Ask the Expert

SAS/ACCESS: Universal SAS Methods to Access just about any data, anywhere

Ask the Expert

SAS/ACCESS: Universal SAS Methods to Access just about any data, anywhere

Presenter: David Ghan

Principal Technical Training Consultant, SAS Education

Q&A: Kathy Kiraly

Principal Curriculum Consultant, SAS Education

Get Connected with SAS

Follow us on Twitter
@SASSoftware

Like SAS on Facebook
SAS Software, SASUsersgroup

Subscribe to SAS on YouTube
SASSoftware

Join on LinkedIn
SAS, SAS Users Group

Join SAS Communities
communities.sas.com

Read SAS Blogs
blogs.sas.com/content

Please allow 10 days for post webinar assets to surface in the SAS Communities at:
communities.sas.com/ask-the-expert

SAVE THE DATE

Analytics Experience 2018

sas

Sept. 17 - 19 | Manchester Grand Hyatt | San Diego

sas.com/analyticsx

ANALYTICS EXPERIENCE

Learn SAS with SAS Education

SAS Education will support you in continual learning to grow your career.

- SAS Training Courses
  support.sas.com/training

- Get SAS Certified
  support.sas.com/certify

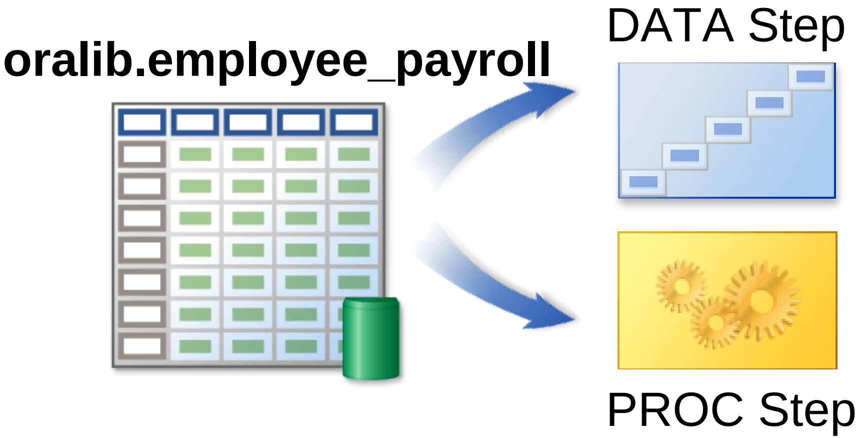
- SAS Books
  support.sas.com/books

Contact SAS Training Customer Service
(800) 727-0025 or training@sas.com

SAS/ACCESS: Universal SAS Methods to Access Just About Any Data, Anywhere

SAS/ACCESS Software

- SAS/ACCESS enables you to read data from other database management systems from within SAS.



oralib.employee_payroll

DATA Step

PROC Step

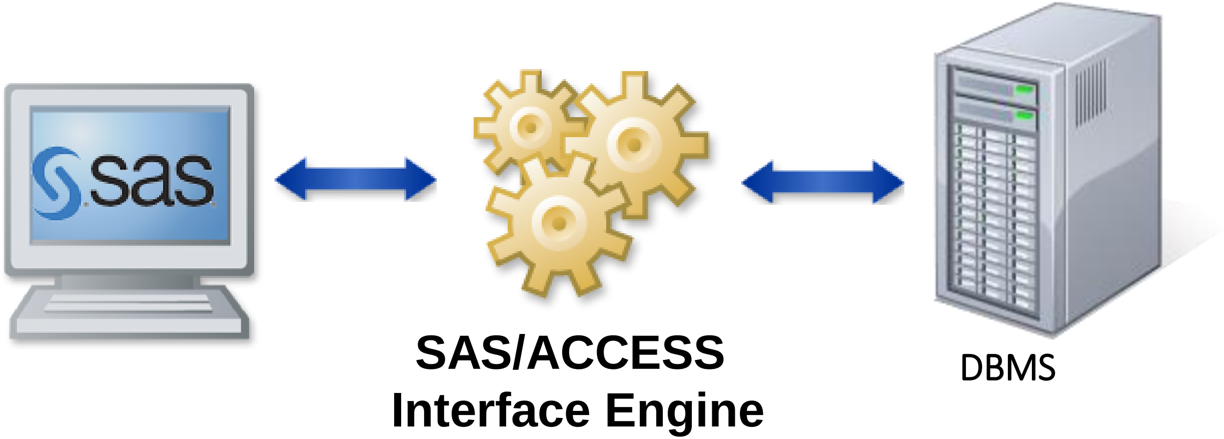
- If you have appropriate authority, SAS/ACCESS enables you to also write to other database management systems.

