Lesson 1: Essentials

1.1 Setting Up for the Course
1.2 What Is SQL?
1.3 Introduction to the SQL Procedure

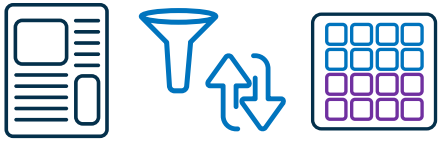
Lesson 1: Essentials

1.1 Setting Up for the Course

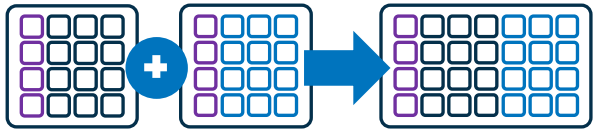
1.2 What Is SQL?

1.3 Introduction to the SQL Procedure

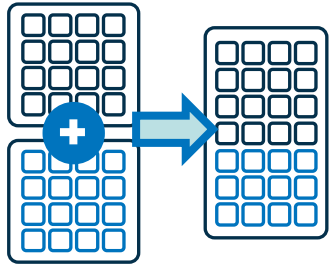
Course Overview



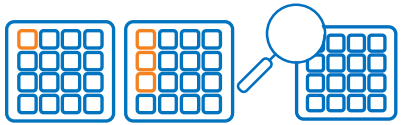
Fundamentals



Joins

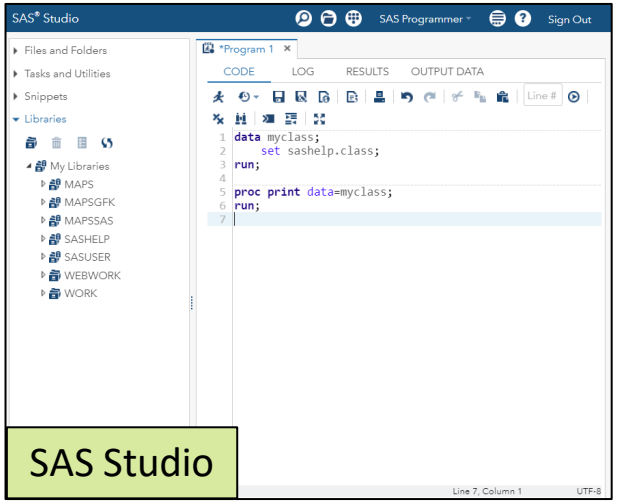


Set Operators

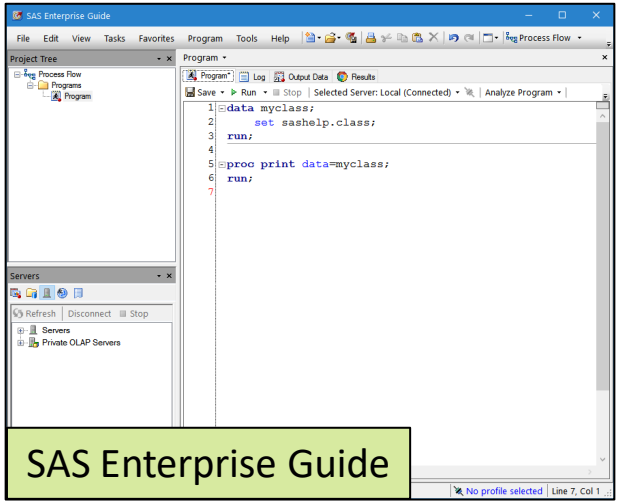


Subqueries

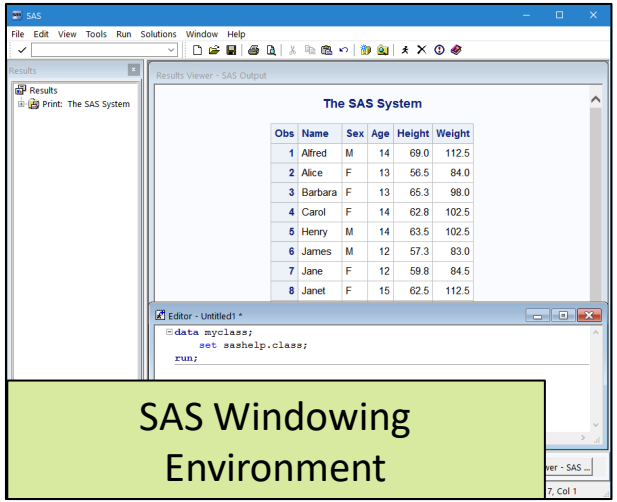
SAS Programming Interfaces



SAS Studio

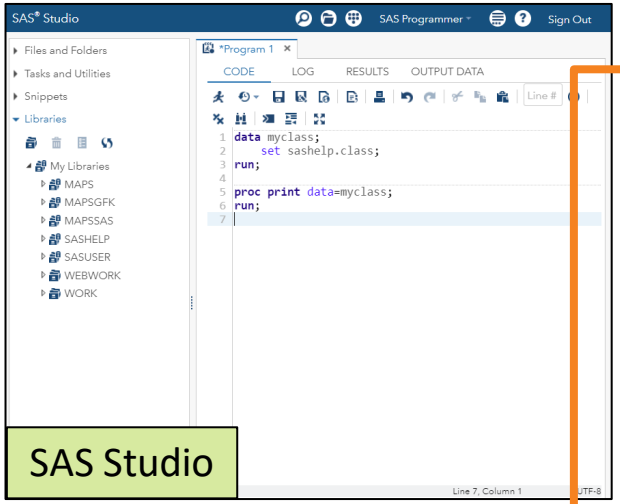


SAS Enterprise Guide

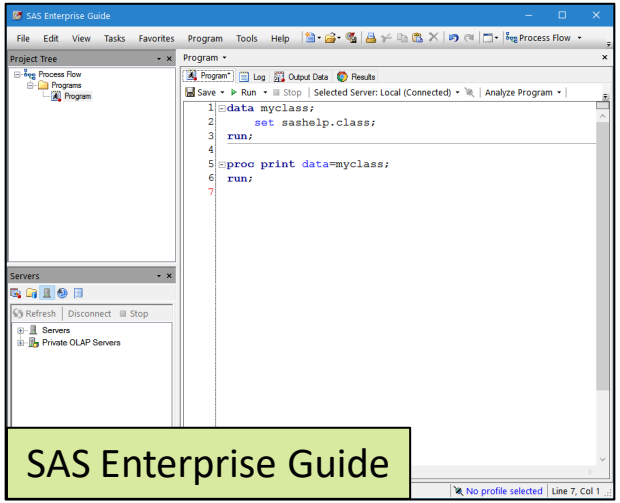


SAS Windowing Environment

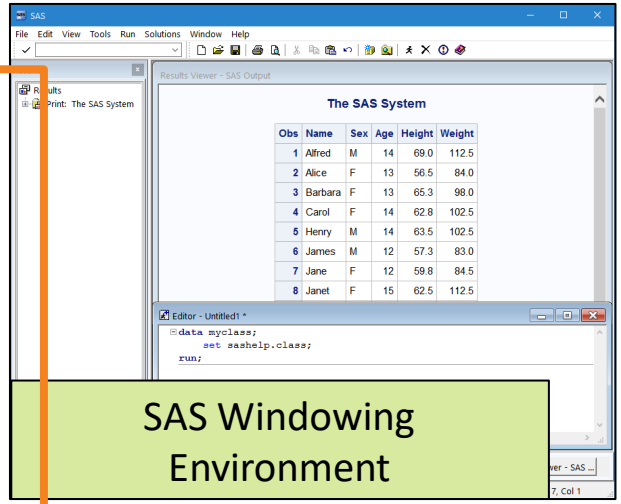
SAS Programming Interfaces



SAS Studio



SAS Enterprise Guide



SAS Windowing Environment

SAS Programming Windows

write and submit code

```
proc sql;
create table targetcust as
select *
   from sq.customer
   where bankid=.;
quit;

proc sql;
select *
   from work.targetcust;
quit;
```

Editor

view messages from SAS

```
31      proc sql;
32      create table targetcust as
