FIRST- AND LAST-TOUCH ATTRIBUTION

Introduction

Think of your favorite website: how did you find it? Did you use a search engine? Or click on an ad? Or follow a link in a blog post?

Web developers, marketers, and data analysts use that information to improve their *sources* (sometimes called channels or touchpoints) online. If an ad campaign drives a lot of visits to their site, then they know that source is working! We say that those visits are **attributed** to the ad campaign.

But how do websites capture that information? The answer is *UTM parameters*. These parameters capture when and how a user finds the site. Site owners use special links containing UTM parameters in their ads, blog posts, and other sources. When a user clicks one, a row is added to a database describing their page visit. You can see a common schema for a "page visits" table below and at this link.

- `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 touchpoint sent the traffic (e.g. google, email, or facebook)
- `utm_medium` - Identifies what type of link was used (e.g. cost-per-click or email)
- `utm_campaign` - Identifies the specific ad or email blast (e.g. retargetting-ad or weekly-newsletter)

In this lesson, you will learn how to use SQL, UTM parameters, and touch attribution to draw insights from this data!

1. Select the first 10 rows of the `page_visits` table. Make sure you understand the significance of each column before moving on!

```
select *
from page_visits
limit 10;
```

First Touch Example

Imagine June. She wants to buy a new t-shirt for her mother, who is visiting from out of town. She reads about CoolTShirts.com in a Buzzfeed article, and clicks a link to their landing page. June finds a fabulous Ninja Turtle t-shirt and adds it to her cart. Before she can advance to the checkout page her mom calls, asking for directions. June navigates away from CoolTShirts.com to look up directions. June's initial visit is logged in the `page_visits` table as follows:

user_id	timestamp	page_name	utm_source
10069	2018-01-02 23:14:01	1 landing_page	buzzfeed

user_id	timestamp	page_name	utm_source
10069	2018-01-02 23:55:01	2 - shopping_cart	buzzfeed

- June's *first touch* — the first time she was exposed to CoolTShirts.com — is attributed to `buzzfeed`
- June is assigned a user id of `10069`
- She visited the landing page at `23:14:01` and the shopping cart at `23:55:01`

1. Find June's rows in the table! Select all columns from the `page_visits` table, using a `WHERE` clause with:

   - `user_id` equals `10069` and
   - `utm_source` equals `'buzzfeed'`.

```
select *
from page_visits
where user_id = 10069
and utm_source = 'buzzfeed'
limit 10;
```

Last Touch Example

Two days later, CoolTShirts.com runs an ad on June's Facebook page. June remembers how much she wanted that Ninja Turtles t-shirt, and follows the ad back to CoolTShirts.com. She now has the following rows in `page_visits` table:

user_id	timestamp	page_name	utm_source
10069	2018-01-02 23:14:01	1 - landing_page	buzzfeed
10069	2018-01-02 23:14:01	2 - shopping_cart	buzzfeed
10069	2018-01-04 08:12:01	3 - checkout	facebook
10069	2018-01-04 08:13:01	4 - purchase	facebook

- June's *last touch* — the exposure to CoolTShirts.com that led to a purchase — is attributed to `facebook`
- She visited the checkout page at `08:12:01` and the purchase page at `08:13:01`

1. Find all of of June's rows, using a `WHERE` clause with just `user_id`.

```
select *
from page_visits
where user_id = 10069;
```

First versus Last

If you want to increase sales at CoolTShirts.com, would you count on `buzzfeed` or increase `facebook` ads? The real question is: should June's purchase be attributed to `buzzfeed` or to `facebook`?

There are two ways of analyzing this:

- *First-touch attribution* only considers the first `utm_source` for each customer, which would be `buzzfeed` in this case. This is a good way of knowing how visitors initially discover a website.
- *Last-touch attribution* only considers the last `utm_source` for each customer, which would be `facebook` in this case. This is a good way of knowing how visitors are drawn back to a website, especially for making a final purchase.

The results can be crucial to improving a company's marketing and online presence. Most companies analyze both first- and last-touch attribution and display the results separately.

1. June's sister also visited CoolTShirts.com! Select all columns from the `page_visits` table, using a `WHERE` clause with `user_id = 10329`. What are her first- and last-touch attributions?

```
select *
from page_visits
where user_id = 10329;
```

The Attribution Query I

We just learned how to attribute a user's first and last touches. What if we want to attribute the first and last touches for ALL users? This is where SQL comes in handy — with one query we can find all first- or last-touch attributions (the first and last versions are nearly identical). We can save this query to run it later, or modify it for a subset of users. Let's learn the query...

