Attribution Queries

CoolTShirts sells shirts of all kinds, as long as they are T-shaped and cool. Recently, CTS started a few marketing campaigns to increase website visits and purchases. Using touch attribution, they'd like to map their customers' journey: from initial visit to purchase. They can use that information to optimize their marketing campaigns. This project will guide you through some of that process.

At this point you should have taken the lesson on touch attribution. This project will guide you through a number of queries on CoolTShirts' user data (see the schema here).

Check off the steps as you go. In the rare chance that your project gets stuck in a bad state — maybe you accidentally edited the data —, refresh the page.

1. 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.

```
select count(distinct utm_campaign)
from page_visits
limit 5;
```

```
select count(distinct utm_source)
from page_visits
limit 5;
```

```
select distinct utm_campaign, utm_source
from page_visits
group by 1
limit 10;
```

2. What pages are on the CoolTShirts website? Find the distinct values of the `page_name` column.

```
select distinct page_name
from page_visits
limit 5;
```

3. How many first touches is each campaign responsible for? You'll need to use the first-touch query from the lesson (also provided in the hint below). Group by campaign and count the number of first touches for each.

```
 WITH first_touch AS (
    SELECT user_id,
        MIN(timestamp) as first_touch_at
    FROM page_visits
    GROUP BY user_id)
SELECT pv.utm_campaign, count(ft.first_touch_at) as first_touch_count
FROM first_touch ft
JOIN page_visits pv
    ON ft.user_id = pv.user_id
    AND ft.first_touch_at = pv.timestamp
    group by 1
    order by 2 desc
  limit 5;
```

```
 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? Starting with the last-touch query from the lesson, group by campaign and count the number of last touches for each.

```
WITH last_touch AS (
   SELECT user_id,
        max(timestamp) as last_touch_at
    FROM page_visits
    GROUP BY user_id),
ft_attr AS (
  SELECT ft.user_id,
         ft.last_touch_at,
         pv.utm_source,
         pv.utm_campaign
  FROM last_touch ft
  JOIN page_visits pv
    ON ft.user_id = pv.user_id
    AND ft.last_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;
```

5. How many visitors make a purchase? Count the distinct users who visited the page named `4 - purchase`.

```
select count(distinct(user_id))
from page_visits
where page_name = '4 - purchase';
```

6. 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.

```
WITH last_touch AS (
   SELECT user_id,
        max(timestamp) as last_touch_at
    FROM page_visits
    GROUP BY user_id),
ft_attr AS (
  SELECT ft.user_id,
         ft.last_touch_at,
         pv.utm_source,
         pv.utm_campaign,
        pv.page_name
  FROM last_touch ft
  JOIN page_visits pv
    ON ft.user_id = pv.user_id
    AND ft.last_touch_at = pv.timestamp
)
SELECT ft_attr.utm_source,
       ft_attr.utm_campaign,
       ft_attr.page_name,
       COUNT(*)
FROM ft_attr
where page_name = '4 - purchase'
GROUP BY 1, 2
ORDER BY 4 DESC;
```

7. CoolTShirts can re-invest in 5 campaigns. Given your findings in the project, which should they pick and why?

```
-- case 1

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 asc;

-- recommendation: google

--case 2

WITH last_touch AS (
   SELECT user_id,
        max(timestamp) as last_touch_at
    FROM page_visits
    GROUP BY user_id),
ft_attr AS (
  SELECT ft.user_id,
         ft.last_touch_at,
         pv.utm_source,
         pv.utm_campaign,
        pv.page_name
  FROM last_touch ft
  JOIN page_visits pv
    ON ft.user_id = pv.user_id
    AND ft.last_touch_at = pv.timestamp
)
SELECT ft_attr.utm_source,
       ft_attr.utm_campaign,
       ft_attr.page_name,
       COUNT(*)
FROM ft_attr
where page_name = '3 - checkout'
```

```
GROUP BY 1, 2
ORDER BY 4 asc;

-- recommendation:
-- cool-tshirts-search
-- ten-crazy-cool-tshirts-facts
-- interview-with-cool-tshirts-founder
-- getting-to-know-cool-tshirts
-- paid-search
```