33      select *
34        from sq.customer
35        where bankid=.;
NOTE: Table WORK.TARGETCUST created, with [illegible] and
22 columns.

36      quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time          0.11 seconds
      cpu time           0.03 seconds


37
38      proc sql;
39      select *
40        from work.targetcust;
41      quit;
NOTE: PROCEDURE SQL used (Total process time):
      real time          2.48 seconds
      cpu time           2.43 seconds
```

Log

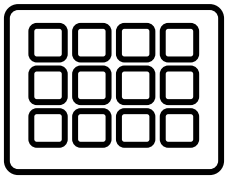
view results

FirstName	MiddleName	LastName	Gender	
James	Arthur	Ramos	M	[illegible]
Mary	Sheilah	Strong	F	11513
Frank	Joseph	Laducer	M	13952
Amy	Valentina	Lusk	F	11384
James	Daniel	Cuevas	M	-2273
Donna	Mary	Brasel	F	7750
Gregory	John	Brooks	M	8487
Richard	Christopher	Malley	M	6857
Charles				
Kelvin				

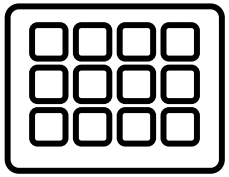
First Name	Middle Name	Last Name	Gender	Date of Birth	Employed	Race	Married	Street Number
James	Arthur	Ramos	M	6710	Y	W	M	363
Mary	Sheilah	Strong	F	11513	Y	W	M	696
Frank	Joseph	Laducer	M	13952	Y	H	M	1
Amy	Valentina	Lusk	F	11384	Y	B	M	379
James	Daniel	Cuevas	M	-2273	N	W	M	158
Donna	Mary	Brasel	F	7750	Y	H	D	28

