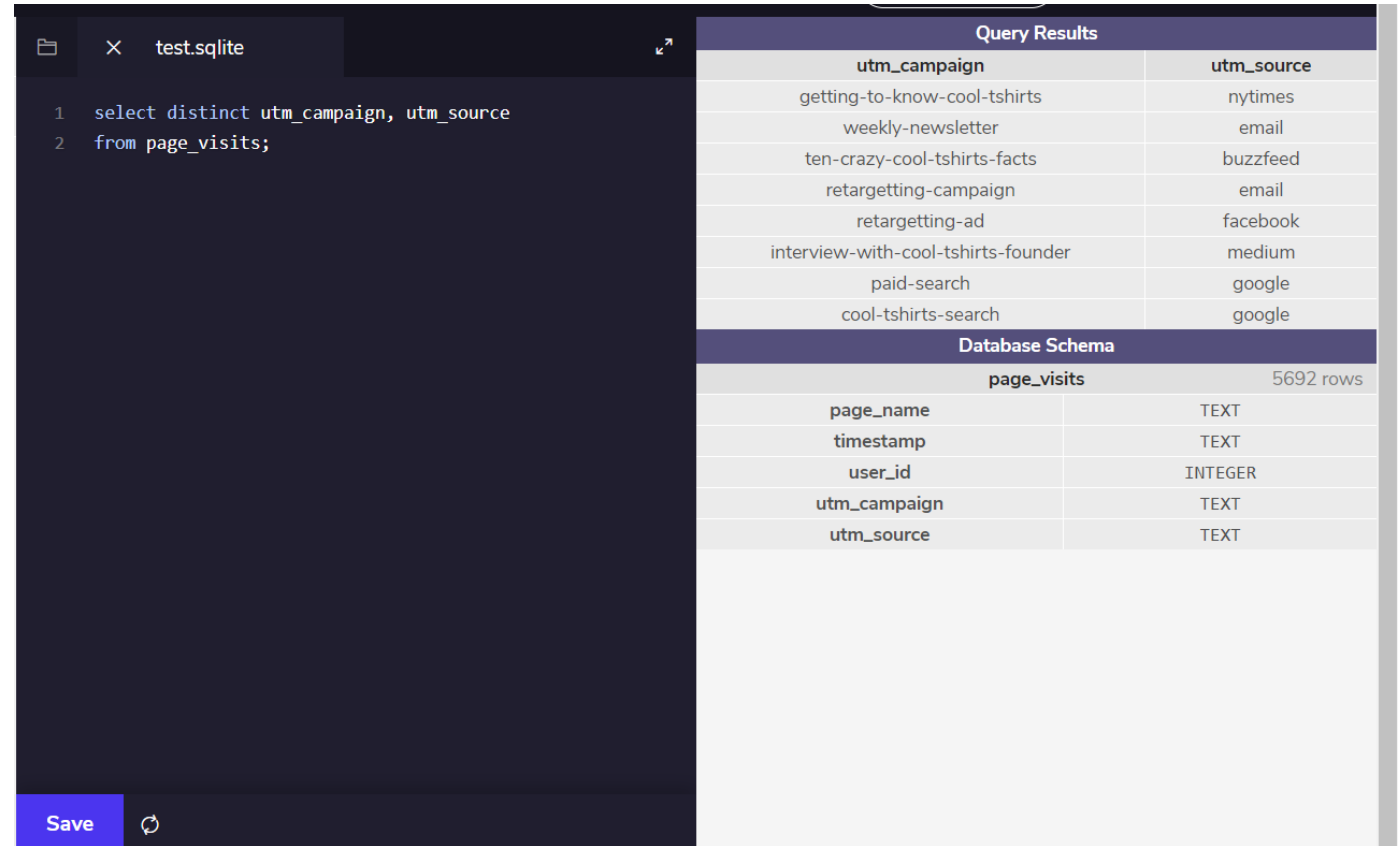


First Touch Last Touch

Capstone by Anmol Hegde

Campaigns and Sources

- CoolTShirts.com has 8 distinct campaigns:
 - Getting to Know Cool T-Shirts
 - Weekly Newsletter
 - Ten Cool T-Shirts Facts
 - Retargeting Campaign
 - Retargeting Ad
 - Interview with Cool T-Shirts Founder
 - Paid Search
 - Cool T-Shirts Search
- CoolTShirts.com has 6 distinct sources:
 - New York Times
 - Email
 - Buzzfeed
 - Facebook
 - Medium
 - Google
- Campaigns are specific ads or emails sent out by the site while sources are the outside platform where the user found the ad



