Know Thy Data – Techniques For Data Exploration

MSUG, 12 June 2024

Charu Shankar

SAS Education

Copyright © SAS Institute Inc. All rights reserved.

Charu Shankar, SAS® Institute

With a background in computer systems management. SAS Instructor Charu Shankar engages with logic, visuals, and analogies to spark critical thinking since 2007.

Charu curates and delivers unique content on SAS, SQL, Viya, etc. to support users in the adoption of SAS software.

When not coding, Charu teaches yoga and loves to explore Canadian trails with her husky Miko.

Copyright © SAS Institute Inc. All rights reserved.

Agenda

Introduction to Know Thy Data

The Population of Interest

Quick and Easy Ways to Know Your Data

Powerful PROC SQL Dictionary Tables

Analytical SAS Procedures to know your data

Handy Links

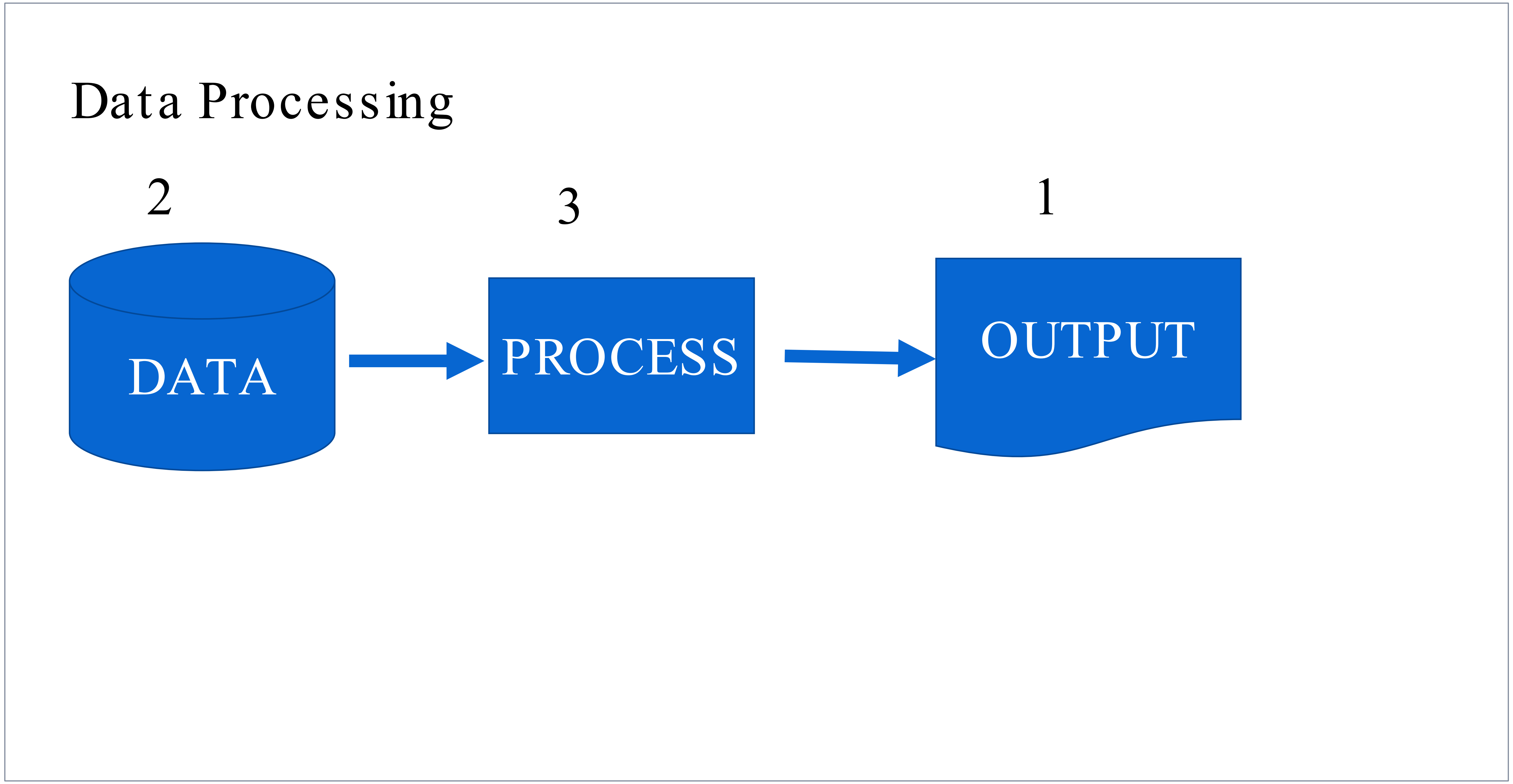
Copyright © SAS Institute Inc. All rights reserved.

1 Introduction

The 80-20 Rule

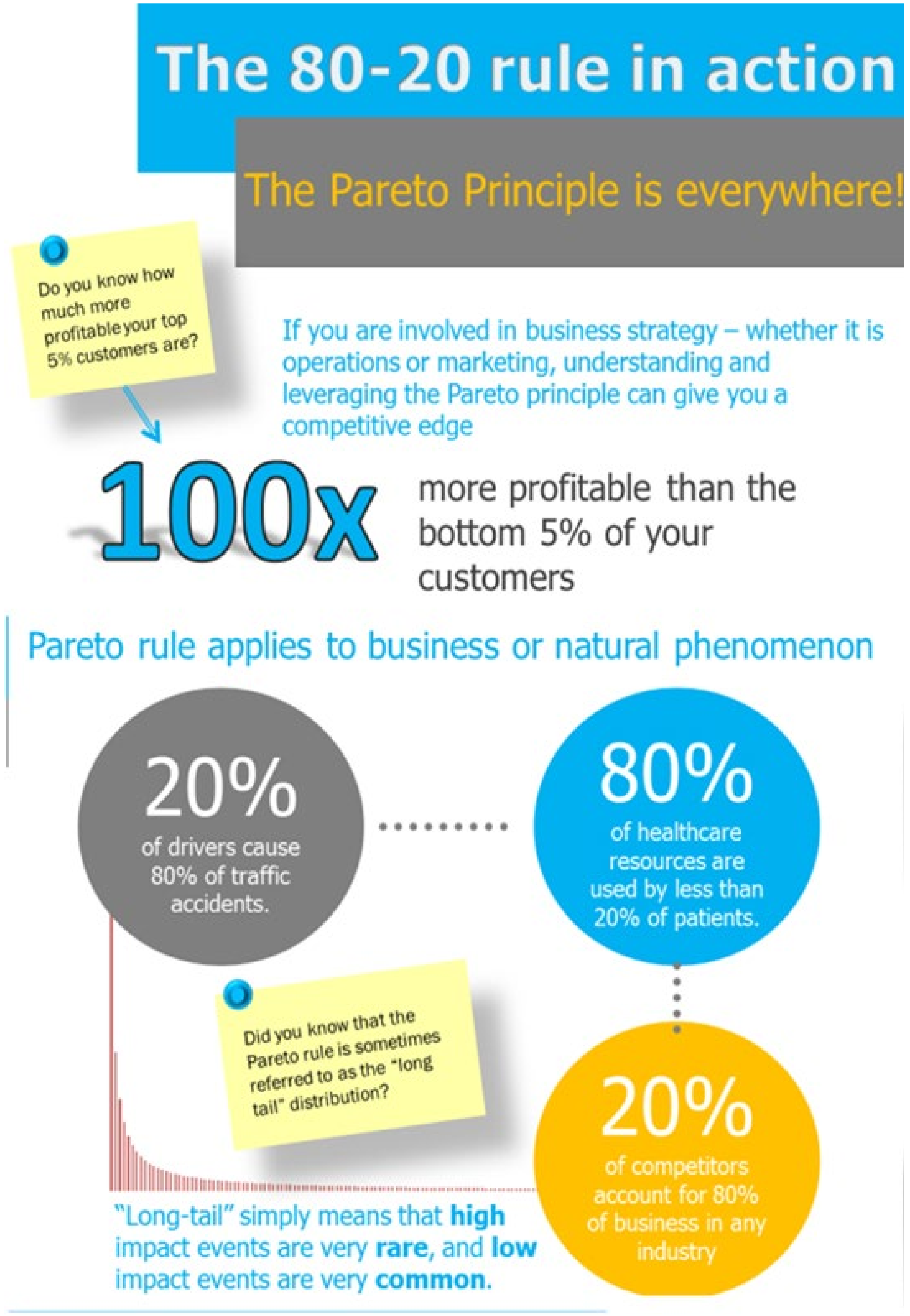
Copyright © SAS Institute Inc. All rights reserved.

