

Just Give Me the File: The Power of PROC SQL with SAS/ACCESS® to Pull Data from Various Files and Calculate Actionable Results

John Standefer, Lighthouse Consulting, Inc.

Abstract

In today's business environment, companies are highly sensitive to efficiency ratings, such as return on investment (ROI). One of the mainstays of the SAS® System is its ability to read data from nearly all sources. This presentation will discuss one example of how to use the power of PROC SQL and SAS/ACCESS® to pull data from three various sources and calculate the ROI for a sales promotion. The source files used for this presentation are the following:

- 1 - MS Access database tables from Sales
- 2 - Lotus spreadsheets from Accounting
- 3 - SAS data sets from Marketing

Introduction

Knowledge is power, and knowledge consists of your competency over information. In the world of the SAS programmer, information is data, and data consists of a multitude of file types and formats. Precious time is spent massaging data into a readable format before you can perform SAS steps to accomplish your objective. In a perfect world everyone would use SAS and all data would be in SAS files. In the real world, however, it is the "Information Age" with more file types and formats than we could cover in this presentation.

The scope of this presentation will cover the use of PROC SQL via the Pass-Through facility and SAS/ACCESS software to pull data from two different file types (database tables from MS Access and spreadsheet files from Lotus) and

share the data with SAS data sets to provide actionable results in a timely manner.

Background

The most common approach to get various vendor software files into a SAS file is to create a text or ASCII file from the source file. Then read the flat file in using an INPUT statement. This process is tedious and can waste critical time. DDE (Dynamic Data Exchange) works for exchanging data with Windows applications and is similar to the text file approach in that you have to use an INPUT statement. SAS/ACCESS is superior to those mentioned because flexibility and speed the interface engine gives you.

SAS/ACCESS

SAS/ACCESS software provides a means to share data between the SAS System and database management systems (DBMSs), open database connectivity (ODBC) files and spreadsheet files. The power behind SAS/ACCESS software is in the interface view engine. The interface view engine is utilized by the Pass-Through facility and is discussed at greater length in the first example.

There are two PROC ACCESS procedures (discussed in the second example). The first builds an access descriptor which tells the SAS System how and where to connect and exchange data with the external data source. An access descriptor does not contain any data; it merely describes data. The second PROC ACCESS procedure builds a view descriptor which contains data in a SAS data view.

ODBC

ODBC is the standard most Windows applications follow that allows data sharing. To gain the benefits of exchanging data between SAS and other databases via ODBC, you must install the appropriate ODBC driver. For example, this presentation pulls data from MS Access database tables. Before SAS/ACCESS could query the database the *MS Access ODBC driver* had to be installed on my system. In Windows 95 under the Windows Control Panel there is an icon called 32bit ODBC. You go into that set up menu and install the appropriate driver.

Accounting Terms

Return on Investment (ROI) is income divided by investment:

$$\text{ROI} = \frac{\text{Income}}{\text{Investment}}$$

ROI is a popular performance ratio because it combines all significant parts of profitability:

- Revenue
- Expenses
- Investment

Income = Revenue - Expenses

Investment = Assets - Liabilities

For this example, investment is the cost of investing in a sales promotion. Revenue is calculated as Total Sales net of expenses; that is, the unit sales price given in example 2 is less the unit cost of goods sold.

Pass-Through Query

Title "Sales Promotion Data from MS Access Tables";

```
PROC SQL;
  connect to odbc(
    "DBQ=C:\MWSUG\Sales Promotion.mdb;" ||
    "Driver={Microsoft Access Driver (*.mdb)};"
  );
  Create View mwsug.promo as
  Select
    EVNTNUM,
    PRODCODE,
    EVNTDT,
    CITY,
    STATE,
    CUSTNUM

  from connection to odbc(select
    Promo."Evt_ven_nbr",
    Promo."Prodcode",
    Promo."Evt Date",
    Promo."City",
    Promo."State",
    Custmrs."Lmp_nbr"
  from "Promo" Promo,
       "Promo Received" Custmrs
  where Promo.Evt_ven_nbr=Custmrs.Evt_ven_nbr
  order by Custmrs.Lmp_nbr)
  as Promo (EVNTNUM, PRODCODE, EVNTDT, CITY, STATE, CUSTNUM )
;
  disconnect from odbc;
quit;
```

Output - Pass-Through Query

Sales Promotion Data from MS Access Tables

Evt_ven_nbr	Prodcode	Evt Date	City	State	Lmp_nbr
0035	AB00	02NOV1997	Denver	CO	0000013090
0035	AB00	02NOV1997	Denver	CO	0000024410
0035	AB00	02NOV1997	Denver	CO	0000024548
0035	AB00	02NOV1997	Denver	CO	0000024842
0035	AB00	02NOV1997	Denver	CO	0000028634
0483	AB00	19DEC1997	Boise	ID	0000028716
0035	AB00	02NOV1997	Denver	CO	0000030057
0035	AB00	02NOV1997	Denver	CO	0000030225
0299	AB00	25OCT1997	Billings	MT	0051290121
0299	AB00	25OCT1997	Billings	MT	0051985103