The screenshot shows a SQLite database interface with a query editor on the left and query results on the right. The query editor contains the following SQL code:

```
1 select distinct utm_campaign, utm_source
2 from page_visits;
```

The query results are displayed in a table with two columns: **utm_campaign** and **utm_source**. The results are as follows:

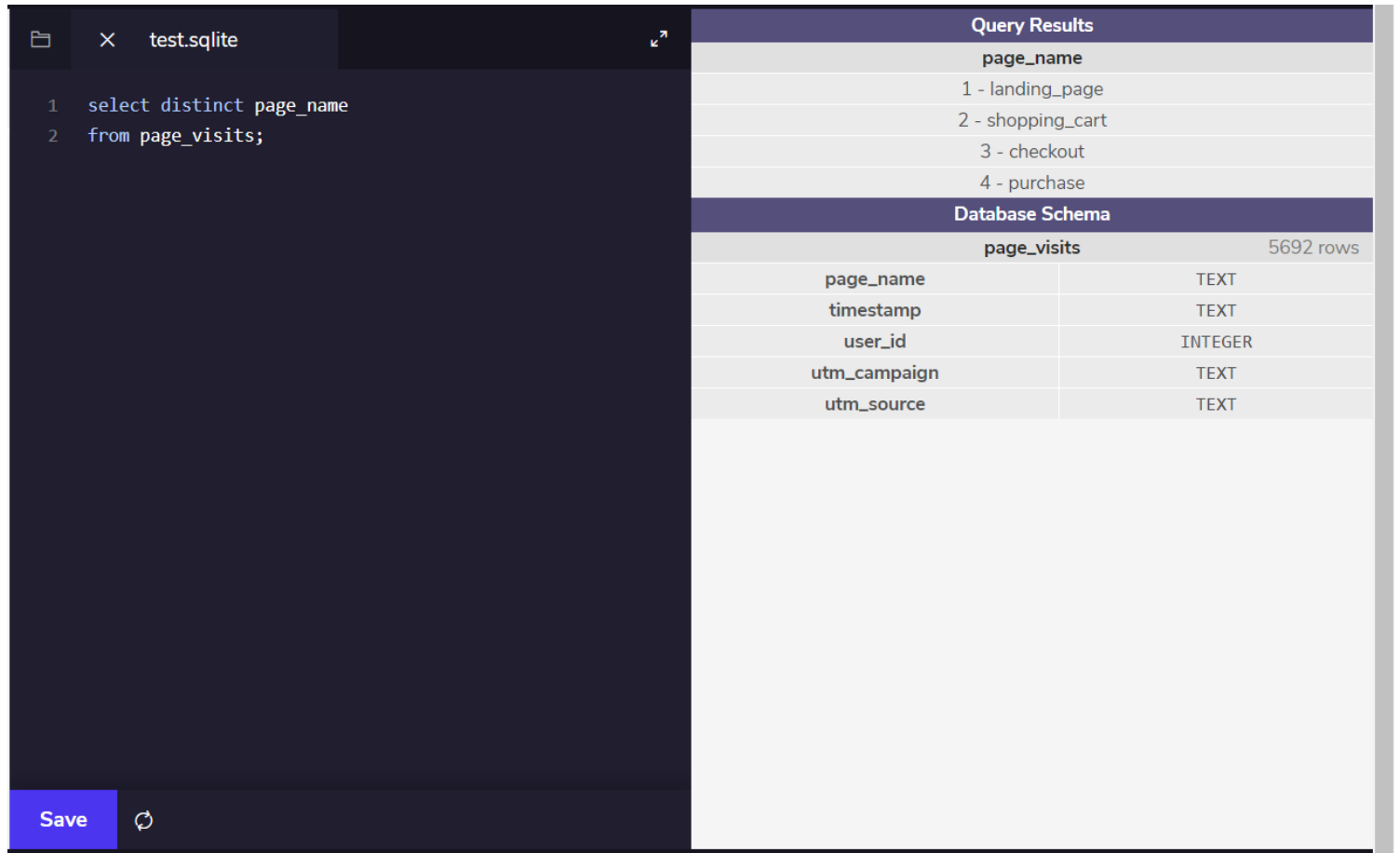
utm_campaign	utm_source
getting-to-know-cool-tshirts	nytimes
weekly-newsletter	email
ten-crazy-cool-tshirts-facts	buzzfeed
retargeting-campaign	email
retargeting-ad	facebook
interview-with-cool-tshirts-founder	medium
paid-search	google
cool-tshirts-search	google

