Sugar Rush: SQL Master Class for Pharma Professionals

Pharmasug

June 2025

Charu Shankar

SAS Education

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.

Agenda

Nuts & Bolts - PROC SQL Overview

Data

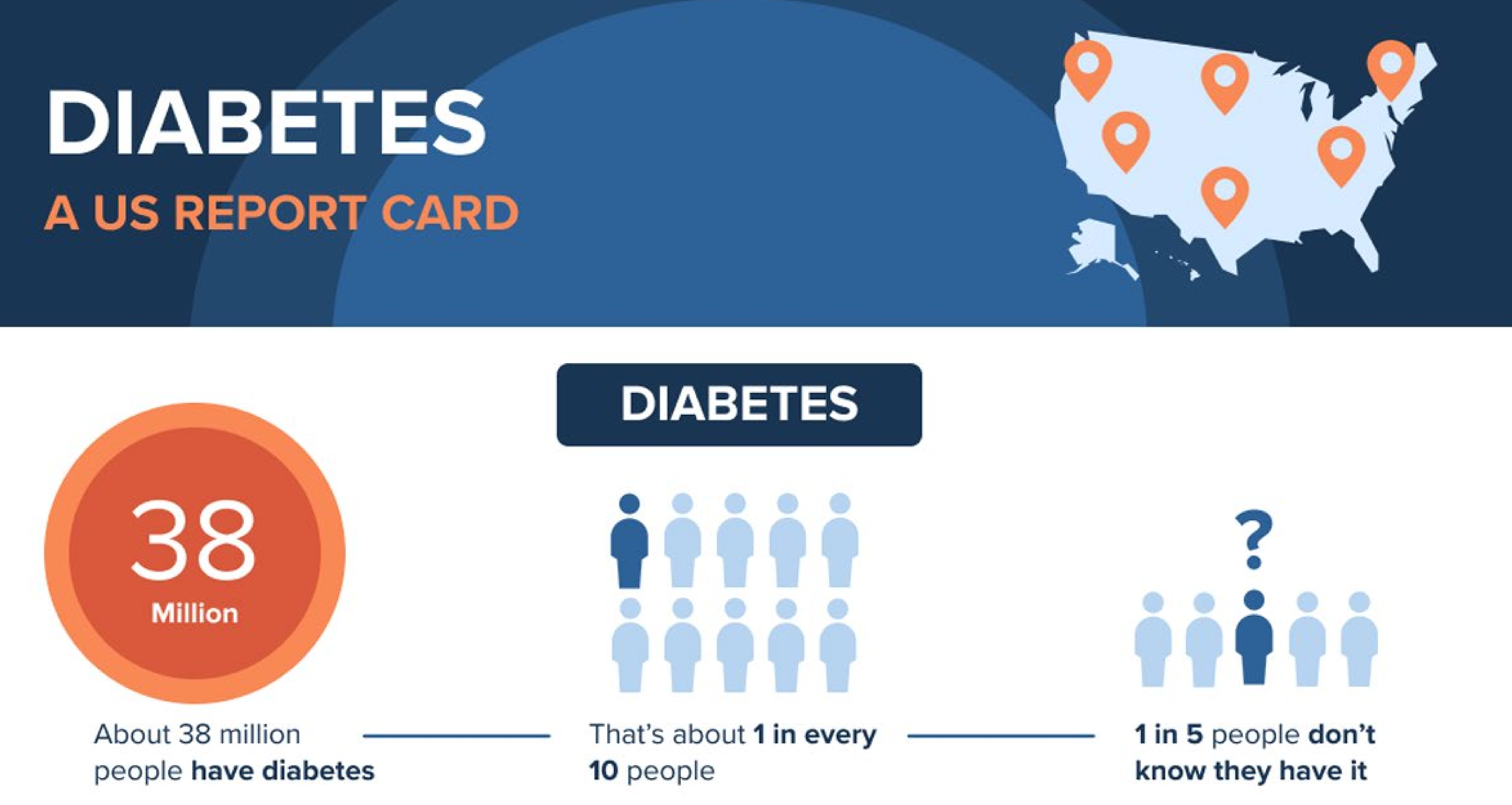
Specifying Rows– Filtering for Focus

Choose, rename, and derive columns in queries

Summarizing Data– Roll it Up: COUNT, AVG, MIN, MAX, GROUP BY

Joining Tables– Connecting the Dots

Handy Links



DIABETES

A US REPORT CARD

DIABETES

