Table of Contents

1. About CoolTShirts

1.1 The Data - exemplary selection of all page_visits

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
4 - purchase	2018-01-28 13:38:02	10030	retargetting-campaign	email

* The display of data in the table is limited to all four of CoolTShirt's website pages

What is the **difference** between **utm_ campaign** & **utm_source**?

- UTM Campaign = **Identifies** the specific **ad** or email **blast** (e.g getting-to-know-cool-tshirts or weekly-newsletter)

- UTM Source = **Identifies** which **touchpoint sent** the **traffic** (e.g nytimes, email, facebook)

The Query

```
SELECT *
FROM page_visits
LIMIT 10;
```

1.2 Campaigns & Sources

How many are there?

- CoolTShirts uses:

- **8 UTM Campaigns:**
1 getting-to-know-cool-tshirts
2 weekly-newsletter
3 ten-crazy-cool-tshirts-facts
4 retargetting-campaign
5 retargetting-ad
6 interview-with-cool-tshirts-founder
7 paid-search
8 cool-tshirts-search

- 6 UTM Sources:
1 nytimes
2 email
3 buzzfeed
4 facebook
5 medium
6 google

How are they related?
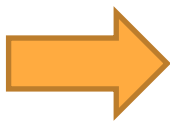
UTM_Campaign	UTM_Source
Getting-to-know-cool-tshirts	NY Times
Weekly-newsletter	EMAIL
Ten-crazy-cool-tshirts-facts	BUZZFEED
Retargetting-campaign	EMAIL
Retargetting-ad	FACEBOOK
Interview-with-cool-tshirts-founder	MEDIUM
Paid-search	GOOGLE
Cool-tshirts-search	GOOGLE

The Query

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

SELECT COUNT (DISTINCT
utm_source)
FROM page_visits;

SELECT DISTINCT
utm_campaign,
utm_source
FROM page_visits;
```

2. The User Journey

2.1 Campaigns' First Touch Attribution (ft_att)

ft_att.utm_source	ft_att.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
facebook	retargetting_ad	0
email	retargetting-campaign	0
email	weekly-newsletter	0
google	paid-search	0

The Query

```
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,
        pv.page_name
  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,
       ft_attr.utm_campaign,
       COUNT(*)
FROM ft_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

2.2 Campaigns' Last Touch Attribution (lt_att)

lt_att.utm_source	lt_att.utm_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-facts	190
medium	interview-with-cool-tshirts-founder	184
google	paid-search	178
google	cool-tshirts-search	60

The Query

```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,
        pv.page_name
  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,
     lt_attr.utm_campaign,
     COUNT(*)
FROM lt_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

2.3 Linking Campaign & Purchase

How many visitors make a purchase?   361

lt_attr.utm_source	lt_att.utm_campaign	COUNT
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

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

The Query

```
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,
         pv.page_name
  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,
       lt_attr.utm_campaign,
       COUNT(*)
FROM lt_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

2.4 The Typical Journey: Part 1 - Calculations

First Touch / Last Touch	Getting To Know Cool Tshirts	Interview with Founder	Ten Crazy Tshirts Facts	Cool Search	Weekly Newsletter	Retargetting Ad	Retargetting Campaign	Paid Search	TOTAL per page
1 - landing page	612 / 612	622 / 623	576 / 576	169 / 168	-	-	-	-	1979 / 1979
2 – shopping cart	675 / 678	513 / 513	563 / 560	131 / 131	-	-	-	-	1882 / 1882
3 - checkout	41 / 41	31 / 31	32 / 32	7 / 7	450 / 450	4445 / 445	246 / 246	179 / 179	1431 / 1431
4 - purchase	9 / 9	7 / 7	9 / 9	2 / 2	115 / 115	113 / 113	54 / 54	52 / 52	361 / 361
TOTAL per campaign	1337 / 1340	1173 / 1174	1180 / 1177	309 / 308	565 / 565	558 / 558	300 / 300	231 / 231	

2.4 The Typical Journey: Part 2 - Diagram *

1 – Landing_page

2 – Shopping_cart

3 – Checkout

4 – Purchase

**Interview
with Founder**

**Getting To
Know Cool
Tshirts**

**Weekly
Newsletter**

**Weekly
Newsletter**

MEDIUM

NY TIMES

EMAIL

EMAIL

** 623

678

450

115

* The typical journey was created based on the highest number of touch attributions coming from each campaign per webpage. It should be noted that it is not necessarily a linear process in all cases.

**As seen in the previous slide, numbers of first and last touch attributions are virtually identical, which is why only last touches are displayed. This does not affect the journey mapped here.