Programmer Rule #1



Copyright © SAS Institute Inc. All rights reserved.

The 80-20 Rule



The 80-20 rule in action

The Pareto Principle is everywhere!

Do you know how much more profitable your top 5% customers are?

If you are involved in business strategy – whether it is operations or marketing, understanding and leveraging the Pareto principle can give you a competitive edge

100x more profitable than the bottom 5% of your customers

Pareto rule applies to business or natural phenomenon

20% of drivers cause 80% of traffic accidents.

80% of healthcare resources are used by less than 20% of patients.

20% of competitors account for 80% of business in any industry

Did you know that the Pareto rule is sometimes referred to as the "long tail" distribution?

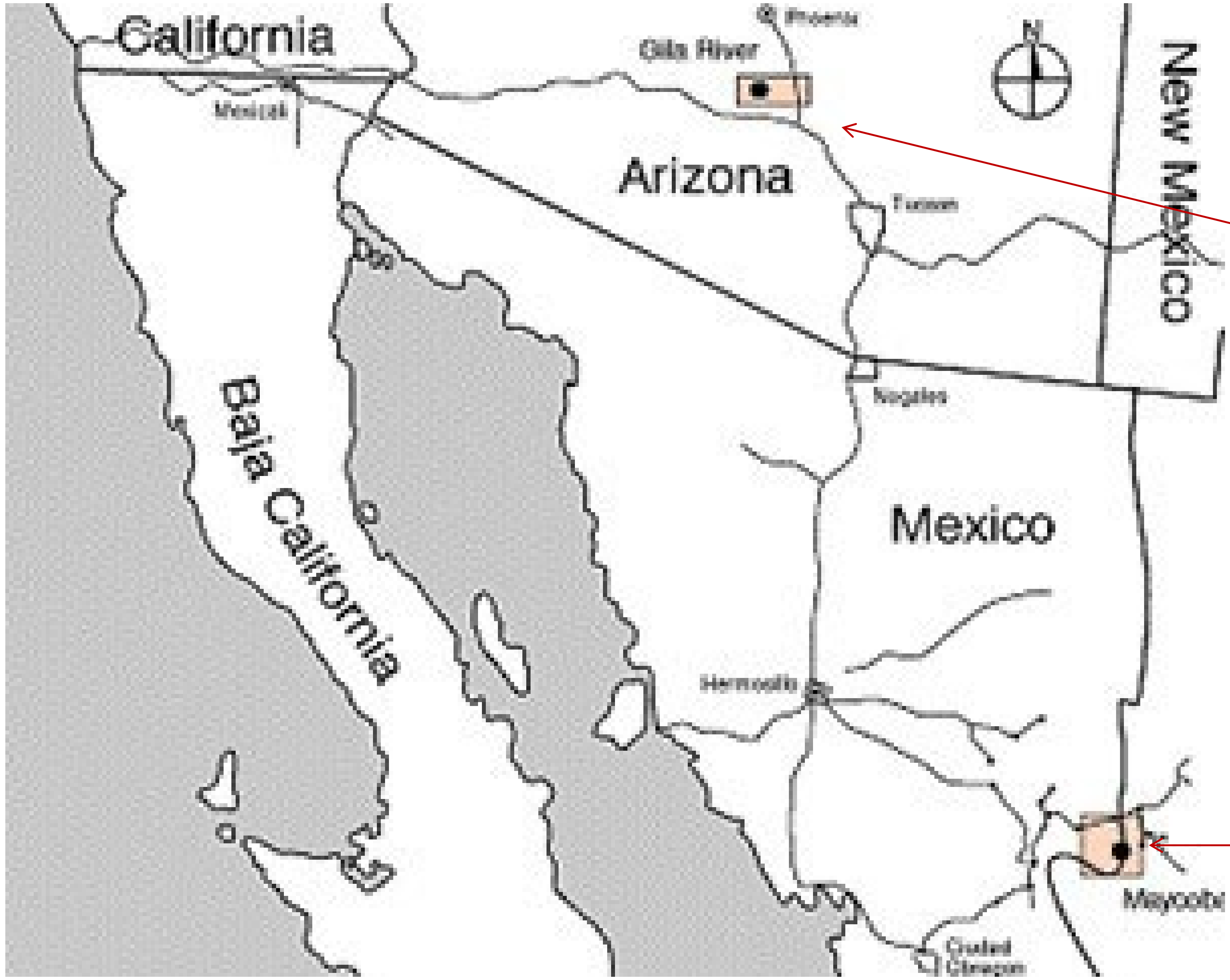
"Long-tail" simply means that **high** impact events are very **rare**, and **low** impact events are very **common**.

Copyright © SAS Institute Inc. All rights reserved.

2 The Population Of Interest

Copyright © SAS Institute Inc. All rights reserved.

The Population Of Interest



Arizona Pima

Mexican Pima
Sierra Madre
Mountains

Copyright © SAS Institute Inc. All rights reserved.

The Variables

Variable Name	Description
glucose	glucose
dbp	diastolic blood pressure
triceps	tricep skin fold thickness
insulin	2-hour serum insulin
pedigree	diabetes pedigree
Diabetes	1 = tested positive for diabetes 0 = tested negative for diabetes
ID	identification number
Pregnancies	number of times pregnant
BMI	body mass index
age	age
ID	identification number

Copyright © SAS Institute Inc. All rights reserved.

3 Metadata

Quick and Easy Ways to Know your Data

Copyright © SAS Institute Inc. All rights reserved.

