codecademy

CodeFlix Churn Analysis

Learn SQL from Scratch

Paul Thorley

25 Jan 2019

Key Questions

How many months has the company been operating?	Slide 3
Which months do you have enough information to calculate a churn rate?	Slide 3
What segments of users exist?	Slide 3
What is the overall churn trend since the company started?	Slide 4
Compare the churn rates between user segments.	Slide 4
Which segment of users should the company focus on expanding?	Slide 4
SQL Screenshots of excuted code	Slide 6 & 7

1.1 Company trading history

Based on the data provided for analysis Codeflix has been operating for four months.

- *The first subscriber came on board on 1 December 2016*
- *The latest subscriber came on board on 30 March 2017*
- *Cancellations are recorded in the data through Jan to Mar 2017*

No cancellations occured in December due to the minimum 31 day subscription and the earliest subscription being 1 Dec 2016.

Churn rates can therefore only be calculated for Jan, Feb, Mar 2017 with the data provided.

Two customer segments are identified in the data, labelled 87 and 30.

1.2 Churn Discussion

In both segments churn is increasing over the period analysed.

- *From a low base in January the customer segment 30 (approx 7%) dropped marginally in February but then rose above prior levels in March (11%).*
- *Customer segment 87 exhibits concistently higher churn than segment 30 and has risen steadily in each month (starting at 25% in January nearly doubling to 48% in March).*

Clearly segment 30 contains a stronger client base and potentially can be expanded if additional investment is applied. Market research to identify possible reasons for the difference between segements would also add value to future marketing activity.

A table of the key calculation output is provided below and the code used for the analysis is provided in an embedded file.