38 Million

About 38 million people **have diabetes**

That's about **1 in every 10** people

1 in 5 people **don't know they have it**

Source CDC

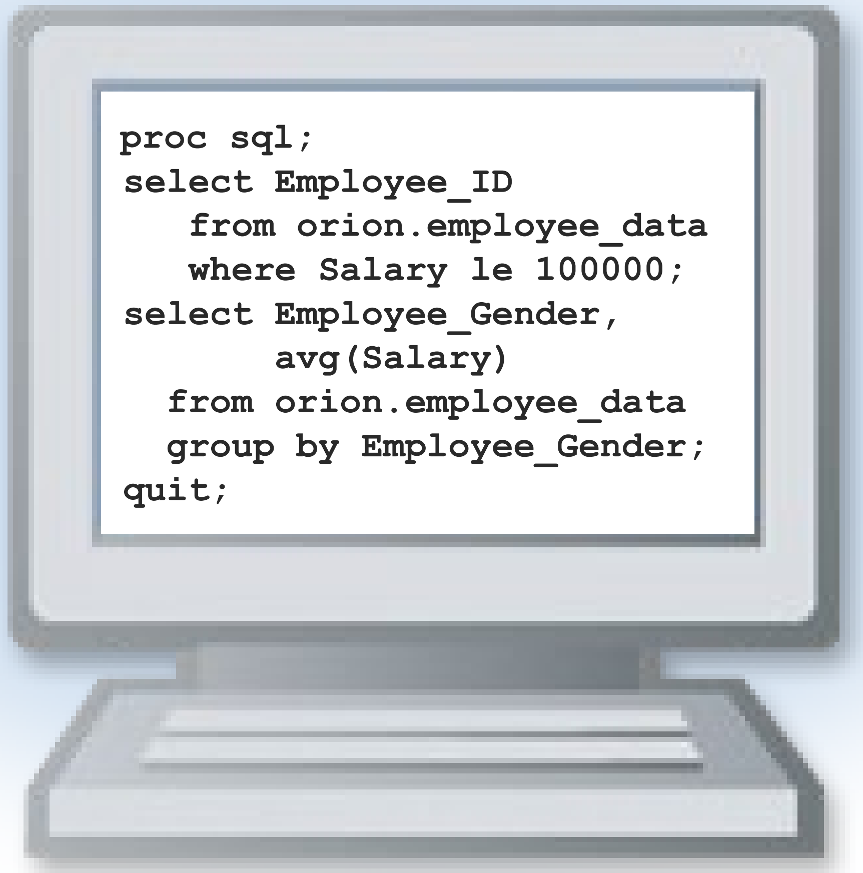
A Report Card: Diabetes in the United States Infographic | Diabetes | CDC

Nuts & Bolts - PROC SQL Overview

Structured Query Language

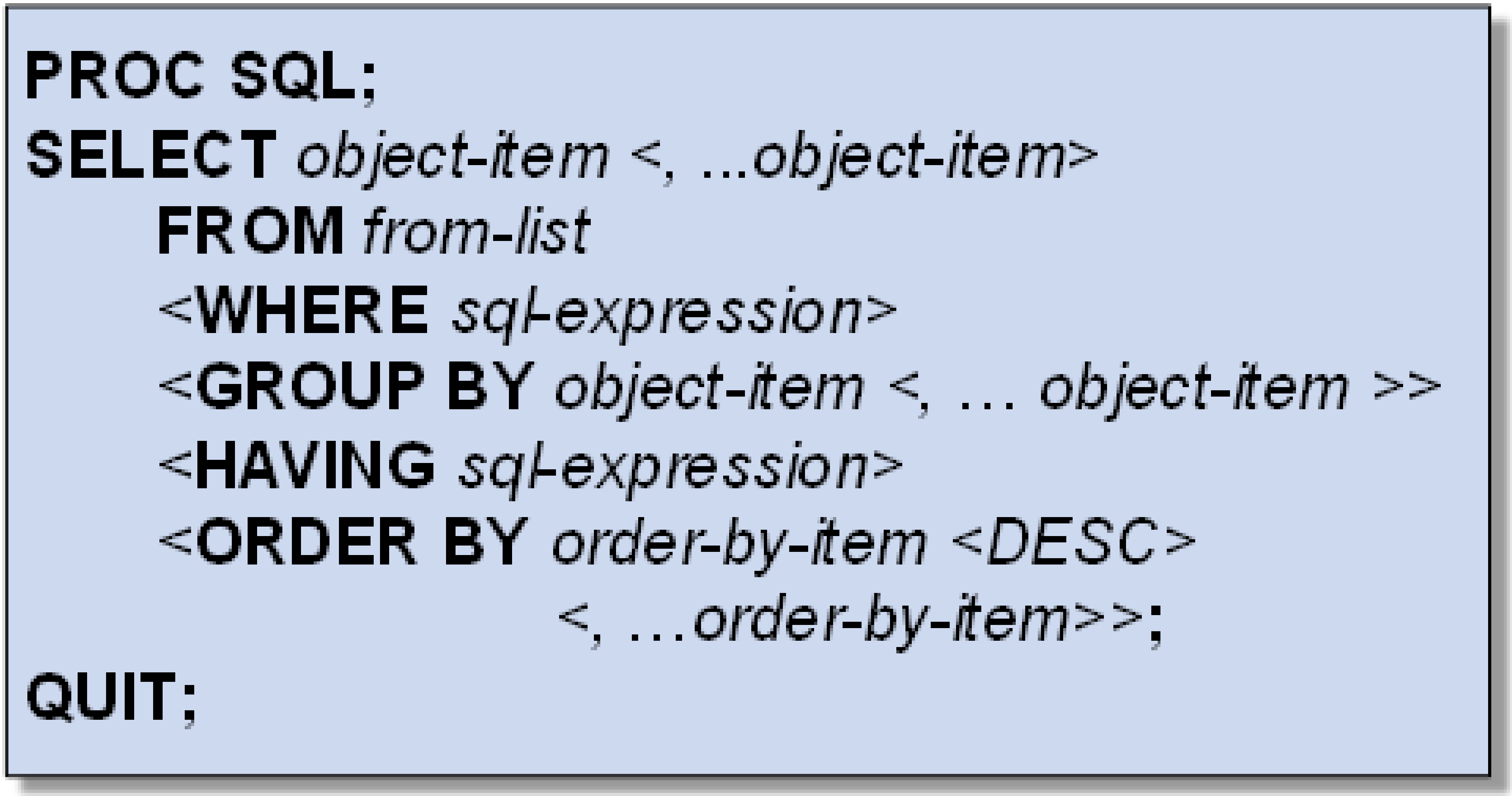
Structured Query Language(SQL) is a standardized language originally designed as a relational database query tool.

