Learn SQL by Calculating Customer Lifetime Value

Part 1: Introduction

The purpose of this project is to learn fundamental SQL queries using a typical data analysis scenario. As I am a product manager at an e-commerce (or SaaS) startup, I calculate the Customer Lifetime Value (CLV) – how much revenue they bring to the company over their lifetime. We can use this critical metric to allow us to make stronger business decisions and customize our products effectively for each customer.

This project uses SQLite3 and the two tables of interest ("users" and "payments") are read from bootstrap.sql taken from http://bit.ly/sql-starter. We conclude that a direct campaign and customers with user_id of 8 and 3 generate the least revenue. Thus the company needs to focus on improving services offered to or produced through these elements.

Part 2: Project

Loading windows command prompt and running sqlite3.exe

> cd C:/sqlite

> sqlite3.exe

Loading tables

> .read bootstrap.sql

> .tables payments users

> SELECT * FROM users;   > SELECT * FROM payments;

```
id          campaign    signed_up_on
----------  ----------  ------------
1           facebook    2014-10-01
2           twitter     2014-10-02
3           direct      2014-10-02
4           facebook    2014-10-03
5           organic     2014-10-03
6           organic     2014-10-03
7           organic     2014-10-04
8           direct      2014-10-05
9           twitter     2014-10-05
10          organic     2014-10-05
```

```
id          amount      paid_on     user_id
----------  ----------  ----------  ----------
1           40          2014-10-02  1
2           30          2014-10-03  1
3           100         2014-10-04  1
4           30          2014-10-05  1
5           30          2014-10-06  1
6           40          2014-10-07  1
7           50          2014-10-08  1
8           50          2014-10-09  1
9           40          2014-10-10  1
10          100         2014-10-11  1
11          30          2014-10-12  1
12          30          2014-10-13  1
13          40          2014-10-14  1
14          30          2014-10-15  1
15          50          2014-10-16  1
16          30          2014-10-17  1
17          40          2014-10-18  1
18          100         2014-10-19  1
19          40          2014-10-20  1
20          40          2014-10-21  1
21          30          2014-10-22  1
```

Ordering by campaign in alphabetical order

> SELECT * FROM users ORDER BY campaign;

```
id          campaign    signed_up_on
----------  ----------  ------------
3           direct      2014-10-02
8           direct      2014-10-05
1           facebook    2014-10-01
4           facebook    2014-10-03
5           organic     2014-10-03
6           organic     2014-10-03
7           organic     2014-10-04
10          organic     2014-10-05
2           twitter     2014-10-02
9           twitter     2014-10-05
```

Filtering with WHERE

> SELECT * FROM users WHERE campaign = 'organic';

```
id          campaign    signed_up_on
----------  ----------  ------------
5           organic     2014-10-03
6           organic     2014-10-03
7           organic     2014-10-04
10          organic     2014-10-05
```

> SELECT * FROM users WHERE campaign IN ('facebook', 'twitter');

```
id          campaign    signed_up_on
----------  ----------  ------------
1           facebook    2014-10-01
2           twitter     2014-10-02
4           facebook    2014-10-03
```

> SELECT * FROM users WHERE campaign NOT IN ('facebook', 'twitter');

```
id          campaign    signed_up_on
----------  ----------  ------------
3           direct      2014-10-02
5           organic     2014-10-03
6           organic     2014-10-03
7           organic     2014-10-04
8           direct      2014-10-05
10          organic     2014-10-05
```

> SELECT * FROM users WHERE campaign IN ('facebook', 'twitter') AND signed_up_on = '2014-10-01';

```
id          campaign    signed_up_on
----------  ----------  ------------
1           facebook    2014-10-01
```

> SELECT * FROM users WHERE campaign = 'organic' OR signed_up_on < '2014-10-04';

```
id          campaign    signed_up_on
----------  ----------  ------------
1           facebook    2014-10-01
2           twitter     2014-10-02
3           direct      2014-10-02
4           facebook    2014-10-03
5           organic     2014-10-03
6           organic     2014-10-03
7           organic     2014-10-04
10          organic     2014-10-05
```

Filtering and sorting

> SELECT * FROM users WHERE campaign in ('facebook', 'twitter') ORDER BY campaign;

```
id          campaign    signed_up_on
----------  ----------  ------------
1           facebook    2014-10-01
4           facebook    2014-10-03
2           twitter     2014-10-02
9           twitter     2014-10-05
```

Group by: SQL's PivotTable

> SELECT user_id, SUM(amount) FROM payments GROUP BY user_id;

```
user_id     SUM(amount)
----------  ----------
1           1410
2           1580
3           35
4           140
5           135
6           1240
7           105
8           90
9           125
10          105
```

JOIN: connecting multiple sources of information

> SELECT user_id, SUM(amount) as clv, campaign FROM users JOIN payments ON users.id = user_id GROUP by user_id ORDER BY clv DESC;

Figure 1: CLV per user

The Other CLV: Campaign Lifetime Value

> SELECT campaign, SUM(amount) as campaign_value FROM users JOIN payments ON users.id = user_id GROUP by campaign ORDER BY campaign_value DESC;

Figure 2: Value according to campaign method

Part 3: Conclusion

I conclude from…

- figure 1 that customers with user_id of 2 and 1 generate the most revenue over their lifetime. Whilst customers with user_id of 8 and 3 generate the least revenue. Hence the company would do best to examine these customers' preferences and produce more tailored products and services more favourable to them.

- figure 2 that twitter generates the most revenue, whilst direct the least. Hence the company would do best to examine why a direct campaign is not working successfully, and strive to rework it so as to be more profitable.