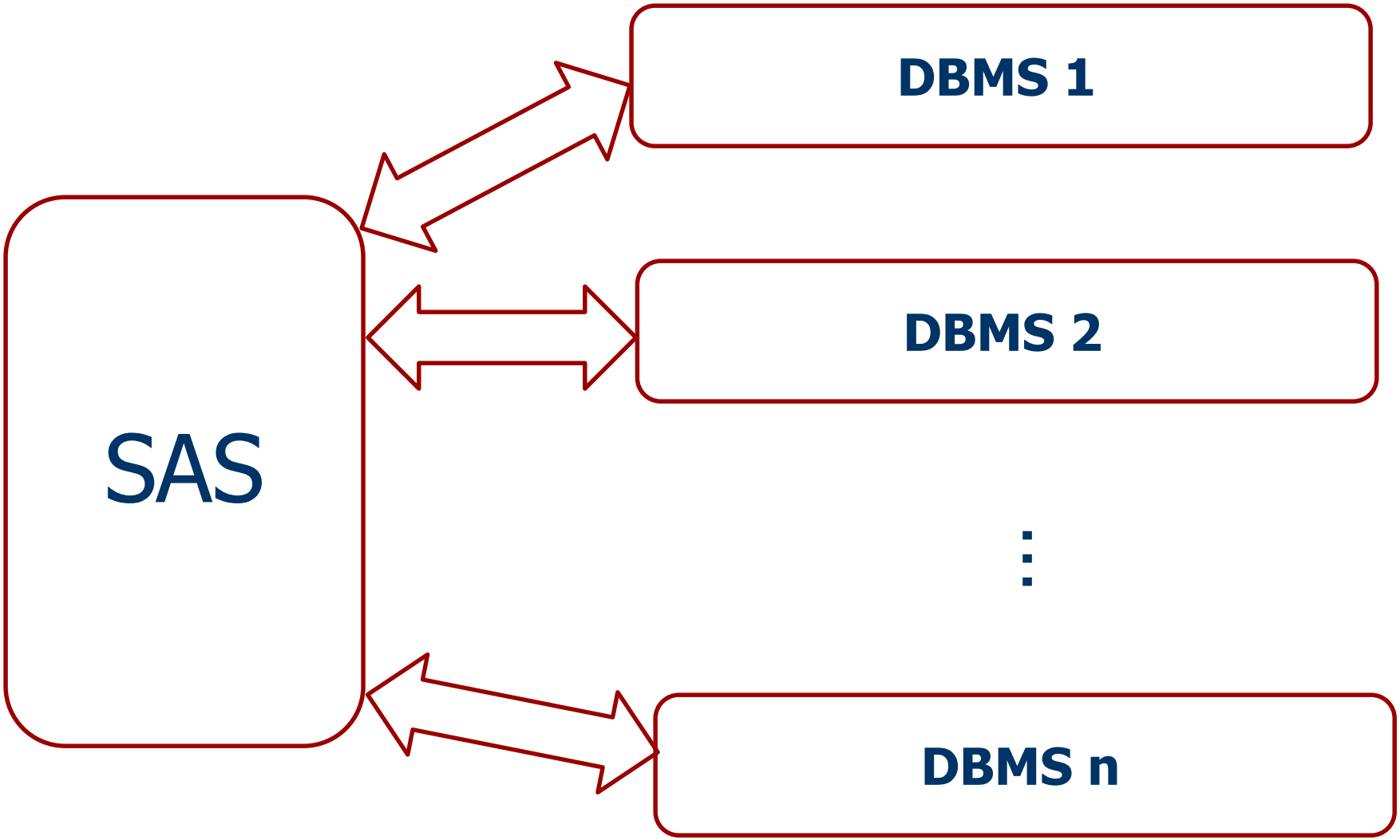