SQL is currently used in many software products to retrieve and update data.



```
proc sql;
select Employee_ID
   from orion.employee_data
   where Salary le 100000;
select Employee_Gender,
       avg(Salary)
  from orion.employee_data
  group by Employee_Gender;
quit;
```

SELECT Statement Syntax



```
PROC SQL;
SELECT object-item <, ...object-item>
      FROM from-list
      <WHERE sql-expression>
      <GROUP BY object-item <, … object-item >>
      <HAVING sql-expression>
      <ORDER BY order-by-item <DESC>
                  <, …order-by-item>>;
QUIT;
```

The specified order of the above clauses within the SELECT statement is required.

SELECT Statement

A SELECT statement contains smaller building blocks called *clauses*

```
proc sql;
select Employee_ID, Employee_Gender, Salary
   from orion.employee_information
   where Employee_Gender='F'
   order by Salary desc;
quit;
```

clauses

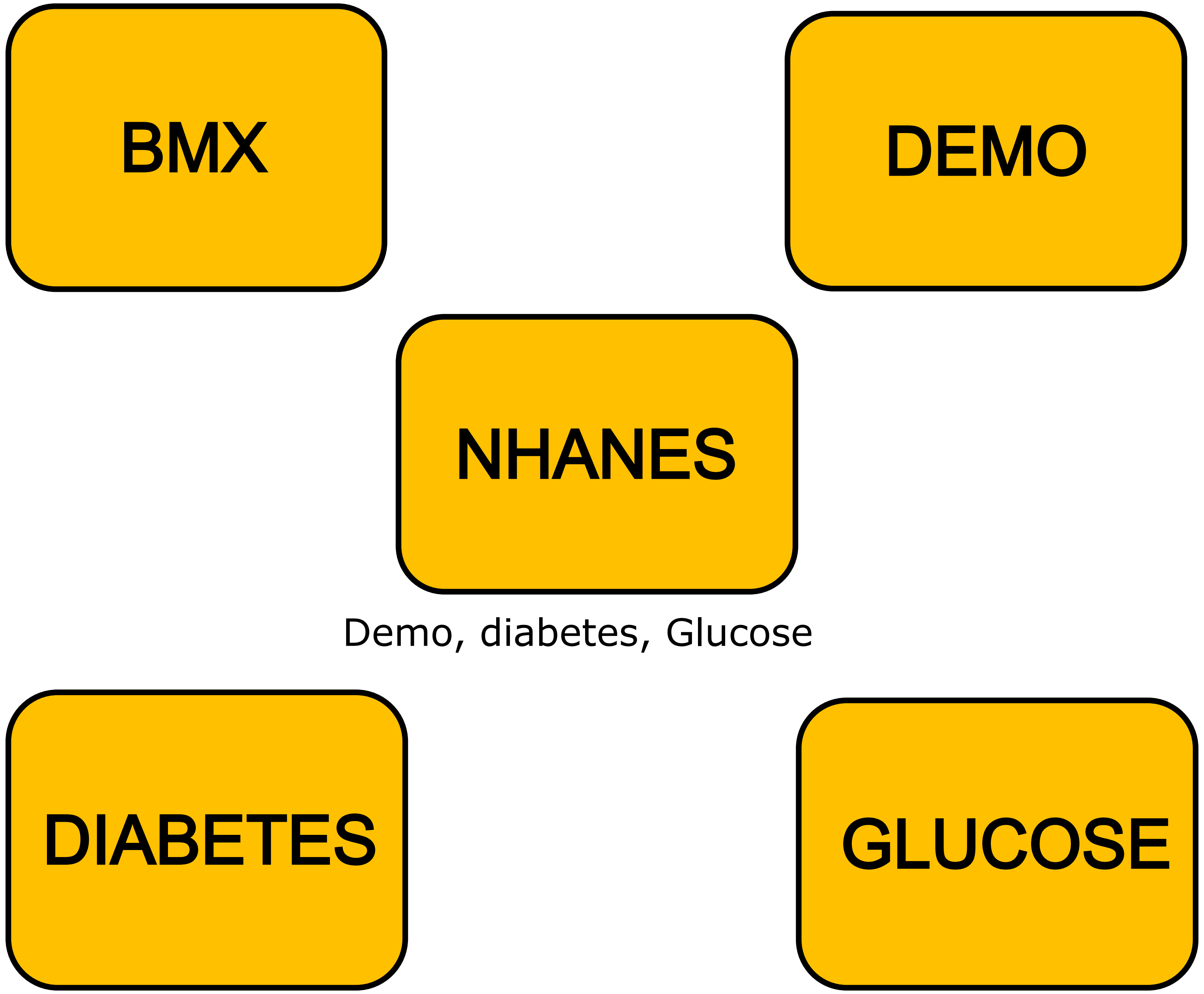
Although it can contain multiple clauses, each SELECT statement begins with the SELECT keyword and ends with a semicolon.

s102d01

The Data

NHANES body measures data track growth trends and obesity rates, and assess how body weight relates to health and nutrition across the U.S. population.

The Demographics public release file includes information that was collected using the Sample Person and Family Demographics questionnaires.