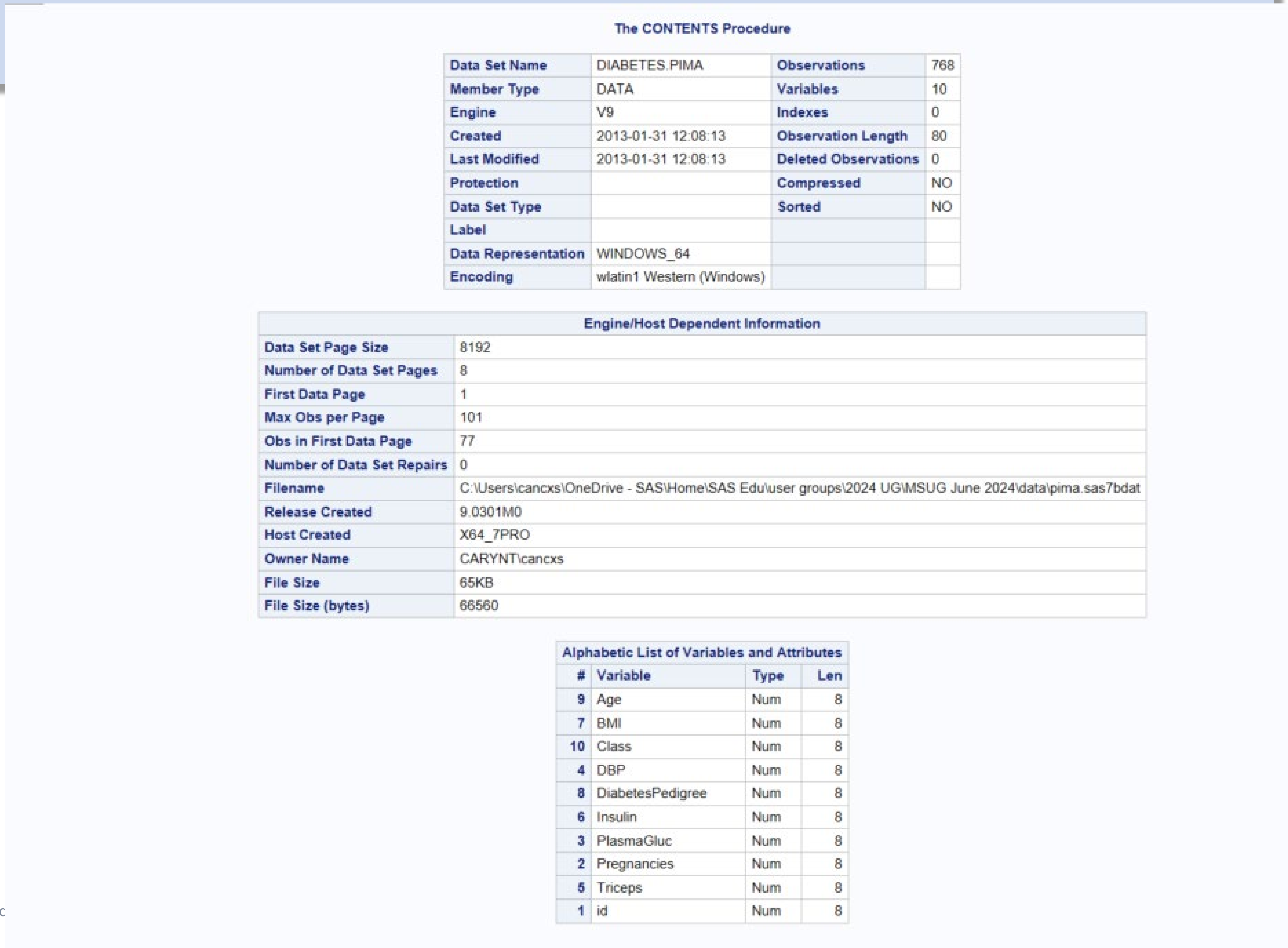
Metadata - The SAS® Explorer

- There is a lot of information available to you with a simple click of the mouse …OK, sometimes a double-click …
- Information about a SAS file.
- Information about the individual fields that make up the file.

Copyright © SAS Institute Inc. All rights reserved.

Metadata - PROC CONTENTSLook At a Single Table

```
proc contents data=diabetes.pima out=test;
run;
```



The CONTENTS Procedure

Data Set Name	DIABETES.PIMA	Observations	768
Member Type	DATA	Variables	10
Engine	V9	Indexes	0
Created	2013-01-31 12:08:13	Observation Length	80
Last Modified	2013-01-31 12:08:13	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_64		
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information	
Data Set Page Size	8192
Number of Data Set Pages	8
First Data Page	1
Max Obs per Page	101
Obs in First Data Page	77
Number of Data Set Repairs	0
Filename	C:\Users\cancxs\OneDrive - SAS\Home\SAS Edu\user groups\2024 UG\MSUG June 2024\data\pima.sas7bdat
Release Created	9.0301M0
Host Created	X64_7PRO
Owner Name	CARYNT\cancxs
File Size	65KB
File Size (bytes)	66560

Alphabetic List of Variables and Attributes

#	Variable	Type	Len
9	Age	Num	8
7	BMI	Num	8
10	Class	Num	8
4	DBP	Num	8
8	DiabetesPedigree	Num	8
6	Insulin	Num	8
3	PlasmaGluc	Num	8
2	Pregnancies	Num	8
5	Triceps	Num	8
1	id	Num	8

Copyright © SAS Institute Inc. All rights reservec

4 Powerful PROC SQL Dictionary Tables

Copyright © SAS Institute Inc. All rights reserved.

Examine Dictionary Tables

```
proc sql ;
   describe table dictionary.dictionaries;
   select distinct memname, memlabel
       from dictionary.dictionaries;
quit;
```

DICTIONARY Table	SASHELP View	Description
COLUMNS	VCOLUMN	Contains information about columns in all known tables.
MEMBERS	VMEMBER	Contains general information about SAS library members
TABLES	VTABLE	Contains detailed information about tables

Copyright © SAS Institute Inc. All rights reserved.

Querying Dictionary Information

Display information about the columns in **DIABETES.PIMA**.

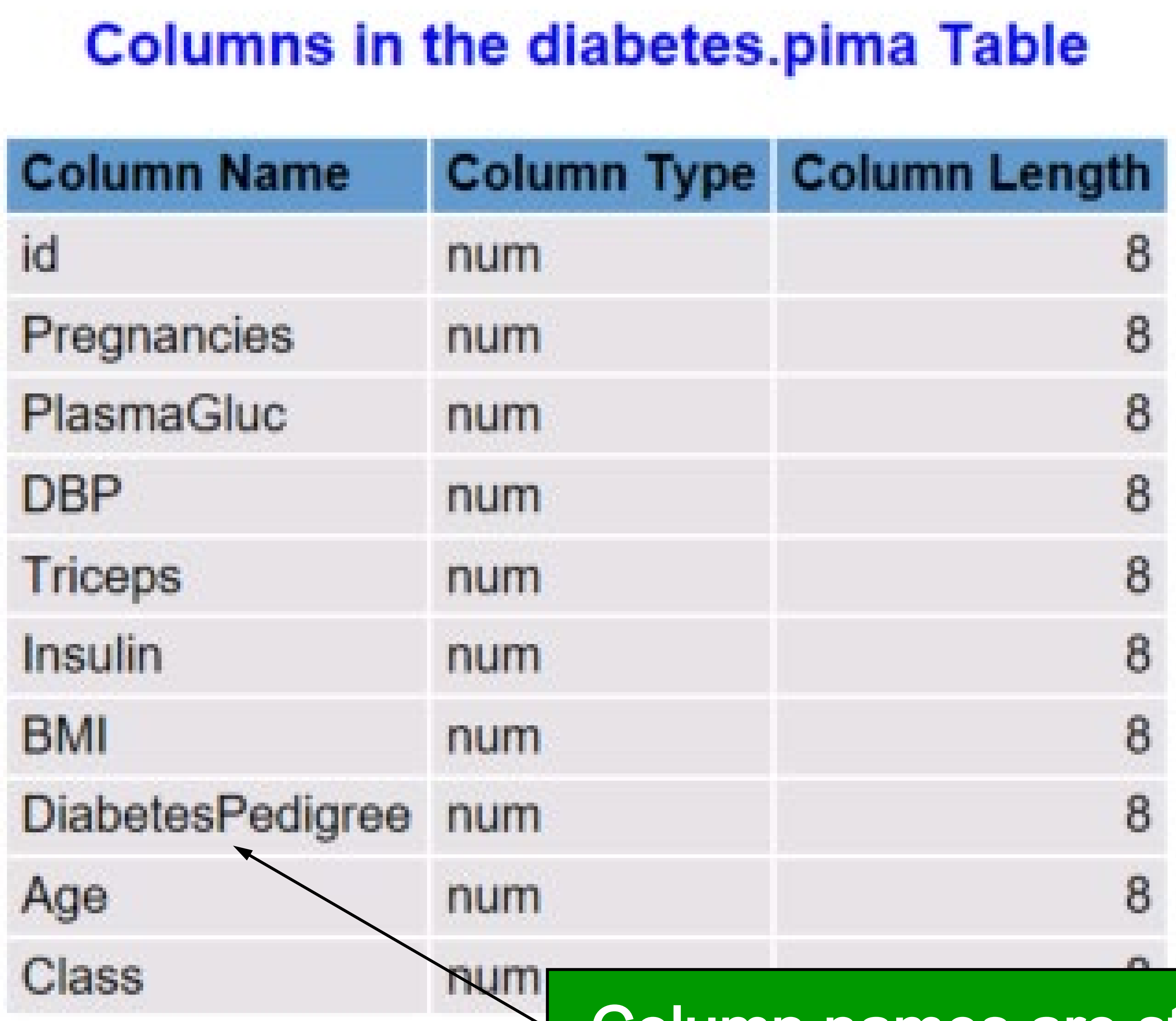
```
title 'Columns in the Diabetes.Pima Table';
proc sql;
select Name,Type,Length
   from dictionary.columns
   where libname=‘DIABETES’
         and memname=‘PIMA’ ;
quit;
```

