CoolTShirts can re-invest in 5 campaigns.

Given your findings in the project, which should they pick and why?

# 7.2 First five campaigns with the highest number of purchases

When taking a quick look at the query from slide 6.3, we can see, that the 5 first campaigns with the highest numbers of last touches purchases are:

- 1. weekly-newsletter
- 2. retargeting-ad
- 3. retargeting-campaign
- 4. paid-search

With a tie for the 5<sup>th</sup> position between

- 5. getting-to-know-cool-tshirts
- 5. ten-crazy-cool-tshirts-facts

It seems that CoolTShirts should re-invest in the first 4 campaigns, but with a better understanding of UTM attribution, we will find out that the first 4 campaigns are retargeting campaigns, they are not first touch campaigns. Let's analyze the data further before making a recommendation.

Query output from slide 6.3

utm_campaign	number_of_last_touch_purchases
weekly-newsletter	114
retargeting-ad	112
retargeting-campaign	53
paid-search	52
getting-to-know-cool-tshirts	9
ten-crazy-cool-tshirts-facts	9
interview-with-cool-tshirts-founder	7
cool-tshirts-search	2

## 7.3 The costumer journey from first touch to purchase

The campaigns:

- weekly-newsletter
- retargeting-ad
- retargeting-campaign
- paid-search

They are retargeting campaigns, meaning they target a potential customer after a first touch happened.

The campaigns: see query output from slide 32 and slide 6.3

- interview-with-cool-tshirts-founder
- getting-to-know-cool-tshirts
- ten-crazy-cool-tshirts-facts
- cool-tshirts-search

They are first touch campaigns having a last touch purchase outcome, with no retargeting campaign involvement. Going forward, I will define those campaigns results as *first-last touch purchases*.

Query output slide 6.3

utm_campaign	number_of_last_touch_purchases
weekly-newsletter	114
retargeting-ad	112
retargeting-campaign	53
paid-search	52
getting-to-know-cool-tshirts	9
ten-crazy-cool-tshirts-facts	9
interview-with-cool-tshirts-founder	7
cool-tshirts-search	2

Query output slide 3.2

utm_campaign	utm_source	number_of_first_touches
interview-with-cool-tshirts-founder	medium	622
getting-to-know-cool-tshirts	nytimes	612
ten-crazy-cool-tshirts-facts	buzzfeed	576
cool-tshirts-search	google	169

## 7.4 The costumer journey from first touch to purchase, Table query results

Query output

user_id	first_touch	purchase_date	ft_source	ltp_source	ft_campaign	ltp_campaign
45453	2018-01-01 01:42:56	2018-01-02 15:09:56	medium	email	interview-with-cool-tshirts-founder	retargeting-campaign
72411	2018-01-01 02:51:09	2018-01-04 20:29:09	nytimes	email	getting-to-know-cool-tshirts	weekly-newsletter
11214	2018-01-01 10:29:53	2018-01-02 18:10:53	nytimes	facebook	getting-to-know-cool-tshirts	retargeting-ad
72723	2018-01-01 11:00:44	2018-01-06 08:24:44	medium	email	interview-with-cool-tshirts-founder	weekly-newsletter
52442	2018-01-01 14:01:19	2018-01-04 01:46:19	buzzfeed	email	ten-crazy-cool-tshirts-facts	weekly-newsletter
36031	2018-01-01 14:52:52	2018-01-05 23:48:52	buzzfeed	email	ten-crazy-cool-tshirts-facts	weekly-newsletter
29982	2018-01-01 16:40:07	2018-01-04 11:43:07	buzzfeed	email	ten-crazy-cool-tshirts-facts	weekly-newsletter
52350	2018-01-01 18:21:15	2018-01-01 23:35:15	nytimes	nytimes	getting-to-know-cool-tshirts	getting-to-know-cool-tshirts
97680	2018-01-01 18:34:28	2018-01-06 15:03:28	medium	facebook	interview-with-cool-tshirts-founder	retargeting-ad
73133	2018-01-01 20:31:15	2018-01-06 08:15:15	nytimes	email	getting-to-know-cool-tshirts	retargeting-campaign

I made a query outputting the first and last touch attributions for every costumer that made a purchase.

When taking a quick look at the query rows:

- We can see that most of the first touch campaigns, ft\_campaign, have a different result than the last touch purchases campaigns, ltp\_campaign. Meaning that those costumers were retargeted, with a different campaign before making the purchase.
- We can also see that a few first touch campaigns, ft\_campaign, have the same result than the last touch purchases campaigns, ltp\_campaign. Meaning that those costumers were not retargeted, and they made a purchase directly from a first touch campaign. I define the purchases results of first touch campaigns having a last touch purchase outcome, with no retargeting campaign involvement, as *first-last touch purchases results*.