BMX

DEMO

NHANES

Demo, diabetes, Glucose

DIABETES

GLUCOSE

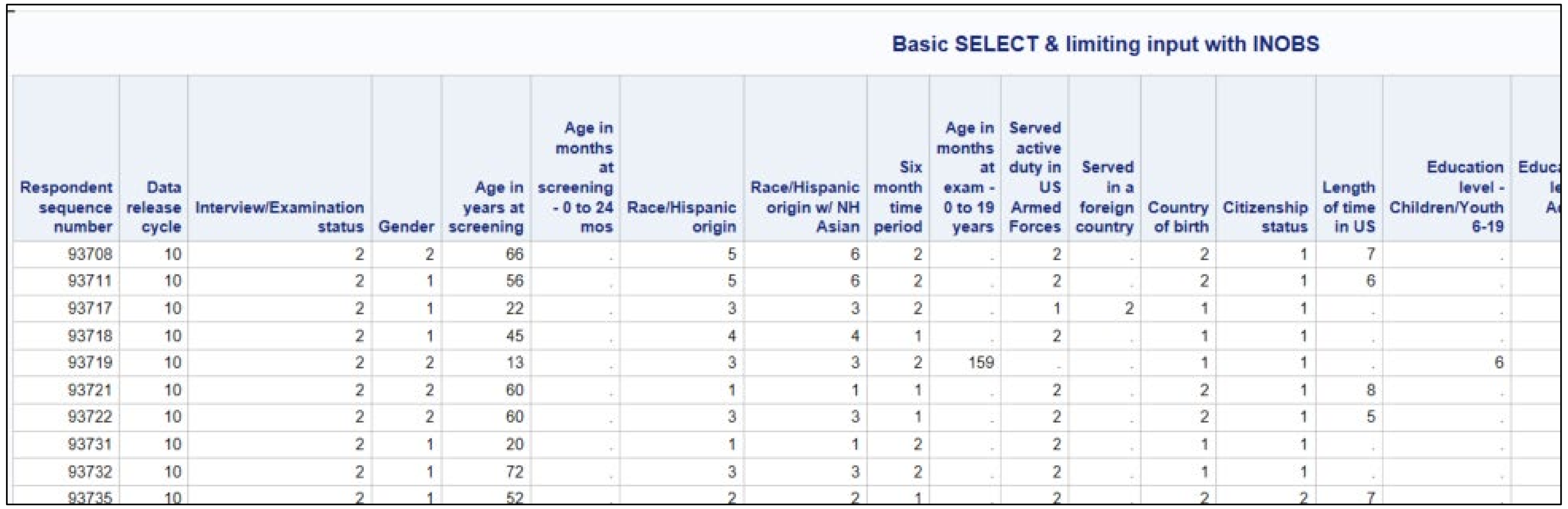
The diabetes section (DIQ) includes interview data on diabetes, prediabetes, treatments, retinopathy, and self-reported awareness of risks, complications, and care practices.

The glucose dataset provides lab-measured blood sugar levels to assess diabetes and metabolic health in the U.S. population.

1. Nuts & Bolts – PROC SQL Overview

Basic SELECT & limiting input with INOBS

```
Title 'Basic SELECT & limiting input with INOBS';
proc sql inobs=100;
    select *
        from sugar.NHANES
    ;
quit;
```



Basic SELECT & limiting input with INOBS

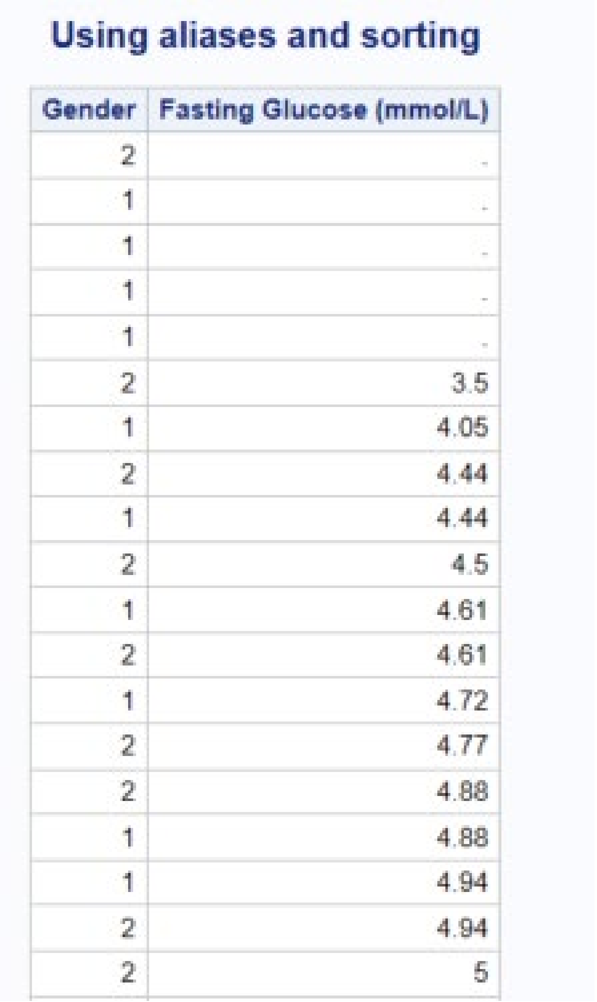
Respondent sequence number	Data release cycle	Interview/Examination status	Gender	Age in years at screening	Age in months at screening - 0 to 24 mos	Race/Hispanic origin	Race/Hispanic origin w/ NH Asian	Six month time period	Age in months at exam - 0 to 19 years	Served active duty in US Armed Forces	Served in a foreign country	Country of birth	Citizenship status	Length of time in US	Education level - Children/Youth 6-19	Educa le A
93708	10	2	2	66	.	5	6	2	.	2	.	2	1	7	.	
93711	10	2	1	56	.	5	6	2	.	2	.	2	1	6	.	
93717	10	2	1	22	.	3	3	2	.	1	2	1	1	.	.	
93718	10	2	1	45	.	4	4	1	.	2	.	1	1	.	.	
93719	10	2	2	13	.	3	3	2	159	.	.	1	1	.	6	
93721	10	2	2	60	.	1	1	1	.	2	.	2	1	8	.	
93722	10	2	2	60	.	3	3	1	.	2	.	2	1	5	.	
93731	10	2	1	20	.	1	1	2	.	2	.	1	1	.	.	
93732	10	2	1	72	.	3	3	2	.	2	.	1	1	.	.	
93735	10	2	1	52	.	2	2	1	.	2	.	2	2	7	.	