Table names (*memnames*) are also stored in uppercase in DICTIONARY tables.

Copyright © SAS Institute Inc. All rights reserved.

Viewing the Output

PROC SQL Output



Columns in the diabetes.pima Table

Column Name	Column Type	Column Length
id	num	8
Pregnancies	num	8
PlasmaGluc	num	8
DBP	num	8
Triceps	num	8
Insulin	num	8
BMI	num	8
DiabetesPedigree	num	8
Age	num	8
Class	num	8

Column names are stored in mixed case.

Copyright © SAS Institute Inc. All rights reserved.

Using Dictionary Information

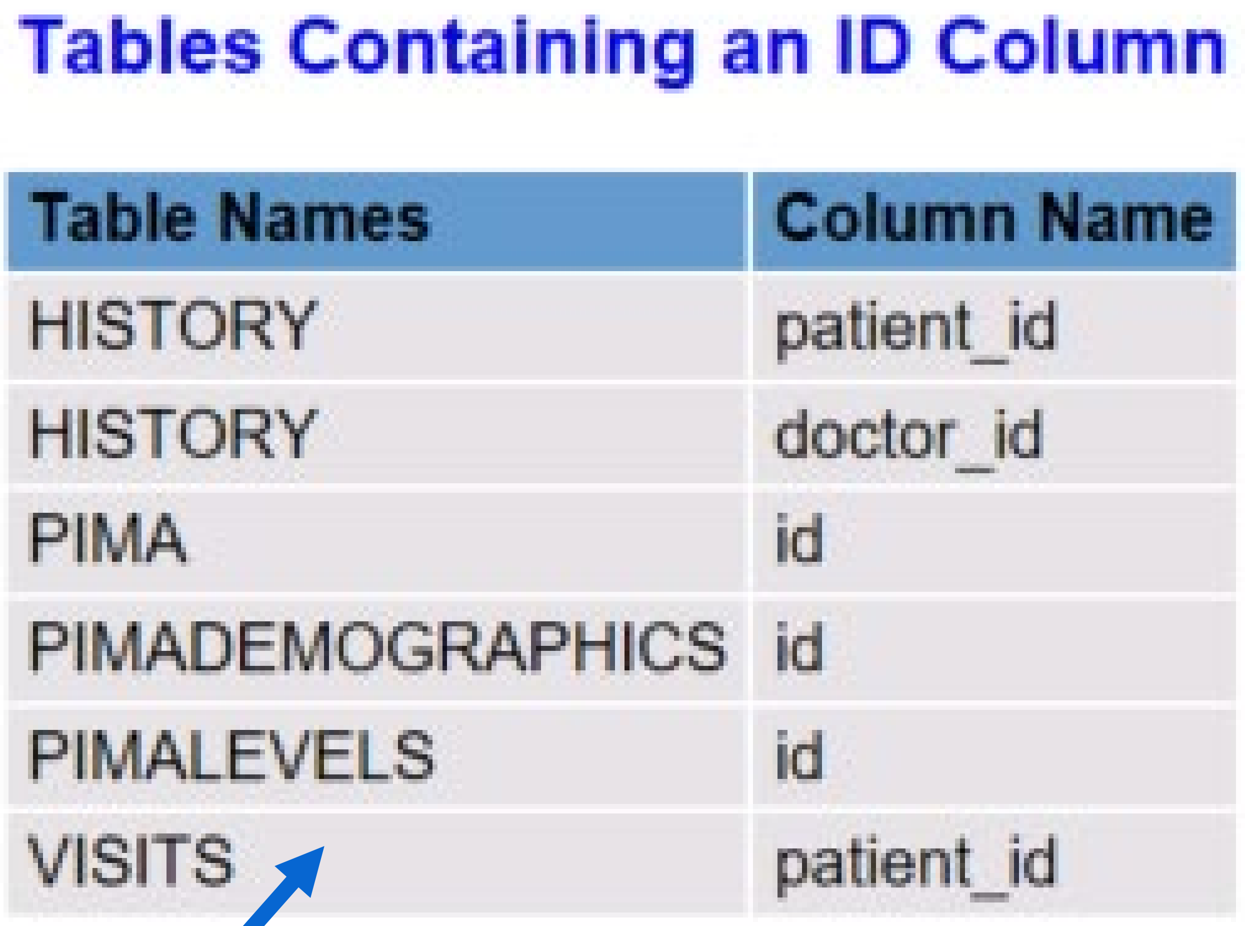
Which tables contain an ID column?

```
title 'Tables Containing an ID Column';
proc sql;
select memname 'Table Names', name
   from dictionary.columns
   where libname=‘DIABETES' and
         upcase(name) contains 'ID';
quit;
```

Because different tables might use different cases for same-named columns, you can use the UPCASE function for comparisons. However, this significantly degrades the performance of the query.

Copyright

Viewing the Output



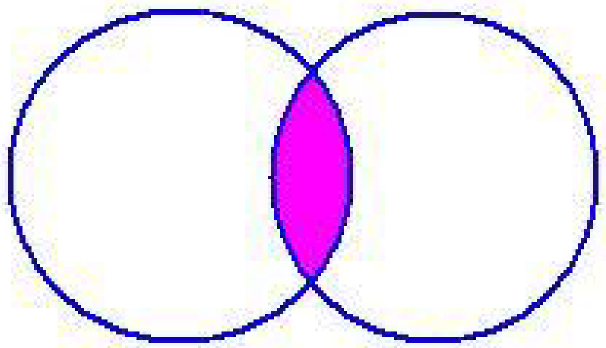
Tables Containing an ID Column

Table Names	Column Name
HISTORY	patient_id
HISTORY	doctor_id
PIMA	id
PIMADEMOGRAPHICS	id
PIMALEVELS	id
VISITS	patient_id

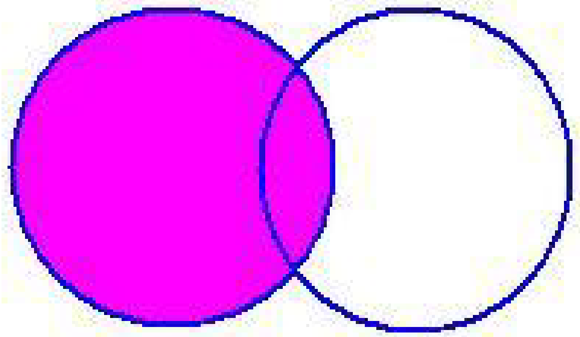
All **ID** column names are stored in uniform uppercase, so the UPCASE function is not needed the next time that a query such as this is executed.