In order to get first-touch attributions, we need to find the first time that a user interacted with our website. We do this by using a `GROUP BY`. Let's call this table `first_touch`:

```
SELECT user_id,
   MIN(timestamp) AS 'first_touch_at'
FROM page_visits
GROUP BY user_id;
```

This tells us the first time that each user visited our site, but does not tell us how they got to our site — the query results have no UTM parameters! We'll see how do get those in the next exercise.

1. Find all last touches. Your query will look similar to the `first_touch` query above.

```
SELECT user_id,
MAX(timestamp) as last_touch_at
FROM page_visits
GROUP BY 1;
```

2. Make sure June's last touch at `08:13:01` is in the data.
Add a `WHERE` clause for `user_id = 10069` to your existing query.

```
SELECT user_id,
MAX(timestamp) as last_touch_at
FROM page_visits
WHERE user_id = 10069
GROUP BY 1;
```

The Attribution Query II

To get the UTM parameters, we'll need to `JOIN` these results back with the original table.
We'll join tables `first_touch`, aka`ft`, and `page_visits`, aka `pv`, on `user_id` and `timestamp`.

```
ft.user_id = pv.user_id
AND ft.first_touch_at = pv.timestamp
```

Remember that `first_touch_at` is the earliest timestamp for each user. Here's the simplified query:

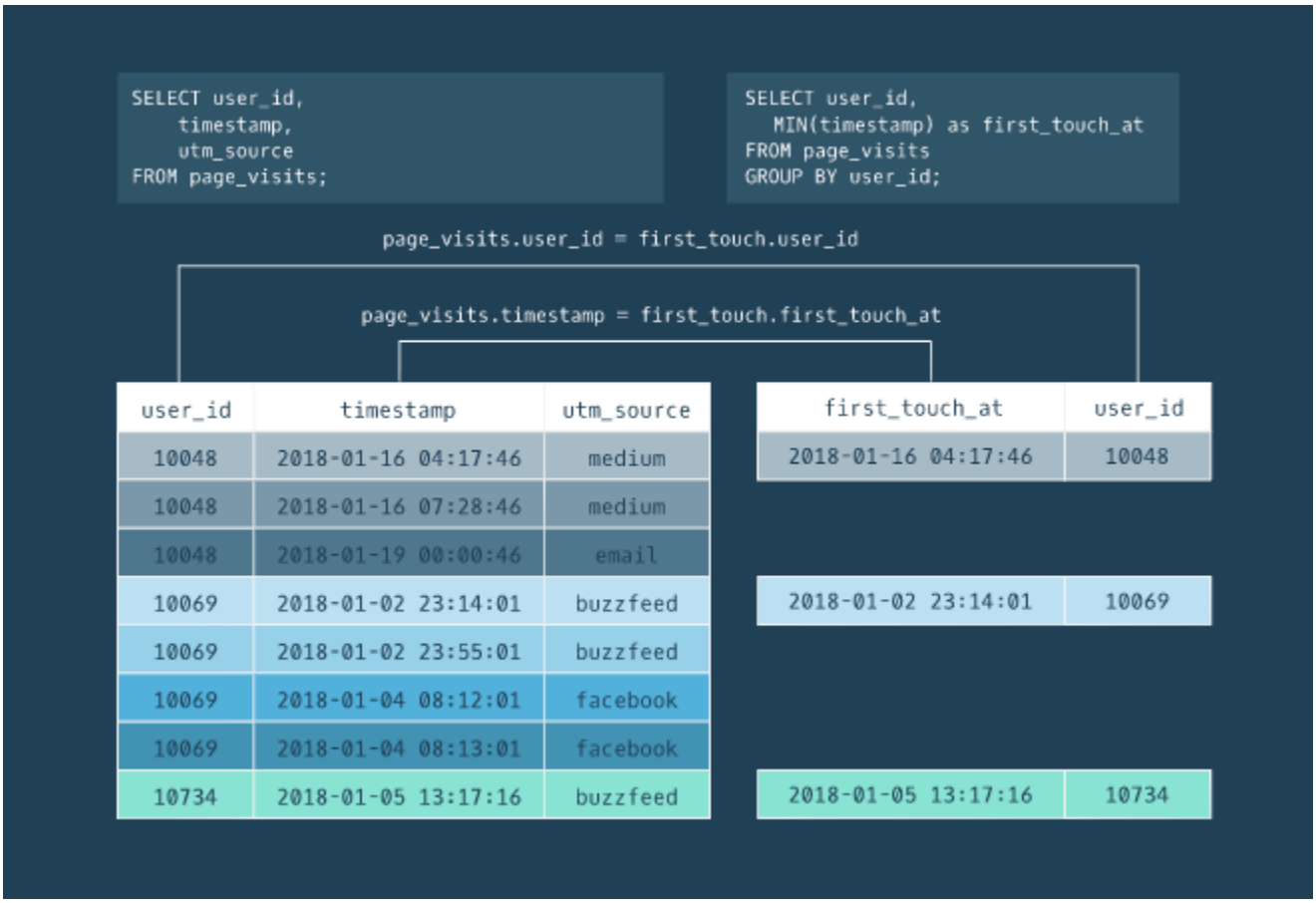
```
WITH first_touch AS (
      /* ... */
    )
SELECT *
FROM first_touch AS 'ft'
JOIN page_visits AS 'pv'
  ON ft.user_id = pv.user_id
  AND ft.first_touch_at = pv.timestamp;
```