Using aliases and sorting

*DMDHRGND - Gender
1 Male
2 Female ;

*LBDGLUSI - Fasting Glucose (mmol/L)
Code or Value 3.28 to 31.1 ;

```
Title 'Using aliases and sorting';
proc sql inobs=100;
    select RIAGENDR as Gender, LBDGLUSI as Glucose_Level
        from sugar.NHANES
            order by 2;
quit;
```



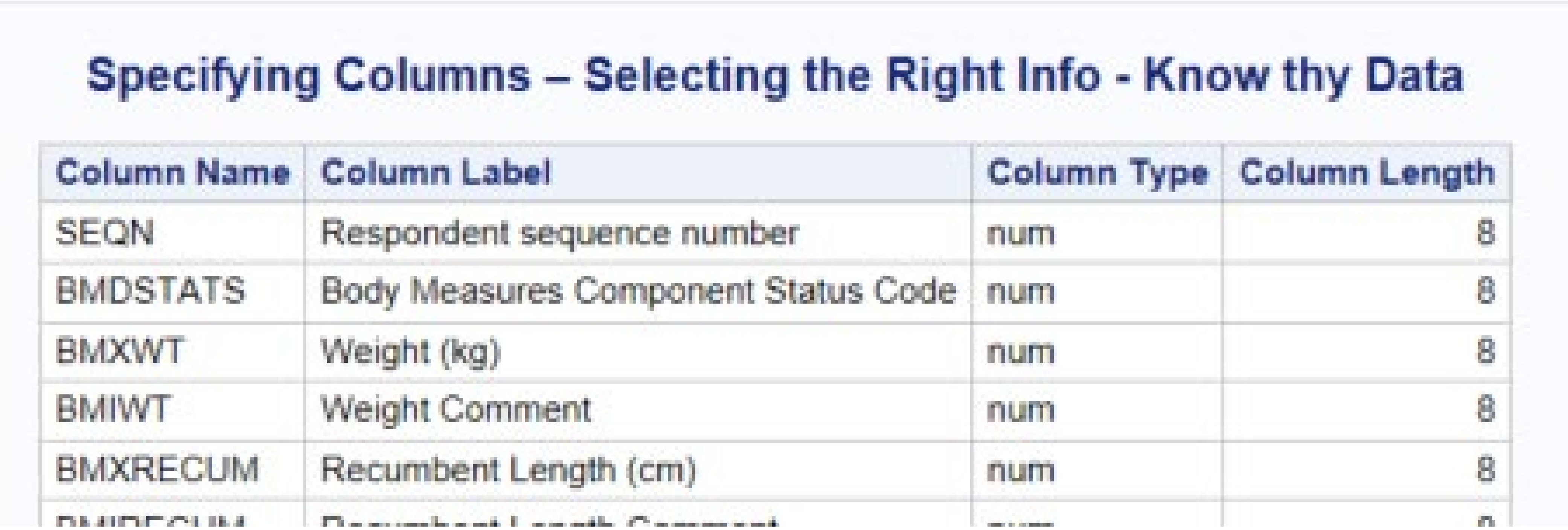
Using aliases and sorting

Gender	Fasting Glucose (mmol/L)
2	.
1	.
1	.
1	.
1	.
2	3.5
1	4.05
2	4.44
1	4.44
2	4.5
1	4.61
2	4.61
1	4.72
2	4.77
2	4.88
1	4.88
1	4.94
2	4.94
2	5

2 Choose, rename, and derive columns in queries

Specifying Columns – Selecting the Right Info - Know thy Data

```
title 'Specifying Columns – Selecting the Right Info - Know thy
Data';
proc sql;
     select name, label, type, length
          from dictionary.columns
               where libname="SUGAR" and memname="BMX";
quit;
```

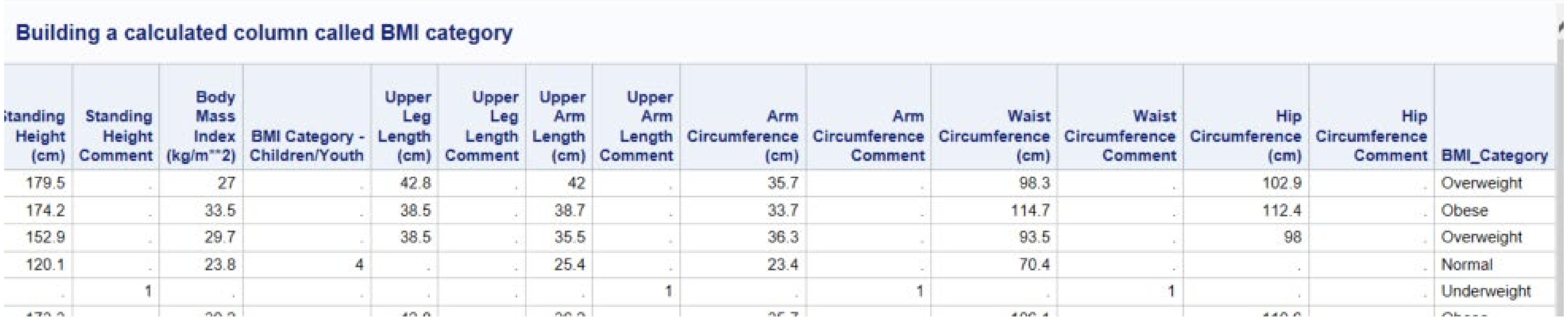


Specifying Columns – Selecting the Right Info - Know thy Data

Column Name	Column Label	Column Type	Column Length
SEQN	Respondent sequence number	num	8
BMDSTATS	Body Measures Component Status Code	num	8
BMXWT	Weight (kg)	num	8
BMIWT	Weight Comment	num	8
BMXRECUM	Recumbent Length (cm)	num	8
BMIRECUM	Recumbent Length Comment	num	8