Copyright © SAS Institute Inc. All rights reserved.

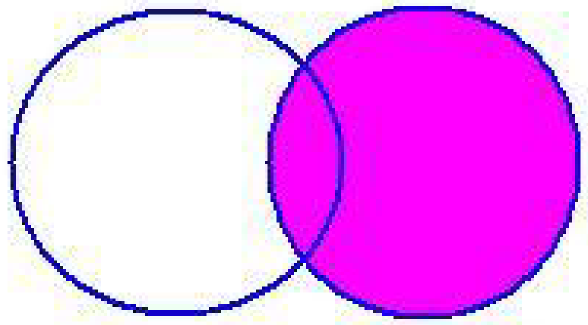
Investigate Common Columns



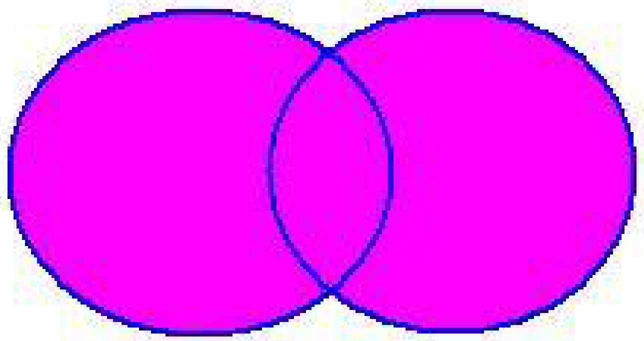
Inner join (result similar to Intersect)



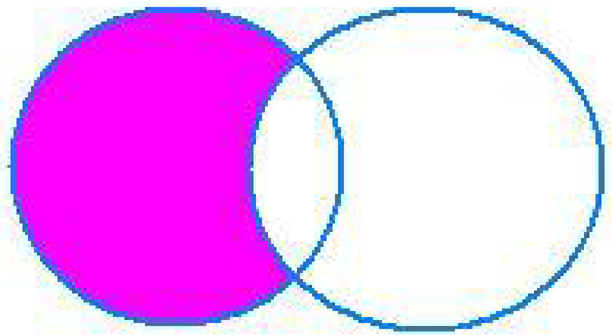
Left outer join



Right outer join



Full outer join



Minus

Copyright © SAS Institute Inc. All rights reserved.

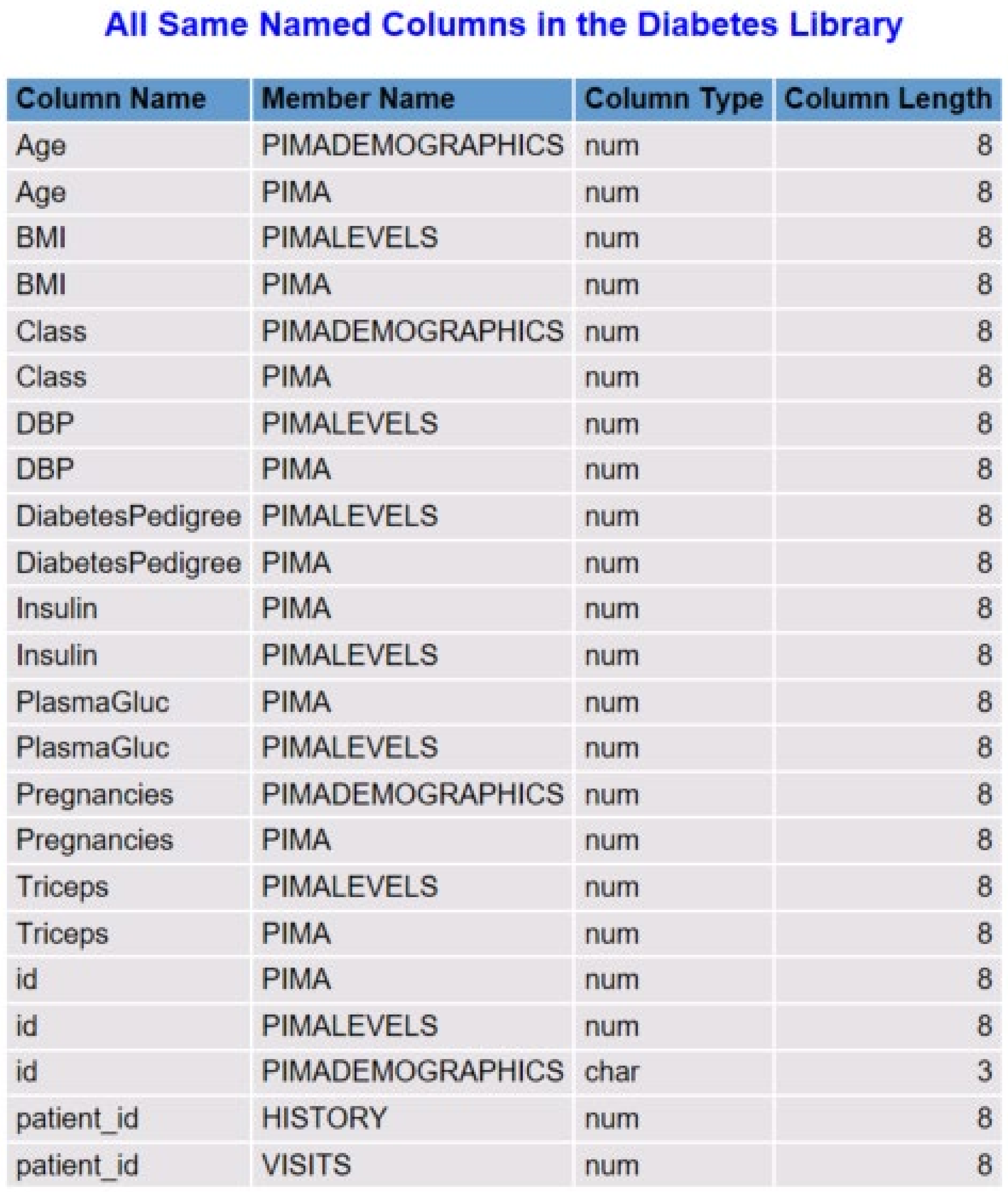
Finding Common Column Names Dynamically

- All of the previous techniques to explore DICTIONARY tables work when you know the names of columns.
- What happens if you do not know your data, and you want SAS to retrieve all same-named columns in a library?
- Use the following code

```
title 'All Same Named Columns in the Diabetes Library';
proc sql;
   select name, memname, type, length
       from dictionary.columns
          where Libname ='DIABETES'
             group by name
                having count(name) > 1
                   order by name;
quit;
```

Copyright © SAS Institute Inc. All rights reserved.

Viewing the Output



All Same Named Columns in the Diabetes Library

Column Name	Member Name	Column Type	Column Length
Age	PIMADEMOGRAPHICS	num	8
Age	PIMA	num	8
BMI	PIMALEVELS	num	8
BMI	PIMA	num	8
Class	PIMADEMOGRAPHICS	num	8
Class	PIMA	num	8
DBP	PIMALEVELS	num	8
DBP	PIMA	num	8
DiabetesPedigree	PIMALEVELS	num	8
DiabetesPedigree	PIMA	num	8
Insulin	PIMA	num	8
Insulin	PIMALEVELS	num	8
PlasmaGluc	PIMA	num	8
PlasmaGluc	PIMALEVELS	num	8
Pregnancies	PIMADEMOGRAPHICS	num	8
Pregnancies	PIMA	num	8
Triceps	PIMALEVELS	num	8
Triceps	PIMA	num	8
id	PIMA	num	8
id	PIMALEVELS	num	8
id	PIMADEMOGRAPHICS	char	3
patient_id	HISTORY	num	8
patient_id	VISITS	num	8

Joins are easier because the structure of each table does not have to be examined before determining common columns. Let SAS bring common columns dynamically by looking up DICTIONARY tables.