Results and Output Data

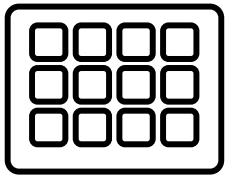
Sample of Data Used in This Course



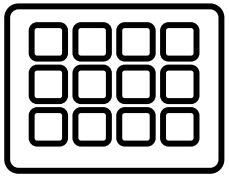
Customers



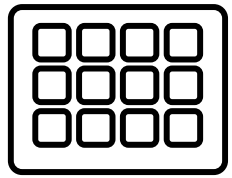
ShoeVendors



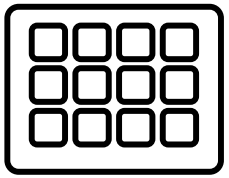
Orders



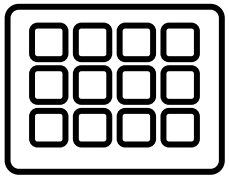
Employee_
donations



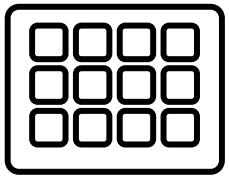
Quarterly_
purchases



Products

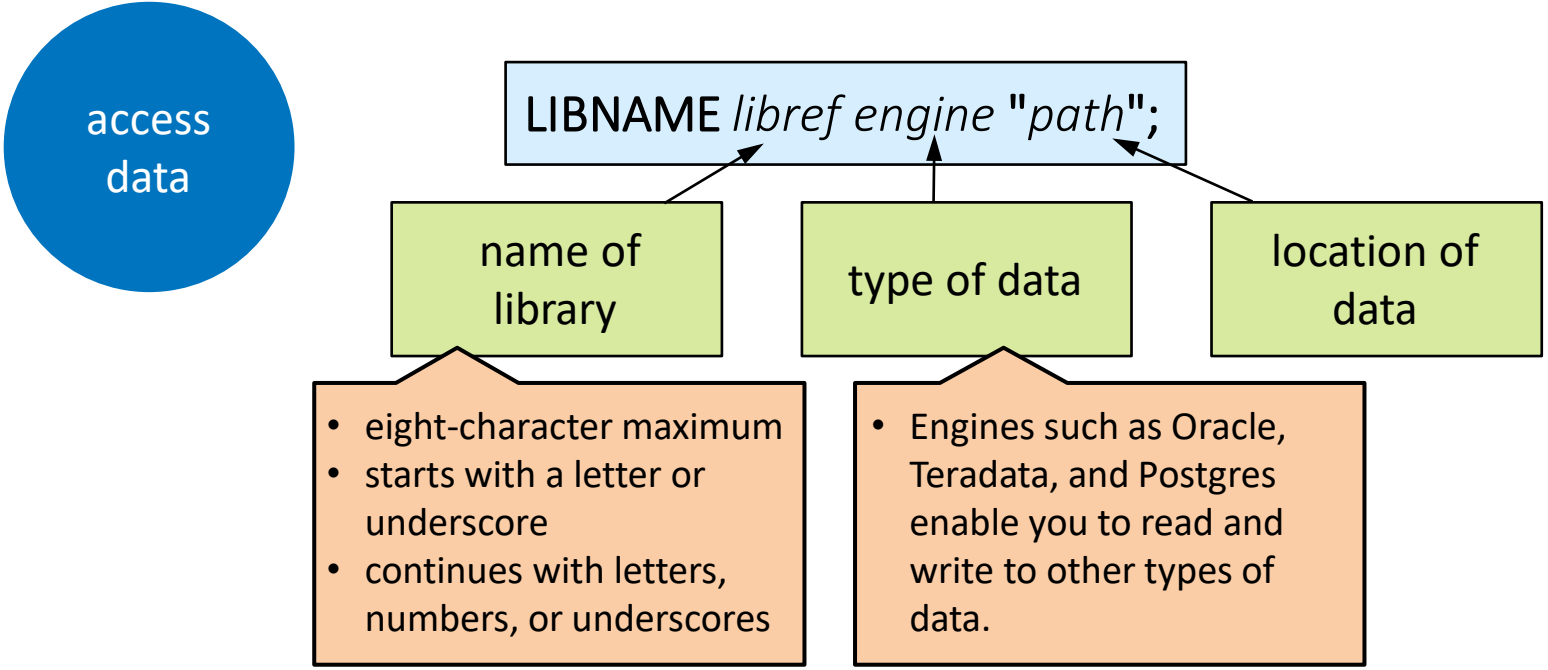


Employee_detail



Employee_
organization

Connecting to a Library to Read SAS Files (Review)



Connecting to a Library to Read SAS Files (Review)

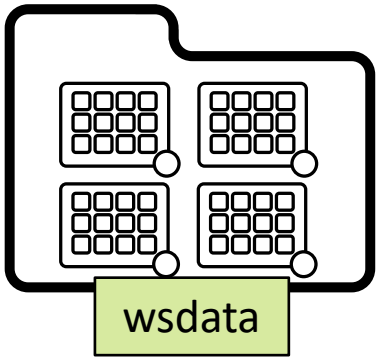
connect to the library

```
libname wsdata base "s:/workshop/data";
```

library name

SAS tables (*default engine*)

data located in s:/workshop/data



LIBNAME is a global statement and does not require a RUN statement.