Specifying Columns – Selecting the Right Info - Know thy Data

```
Title 'Building a calculated column called BMI category';
proc sql inobs=100;
select *,
       case
           when BMXBMI < 18.5 then 'Underweight'
           when BMXBMI between 18.5 and 24.9 then 'Normal'
           when BMXBMI between 25 and 29.9 then 'Overweight'
           else 'Obese'
       end as BMI_Category
                 from sugar.bmx;
quit;
```

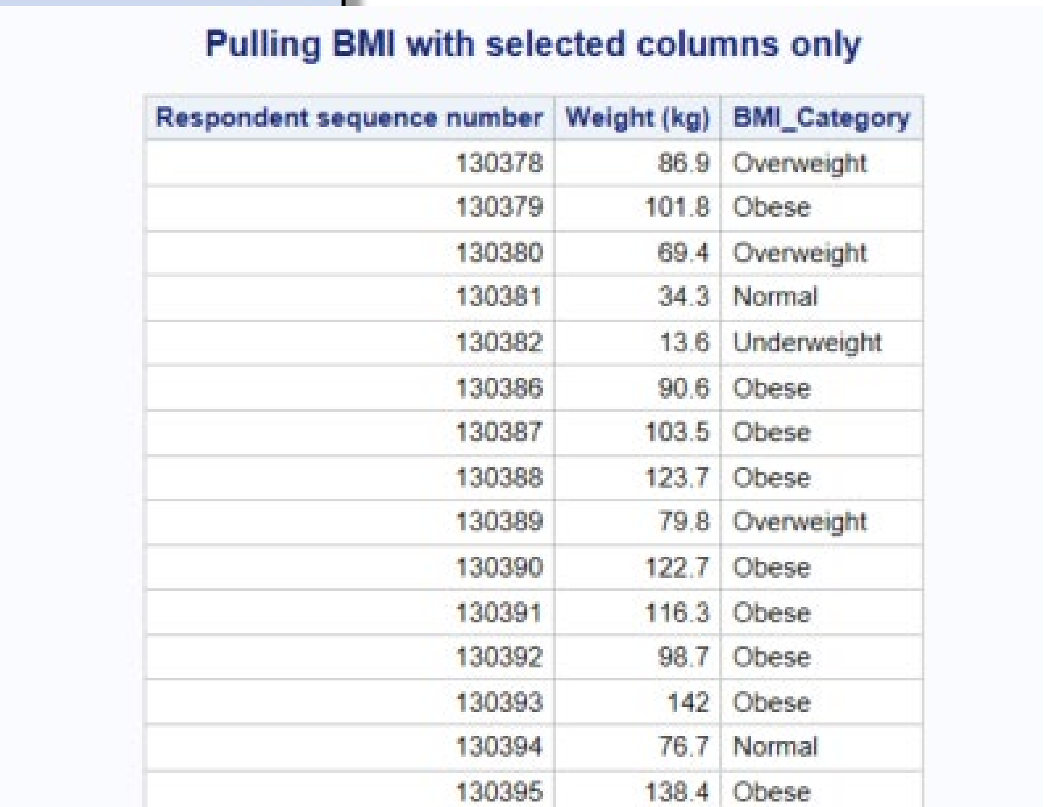


Building a calculated column called BMI category

tanding Height (cm)	Standing Height Comment	Body Mass Index (kg/m**2)	BMI Category - Children/Youth	Upper Leg Length (cm)	Upper Leg Length Comment	Upper Arm Length (cm)	Upper Arm Length Comment	Arm Circumference (cm)	Arm Circumference Comment	Waist Circumference (cm)	Waist Circumference Comment	Hip Circumference (cm)	Hip Circumference Comment	BMI_Category
179.5	.	27	.	42.8	.	42	.	35.7	.	98.3	.	102.9	.	Overweight
174.2	.	33.5	.	38.5	.	38.7	.	33.7	.	114.7	.	112.4	.	Obese
152.9	.	29.7	.	38.5	.	35.5	.	36.3	.	93.5	.	98	.	Overweight
120.1	.	23.8	4	.	.	25.4	.	23.4	.	70.4	.	.	.	Normal
.	1	.	.	.	.	.	1	.	1	.	1	.	.	Underweight

Pulling BMI with selected columns only

```
Title 'Building a calculated column called BMI category';
proc sql inobs=100;
select *,
       case
           when BMXBMI < 18.5 then 'Underweight'
           when BMXBMI between 18.5 and 24.9 then 'Normal'
           when BMXBMI between 25 and 29.9 then 'Overweight'
           else 'Obese'
       end as BMI_Category
              from sugar.bmx;
quit;
```



Pulling BMI with selected columns only

Respondent sequence number	Weight (kg)	BMI_Category
130378	86.9	Overweight
130379	101.8	Obese
130380	69.4	Overweight
130381	34.3	Normal
130382	13.6	Underweight
130386	90.6	Obese
130387	103.5	Obese
130388	123.7	Obese
130389	79.8	Overweight
130390	122.7	Obese
130391	116.3	Obese
130392	98.7	Obese
130393	142	Obese
130394	76.7	Normal
130395	138.4	Obese