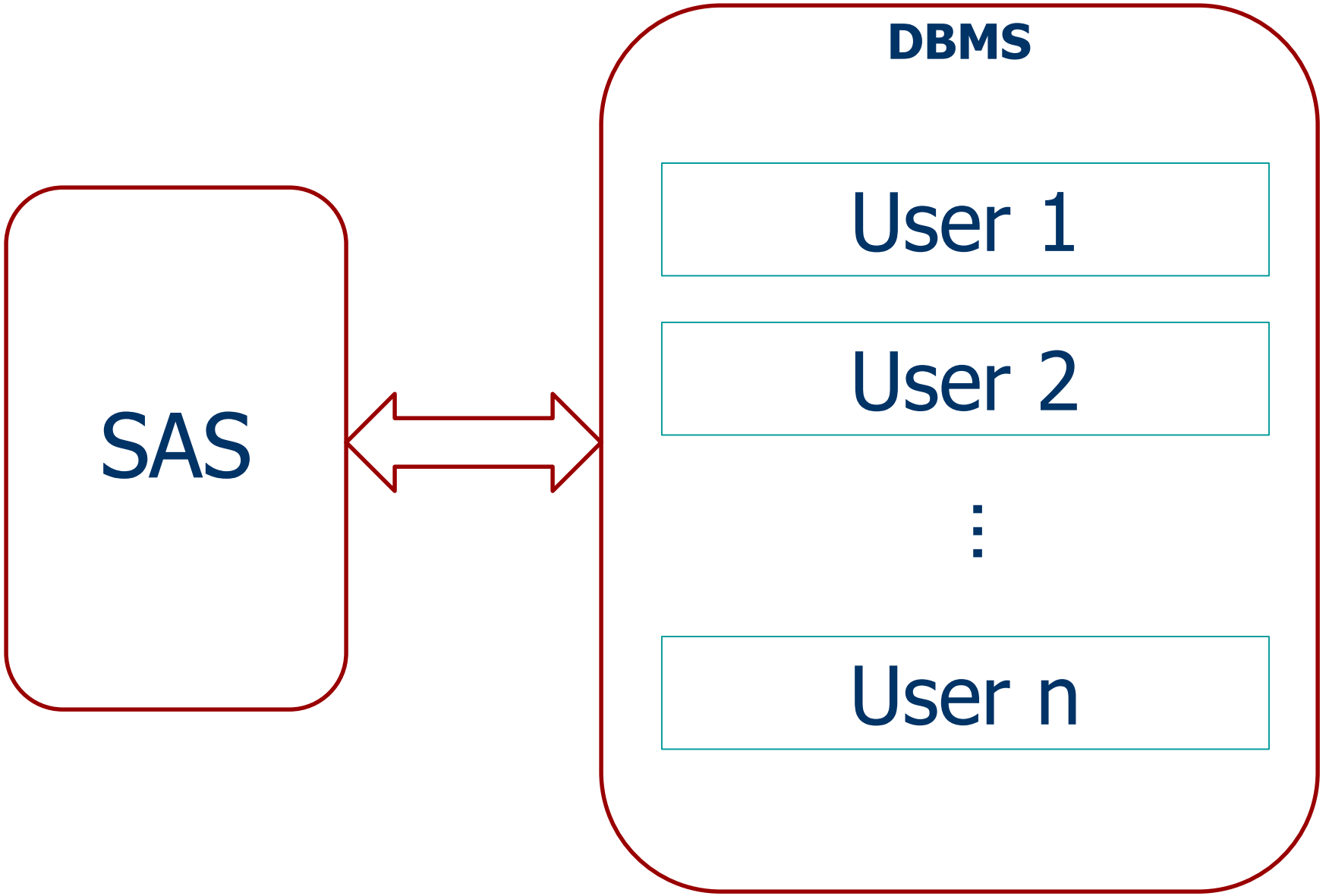
Last Comments about SAS HPC

