

SESUG Paper 162-2018  
**Learnin' about Churnin'**

**(with Examples from SAS® Enterprise Guide®)**

Victoria Garcia, Florida Fish and Wildlife Conservation Commission

## ABSTRACT

What is churn? According to sources it's; "The percentage of customers who cut ties with your company". Churn rate tell a story. If you listen, you'll learn a lot more about your customers and company than you think. In this presentation, we will be looking at our agencies churn rate; how we got it, what we learned from it, and what we are doing to improve it. For this presentation, we will be working data from the Florida Fish and Wildlife Conservation Commission.

## INTRODUCTION

Here we are calculating churn rate for recreational fishing and hunting license sales. There are licenses that range from three days to a lifetime and they fall into certain categories; freshwater fishing, saltwater fishing, and hunting are the three big ones. We're going to calculate churn based on our agencies fiscal year July 1<sup>st</sup> to June 30<sup>th</sup> and we're going to calculate churn based on these three major categories. "How many customers aren't returning the next year and which category?" that's our question.

## CALCULATION

There are several ways we can calculate churn. We're going to start with a simple calculation.

$$\frac{100 - 90}{100} = 10\%$$

If we had 100 customers who were active in the saltwater category in 2014 and 90 of those customers were active in the saltwater category in 2015, we have a 10% churn rate.

## THE CALCULATION IN ACTION

```
( IF ( ( Sum _ByGroup_ ( Saltwater_14 ) - Sum _ByGroup_ ( Saltwater_14_15 ) ) < 0 ) / Sum _ByGroup_ ( Saltwater_14 ) )  
RETURN 0  
ELSE ( Sum _ByGroup_ ( Saltwater_14 ) - Sum _ByGroup_ ( Saltwater_14_15 ) ) )
```

Using the code above.

To calculate Saltwater churn for 2014, we take the difference between customers active in the saltwater category and the same customers who were active in that category 2015.

We are looking at the same customers each time and not counting new customers of 2015(they will be counted in 2015 churn calculation). If the difference is zero or less, the measure returns zero. Zero or

less means that all customers from one year were active the next (zero difference), meaning zero percent churn.

If there is a difference between Saltwater\_14 and Saltwater\_14\_15, then we divide by Saltwater\_14. The difference between the two represent how many customers did not renew in that category from the previous year.

## THE PROCESS

How do we get from order data to churn? This is the challenge we faced. We had table of orders that contained the customer identification number, demographic information, etc... Below is generally what we had. For the category columns, (1 for yes, 0 for no) indicates that the license fell into that category.

Customer_ID	Demographic_Info	Order_Number	Item_ID	Item_Ordered	Start_Date	End_Date	Salwater	Freshwater	Hunting
123	General info	789	111	Saltwater License	11/11/2015	11/11/2016	1	0	0
456	General info	456	222	Hunting License	12/12/2016	12/12/2017	0	0	1
789	General info	123	333	Freshwater License	4/5/2014	4/5/2015	0	1	0

## THE TALE OF TWO TABLES:

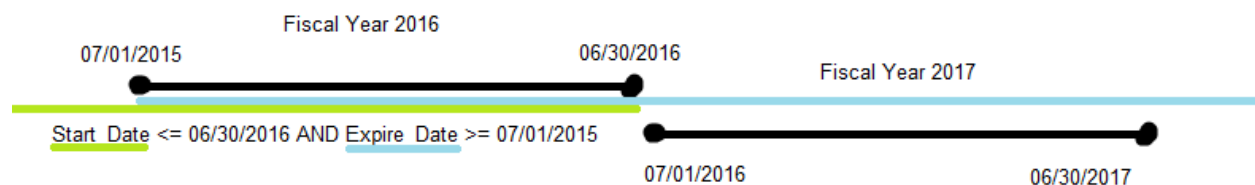
Using the code below for each fiscal year, we created columns to identify when the license was valid.

```

Enter an expression:
case
when (t1.Start_Date<='30Jun2016'd & t1.Expire_Date>='1Jul2015'd)
then '1'
else '0'
end

```

Visualization of the code:



This produced our Base table, from this table we can see what year the license was active in and which category the license fell into.