Below the query results, the database schema is shown. It includes a table named **page_visits** with 5692 rows. The schema details are as follows:

Database Schema	
page_visits 5692 rows	
page_name	TEXT
timestamp	TEXT
user_id	INTEGER
utm_campaign	TEXT
utm_source	TEXT

Pages

- The Cool T-Shirts website has four pages:
 1. Landing Page
 2. Shopping Cart
 3. Checkout
 4. Purchase



The screenshot shows a SQLite database interface with a dark theme. On the left, a code editor displays a SQL query: `1 select distinct page_name` and `2 from page_visits;`. Below the editor is a blue 'Save' button and a refresh icon. On the right, the 'Query Results' panel shows the output of the query as a table with four rows: '1 - landing_page', '2 - shopping_cart', '3 - checkout', and '4 - purchase'. Below this, the 'Database Schema' section shows the 'page_visits' table with 5692 rows and five columns: 'page_name' (TEXT), 'timestamp' (TEXT), 'user_id' (INTEGER), 'utm_campaign' (TEXT), and 'utm_source' (TEXT).

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

Database Schema	
page_visits	
	5692 rows
page_name	TEXT
timestamp	TEXT
user_id	INTEGER
utm_campaign	TEXT
utm_source	TEXT

First Touches

- Each campaign is responsible for the following number of first touches:
 - Interview with Cool T-Shirt Founder: 622
 - Getting to Know Cool T-Shirts: 612
 - Ten Crazy Cool T-Shirt Facts: 576
 - Cool T-Shirts Search: 169

The screenshot shows a database interface with a SQL editor on the left and a results panel on the right. The SQL editor contains a query to find the first touch for each user by joining the `first_touch` and `page_visits` tables. The results panel displays the query results as a table with two columns: `Campaign` and `First Touches`. Below the results, the database schema for the `page_visits` table is shown, indicating it has 5692 rows. The schema lists columns: `page_name` (TEXT), `timestamp` (TEXT), `user_id` (INTEGER), `utm_campaign` (TEXT), and `utm_source` (TEXT).

```
1 WITH first_touch AS (  
2     SELECT user_id,  
3           MIN(timestamp) as first_touch_at  
4     FROM page_visits  
5     GROUP BY user_id),  
6 first_touch_join AS (SELECT ft.user_id,  
7                             ft.first_touch_at,  
8                             pv.utm_source,  
9                             pv.utm_campaign  
10    FROM first_touch AS 'ft'  
11   JOIN page_visits AS 'pv'  
12     ON ft.user_id = pv.user_id  
13     AND ft.first_touch_at = pv.timestamp)  
14   SELECT first_touch_join.utm_campaign as 'Campaign',  
15          COUNT(*) as 'First Touches'  
16   FROM first_touch_join  
17  GROUP BY 1  
18  ORDER BY 2 DESC;
```

Query Results	
Campaign	First Touches
interview-with-cool-tshirts-founder	622
getting-to-know-cool-tshirts	612
ten-crazy-cool-tshirts-facts	576
cool-tshirts-search	169

Database Schema	
page_visits 5692 rows	
page_name	TEXT
timestamp	TEXT
user_id	INTEGER
utm_campaign	TEXT
utm_source	TEXT

Save ↺

Last Touches

- Each campaign is responsible for the following number of last touches:

- Weekly Newsletter: 447
- Retargeting Ad: 443
- Retargeting Campaign: 245
- Getting to Know Cool T-Shirts: 232
- Ten Cool T-Shirts Facts: 190
- Interview with Cool T-Shirts Founder: 184
- Paid Search: 178
- Cool T-Shirts Search: 60

test.sqlite		Query Results	
		Campaign	Last Touches
1 WITH last_touch AS (2 SELECT user_id, 3 MAX(timestamp) as last_touch_at 4 FROM page_visits 5 GROUP BY user_id), 6 last_touch_join AS (SELECT lt.user_id, 7 lt.last_touch_at, 8 pv.utm_source, 9 pv.utm_campaign 10 FROM last_touch AS 'lt' 11 JOIN page_visits AS 'pv' 12 ON lt.user_id = pv.user_id 13 AND lt.last_touch_at = pv.timestamp) 14 SELECT last_touch_join.utm_campaign as 'Campaign', 15 COUNT(*) as 'Last Touches' 16 FROM last_touch_join 17 GROUP BY 1 18 ORDER BY 2 DESC;		weekly-newsletter	447
		retargeting-ad	443
		retargeting-campaign	245
		getting-to-know-cool-tshirts	232
		ten-crazy-cool-tshirts-facts	190
		interview-with-cool-tshirts-founder	184
		paid-search	178
		cool-tshirts-search	60
		Database Schema	
		page_visits 5692 rows	
		page_name	TEXT
		timestamp	TEXT
		user_id	INTEGER
		utm_campaign	TEXT
		utm_source	TEXT
Save			

Purchases

- 361 users make a purchase
- Each campaign is responsible for the following number of last touches leading to purchases:
 - Cool T-Shirts Search: 2
 - Getting to Know Cool T-Shirts: 9
 - Interview with Cool T-Shirts Founder: 7
 - Paid Search: 52
 - Retargeting Ad: 113
 - Retargeting Campaign: 54
 - Ten Crazy Cool T-Shirts Facts: 9
 - Weekly Newsletter: 115

The image displays two SQL queries and their corresponding results in a database interface.

Top Query:

```
1 WITH last_touch AS (  
2   SELECT user_id,  
3     MAX(timestamp) as last_touch_at  
4   FROM page_visits  
5   WHERE page_name = '4 - purchase'  
6   GROUP BY user_id),  
7 last_touch_join AS (SELECT lt.user_id,  
8   lt.last_touch_at,  
9   pv.utm_source,  
10  pv.utm_campaign  
11 FROM last_touch AS 'lt'  
12 JOIN page_visits AS 'pv'  
13   ON lt.user_id = pv.user_id  
14   AND lt.last_touch_at = pv.timestamp)  
15 SELECT COUNT(DISTINCT user_id) AS 'Purchase Users'  
16 FROM last_touch_join;
```

Query Results (Top):

Purchase Users	
361	

Database Schema (Top):

page_visits	
page_name	TEXT
timestamp	TEXT
user_id	INTEGER
utm_campaign	TEXT
utm_source	TEXT

Bottom Query:

```
1 WITH last_touch AS (  
2   SELECT user_id,  
3     MAX(timestamp) as last_touch_at  
4   FROM page_visits  
5   WHERE page_name = '4 - purchase'  
6   GROUP BY user_id),  
7 last_touch_join AS (SELECT lt.user_id,  
8   lt.last_touch_at,  
9   pv.utm_source,  
10  pv.utm_campaign  
11 FROM last_touch AS 'lt'  
12 JOIN page_visits AS 'pv'  
13   ON lt.user_id = pv.user_id  
14   AND lt.last_touch_at = pv.timestamp)  
15 SELECT utm_campaign, COUNT(*) AS 'Purchase Users'  
16 FROM last_touch_join  
17 GROUP BY 1;
```

Query Results (Bottom):

utm_campaign	Purchase Users
cool-tshirts-search	2
getting-to-know-cool-tshirts	9
interview-with-cool-tshirts-founder	7
paid-search	52
retargeting-ad	113
retargeting-campaign	54
ten-crazy-cool-tshirts-facts	9
weekly-newsletter	115

