PROC SQL (structured query language—usually pronounced *sequel*)

offers an alternative to the DATA step for querying and combining SAS data sets. There are some tasks that PROC SQL can perform much better and easier than the DATA step. Other tasks may be easier or more efficient using a DATA step. The best advice is to learn both and use the tool that works best for you in each situation.

This section touches only on the basics of PROC SQL.

The following table lists SAS terms and the corresponding SQL terms:

SAS Term	SQL Equivalent
Data set	Table
Observation	Row
Variable	Column

SELECT Statement

The SELECT statement is the primary tool of PROC SQL. You can use it to retrieve data from a table. You can use several optional clauses within the SELECT statement to place restrictions on a query.

When you construct a SELECT statement, you must specify the clauses in the following **order**:

1. SELECT
2. FROM
3. WHERE
4. GROUP BY
5. HAVING
6. ORDER BY

Note: Only the SELECT and FROM clauses are required.

Example:

```
PROC SQL;
   CREATE TABLE scoredata0 AS
   SELECT stu_id, gender, name
   FROM score_data
   WHERE gender in ('m');
QUIT;
```

Note:

- To create a PROC SQL table from a query result, use a CREATE TABLE statement with the AS keyword, and place it before the SELECT statement. scoredata0 in this case is the table (data set) that will be created.
- a SELECT clause is where you list the variables you want. Notice that the variables in this list are separated by commas (spaces do not work).
- The clause FROM names the data set you want to read.
- Finally, a WHERE clause describes the particular subset you want.
- SELECT, FROM, and WHERE form a single query, which you end with a single semicolon.
- the query ends with a QUIT statement. You do not need a RUN statement because PROC SQL executes as soon as a complete query has been specified. If you don't include a QUIT statement, PROC SQL remains in memory for another query.

Optional clauses:

ORDER BY Clause

The ORDER BY clause enables you to sort the output from a table by one or more columns.

GROUP BY Clause

The GROUP BY clause enables you to break query results into subsets of rows. When you use the GROUP BY clause, you use an aggregate function in the SELECT clause or a HAVING clause to instruct PROC SQL how to group the data.

HAVING Clause

The HAVING clause works with the GROUP BY clause to restrict the groups in a query's results based on a given condition.