Customer_ID	Demographic_Info	Item_Ordered	Start_Date	End_Date	2015	2016	Salwater	Freshwater	Hunting
123	General info	Saltwater License	7/1/2015	7/1/2016	0	1	1	0	0
123	General info	Hunting License	7/1/2015	7/1/2016	0	1	0	0	1
456	General info	Hunting License	7/1/2014	7/1/2015	1	0	0	0	1
456	General info	Saltwater License	7/1/2015	7/1/2016	0	0	1	0	0
789	General info	Freshwater License	7/1/2014	7/1/2015	1	0	0	1	0

From this base table we create the Customer Table.

Customer_ID	Demographic_Info
123	General info
456	General info
789	General info

## PHASE 1:

*/\*This program creates 14 tables. One for each fiscal year. Filtered for whatever license is active during that time period. Example: work.Q2012 shows all licenses active for fiscal year 2012\*/*

In phase one, we filtered our customers by which year their license was active. Some customers appeared in multiple tables because they held a license that was valid in one or more year.

Customer_ID	2012	2013	2014	Saltwater	Freshwater	Hunting
123	1	1	1	1	0	0
456	1	1	0	0	0	1

```
PROC SQL;
CREATE TABLE WORK.Q2012 AS
SELECT t1.customerId,
       t1.'2012'n,
       t1.'2013'n,
       t1.'2014'n,
       t1.'2015'n,
       t1.Saltwater,
       t1.Freshwater,
       t1.Hunting,
FROM WORK.Base_Customer_Table t1
WHERE t1.'2012'n = '1'
```

## PHASE 2:

After creating tables for each fiscal year, we used the MAX function on all the columns. This grouped the customer by their identification number and category. So, if the customer would only appear once per fiscal year no matter how many items they bought.

```
PROC SQL;
CREATE TABLE WORK.Max2012 AS
SELECT DISTINCT t1.customerId,

       (MAX(t1.'2012'n)) AS M2012,

       (MAX(t1.'2013'n)) AS M2013,

       (MAX(t1.'2014'n)) AS M2014,

       (MAX(t1.Saltwater)) FORMAT=BEST12. AS Saltwater,

       (MAX(t1.Freshwater)) FORMAT=BEST12. AS Freshwater,

       (MAX(t1.Hunting)) FORMAT=BEST12. AS Hunting,
FROM WORK.Q2012 t1
GROUP BY t1.customerId;
```

## PHASE 3:

We now have tables for each fiscal year. In phase three, we add the customer demographics column from the Customer Table then using filters on the fiscal year columns we create comparison tables. Because we are calculating churn based on fiscal year, we are comparing customers year to year.

C2012 Table:

The data in this table shows customers that were active in 2012 and indicates what category they were active in 2012. The category columns represent what category the licenses in 2012 fell into.

The customer ID does not repeat. If the customer held an active hunting license and saltwater license in 2012 then that customer would have a 1 in the Saltwater column and 1 in the Hunting column. If the customer held two licenses in the same category then the Category column would still indicate 1, as it does not count the number of licenses just if there was an active license in that category

```
PROC SQL;
CREATE TABLE WORK.C2012 AS
SELECT t1.customerId,
       t2.Demographics,
       t1.M2012,
       t1.M2013,
       t1.M2014,
       t1.Saltwater,
       t1.Freshwater,
       t1.Hunting,
FROM WORK.MAX2012 t1
INNER JOIN OLPLIB.RLIS_CUST_DEMO_CHURN t2 ON (t1.customerId =
t2.customerId)
WHERE t1.2012 = '1';
```

C2012\_Compare\_2013 Table:

Like the 2012 table, the 2012\_Compare\_2013 shows only customers who were active in 2012 but instead of showing the category of the license bought in 2012, it represents the category the 2013 purchase.

```
PROC SQL;
CREATE TABLE WORK.C2012_Compare_2013 AS
SELECT t1.customerId,
       t2.Demographics,
       t1.M2012 AS '2012'n,
       t1.M2013 AS '2013'n,
       t1.M2014 AS '2014'n,
       t1.M2015 AS '2015'n,
       t1.M2016 AS '2016'n,
       t1.Saltwater,
       t1.Freshwater,
       t1.Hunting
FROM WORK.MAX2013 t1
INNER JOIN OLPLIB.RLIS_CUST_DEMO_CHURN t2 ON
(t1.customerId = t2.customerId)
WHERE t1.M2012 = '1' AND t1.M2013 = '1';
```

After we create these tables, we combine them and label them accordingly.

```
PROC SQL;
CREATE TABLE OLPLIB.RLIS_Churn2012 AS
SELECT t1.customerId,
       t1.Demographics,
       t1.'2012'n LABEL="2012_12" AS '2012_12'n,
```

```

t1.Saltwater AS Saltwater_12,
t1.Freshwater AS Freshwater_12,
t1.Hunting AS Hunting_12,
t1.'2013'n AS '2012_13'n,
t2.Saltwater AS Saltwater_12_13,
t2.Freshwater AS Freshwater_12_13,
t2.Hunting AS Hunting_12_13
FROM WORK.RLIS_VG_CHURN_C_2012_S_2012 t1
FULL JOIN WORK.RLIS_VG_CHURN_C_2012_S_2013 t2 ON
(t1.customerId = t2.customerId)
ORDER BY t1.customerId;
QUIT;

```

#### PHASE 4:

Now we have a lot of tables! This is the last phase of this project. We append all the tables that we created. After that we use the MAX function again to create our final table.

```

PROC SQL;
CREATE TABLE LASR_TGT.RLIS_VG_CHURN_TABLES AS
SELECT DISTINCT
    t1.customerId,
    t1.Demographics,

    (MAX(t1.'2012_12'n)) AS '2012_12'n,

    (MAX(t1.Saltwater_12)) FORMAT=BEST12. AS Saltwater_12,

    (MAX(t1.Freshwater_12)) FORMAT=BEST12. AS Freshwater_12,

    (MAX(t1.Hunting_12)) FORMAT=BEST12. AS Hunting_12,

    (MAX(t1.'2012_13'n)) AS '2012_13'n,

    (MAX(t1.Saltwater_12_13)) FORMAT=BEST12. AS
Saltwater_12_13,

    (MAX(t1.Freshwater_12_13)) FORMAT=BEST12. AS
Freshwater_12_13,

    (MAX(t1.Hunting_12_13)) FORMAT=BEST12. AS Hunting_12_13,

    (MAX(t1.'2013_13'n)) AS '2013_13'n,

    (MAX(t1.Saltwater_13)) FORMAT=BEST12. AS Saltwater_13,

    (MAX(t1.Freshwater_13)) FORMAT=BEST12. AS Freshwater_13,

    (MAX(t1.Hunting_13)) FORMAT=BEST12. AS Hunting_13,

```

This is how our final table looks:

Customer_ID	2012	Saltwater_12	Freshwater_12	Hunting_12	2012_2013	Saltwater_12_13	Freshwater_12_13	Hunting_12_13
123	1	1	0	0	0	0	0	0
456	1	0	0	1	1	0	0	1
789	0	1	0	1	0	0	0	0
1111	1	1	1	1	1	1	0	1
22222	0	1	1	0	1	1	1	0

From this table we can calculate our churn rates for each year and category.

```

( IF ( ( Sum [ByGroup_] ( Saltwater_14 ) - Sum [ByGroup_] ( Saltwater_14_15 ) ) < 0 ) / Sum [ByGroup_] ( Saltwater_14 ) )
RETURN 0
ELSE ( Sum [ByGroup_] ( Saltwater_14 ) - Sum [ByGroup_] ( Saltwater_14_15 ) )

```

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Victoria Garcia  
Florida Fish and Wildlife Conservation Commission  
Victoria.Garcia@myfwc.com  
LinkedIn.com/i-love-data