Lesson 1: Essentials

1.1 Setting Up for the Course

1.2 What Is SQL?

1.3 Introduction to the SQL Procedure

Collection of Related Tables

Customers

#	Variable	Type	Len
11	Customer_Age	Num	8
8	Customer_Age_Group	Char	12
7	Customer_BirthDate	Num	8
2	Customer_Country	Char	2
5	Customer_FirstName	Char	20
3	Customer_Gender	Char	1
10	Customer_Group	Char	40
1	Customer_ID		
6	Customer_LastName	Char	30
4	Customer_Name	Char	40
9	Customer_Type	Char	40

Orders

#	Variable	Type	Len
10	CostPrice_Per_Unit	Num	8
2	Customer_ID	Num	8
5	Delivery_Date	Num	8
11	Discount	Num	8
3	Employee_ID		
4	Order_Date	Num	8
1	Order_ID	Num	8
6	Order_Type	Num	8
7	Product_ID		
13	Profit	Num	8
8	Quantity	Num	8
12	Shipping	Num	8
9	Total_Retail_Price	Num	8

SalesAssociates

#	Variable	Type	Len
5	Birth_Date	Num	8
6	Emp_Hire_Date	Num	8
7	Emp_Term_Date	Num	8
1	Employee_ID	Num	8
10	Employee_Name	Char	40
4	Gender	Char	1
2	Job_Title	Char	25
8	Manager_ID	Num	8
9	SSN	Char	16
3	Salary	Num	8

Products

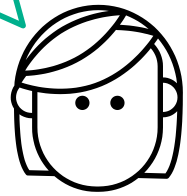
#	Variable	Type	Len
3	Product_Category	Char	24
4	Product_Group	Char	24
1	Product_ID		
2	Product_Line	Char	15
5	Product_Name	Char	45
6	Supplier_Country	Char	2
8	Supplier_ID	Num	8
7	Supplier_Name	Char	28

ShoeVendors

#	Variable	Type	Len
10	Category_Name	Char	45
9	Group_Name	Char	45
11	Line_Name	Char	45
12	Mfg_Suggested_Retail_Price	Num	8
2	Product_Category	Num	8
3	Product_Group	Num	8
4	Product_ID	Num	8
1	Product_Line	Num	8
5	Product_Name	Char	45
8	Supplier_Country	Char	2
6	Supplier_ID	Num	4
7	Supplier_Name	Char	30

Structured Query Language (SQL)

Structured Query Language (***SQL***) is a standardized, widely used language designed to query and manage data in relational databases.



Some developers pronounce it as “S-Q-L” [ˈɛs kjuː ˈɛl] and others use “sequel” [ˈsiːkwəl]. Both are acceptable.

Timeline

Relational Database Model Proposed

1970

IBM

SEQUEL Developed

1973

ANSI

First ANSI Standard SQL (SQL-87)

1986

ANSI

First ***major*** revision to the ANSI Standard (SQL-2)

1992

sas

SAS PROC SQL

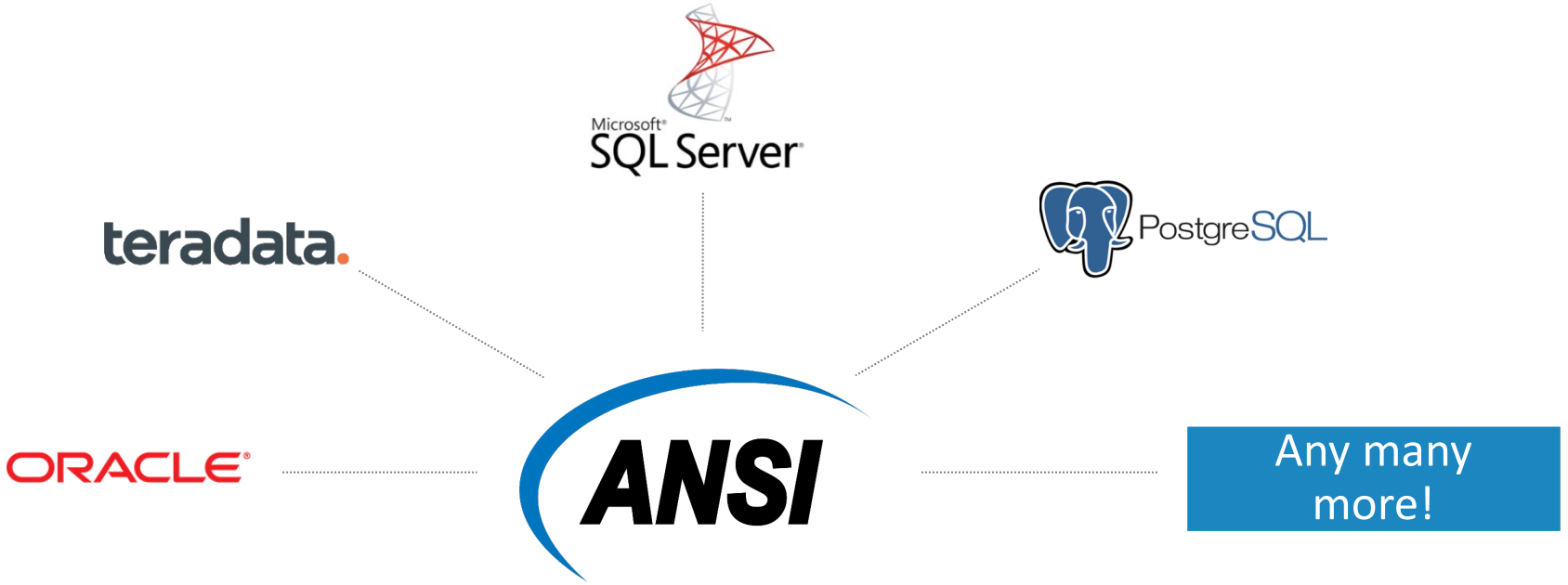
1993

ANSI

Last ***major*** revision to the ANSI Standard (SQL-3)

1999

SQL Implementations



Database Management Systems (***DBMS***) offer different implementations of SQL.