Database Schema (Bottom):

page_visits	
page_name	TEXT
timestamp	TEXT
user_id	INTEGER
utm_campaign	TEXT
utm_source	TEXT

User Journey

- The typical user finds their way to the site primarily through the Interview with Cool T-Shirts Founder or Getting to Know Cool T-Shirts
- This user ultimately makes a purchase when sent the Weekly Newsletter or Retargeting Ad

Query Results	
Campaign	First Touches
interview-with-cool-tshirts-founder	622
getting-to-know-cool-tshirts	612
ten-crazy-cool-tshirts-facts	576
cool-tshirts-search	169
Database Schema	
page_visits	
5692 rows	
page_name	TEXT
timestamp	TEXT
user_id	INTEGER
utm_campaign	TEXT
utm_source	TEXT

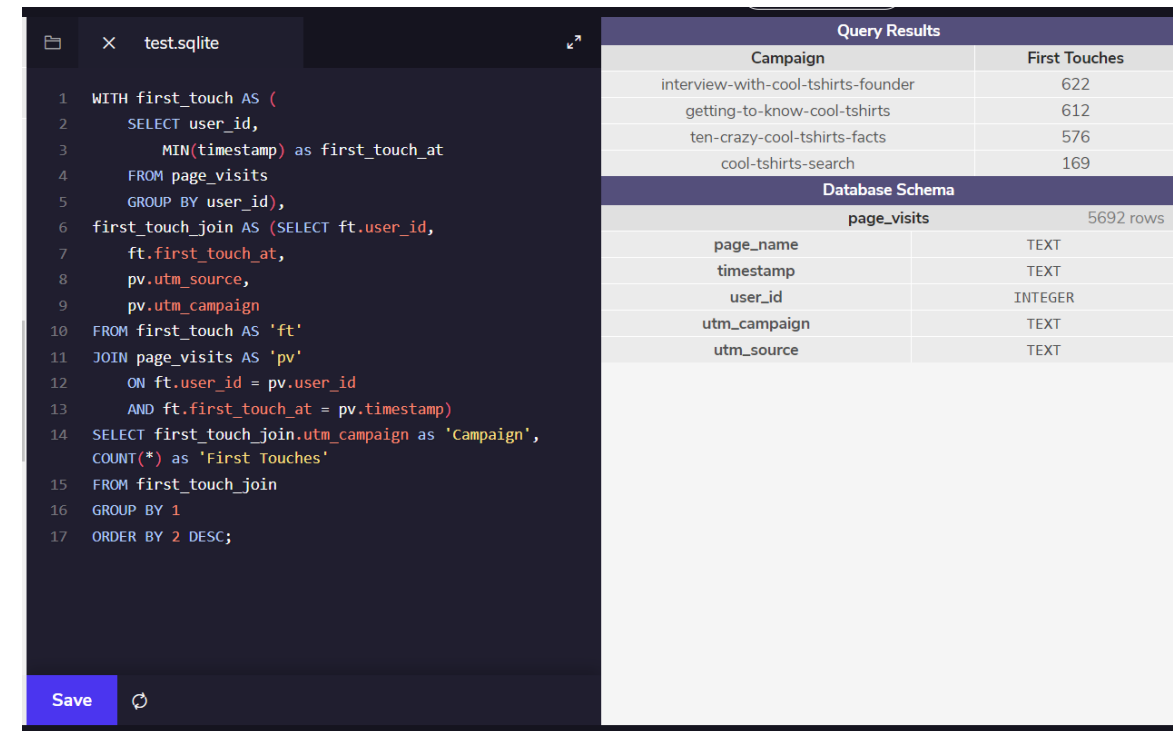
Query Results	
Campaign	First Touches
interview-with-cool-tshirts-founder	622
getting-to-know-cool-tshirts	612
ten-crazy-cool-tshirts-facts	576
cool-tshirts-search	169
Database Schema	
page_visits	
5692 rows	
page_name	TEXT
timestamp	TEXT
user_id	INTEGER
utm_campaign	TEXT
utm_source	TEXT