Using DICTIONARY Tables in Other SAS Code

- SAS provides views based on the DICTIONARY tables in the **SASHELP**library.
- Most of the **SASHELP**library DICTIONARY view names are similar to DICTIONARY table names, but they are shortened to eight characters or less. They begin with the letter **v** and do not end in **s**. For example:

dictionary.tables = sashelp.vtable

- The following code executes successfully:

```
options fullstimer;
title 'Proc Print Output - Tables Containing an ID Column';
proc print data=sashelp.vcolumn noobs;
    var libname memname name type length;
        where libname='DIABETES' and upcase(name) contains 'ID';
run;
```

Copyright © SAS Institute Inc. All rights reserved.

An Efficiency Question: PROC SQL or PRINT?

```
options fullstimer;
proc sql;
select libname, memname, name, type, length
    from dictionary.columns
        where libname='DIABETES' and  upcase(name) contains 'ID';
quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time                0.73 seconds
      user cpu time            0.42 seconds
      system cpu time          0.29 seconds
      memory                   5584.18k
      OS Memory                24672.00k
```

Copyright © SAS Institute Inc. All rights reserved.

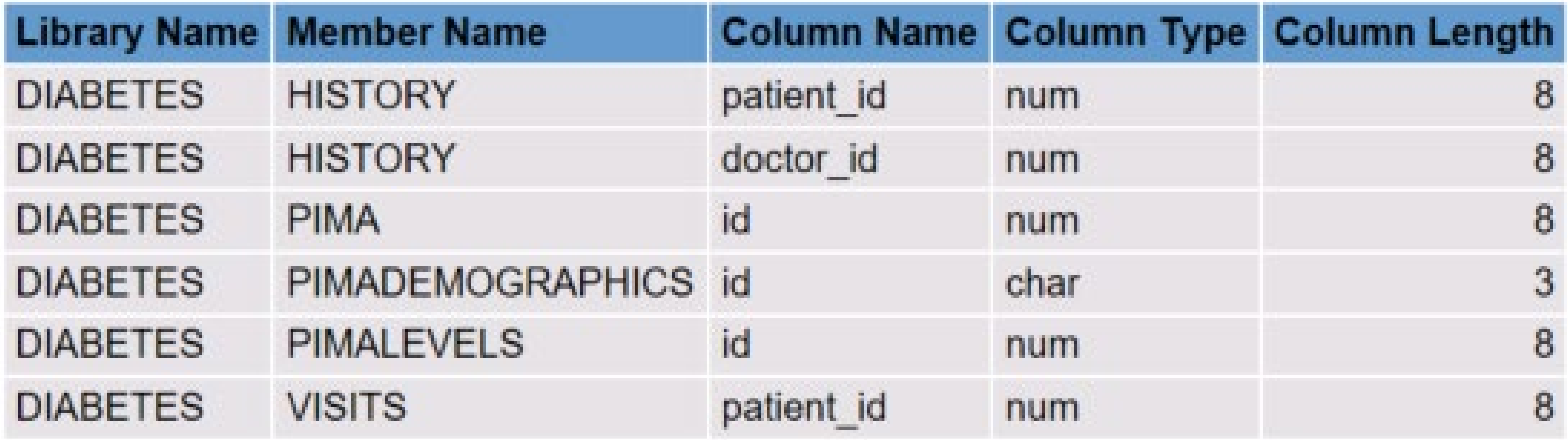
An Efficiency Question: PROC SQL or PRINT?

Can I use PROC PRINT instead?

```
options fullstimer;

title 'Proc Print Output - Tables Containing an ID Column';
proc print data=sashelp.vcolumn noobs;
    var libname memname name type length;
    where libname='DIABETES' and upcase(name) contains 'ID';
run;
NOTE: There were 6 observations read from the data set
SASHELP.VCOLUMN. WHERE (Libname=‘DIABETES’) and UPCASE(name) contains
‘ID’;
NOTE: PROCEDURE PRINT used (Total process time):
      real time             2.19 seconds
      user cpu time         0.92 seconds
      system cpu time       1.18 seconds
      memory                6738.81k
      OS Memory             25440.00k
```