## 7.4.1 The costumer journey from first touch to purchase, Table results query code

```
----- first_touch table
WITH first_touch AS (
  SELECT
    user_id,
    MIN(timestamp) AS first_touch_at
  FROM page_visits
  GROUP BY user_id
),
----- ft.attr table
ft_attr AS (
  SELECT
    ft.user_id,
    ft.first_touch_at,
    pv.utm_source,
    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
),
----- last_touch table
last_touch AS (
  SELECT
    user_id,
    MAX(timestamp) AS last_touch_at
  FROM page_visits
  GROUP BY user_id
),
-----
lt_attr_purchases table
lt_attr_purchases AS (
  SELECT
    lt.user_id,
    lt.last_touch_at,
    pv.utm_source,
    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
  WHERE pv.page_name = '4 - purchase'
),
----- ft_attr_purchases table
ft_attr_purchases AS (
  SELECT
    lap.user_id,
    fa.first_touch_at AS first_touch,
    lap.last_touch_at AS purchase_date,
    fa.utm_source AS ft_source,
    lap.utm_source AS ltp_source,
    fa.utm_campaign AS ft_campaign,
    lap.utm_campaign AS ltp_campaign
  FROM lt_attr_purchases lap
  JOIN ft_attr fa
    ON lap.user_id = fa.user_id
)
----- query
SELECT *
FROM ft_attr_purchases
ORDER BY 2
LIMIT 10;
```

## 7.5 Purchase numbers for first touch campaigns

First-last touch purchases results, are the results from first touch campaigns having a last touch purchase outcome, with no retargeting campaign involvement.

When taking a quick look at the query:

We can see that the results of the Numbers of Last Touch resulting in a Purchase are lower compared to the Numbers of First Touch Resulting in a Purchase (First-last touch purchases results).

Even if it is a helpful comparison, a more interesting comparison will be to compare the Numbers of First Touch Resulting in a Purchase from the first touch campaigns and the number\_of\_last\_touch\_purchases from the retargeting campaigns.

Query output

```
.....
----- query
SELECT
  ft_campaign AS 'First-Last Touch Purchase Campaign',
  ft_source AS 'First-Last Touch Purchase Campaign's Source',
  COUNT (*) AS 'Number of First Touch Resulting in a Purchase',
  COUNT(DISTINCT CASE
    WHEN ft_campaign = ltp_campaign
    THEN user_id
  END) AS 'Number of Last Touch resulting in a Purchase'
FROM ft_attr_purchases
GROUP BY ft_campaign
ORDER BY 3 DESC;
```

Query

First-Last Touch Purchase Campaign	First-Last Touch Purchase Campaign's Source	Number of First Touch Resulting in a Purchase	Number of Last Touch resulting in a Purchase
interview-with-cool-tshirts-founder	medium	118	7
ten-crazy-cool-tshirts-facts	buzzfeed	107	8
getting-to-know-cool-tshirts	nytimes	102	8
cool-tshirts-search	google	31	2

## 7.6 Purchase numbers for last touch campaigns.

Last touch purchases results, can be the results from first touch campaigns having a last touch purchase outcome, with no retargeting campaign involvement, but more often than not, last touch purchases results are from retargeting campaigns.

I created a temporary table, `retargeting_campaign_table`, to isolate the targeting campaigns from the first touch campaigns and created a query to output the Number of purchases for each campaign.

We can see from the output query, that the `weekly-newsletter` campaign and `retargeting-campaign` campaign have email as a utm source. Next step is to compare the Number of purchases from Retargeting Campaigns and from First Touch Campaigns.

Retargeting Campaigns:

weekly-newsletter
retargeting-ad
retargeting-campaign
paid-search

Query

```
.....
----- retargeting_campaign_table
retargeting_campaign_table AS (
  SELECT
    user_id,
    CASE
      WHEN ltp_campaign = 'weekly-newsletter' THEN ltp_campaign
      WHEN ltp_campaign = 'retargeting-ad' THEN ltp_campaign
      WHEN ltp_campaign = 'retargeting-campaign' THEN ltp_campaign
      WHEN ltp_campaign = 'paid-search' THEN ltp_campaign
    END AS retargeting_campaign,
    CASE
      WHEN ltp_campaign = 'weekly-newsletter' THEN ltp_source
      WHEN ltp_campaign = 'retargeting-ad' THEN ltp_source
      WHEN ltp_campaign = 'retargeting-campaign' THEN ltp_source
      WHEN ltp_campaign = 'paid-search' THEN ltp_source
    END AS retargeting_campaign_source
  FROM ft_attr_purchases
  WHERE retargeting_campaign IS NOT NULL
)
----- Query
SELECT
  retargeting_campaign AS 'Retargeting Campaign',
  retargeting_campaign_source AS 'Retargeting Campaign's Source',
  COUNT(*) AS 'Number of purchases'
FROM retargeting_campaign_table
GROUP BY retargeting_campaign
ORDER BY 3 DESC;
```

Query output

Retargeting Campaign	Retargeting Campaign's Source	Number of purchases
weekly-newsletter	email	114
retargeting-ad	facebook	112
retargeting-campaign	email	53
paid-search	google	52

# 7.7 Number of purchases outcomes for each campaign

The following query outputs the number of purchases outcomes for each campaign, it also assign to the campaign a type, first touch or retargeting.

the 5 first campaigns with the highest number of purchase outcomes are:

- interview-with-cool-tshirts-founder first touch campaign
- weekly-newsletter retargeting campaign
- retargeting-ad retargeting campaign
- ten-crazy-cool-tshirts-facts first touch campaign
- getting-to-know-cool-tshirts first touch campaign

I will recommend CoolTShirts to re-invest in all five of them, together they strike a good balance between first touch and retargeting campaigns

Query

campaign	source	campaign_type	Number of Purchases
interview-with-cool-tshirts-founder	medium	first touch	118
weekly-newsletter	email	retargeting	114
retargeting-ad	facebook	retargeting	112
ten-crazy-cool-tshirts-facts	buzzfeed	first touch	107
getting-to-know-cool-tshirts	nytimes	first touch	102
retargeting-campaign	email	retargeting	53
paid-search	google	retargeting	52
cool-tshirts-search	google	first touch	31



## 7.7.1 Number of purchases outcomes for each campaign, Table results query code

```
.....
----- retargeting_campaign_table
retargeting_campaign_table AS (
  SELECT
    user_id,
  CASE
    WHEN ltp_campaign = 'weekly-newsletter' THEN ltp_campaign
    WHEN ltp_campaign = 'retargeting-ad' THEN ltp_campaign
    WHEN ltp_campaign = 'retargeting-campaign' THEN ltp_campaign
    WHEN ltp_campaign = 'paid-search' THEN ltp_campaign
  END AS retargeting_campaign,
  CASE
    WHEN ltp_campaign = 'weekly-newsletter' THEN ltp_source
    WHEN ltp_campaign = 'retargeting-ad' THEN ltp_source
    WHEN ltp_campaign = 'retargeting-campaign' THEN ltp_source
    WHEN ltp_campaign = 'paid-search' THEN ltp_source
  END AS retargeting_campaign_source
  FROM ft_attr_purchases
  WHERE retargeting_campaign IS NOT NULL
),
----- ft_campaign_purchases table
ft_campaign_purchases AS (
  SELECT
    user_id,
    ft_campaign,
    ft_source
  FROM ft_attr_purchases
),
.....
```

```
.....
----- retargeting_campaign_purchases table
retargeting_campaign_purchases AS (
  SELECT
    user_id,
    retargeting_campaign,
    retargeting_campaign_source
  FROM retargeting_campaign_table
),
----- campaign_purchases table
campaign_purchases AS (
  SELECT
    user_id,
    ft_campaign AS campaign,
    ft_source AS source
  FROM ft_attr_purchases
  UNION
  SELECT
    user_id,
    retargeting_campaign AS campaign,
    retargeting_campaign_source AS source
  FROM retargeting_campaign_purchases
),
.....
```

```
.....
----- campaign_purchase_type table
campaign_purchase_type as (
  SELECT
    user_id,
    campaign,
    source,
  CASE
    WHEN campaign = 'weekly-newsletter' THEN 'retargeting'
    WHEN campaign = 'retargeting-ad' THEN 'retargeting'
    WHEN campaign = 'retargeting-campaign' THEN 'retargeting'
    WHEN campaign = 'paid-search' THEN 'retargeting'
    ELSE 'first touch'
  END AS campaign_type
  FROM campaign_purchases
)
SELECT
  campaign,
  source,
  campaign_type,
  COUNT(*) AS 'Number of Purchases'
FROM campaign_purchase_type
GROUP BY 1
ORDER BY 4 DESC;
```