Lesson 1: Essentials

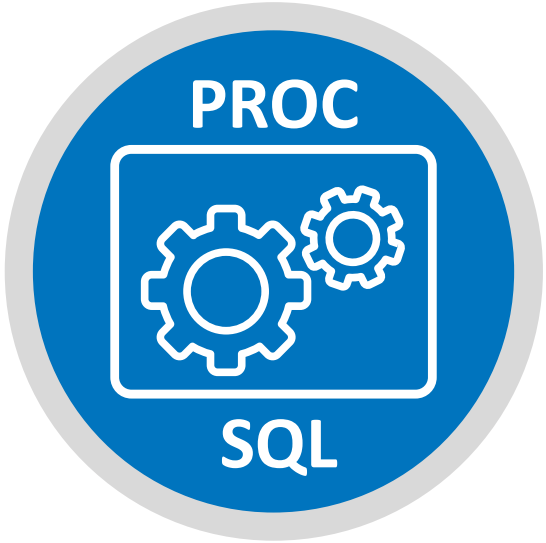
1.1 Setting Up for the Course

1.2 What Is SQL?

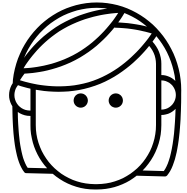
1.3 Introduction to the SQL Procedure

SQL Procedure

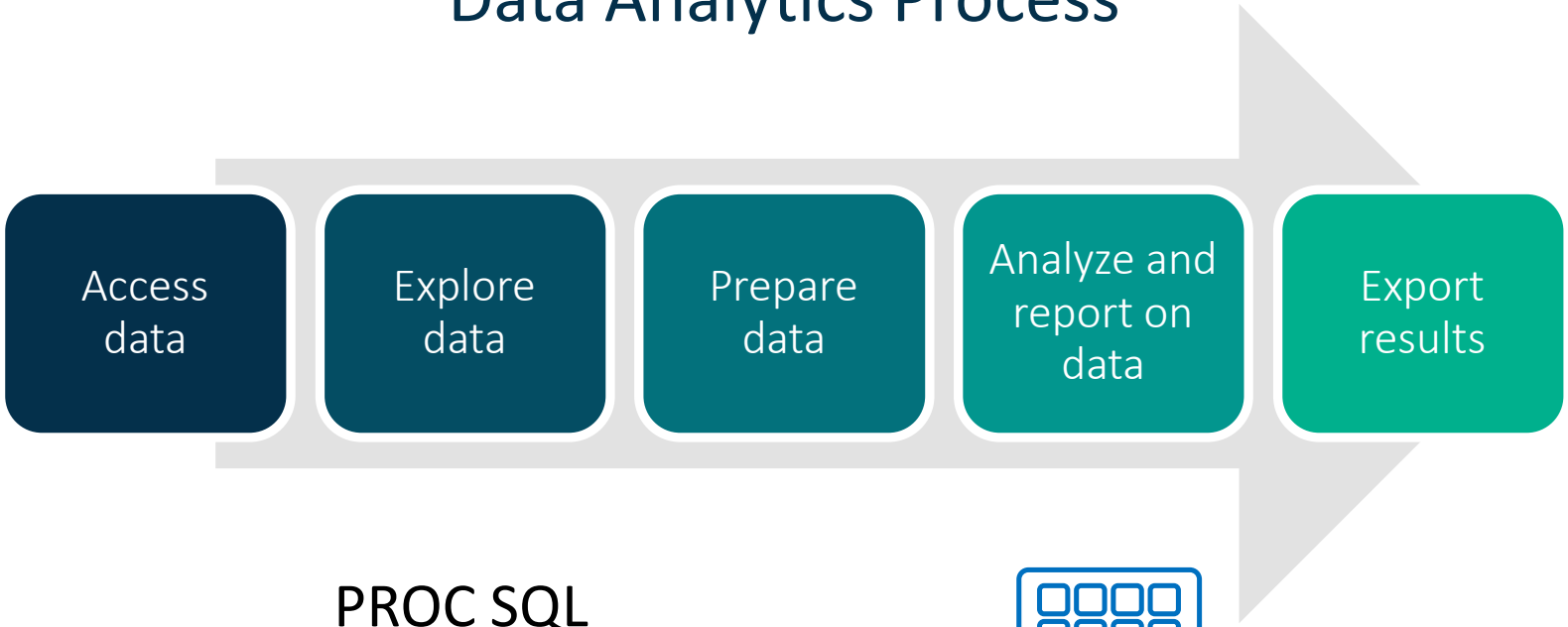
SQL + SAS



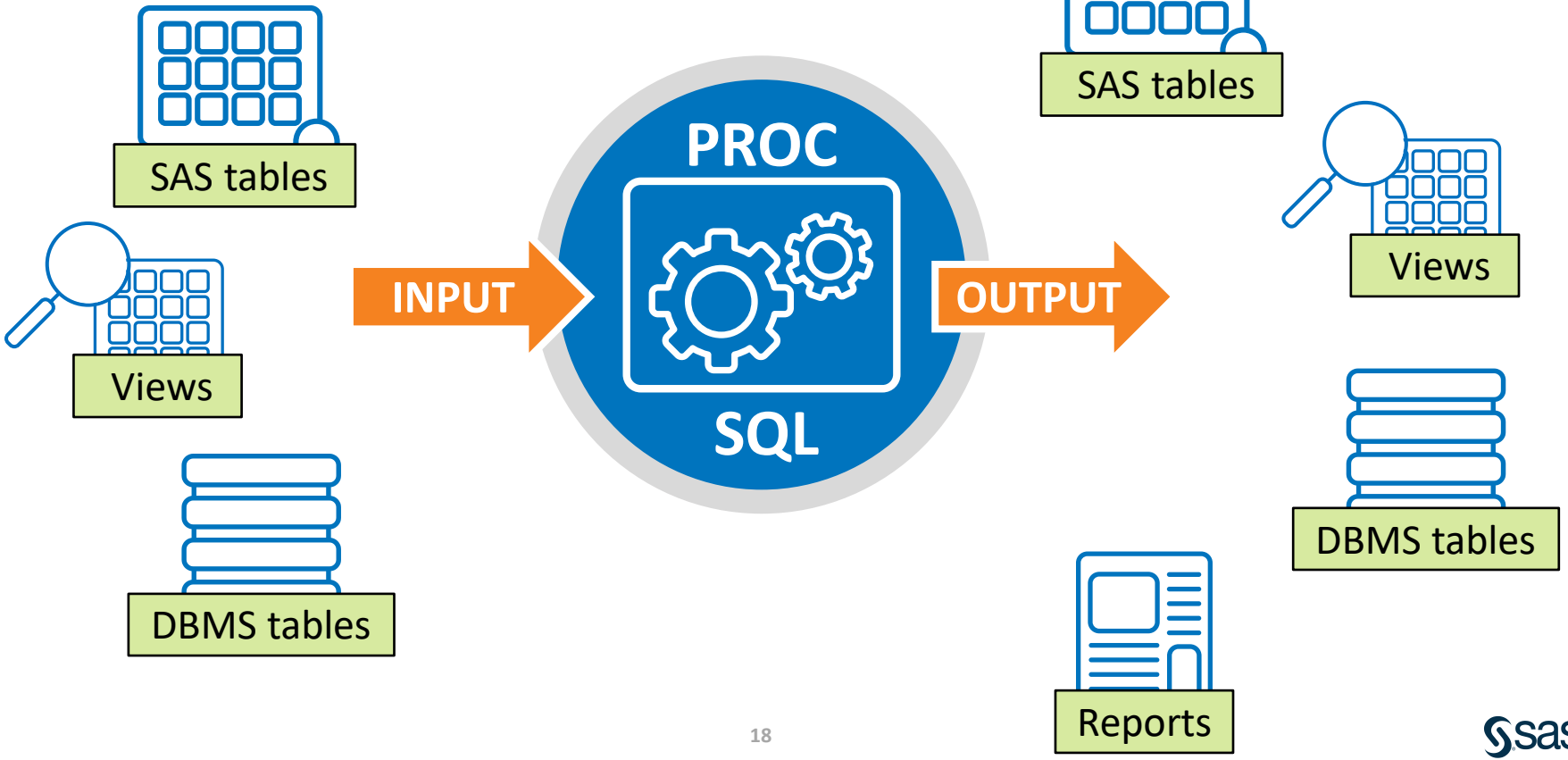
PROC SQL enables the use of SQL in SAS and includes non-ANSI-compliant ***SAS enhancements***.



Data Analytics Process



SQL Procedure



SQL Procedure

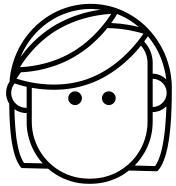
PROC SQL <*options*>**;**

The SQL procedure is initiated with a **PROC SQL** statement.

SELECT *col-name, col-name*
FROM *input-table*
<***WHERE*** *clause*>
<***GROUP BY*** *clause*>
<***HAVING*** *clause*>
<***ORDER BY*** *clause*>**;**

QUIT;

The SQL procedure is terminated with a **QUIT** statement.



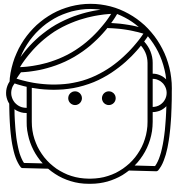
SELECT Statement in PROC SQL

```
PROC SQL <options>;
SELECT col-name, col-name
      FROM input-table
      <WHERE clause>
      <GROUP BY clause>
      <HAVING clause>
      <ORDER BY clause>;
QUIT;
```

query

A **SELECT** statement or query consists of ***clauses*** and ends with a ***semicolon***.