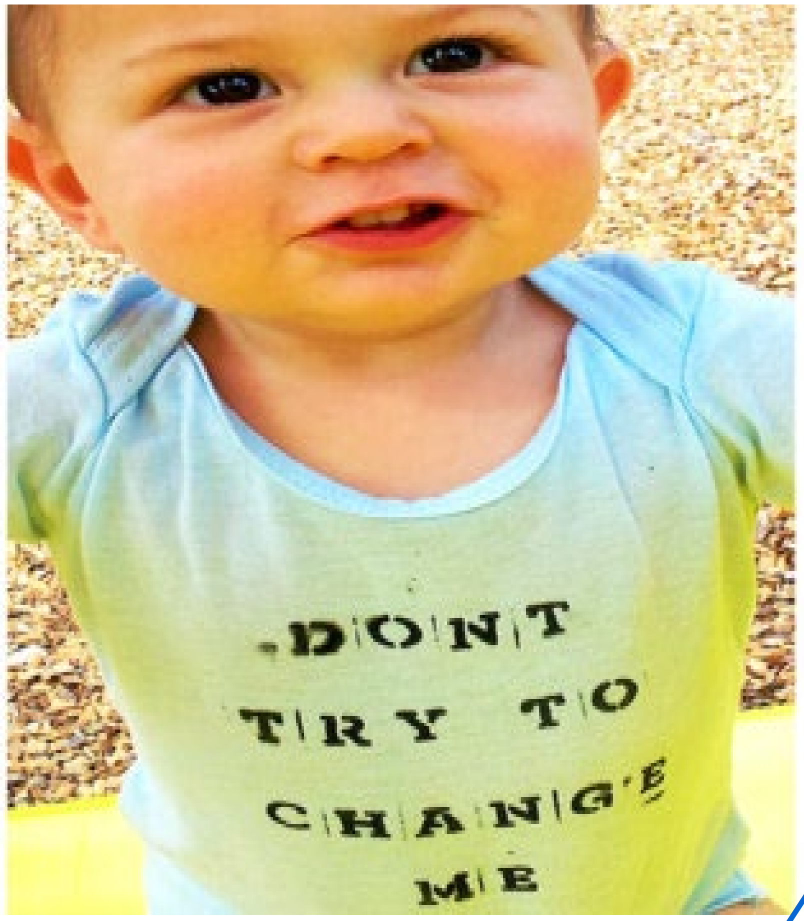
Investigate Common Columns - Output



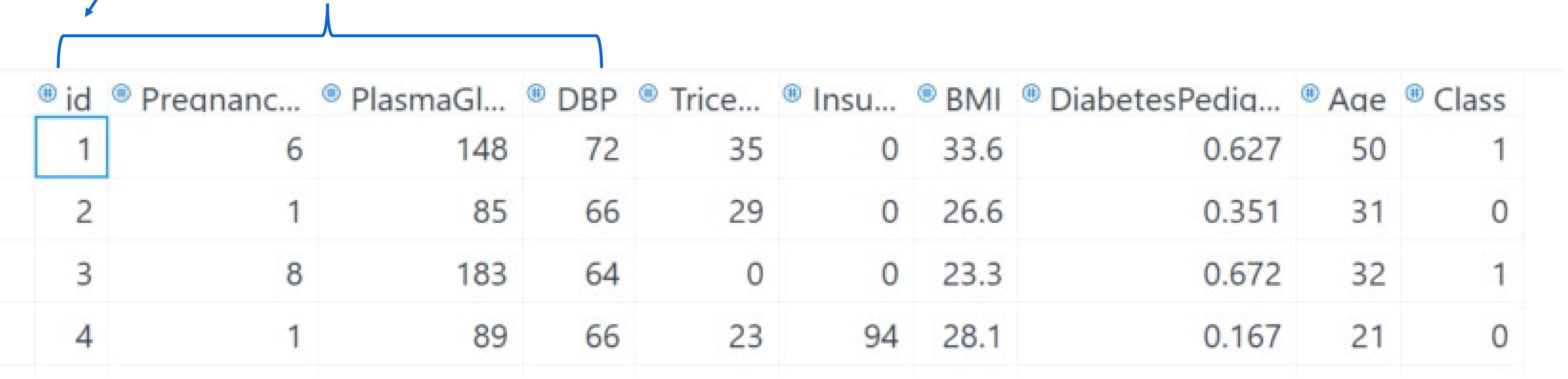
Library Name	Member Name	Column Name	Column Type	Column Length
DIABETES	HISTORY	patient_id	num	8
DIABETES	HISTORY	doctor_id	num	8
DIABETES	PIMA	id	num	8
DIABETES	PIMADEMOGRAPHICS	id	char	3
DIABETES	PIMALEVELS	id	num	8
DIABETES	VISITS	patient_id	num	8

Copyright © SAS Institute Inc. All rights reserved.

Reorder Dataset Variables



```
proc print data=diabetes.pima;
var dbp--id;
ERROR: Starting variable after ending variable in data
set.
212  run;
```



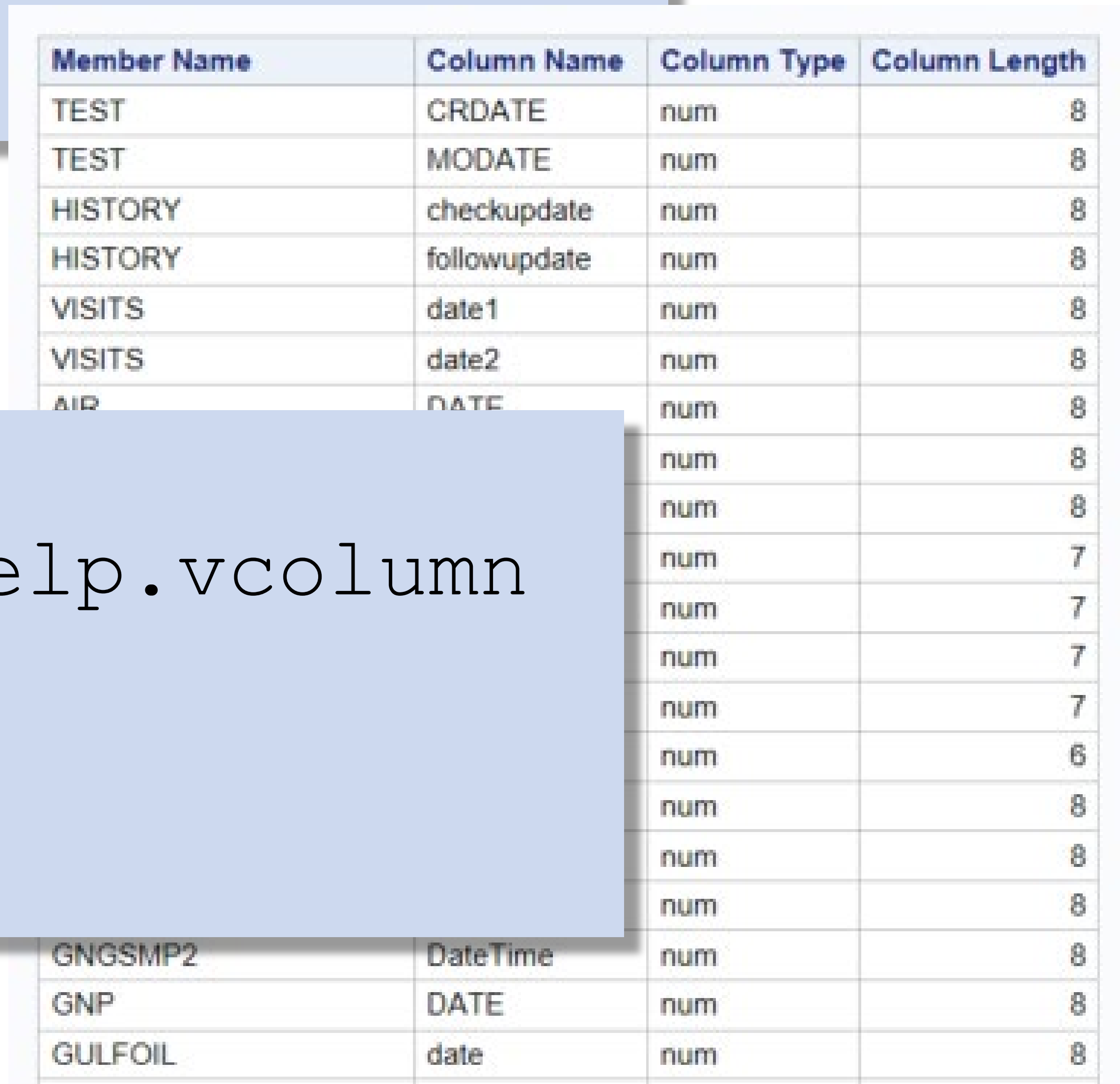
id	Pregnanc...	PlasmaGl...	DBP	Trice...	Insu...	BMI	DiabetesPedia...	Age	Class
1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0

Copyright © SAS Institute Inc. All rights reserved.

Recall Date Column Names

Date Column Names have changed and its hard to remember where all the date columns are now and what they are called

```
proc sql;
    select memname, name, type, length from sashelp.vcolumn
        where upcase(name)like '%DATE%';
quit;
```



Member Name	Column Name	Column Type	Column Length
TEST	CRDATE	num	8
TEST	MODATE	num	8
HISTORY	checkupdate	num	8
HISTORY	followupdate	num	8
VISITS	date1	num	8
VISITS	date2	num	8
AIR	DATE	num	8
		num	8
		num	8
		num	7
		num	7
		num	7
		num	7
		num	6
		num	8
		num	8
		num	8
GNGSMP2	DateTime	num	8
GNP	DATE	num	8
GULFOIL	date	num	8

```
proc sql;
    select memname, name, type, length from sashelp.vcolumn
        where libname='DIABETES' and upcase(name)
        like '%DATE%';
quit;
```

Copyright © SAS Institute Inc. All rights reserved.