- It's a new and evolving topic.
- You need to meet both software and hardware requirements.
- The SAS system can make use of the available capability when possible.
- Associating SAS with a DBMS can improve computing efficiency through the LIBNAME statement and SQL Pass-through Facility. This approach can take advantage of DBMS' parallel computing capabilities. ACCESS and DBLOAD procedures are not recommended but still functional.
- Using In-Database Processing for the supported SAS procedures, especially when you are handling a big amount of data.
- Optimizing your code and/or processing plan can also help improve the efficiency, e.g., sorting outside a DBMS, using indexes and (materialized) views, avoiding correlated subqueries when possible.

Associating SAS with Multiple DBMSs



Associating SAS with multiple users



Additional Notes

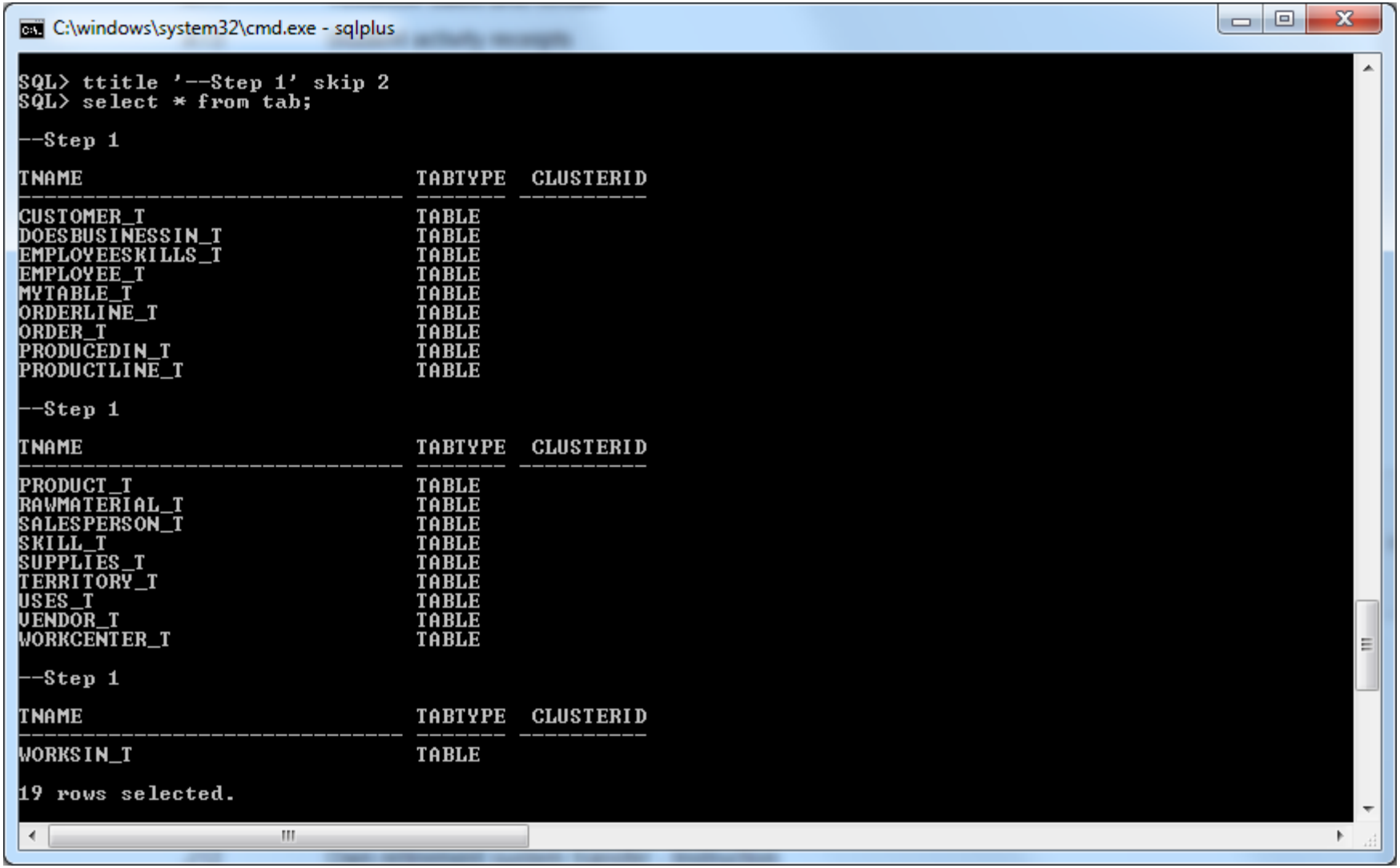
Rename Columns of a SAS Dataset

```
proc sql;
    create table new_table_name
    (rename=(old_name1 = new_name1 old_name2 =
    new_name2 …)) as select * from old_table_name;
quit;
```