Database Management Systems

- Database management systems (DBMS) typically use Structured Query Language (SQL) as the interface to access and manage database tables.
- The SAS/ACCESS interface engines use the SQL language to communicate with the database tables.

Interface Engine

- The SAS/ACCESS interface engines
  - are software technologies that transfer data between DBMS and SAS
  - provide transparent Read and Write capabilities.



Two Types of SAS/ACCESS Methods

SQL pass-through	A SAS programmer uses the SQL procedure in a SAS session to write and submit native database SQL to the database.
SAS/ACCESS LIBNAME	A SAS programmer uses a LIBNAME statement to connect to the DBMS. DBMS tables can be named in a SAS program wherever SAS data sets can be named. SAS implicitly converts the SAS code into native database SQL statements.

Examples of the SAS/ACCESS Methods

SQL pass-through

```
proc sql;
connect to dbms (<dbms connection parameters>);
select * from connection to dbms
        (select state, avg(salary) as avsal
            from hivetable
                group by state);
disconnect from dbms;
quit;
```

SAS/ACCESS LIBNAME

```
libname mydbms dbms <dbms connection parameters>;
proc means data = mydbms.dbms_table;
    class state;
    var salary;
run;
```

Examples of the SAS/ACCESS Methods

SQL pass-through

```
proc sql;
connect to dbms (<dbms connection parameters>);
select * from connection to dbms
        (select state, avg(salary) as avsal
           from hivetable
               group by state);
disconnect from dbms;
quit;
```

native database SQL executed by the DBMS and results returned to SAS

SAS/ACCESS LIBNAME

```
libname mydbms dbms <dbms connection parameters>;
proc means data = mydbms.dbms_table;
    class state;
    var salary;
run;
```

SAS converts to a native database SQL summary query executed by the DBMS and results returned to SAS

Components of the Pass-Through Facility

- Here are two types of SQL pass-through components that you can submit to the DBMS:
  - SELECT statements
  - EXECUTE statements

Executing Pass-Through Statements

- Example of an SQL pass-through SELECT statement:

```
proc sql;
connect to hadoop (server="server2" port=10000 schema=DIACHD
                   user='student' passwd='Metadata0');
select * from connection to hadoop
       (select employee_name,salary
          from salesstaff
          where salary > 50000);
disconnect from hadoop;
quit;
```

Executing Pass-Through Statements

- Example of an SQL pass-through EXECUTE statement:

```
proc sql;
connect to hadoop (server="server2" port=10000 schema=DIACHD
                   user='student' passwd='Metadata0');

execute (drop table salesstaff) by hadoop;
disconnect from hadoop;
quit;
```

Universal Methodology

- SAS/ACCESS interfaces enable SAS programmers to apply ***consistent*** techniques to access a large number of data sources in different formats in a ***consistent*** way.

Using SAS/ACCESS Methods for a Variety of Data Sources

This demonstration illustrates how similar SAS code can be used to query and process data from a variety of different data sources.

SQL Pass-Through Method versus LIBNAME Method

Method	Characteristics
SQL Pass-Through	Explicit control of what executes in the DBMS versus SAS.
	Must construct native DBMS SQL
LIBNAME	Can use any SAS programming methods and name DBMS tables as input or output data sets.
	Implicitly generated SQL might cause all data from DBMS to be returned to SAS. Should turn on SASTRACE and examine the SAS log during development to produce efficient code.

Using the SASTRACE Option with the SAS/ACCESS LIBNAME Method

This demonstration illustrates how to use the SASTRACE system option to evaluate the efficiency of the SAS/ACCESS LIBNAME method when accessing DBMS tables.

SAS/ACCESS Documentation

This demonstration shows you where to find the SAS Online Documentation for SAS/ACCESS, how to use it to discover what ACCESS engines are available, and how to navigate the documentation to find the information that you need.

support.sas.com/documentation/onlinedoc/access/index.html

Documentation and Training

support.sas.com/documentation/onlinedoc/access/index.html

support.sas.com/training/us/paths/dmgt.html#acc