Reorder Dataset Variables

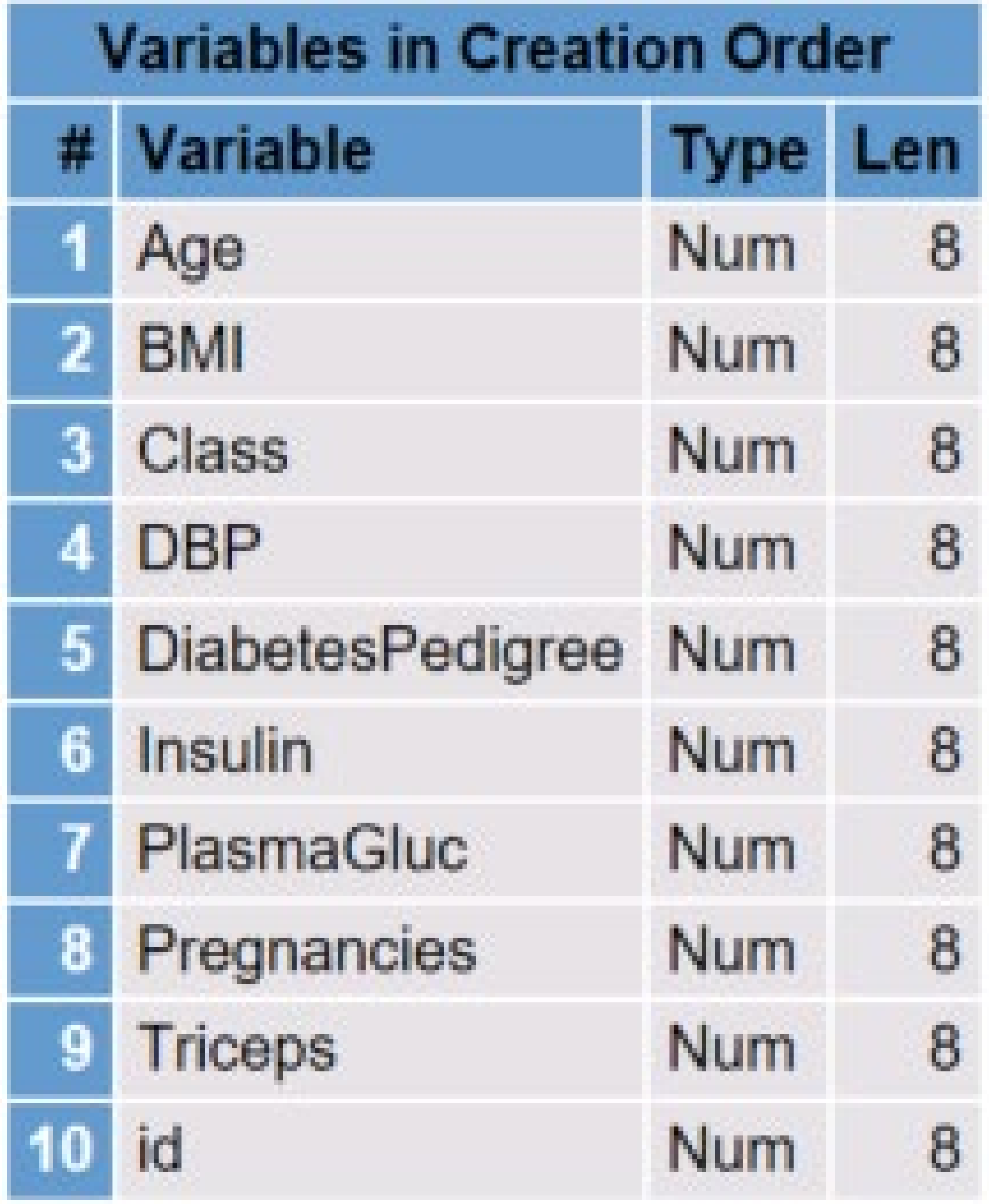
```
proc contents data=diabetes.pima varnum;
run;
proc sql noprint;
   select name into :newname separated by ","
       from dictionary.columns
          where libname ='DIABETES' and
             upcase(memname) ='PIMA'
                order by name
;
%put &=newname;
```

Copyright © SAS Institute Inc. All rights reserved.

Reorder Dataset Variables

```
proc sql;
create table ordered as
    select &newname
        from diabetes.Pima;
quit;
```

```
proc contents data=ordered varnum;
run;
```

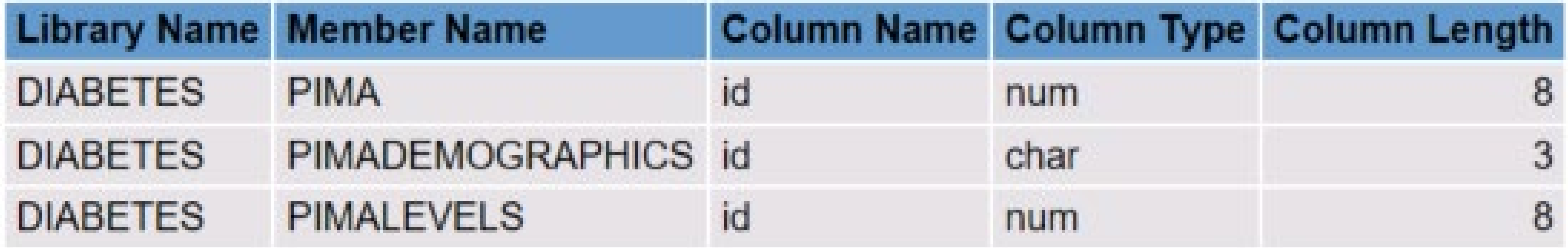


Variables in Creation Order			
#	Variable	Type	Len
1	Age	Num	8
2	BMI	Num	8
3	Class	Num	8
4	DBP	Num	8
5	DiabetesPedigree	Num	8
6	Insulin	Num	8
7	PlasmaGluc	Num	8
8	Pregnancies	Num	8
9	Triceps	Num	8
10	id	Num	8

Copyright © SAS Institute Inc. All rights reserved.

Isolate Variable Type Conflicts

```
proc sql;
    select libname, memname, name, type, length
        from dictionary.columns
            where upcase(name) contains 'ID' and
            libname='DIABETES'
                group by name
                    having count(distinct length) > 1
                        AND count(distinct type) > 1
                            order by 1, 2
;
quit;
```



Library Name	Member Name	Column Name	Column Type	Column Length
DIABETES	PIMA	id	num	8
DIABETES	PIMADEMOGRAPHICS	id	char	3
DIABETES	PIMALEVELS	id	num	8

Copyright © SAS Institute I

Demonstration

Copyright © SAS Institute Inc. All rights reserved.

Handy Links

- PROC SQL INTO Clause
- SAS 9.4 PROC SQL User's Guide
- The Power Of SAS SQL – SAS YouTube Video
- Libeg, Linda. "The SAS®Magical Dictionary Tour".
- Go, Imelda C. "Reordering Variables in a SAS®Data Set".
- SAS Tutorial | Step-By-Step PROC SQL – SAS YouTube Video
- Droogendyk, Harry. "QCYour SAS ®and RDBMS Data Using Dictionary Tables"
- "Shankar, Charu. "Know Thy Data: Techniques For Data Exploration". Pharmasug 2018,
- Proc Sql Syntax Order: Go Home On Time With These 5 PROC SQL Tips. Shankar, Charu
- Kuligowski, Andrew T. & Shankar, Charu. "Know thy data: Techniques for Data Exploration"
- Ask the Expert Webinar - Why Choose Between SAS Data Step & PROC SQL When You Can Have Both
- Eberhardt, Peter & Brill, Irene. "How Do I Look it Up If I Cannot Spell It: An Introduction to SAS® Dictionary Tables".

Copyright © SAS Institute Inc. All rights reserved.

✓ Did you enjoy this session, Let us know in the evaluation

Charu Shankar

EMAIL	Charu.shankar@sas.com
BLOG	https://blogs.sas.com/content/author/charushankar/
TWITTER	CharuYogaCan
LINKEDIN	https://www.linkedin.com/in/charushankar/

Copyright © SAS Institute Inc. All rights reserved.