3 Specifying Rows – Filtering for Focus

Family History

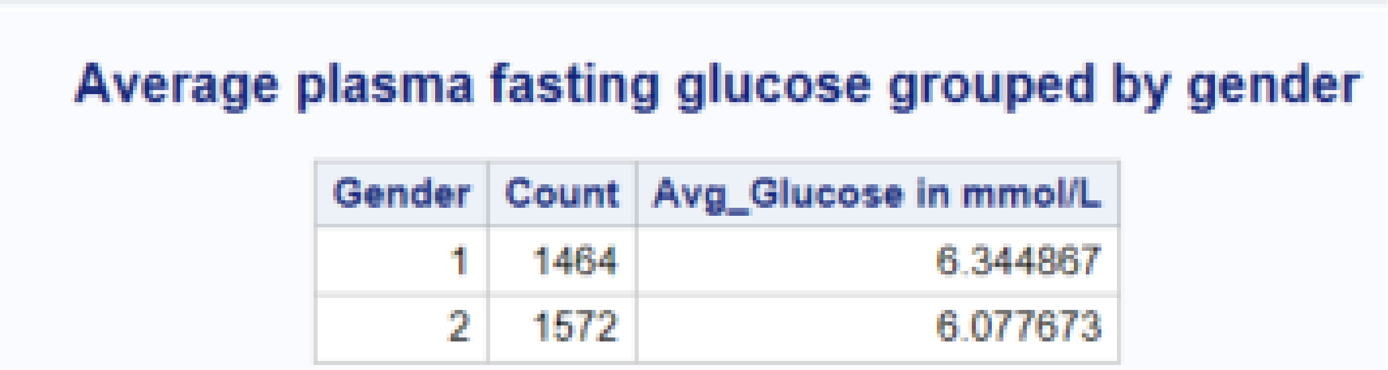
```
Title 'Family history';
proc sql;
     select * from sugar.diabetes
          where DIQ175A =10;
quit;
%put &=sqlobs;

Title 'Family History and High Cholesterol';
proc sql;
     select * from sugar.diabetes
          where DIQ175A =10 and DIQ175J = 19;
quit;
%put &=sqlobs;
```

4. Summarizing Data – Roll it Up: COUNT, AVG, MIN, MAX, GROUP BY

Average plasma fasting glucose grouped by gender

```
title 'Average plasma fasting glucose grouped by gender';
proc sql;
     SELECT RIAGENDR 'Gender', count(*) AS Count, avg(LBDGLUSI)
'Avg_Glucose in mmol/L'
         FROM sugar.nhanes
              group by 1;
quit;
```



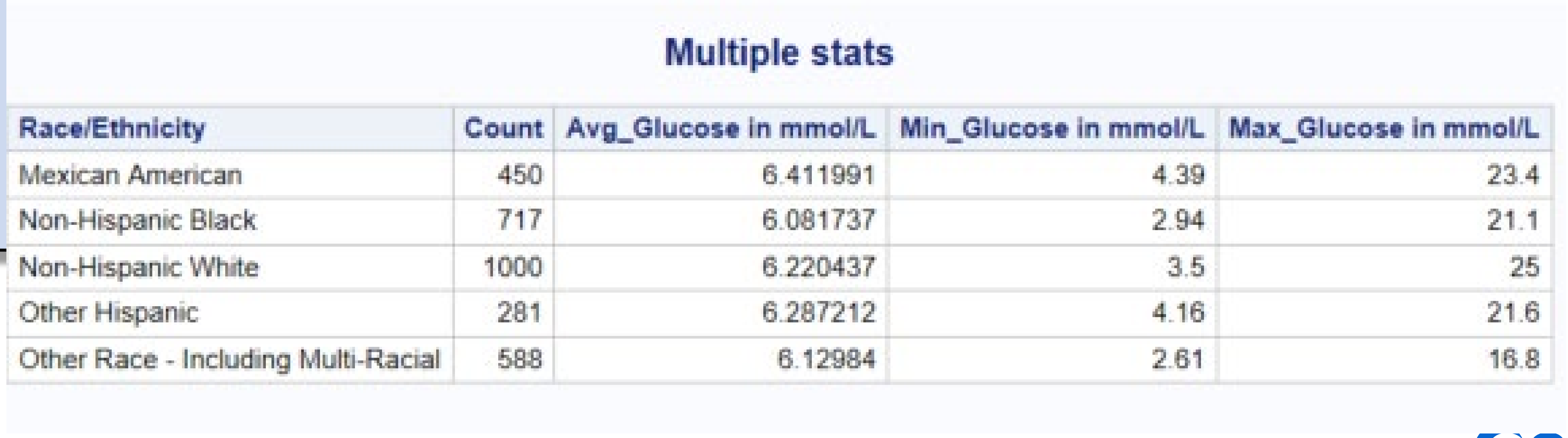
Average plasma fasting glucose grouped by gender

Gender	Count	Avg_Glucose in mmol/L
1	1464	6.344867
2	1572	6.077673

The plasma fasting glucose value in mg/dL (LBXGLU) was converted to mmol/L (LBDGLUSI) by multiplying by 0.05551 (rounded to 3 decimals)

Multiple stats

```
title 'Multiple stats';
proc sql;
     SELECT
          case(RIDRETH1)
               when 1 then 'Mexican American'
               when 2     then 'Other Hispanic'
               when 3     then 'Non-Hispanic White'
               when 4     then 'Non-Hispanic Black'
               when 5     then 'Other Race - Including Multi-Racial'
          end as Race 'Race/Ethnicity',  count(*) as Count, avg(LBDGLUSI)
'Avg_Glucose in mmol/L',
          min(LBDGLUSI) 'Min_Glucose in mmol/L', max(LBDGLUSI) 'Max_Glucose
in mmol/L'
     FROM sugar.nhanes
          group by 1;
quit;
```