codeSQL.txt

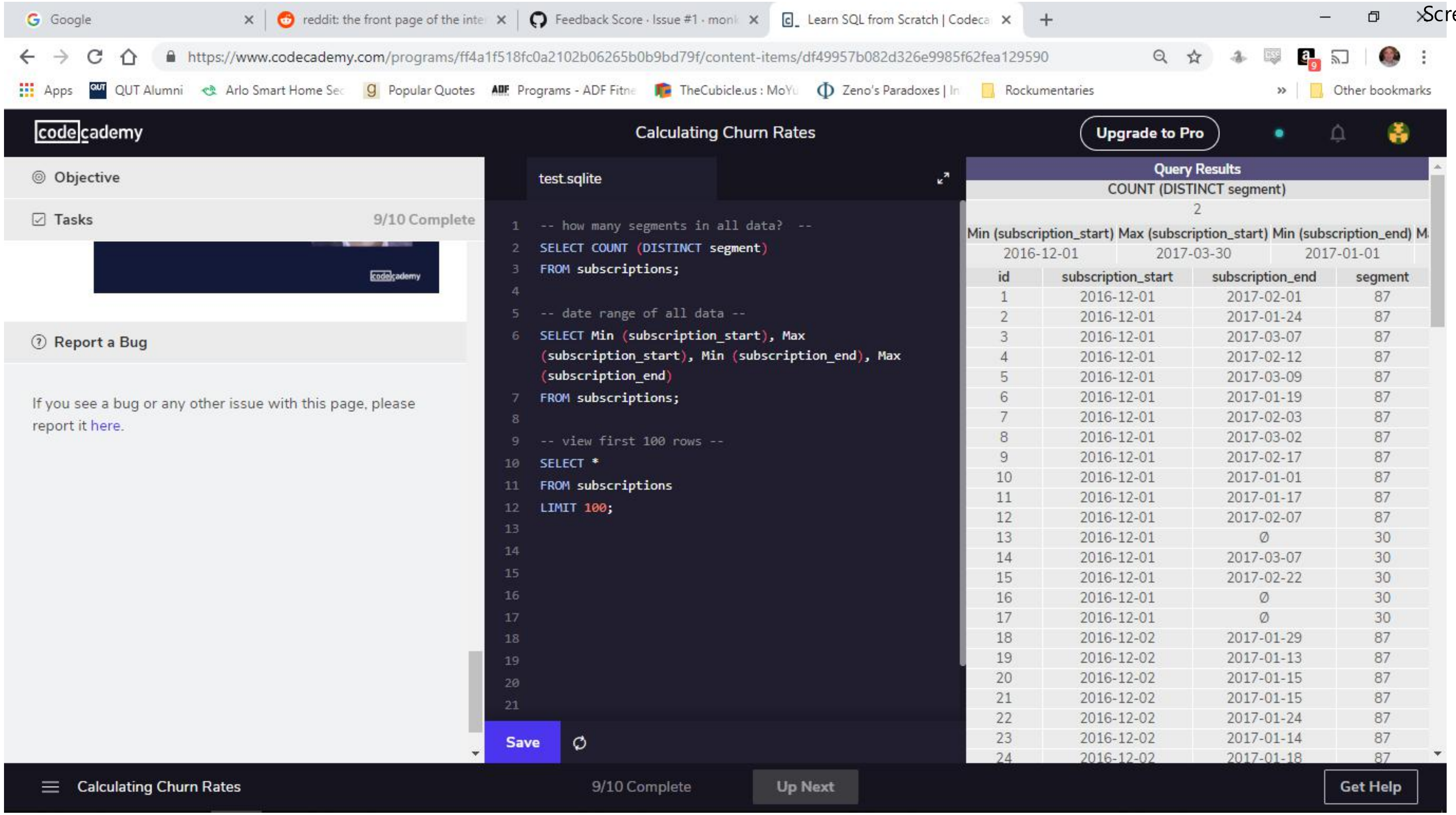
1.2 Churn Calculation Output

The churn_87 and churn_30 variables refer to the calculated chrun rates in each month for the two different customer segments

MONTH	CHURN_87	CHURN_30
Jan 2017	0.25180	0.07560
Feb 2017	0.32035	0.07338
Mar 2017	0.48588	0.11732

Appendix 1 - SQL screenshots - prelim calculations

Other Calcs Screenshot.jpg



If you see a bug or any other issue with this page, please report it here.

```
-- how many segments in all data?  --
SELECT COUNT (DISTINCT segment)
FROM subscriptions;

-- date range of all data --
SELECT Min (subscription_start), Max
(subscription_start), Min (subscription_end), Max
(subscription_end)
FROM subscriptions;

-- view first 100 rows --
SELECT *
FROM subscriptions
LIMIT 100;
```