Think of a *query* as asking your data a question.



SELECT Statement Clauses in PROC SQL

PROC SQL <*options*>**;**

SELECT *col-name*, *col-name*
FROM *input-table*
<*WHERE clause*>
<*GROUP BY clause*>
<*HAVING clause*>
<*ORDER BY clause*>**;**

QUIT;

The **SELECT** clause lists the columns to appear in the results, in the order written.

The **FROM** clause specifies the data sources.

The SELECT statement must contain ***SELECT*** and ***FROM*** clauses.

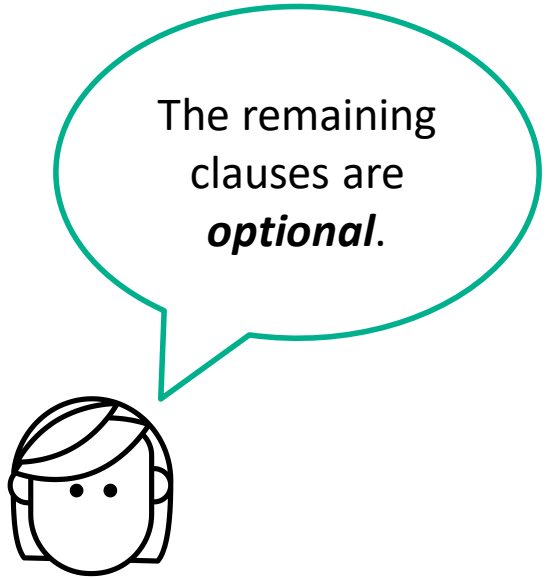
SELECT Statement Clauses in PROC SQL

PROC SQL <*options*>**;**

SELECT *col-name*, *col-name*
FROM *input-table*
<***WHERE*** *clause*>
<***GROUP BY*** *clause*>
<***HAVING*** *clause*>
<***ORDER BY*** *clause*>**;**

QUIT;

If present, the other clauses ***must*** be in this order.



Additional Statements in PROC SQL

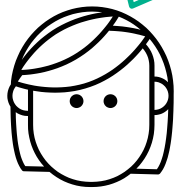
PROC SQL <*options*>**;**

*statement(s)***;**

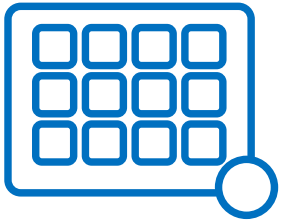
QUIT;

PROC SQL supports additional statements. Each statement begins with the ***keyword*** and ends with a ***semicolon***.

Visit the SAS documentation for all available statements.

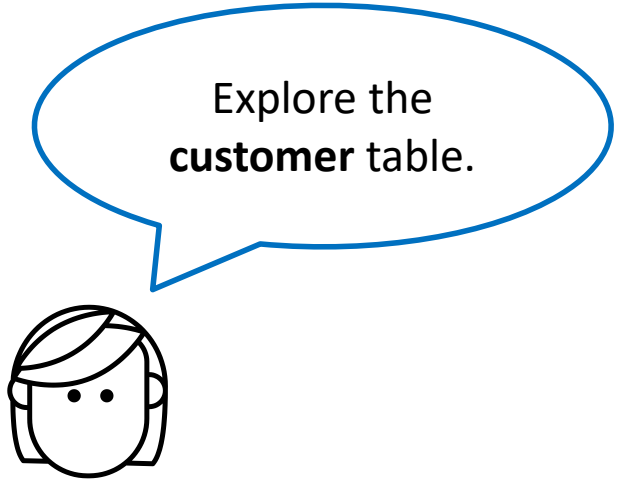


Exploring Tables



customers

- column attributes
- preview first 10 rows



Explore Data

Viewing Column Attributes

DESCRIBE TABLE *table-name*;

```
proc sql;
     describe table orion.customers;
quit;
```

```
31  28          PROC SQL;
32  29          DESCRIBE TABLE ORION.CUSTOMERS;
33  NOTE: SQL table ORION.CUSTOMERS was created like:
34
35  create table ORION.CUSTOMERS( bufsize=16384 )
36    (
37     Customer_ID num format=12. label='Customer ID',
38     Customer_Country char(2) label='Customer Country',
39     Customer_Gender char(1) label='Customer Gender',
40     Customer_Name char(40) label='Customer Name',
41     Customer_FirstName char(20) label='Customer First Name',
42     Customer_LastName char(30) label='Customer Last Name',
43     Customer_BirthDate num format=DATE9. label='Customer Birth Date',
44     Customer_Age_Group char(12) label='Customer Age Group',
45     Customer_Type char(40) label='Customer Type Name',
46     Customer_Group char(40) label='Customer Group Name',
47     Customer_Age num label='Customer Age'
48    );
49
50  30          QUIT;
51  NOTE: PROCEDURE SQL used (Total process time):
52        real time           0.00 seconds
53        cpu time            0.00 seconds
```

The **DESCRIBE TABLE** statement lists all the columns in a table and their properties.