Query Results	
Campaign	Last Touches
weekly-newsletter	447
retargeting-ad	443
retargeting-campaign	245
getting-to-know-cool-tshirts	232
ten-crazy-cool-tshirts-facts	190
interview-with-cool-tshirts-founder	184
paid-search	178
cool-tshirts-search	60
Database Schema	
page_visits	
5692 rows	
page_name	TEXT
timestamp	TEXT
user_id	INTEGER
utm_campaign	TEXT
utm_source	TEXT

Query Results	
Campaign	Last Touches
weekly-newsletter	447
retargeting-ad	443
retargeting-campaign	245
getting-to-know-cool-tshirts	232
ten-crazy-cool-tshirts-facts	190
interview-with-cool-tshirts-founder	184
paid-search	178
cool-tshirts-search	60
Database Schema	
page_visits	
5692 rows	
page_name	TEXT
timestamp	TEXT
user_id	INTEGER
utm_campaign	TEXT
utm_source	TEXT

Campaign Budget Optimization

- Five campaigns I recommend to reinvest in:
 - Interview with Cool T-Shirts Founder
 - This campaign entices a lot of initial viewers perhaps due to its personable nature
 - It would grow exposure to the brand
 - Getting to Know Cool T-Shirts
 - This campaign also entices first time users perhaps due to its catchy name
 - Making the company and its employees relatable



The screenshot shows a database interface with a SQL query editor on the left and query results on the right. The query is a complex JOIN statement designed to identify the first touchpoint for each user by joining 'page_visits' and 'first_touch' tables. The results table shows four campaigns ranked by the number of first touches, with 'interview-with-cool-tshirts-founder' having the highest count at 622.

```
1 WITH first_touch AS (  
2     SELECT user_id,  
3         MIN(timestamp) as first_touch_at  
4     FROM page_visits  
5     GROUP BY user_id),  
6 first_touch_join AS (SELECT ft.user_id,  
7     ft.first_touch_at,  
8     pv.utm_source,  
9     pv.utm_campaign  
10    FROM first_touch AS 'ft'  
11    JOIN page_visits AS 'pv'  
12      ON ft.user_id = pv.user_id  
13     AND ft.first_touch_at = pv.timestamp)  
14    SELECT first_touch_join.utm_campaign as 'Campaign',  
15           COUNT(*) as 'First Touches'  
16    FROM first_touch_join  
17   GROUP BY 1  
18   ORDER BY 2 DESC;
```

Query Results	
Campaign	First Touches
interview-with-cool-tshirts-founder	622
getting-to-know-cool-tshirts	612
ten-crazy-cool-tshirts-facts	576
cool-tshirts-search	169

Database Schema	
page_visits 5692 rows	
page_name	TEXT
timestamp	TEXT
user_id	INTEGER
utm_campaign	TEXT
utm_source	TEXT

Campaign Budget Optimization cont.

- Weekly newsletter
 - This campaign led most people to finally purchase an item
 - Good way to show people weekly catalogues of products
- Retargetting Ads
 - This campaign led to second most final purchases
 - Targets already interested users with specialized content
- Retargetting Campaign
 - Third highest purchase rate of all campaigns which is the ultimate goal

Query Results	
Campaign	Last Touches
weekly-newsletter	447
retargetting-ad	443
retargetting-campaign	245
getting-to-know-cool-tshirts	232
ten-crazy-cool-tshirts-facts	190
interview-with-cool-tshirts-founder	184
paid-search	178
cool-tshirts-search	60
Database Schema	
page_visits	
5692 rows	
page_name	TEXT
timestamp	TEXT
user_id	INTEGER
utm_campaign	TEXT
utm_source	TEXT

Query Results	
utm_campaign	Purchase Users
cool-tshirts-search	2
getting-to-know-cool-tshirts	9
interview-with-cool-tshirts-founder	7
paid-search	52
retargetting-ad	113
retargetting-campaign	54
ten-crazy-cool-tshirts-facts	9
weekly-newsletter	115
Database Schema	
page_visits	
5692 rows	
page_name	TEXT
timestamp	TEXT
user_id	INTEGER
utm_campaign	TEXT
utm_source	TEXT