



# Capstone: Attribution

Learn SQL from Scratch

Vanisha Hirji

July 2018

# Table of Contents

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

# **1. Get Familiar with the CoolTShirts**

# 1. How many campaigns & sources does CoolTShirts use?

This was some simple code, to retrieve distinct campaigns and sources.

```
-- You can put your query here
select count(DISTINCT utm_campaign) as 'utm campaigns'
from page_visits;

select count(DISTINCT utm_source) as 'utm sources'
from page_visits;

select distinct utm_campaign, utm_source
from page_visits;
```

Query Results	
utm campaigns	
8	
utm sources	
6	
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

## 2. What pages are on the CoolTShirts website?

This was again some simple code, to retrieve distinct page name on the website so it was similar to what I wrote for the previous query.

-- You can put your query here

```
select distinct page_name  
from page_visits;
```

### Query Results

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

# **1. What is the User Journey?**

### 3. How many first touches is each campaign responsible for?

I struggled with this, I heavily used the hint, I just could not get my head around how to do this.

Query Results		
ft_attr.utm_source	ft_attr.utm_campaign	COUNT(*)
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

```
-- You can put your query here
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 ft
    JOIN page_visits pv
    ON ft.user_id = pv.user_id
    AND ft.first_touch_at = pv.timestamp
)
SELECT ft_attr.utm_source,
       ft_attr.utm_campaign,
       COUNT(*)
FROM ft_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

## 4. How many last touches is each campaign responsible for?

### 1<sup>st</sup> Attempt

I initially tried writing this from scratch though I got an answer I suspected it maybe incorrect, the next slide shows my second attempt at this question.

```
-- You can put your query here
WITH last_touch AS (
  SELECT user_id,
         MIN(timestamp) as last_touch_at
  FROM page_visits
  GROUP BY user_id)
SELECT count(lt.user_id), 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;
```

#### Query Results

count(lt.user_id)	utm_campaign
169	cool-tshirts-search
612	getting-to-know-cool-tshirts
622	interview-with-cool-tshirts-founder
576	ten-crazy-cool-tshirts-facts



## 4. How many last touches is each campaign responsible for?

### 2<sup>nd</sup> Attempt

I used the first touch code provided from the previous query and amended it to last touch, then added a count and stripped away unnecessary columns.

I got the same answers when I wrote out the query this way so it confirmed what I did initially on my first attempt.

Query Results		
lt_attr.utm_source	lt_attr.utm_campaign	COUNT(*)
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

```
-- You can put your query here
WITH last_touch AS (
    SELECT user_id,
           MIN(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 lt
    JOIN page_visits pv
    ON lt.user_id = pv.user_id
    AND lt.last_touch_at = pv.timestamp
)
SELECT lt_attr.utm_source,
       lt_attr.utm_campaign,
       COUNT(*)
FROM lt_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

# **1. Optimise the Campaign Budget**

## 5. How many visitors make a purchase?

This was a nice simple question after the previous one, counted the number of id that had landed on the purchase page. I also double checked for distinct user ids, however the result was the same.

```
-- You can put your query here
Select count(user_id), page_name
from page_visits
where page_name = '4 - purchase';
```

### Query Results

count(user_id)	page_name
361	4 - purchase

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

This was built up using the learning from the first/last touch questions.

I hope it's correct!

Query Results		
lt_attr.utm_campaign	lt_attr.page_name	COUNT(*)
weekly-newsletter	4 - purchase	114
retargeting-ad	4 - purchase	112
retargeting-campaign	4 - purchase	53
paid-search	4 - purchase	52
getting-to-know-cool-tshirts	4 - purchase	9
ten-crazy-cool-tshirts-facts	4 - purchase	9
interview-with-cool-tshirts-founder	4 - purchase	7
cool-tshirts-search	4 - purchase	2

```
-- You can put your query here
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, pv.page_name, pv.utm_campaign,
lt.last_touch_at
from last_touch lt
join page_visits pv
ON lt.user_id = pv.user_id AND lt.last_touch_at =
pv.timestamp)
SELECT lt_attr.utm_campaign, lt_attr.page_name,
COUNT(*)
FROM lt_attr
where lt_attr.page_name = '4 - purchase'
GROUP BY 1, 2
ORDER BY 3 DESC;
```