Rename Columns of an Oracle Table

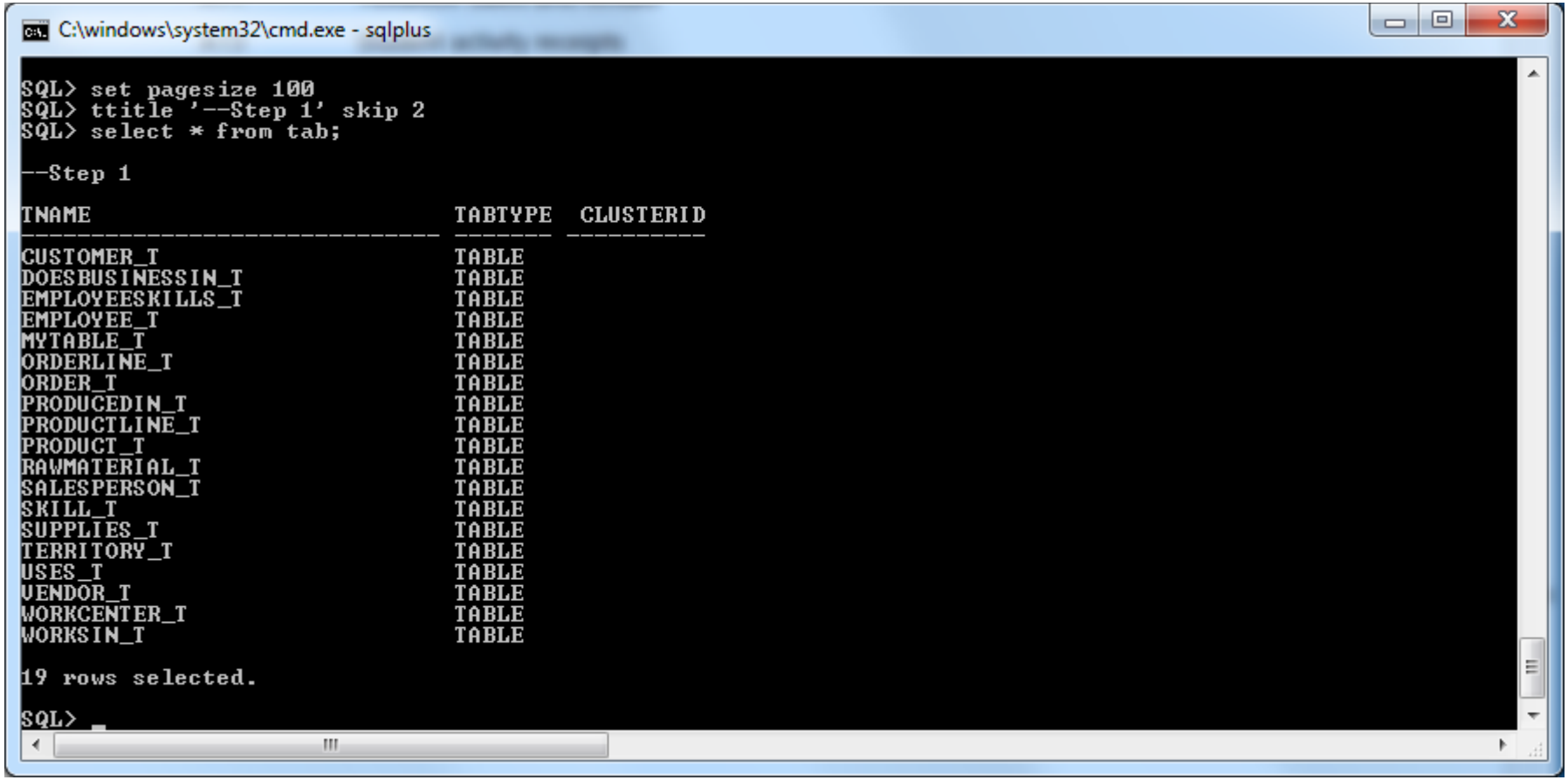
```
alter table  table_name
        rename column
          old_column_name
        TO
          new_column_name;
```

Add a Title to an SQL*PLUS Output Page



Add a Title to an SQL*PLUS Output Page

Set a new page size so that the title just appears once per physical page.



Practice

You will be using Oracle command line interface, Oracle SQLDEVELOPER, and SAS 9.4 to do this practice. Save all your code and outputs to appropriate file formats (use .txt or .sql or an MS Word file if it is a table or screenshot, etc.). For how to save your files in the Oracle command line interface, refer to early lab session instructions.

1. In your old Oracle account you created and have been using, create and populate a large DBMS table called Rawmaterial_t by running the script "Mariterials.sql" available on the course website.
2. Check the number of rows inserted into the Rawmaterial_t table by running query containing a COUNT() function. Do this through the SAS LIBNAME statement method. You just report the number, not the contents of the table.
3. In SAS 9.4, create a SAS dataset, Rawmaterial, in the SASUSER library containing the same contents as the Rawmaterials_t table in your Oracle database using the LIBNAME statement and a PROC SQL procedure. Then use a PROC PRINT procedure to display your result. Only the first 10 rows of the dataset should be reported.
4. In the Oracle command line interface, create a new account or user, **newacnt**, and grant all the privileges to this new account.

Practice, cont.

5. In SAS 9.4, associate SAS with the new Oracle account (newacnt) by creating a libname, **newlib**, using the LIBNAME statement. Then, from within SAS create a table called "**Expensiveoak_t**" which is saved in the Oracle account newacnt by using a PROC SQL procedure through querying the Rawmaterial dataset in SASUSER. The Expensiveoak_t table contains all the Oak materials whose standard prices are greater than $500. Display Expensiveoak_t in SQLDEVELOPER and submit a screenshot. You are required to create a new connection named **newacnt** in the Connections panel for this practice.
6. Use in-database processing and PROC SORT to sort the Rawmaterials_t by materialstandardprice and output your result to a dataset in the Work library. Use PROC PRINT to display the table contents. Only report the first 10 rows of the dataset.