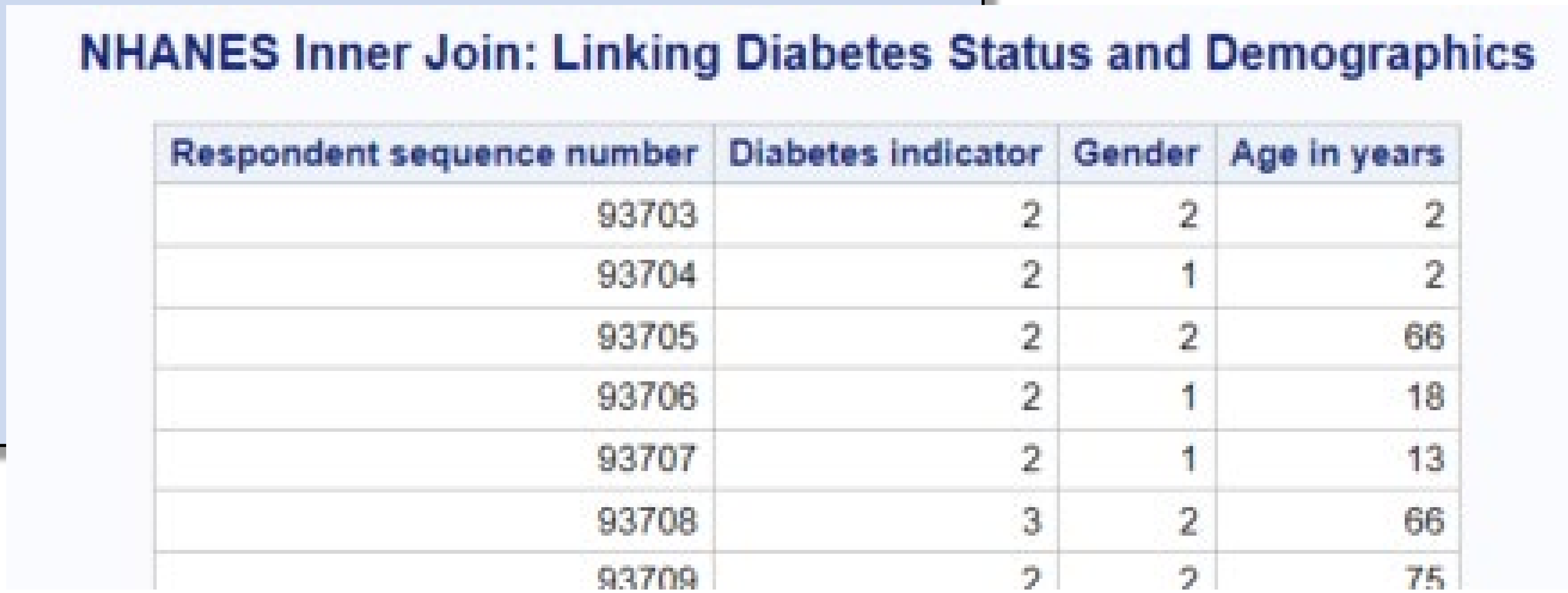
Multiple stats

Race/Ethnicity	Count	Avg_Glucose in mmol/L	Min_Glucose in mmol/L	Max_Glucose in mmol/L
Mexican American	450	6.411991	4.39	23.4
Non-Hispanic Black	717	6.081737	2.94	21.1
Non-Hispanic White	1000	6.220437	3.5	25
Other Hispanic	281	6.287212	4.16	21.6
Other Race - Including Multi-Racial	588	6.12984	2.61	16.8

5 Joining Tables – Connecting the Dots

Multiple stats

```
Title 'NHANES Inner Join: Linking Diabetes Status and
Demographics';
proc sql;
    select
        diab.SEQN,
        diab.DIQ010 'Diabetes indicator',
        demo.RIAGENDR 'Gender',
        demo.RIDAGEYR 'Age in years'
    from sugar.diabetes as diab
    inner join sugar.demo as demo
        on diab.SEQN = demo.SEQN;
quit;
```



NHANES Inner Join: Linking Diabetes Status and Demographics

Respondent sequence number	Diabetes indicator	Gender	Age in years
93703	2	2	2
93704	2	1	2
93705	2	2	66
93706	2	1	18
93707	2	1	13
93708	3	2	66
93709	2	2	75

DIQ010(diabetes_status) is used to determine if a person has been told they
have diabetes by a doctor
or other health professional.
1 (YES): Indicates that the respondent has been told they have diabetes.
2 (NO): Indicates that the respondent has not been told they have diabetes.

Handy Links

- SAS 9.4 PROC SQL user's guide
- Video - Step-by-step PROC SQL
- Go home on time with 5 PROC SQL tips
- Ask The Expert Webinar – Top 5 Handy PROC SQL Tips
- Know thy data: Dictionary tables SAS Global Forum Paper
- SAS YouTube Video - Mastering the WHERE clause in PROG SQL
- SAS YouTube Video - Power of SAS SQL –SAS Global Forum 2021
- SAS YouTube Video - Step by step PROC SQL – SAS Global forum 2020
- "Ask the Expert Webinar - Why choose between SAS data Step & PROC SQL When You Can Have Both
- NHANES Demographic Data
- NHANES Diabetes Data
- NHANES Glucose Data
- NHANES BMX_L Data

Recommended

- Recommended Course - SAS®SQL 1: Essentials

Recommended Presentations		
SI-342 : Tuesday, 10:00 AM – 10:20 AM, Location: Indigo 206	Comparing SQL and Graph Database Query Methods for Answering Clinical Trial Questions with LLM-Powered Pipelines	Jaime Yan, Merck
RW-234 : Wednesday, 9:00 AM – 9:20 AM, Location: Indigo 206	Going from PROC SQL to PROC FedSQ for CAS Processing– Common mistakes to avoid.	Vijayasarathy Govindarajan, SAS Institute

Thank You

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

Charu Shankar
SAS Institute Toronto

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