Now fill in the `WITH` clause using the `first_touch` query from the previous exercise. We'll also specify the columns to `SELECT`.

```
WITH first_touch AS (
   SELECT user_id,
      MIN(timestamp) AS 'first_touch_at'
   FROM page_visits
   GROUP BY user_id)
SELECT ft.user_id,
  ft.first_touch_at,
  pv.utm_source
FROM first_touch AS 'ft'
JOIN page_visits AS 'pv'
  ON ft.user_id = pv.user_id
  AND ft.first_touch_at = pv.timestamp;
```

The diagram on the right illustrates the `JOIN` we need to get the UTM parameters of each first touch:

- On the left is `page_visits` (just three columns from the original table). We get the UTM parameters from there.
- On the right is `first_touch` (the result of the `GROUP BY` query). We get the first touches from there.



The Attribution Query III

We can easily modify the first-touch attribution query to get last-touch attribution: use `MAX(timestamp)` instead of `MIN(timestamp)`. For reference, the first-touch attribution query is shown below.

```
WITH first_touch AS (
    SELECT user_id,
       MIN(timestamp) AS 'first_touch_at'
    FROM page_visits
    GROUP BY user_id)
SELECT ft.user_id,
   ft.first_touch_at,
   pv.utm_source
FROM first_touch AS 'ft'
JOIN page_visits AS 'pv'
   ON ft.user_id = pv.user_id
```

Credit: codeacademy
Compiled by: Sumaia Parveen Shupti

```
   AND ft.first_touch_at = pv.timestamp;
```

1. Using the query above as a guide, write the LAST-touch attribution query and run it.
2. Make sure June's last touch at `08:13:01` is still there! Add a `WHERE` clause for `user_id = 10069` to your existing query.

```
WITH last_touch AS (
    SELECT user_id,
    max(timestamp) AS 'last_touch_at'
    FROM page_visits
    GROUP BY user_id)
SELECT lt.user_id,
   lt.last_touch_at,
   pv.utm_source
FROM last_touch AS 'lt'

JOIN page_visits AS 'pv'
   ON lt.user_id = pv.user_id
   AND lt.last_touch_at = pv.timestamp
   where lt.user_id = 10069
   limit 10;
```

Review

You can now wield SQL to find where, when, and how users are visiting a website. Well done! Here's a summary of what you learned:

- *UTM parameters* are a way of tracking visits to a website. Developers, marketers, and analysts use them to capture information like the time, attribution source, and attribution medium for each user visit.
- *First-touch attribution* only considers the first source for each customer. This is a good way of knowing how visitors initially discover a website.
- *Last-touch attribution* only considers the last source for each customer. This is a good way of knowing how visitors are drawn back to a website, especially for making a final purchase.
- Find first and last touches by grouping `page_visits` by `user_id` and finding the `MIN` and `MAX` of `timestamp`.
- To find first- and last-touch attribution, join that table back with the original `page_visits` table on `user_id` and `timestamp`.