codecademy

CoolTShirts Capstone

Learn SQL from Scratch

Matthew Myers

09/21/18

Example Table of Contents

1. Learn SQL From Scratch : First- and Last-Touch Attribution code and explainations

1. Learn SQL From Scratch : First- and Last-Touch Attribution code and explainations

0.1 Default code page

This is the default code displayed when opening the Capstone Project: First- and Last-touch Attribution with CoolTShirts.com

- WITH AS argument creates a new table as ‘last_touch_at’ and selects the user_id with the last or most recent timestamp
- The AS arguments rename columns (‘last_touch’ and ‘page_visits’) with an abbreviated ‘lt’ and ‘pv’, respectively.
- The JOIN argument joins tables: page_visits and last_touch_at on respective columns: user_id and last_touch_at/timestamp

user_id	last_touch_at	utm_source
10069	2018-01-04 08:13:01	facebook

```
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;
```

1.1 Introduction

Select the first 10 rows of table page_visits

- The SELECT * argument selects all columns FROM table page_visits
- The LIMIT argument limits the result to the number (e.g. 10) of rows identified
- Additional columns not shown are: utm_campaign and utm_source

Page_name	Timestamp	User_id
1 - landing_page	2018-01-24 03:12:16	10006

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

2.1 First Touch Example

Select June's (example user's first name) rows

- The SELECT * argument selects all columns FROM table page_visits
- The WHERE argument selects only the rows matching the column criterion (two criterion)
- Additional columns not shown are: page_name and timestamp

```
SELECT *
FROM page_visits
WHERE user_id = 10069
              AND utm_source = 'buzzfeed';
```

User_id	Utm_campaign	Utm_source
10069	ten-crazy-cool-tshirts-facts	buzzfeed

3.1 Last Touch Example

Select June's (example user's first name) rows

- The SELECT * argument selects all columns FROM table page_visits
- The WHERE argument selects only the rows matching the column criterion (one criterion only)
- Additional columns not shown are: page_name and timestamp

```
SELECT *
FROM page_visits
WHERE user_id = 10069;
```

User_id	Utm_campaign	Utm_source
10069	ten-crazy-cool-tshirts-facts	buzzfeed

4.1 First versus Last

Select June's sister's (example user's first name) rows

- The SELECT * argument selects all columns FROM table page_visits
- The WHERE argument selects only the rows matching the column criterion (one criterion only)
- Additional columns not shown are: page_name and timestamp

```
SELECT *
FROM page_visits
WHERE user_id = 10329;
```

User_id	Utm_campaign	Utm_source
10329	interview-with-cool-tshirts-founder	medium

5.1 The Attribution Query I

What if we want to attribute the first and last touches for ALL users? The SELECT user_id, argument selects all columns FROM table page_visits

- The SELECT argument selects all rows user_id and most recent (or MAX) timestamp
- The GROUP BY argument returns only one row per column specified
- Additional columns not shown because they are not SELECTed

User_id	timestamp	
10006	2018-01-25 23:10:16	

```
SELECT user_id,
            MAX(timestamp) AS 'last_touch_at'
FROM page_visits
GROUP BY user_id;
```

5.2 The Attribution Query I

What if we want just June's last timestamp?

- The SELECT argument selects all rows user_id and most recent (or MAX) timestamp and AS creates new column
- The WHERE argument limits to criterion
- The GROUP BY argument returns only one row per column specified

User_id	Last_touch_at	
10069	2018-01-04 08:13:01	

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

6.1 The Attribution Query II

Select the user's first touch (MIN timestamp) on joined table?

- WITH AS argument creates and names new table
- The SELECT argument selects all rows user_id and most recent (or MIN) timestamp and AS creates new column
- The JOIN argument joins tables ON columns specified
- The GROUP BY argument returns only one row per column specified

User_id	First_touch_at	Utm_source
10006	2018-01-24 03:12:16	nytimes

```
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;
```

7.1 The Attribution Query III

Select the user's last touch (MAX timestamp) on joined table?

- WITH AS argument creates and names new table
- The SELECT argument selects all rows user_id and least recent (or MAX) timestamp and AS creates new column
- The JOIN argument joins tables ON columns specified
- The GROUP BY argument returns only one row per column specified

User_id	First_touch_at	Utm_source
10006	2018-01-25 23:10:16	email

```
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;
```

7.2 The Attribution Query III

Select the June's (WHERE new table user_id column equals user) last touch (MAX timestamp) on joined table?

- WITH AS argument creates and names new table
- The SELECT argument selects all rows user_id and least recent (or MAX) timestamp and AS creates new column
- The JOIN argument joins tables ON columns specified
- The GROUP BY argument returns only one row per column specified

User_id	First_touch_at	Utm_source
10069	2018-01-04 08:13:01	facebook

```
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;
```