PROC ACCESS -> ACCESS DESCRIPTOR

```
libname mwsug V612 'c:\mwsug';

PROC ACCESS DBMS=WK4;
  CREATE mwsug.invest.ACCESS;          /** create access descriptor  **/
```

```
PATH='C:\MWSUG\Invest.WK4';
LIST ALL;
RUN;
```

PROC ACCESS -> VIEW DESCRIPTOR

```
PROC ACCESS DBMS=WK4 ACCDESC=mwsug.invest;
CREATE mwsug.invest.VIEW;          /** create view descriptor **/
SELECT LDGRACCT INVESTMENT PROMODAT;
LIST ALL;
RUN;
```

DESCRIPTOR AND VIEW DESCRIPTOR in the Same PROC ACCESS Step

```
PROC ACCESS DBMS=WK4;
CREATE mwsug.invest.ACCESS;          /** create access descriptor **/
PATH='C:\MWSUG\INVEST.WK4';          /** points to directory and file **/
SCANTYPE = Y;
WORKSHEET = "Promotns";              /** Lotus worksheet name **/
GETNAMES = Y;                        /** assigns SAS var names from columns **/
TYPE 1 = C;
MIXED Y;
LIST ALL;

CREATE mwsug.invest.VIEW;            /** create view descriptor **/
SELECT LDGRACCT INVESTMENT PROMODAT;
SUBSET WHERE SUBSTR(LDGRACCT,7,4) IN ('0299','0483');
LIST ALL;
RUN;

PROC DATASETS LIBRARY=mwsug MEMTYPE=VIEW; /** contents of lotus view **/
CONTENTS DATA=invest;
RUN;
```

OUTPUT - DESCRIPTOR AND VIEW DESCRIPTOR in the Same PROC ACCESS Step

Accounting Unit Price Data from Lotus Spreadsheets

DATASETS PROCEDURE

```
Data Set Name: MWSUG.INVEST      Observations:      24
Member Type:  VIEW              Variables:         3
```

-----Alphabetic List of Variables and Attributes-----

#	Variable	Type	Len	Pos	Format	Informat	Label
2	INVESTME	Num	8	10	BEST10.	BEST10.	INVESTMENT
1	LDGRACCT	Char	10	0	\$10.	\$10.	LDGRACCT
3	PROMODAT	Num	8	18	MMDDYY8.	MMDDYY11.	PROMODAT

INCOME QUERY

```
libname mwsug 'c:\mwsug';
libname sasnkt 'c:\sas\mydata';
title 'INCOME - Unit Sales Increase Times Unit Price';

Proc SQL;
```

```

select distinct b.prodcode,
               sum(slsml4+slsml5+slsml6+slsml7+slsml8) as pstsqty,
               sum(slsml8+slsml9+slsml10+slsml11+slsml12) as presqty,
               calculated pstsqty - calculated presqty as qtyinc,
               calculated qtyinc * unitpric as INCOME format dollar8.
  from sasmtk.sales a, mwsug.promo b, mwsug.revenue c
 where a.custnum=b.custnum and b.Prodcode=c.prodcode
        group by c.prodcode
;
quit;

```

OUTPUT - INCOME QUERY

INCOME - Unit Sales Increase Times Unit Price				
Prodcode	PSTSQTY	PRESQTY	QTYINC	INCOME
AB00	968	238	730	\$36,500

ROI QUERY

Title 'ROI for Sales Promotion';

```

proc sql;
create view temp as
select * ,
       substr(LDGRACCT,7,4) as evntnum
from mwsug.invest
;

create view temp2 as
select distinct a.evntnum ,
               investme
               label = 'Incremental Investment'
from mwsug.promo a , temp b
where a.EVNTNUM = b.evntnum
;

select distinct income ,
               SUM(investme) as tinvest
               label = 'Total Investment',
               (income / calculated tinvest) as ROI format percent7.
  from temp2 , mwsug.income
;

```

OUTPUT - ROI QUERY

ROI for Sales Promotion		
INCOME	Total Investment	ROI
\$36,500	\$150,000	24%

Conclusion

SAS/ACCESS using either PROC SQL Pass-Through facility or PROC ACCESS procedure to build an access descriptor and a view descriptor gives you two options to increase data exchange using the SAS System. The ability to quickly gather and efficiently process data to provide actionable results is a valuable commodity.

Utilizing SAS/ACCESS in your SAS code will increase your efficiency when working with DBMSs, ODBC files and spreadsheet files.

Author Information

John Standefer
Lighthouse Consulting
6303 Freemont
Hanover Park, IL 60103

email: JohnStan@DTNhome.com
phone: (847) 602-5006

References

- SAS/ACCESS Software for Relational Databases, Reference, Version 6, First Edition
- SAS Guide to the SQL Procedure, Usage and Reference, Version 6, First Edition
- Judy Loren, and Gregory S. Barnes Nelson, "SQL Step by Step: An Advanced Tutorial for Business Users", Proceedings of the Twenty-Third Annual SAS Users Group International Conference (1998), pp. 175-183
- Charles T. Horngren, and George Foster, Cost Accounting: A Managerial Emphasis, Seventh Edition (1991)