Query Results

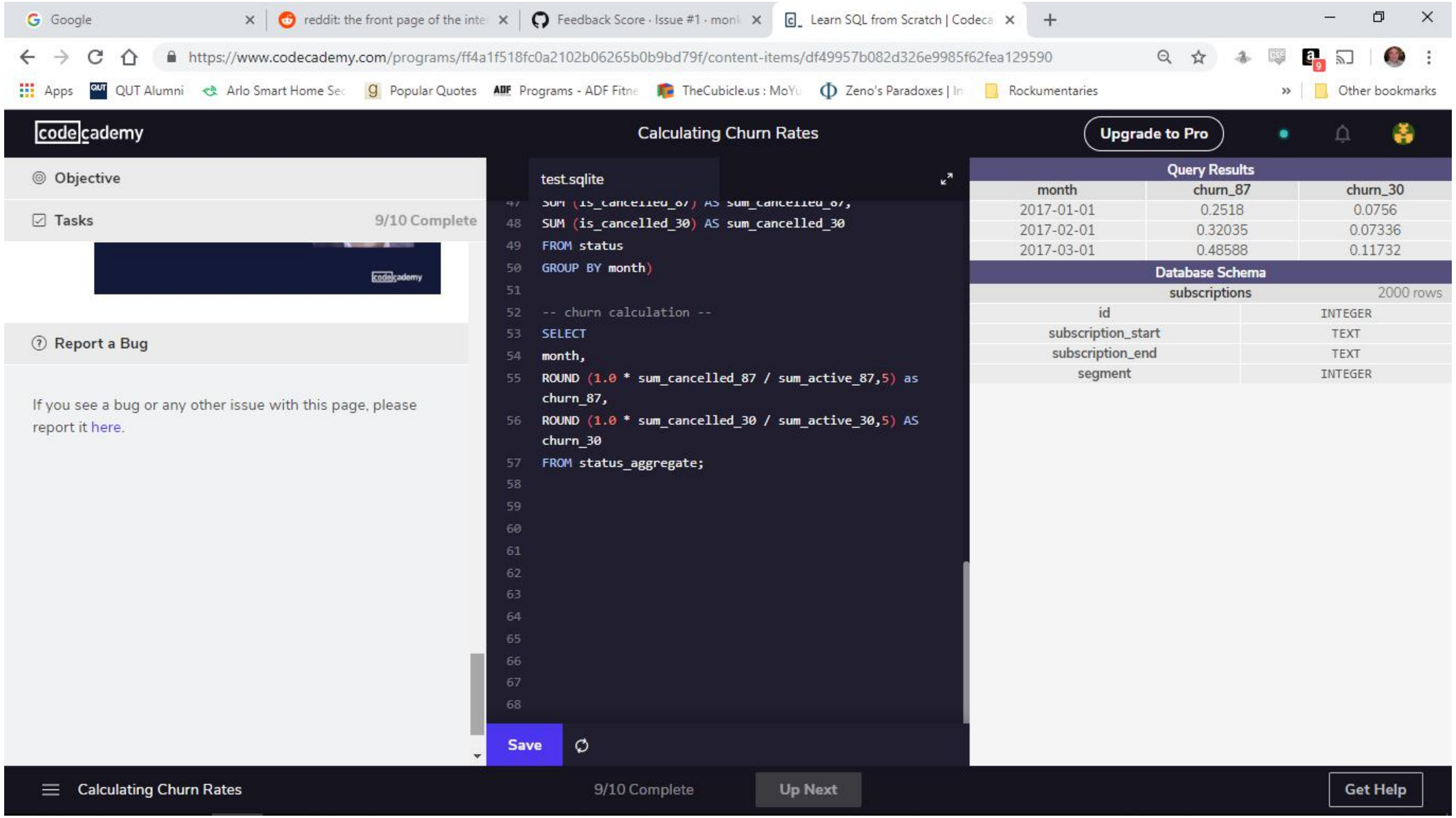
COUNT (DISTINCT segment)
2

Min (subscription_start)	Max (subscription_start)	Min (subscription_end)	M
2016-12-01	2017-03-30	2017-01-01	

id	subscription_start	subscription_end	segment
1	2016-12-01	2017-02-01	87
2	2016-12-01	2017-01-24	87
3	2016-12-01	2017-03-07	87
4	2016-12-01	2017-02-12	87
5	2016-12-01	2017-03-09	87
6	2016-12-01	2017-01-19	87
7	2016-12-01	2017-02-03	87
8	2016-12-01	2017-03-02	87
9	2016-12-01	2017-02-17	87
10	2016-12-01	2017-01-01	87
11	2016-12-01	2017-01-17	87
12	2016-12-01	2017-02-07	87
13	2016-12-01	Ø	30
14	2016-12-01	2017-03-07	30
15	2016-12-01	2017-02-22	30
16	2016-12-01	Ø	30
17	2016-12-01	Ø	30
18	2016-12-02	2017-01-29	87
19	2016-12-02	2017-01-13	87
20	2016-12-02	2017-01-15	87
21	2016-12-02	2017-01-15	87
22	2016-12-02	2017-01-24	87
23	2016-12-02	2017-01-14	87
24	2016-12-02	2017-01-18	87

Appendix 1 - SQL screenshots - churn calculation

Churn Calc Screenshot.jpg



If you see a bug or any other issue with this page, please report it here.

```sql
SUM (is_cancelled_87) AS sum_cancelled_87,
SUM (is_cancelled_30) AS sum_cancelled_30
FROM status
GROUP BY month)

-- churn calculation --
SELECT
month,
ROUND (1.0 * sum_cancelled_87 / sum_active_87,5) as churn_87,
ROUND (1.0 * sum_cancelled_30 / sum_active_30,5) AS churn_30
FROM status_aggregate;
```

Query Results

month	churn_87	churn_30
2017-01-01	0.2518	0.0756
2017-02-01	0.32035	0.07336
2017-03-01	0.48588	0.11732

Database Schema

subscriptions 2000 rows

id	INTEGER
subscription_start	TEXT
subscription_end	TEXT
segment	INTEGER