2.4 The Typical Journey: Part 3 - Supporting Queries (First Touch)

Query 1 – Landing_Page

```sql
WITH first_touch AS (
    SELECT user_id,
           MIN(timestamp) as
first_touch_at
    FROM page_visits
    WHERE page_name = '1 -
landing_page'
    GROUP BY user_id),
ft_attr AS (
  SELECT ft.user_id,
         ft.first_touch_at,
         pv.utm_source,
         pv.utm_campaign,
         pv.page_name
  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,
       ft_attr.utm_campaign,
       COUNT(*)
FROM ft_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

Query 2 – Shopping_Cart

```sql
WITH first_touch AS (
    SELECT user_id,
           MIN(timestamp) as
first_touch_at
    FROM page_visits
    WHERE page_name = '2 -
shopping_cart'
    GROUP BY user_id),
ft_attr AS (
  SELECT ft.user_id,
         ft.first_touch_at,
         pv.utm_source,
         pv.utm_campaign,
         pv.page_name
  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,
       ft_attr.utm_campaign,
       COUNT(*)
FROM ft_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

Query 3 - Checkout

```sql
WITH first_touch AS (
    SELECT user_id,
           MIN(timestamp) as
first_touch_at
    FROM page_visits
    WHERE page_name = '3 -
checkout'
    GROUP BY user_id),
ft_attr AS (
  SELECT ft.user_id,
         ft.first_touch_at,
         pv.utm_source,
         pv.utm_campaign,
         pv.page_name
  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,
       ft_attr.utm_campaign,
       COUNT(*)
FROM ft_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

Query 4 – Purchase

```sql
WITH first_touch AS (
    SELECT user_id,
           MIN(timestamp) as
first_touch_at
    FROM page_visits
    WHERE page_name = '4 -
purchase'
    GROUP BY user_id),
ft_attr AS (
  SELECT ft.user_id,
         ft.first_touch_at,
         pv.utm_source,
         pv.utm_campaign,
         pv.page_name
  FROM first_touch AS 'ft'
  INNER JOIN page_visits AS
'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;
```

2.4 The Typical Journey: Part 3 - Supporting Queries (Last Touch)

Query 1 – Landing_Page   Query 2 – Shopping_Cart   Query 3 - Checkout   Query 4 – Purchase

```sql
WITH last_touch AS (
    SELECT user_id,
        MAX(timestamp) as
last_touch_at
    FROM page_visits
    WHERE page_name = '1 -
landing_page'
    GROUP BY user_id),
lt_attr AS (
  SELECT lt.user_id,
        lt.last_touch_at,
        pv.utm_source,
        pv.utm_campaign,
        pv.page_name
  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,
        lt_attr.utm_campaign,
        COUNT(*)
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
    WHERE page_name = '2 -
shopping_cart'
    GROUP BY user_id),
lt_attr AS (
  SELECT lt.user_id,
        lt.last_touch_at,
        pv.utm_source,
        pv.utm_campaign,
        pv.page_name
  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,
        lt_attr.utm_campaign,
        COUNT(*)
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
    WHERE page_name = '3 -
checkout'
    GROUP BY user_id),
lt_attr AS (
  SELECT lt.user_id,
        lt.last_touch_at,
        pv.utm_source,
        pv.utm_campaign,
        pv.page_name
  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,
        lt_attr.utm_campaign,
        COUNT(*)
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
    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,
        pv.page_name
  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,
        lt_attr.utm_campaign,
        COUNT(*)
FROM lt_attr
GROUP BY 1, 2
ORDER BY 3 DESC;
```

3. Budget Optimization

3.1 How to Reinvest*

- **Original goals of marketing campaigns:** **Increase Purchases + Increase Website visits**

First Touch / Last Touch	Getting To Know Cool Tshirts	Interview with Founder	Ten Crazy Tshirts Facts	Cool Search	Weekly Newsletter	Retargetting Ad	Retargetting Campaign	Paid Search
Total Website Touch Attribution	1337 / 1340	1173 / 1174	1180 / 1177	309 / 308	565 / 565	558 / 558	300 / 300	231 / 231
Total Purchase Touch Attribution	9 / 9	7 / 7	9 / 9	2 / 2	115 / 115	113 / 113	54 / 54	52 / 52

CoolTShirts should reinvest into the 5 in black highlighted campaigns:

➡️ ✓ **Weekly Newsletter** - highest purchases   115
    ✓ **Retargetting Ad** - 2nd highest purchases   113   ➡️   334/ 361 purchases
    ✓ **Retargetting Campaign** - 3rd highest purchases   54
    ✓ **Paid Search** - 4th highest purchases   52

    ✓ **Getting To Know Cool Tshirts** - highest website visits + highest purchases out of campaigns with fewer purchases

* This strategy was conceived based on the argument of reinvesting in (strengthening) those campaigns that we already the most successful