Explore Data

Previewing the Table

SELECT *col-name, col-name*
FROM *input-table <(data set options)>*;

```
proc sql;
select Customer_FirstName, Customer_LastName,
Customer_Country
    from orion.customers (obs=10);
quit;
```

Customer First Name	Customer Last Name	Customer Country
James	Kvarniq	US
Sandrina	Stephano	US
Cornelia	Krahl	DE
Karen	Ballinger	US
Elke	Wallstab	DE
David	Black	US
Markus	Sepke	DE
Ulrich	Heyde	DE
Jimmie	Evans	US
Tonie	Asmussen	US

SQL Options: Controlling Processing

PROC SQL INOBS=*n*;

Limits rows from each source table that contribute to a query

PROC SQL OUTOBS=*n*;

Restricts the number of rows that a query outputs

```
proc sql inobs=10;
select Customer_ID, Customer_LastName, Customer_BirthDate
    from ORION.CUSTOMERS;
quit;
```

SQL Options: Controlling Display

PROC SQL NUMBER;

controls whether the row number is displayed as the first column in the query results

```
proc sql inobs=10 number;
select Customer_ID, Customer_LastName, Customer_BirthDate
    from ORION.CUSTOMERS;
quit;
```

Row	Customer ID	Customer Last Name	Customer Birth Date
1	4	Kvarniq	27JUN1974
2	5	Stephano	09JUL1979
3	9	Krahl	27FEB1974
4	10	Ballinger	18OCT1984
5	11	Wallstab	16AUG1974
6	12	Black	12APR1969
7	13	Sepke	21JUL1988
8	16	Heyde	16JAN1939
9	17	Evans	17AUG1954
10	18	Asmussen	02FEB1954

Previewing the Table

PROC SQL outobs=*n*;

used to specify that the query should return all columns of the queried table(s)

```
proc sql outobs=10;
select *
     from ORION.CUSTOMERS;
quit;
```

Customer ID	Customer Country	Customer Gender	Customer Name	Customer First Name	Customer Last Name	Customer Birth Date	Customer Age Group	Customer Type Name	Customer Group Name	Customer Age
4	US	M	James Kvarniq	James	Kvarniq	27JUN1974	31-45 years	Orion Club members low activity	Orion Club members	33
5	US	F	Sandrina Stephano	Sandrina	Stephano	09JUL1979	15-30 years	Orion Club Gold members medium activity	Orion Club Gold members	28
9	DE	F	Cornelia Krahl	Cornelia	Krahl	27FEB1974	31-45 years	Orion Club Gold members medium activity	Orion Club Gold members	33
10	US	F	Karen Ballinger	Karen	Ballinger	18OCT1984	15-30 years	Orion Club members high activity	Orion Club members	23
11	DE	F	Elke Wallstab	Elke	Wallstab	16AUG1974	31-45 years	Orion Club members high activity	Orion Club members	33
12	US	M	David Black	David	Black	12APR1969	31-45 years	Orion Club members medium activity	Orion Club members	38
13	DE	M	Markus Sepke	Markus	Sepke	21JUL1988	15-30 years	Orion Club Gold members low activity	Orion Club Gold members	19
16	DE	M	Ulrich Heyde	Ulrich	Heyde	16JAN1939	61-75 years	Internet/Catalog Customers	Internet/Catalog Customers	68
17	US	M	Jimmie Evans	Jimmie	Evans	17AUG1954	46-60 years	Orion Club members medium activity	Orion Club members	53
18	US	M	Tonie Asmussen	Tonie	Asmussen	02FEB1954	46-60 years	Orion Club members low activity	Orion Club members	53

Previewing the Table

PROC SQL outobs=*n*;

used to specify the columns you want to drop

```
proc sql outobs=10;
select *
     from ORION.CUSTOMERS (DROP = Customer_FirstName
                           Customer_LastName Customer_Age
                           Customer_Age_Group Customer_Type);
quit;
```

Customer ID	Customer Country	Customer Gender	Customer Name	Customer Birth Date	Customer Group Name
4	US	M	James Kvarniq	27JUN1974	Orion Club members
5	US	F	Sandrina Stephano	09JUL1979	Orion Club Gold members
9	DE	F	Cornelia Krahl	27FEB1974	Orion Club Gold members
10	US	F	Karen Ballinger	18OCT1984	Orion Club members
11	DE	F	Elke Wallstab	16AUG1974	Orion Club members
12	US	M	David Black	12APR1969	Orion Club members
13	DE	M	Markus Sepke	21JUL1988	Orion Club Gold members
16	DE	M	Ulrich Heyde	16JAN1939	Internet/Catalog Customers
17	US	M	Jimmie Evans	17AUG1954	Orion Club members
18	US	M	Tonie Asmussen	02FEB1954	Orion Club members

Syntax Summary

PROC SQL <*options*>;
SELECT *col-name, col-name*
FROM *input-table*;
QUIT;

SELECT Statement

DESCRIBE TABLE *table-name*;

Explore a Table

input-table (**OBS**=*n*)

Data Set Options

PROC SQL INOBS=*n* **OUTOBS**=*n* **NUMBER**;

PROC SQL Options