Chapter 1

Performing Queries Using PROC SQL

PROC SQL Basics

- PROC SQL is the SAS implementation of Structured Query Language (SQL)
- SQL is a standardized language and is widely used to retrieve and update data in tables
- PROC SQL can often be used as an alternative to other SAS procedures or the data step

PROC SQL Basics

- The PROC SQL statement does not need to be repeated with each query
- PROC SQL does not require a run statement
- To end PROC SQL, submit another procedure, a data step, or a quit statement.

Uses for SQL

- Retrieve data from tables
- Add or modify data in a table
- Add, modify, or drop columns in a table
- Create tables and views
- Join tables
- Create reports

Basic PROC SQL Step

```
proc sql;
  select column1, column2,…, columnn
  from table1 (or view1)
  <where expression>
<group by column1,…,columnn>
<having expression>
<order by column1,…,columnn>;
```

SELECT Statement

- The SELECT statement follows the PROC SQL statement
- The SELECT statement retrieves and displays data
- The SELECT statement is composed of clauses (e.g., SELECT, FROM, WHERE)
- The SELECT and FROM clauses are required. The remaining clauses are optional.

The Order of Clauses Within the PROC SQL SELECT Statement Matters

- SELECT (So)
- FROM (Few)
- WHERE (Workers)
- GROUP BY (Go)
- HAVING (Home)
- ORDER BY (On-time)

SELECT Clause

- The first clause in the SELECT statement
- Specifies the column(s) to be selected from the table
- Each column listed must be separated by a comma
- A label can be specified for a column in the SELECT clause

SELECT Clause

- New columns (i.e., columns that are not in the table) can be created with the SELECT clause
- The new columns may contain text or a calculation
- New columns will appear in the output but will not be kept unless a table or view is created

FROM Clause

- The FROM clause specifies which table(s) or view(s) to be queried

2011 SC Baseball Stats (bbstats)

Player	At Bats	Hits	BB
Christian Walker	271	97	36
Scott Wingo	240	81	44
Brady Thomas	231	73	23
Evan Marzilli	220	64	25
Robert Beary	211	61	12
Adrian Morales	249	70	30
Peter Mooney	254	71	44
Jake Williams	209	56	21
Jackie Bradley Jr.	162	40	22

Example – Select and From Clauses

```
proc sql;
  select player,
   atbats
  from bbstats;
quit;
```

player	atbats
Christian Walker	271
Scott Wingo	240
Brady Thomas	231
Evan Marzilli	220
Robert Beary	211
Adrian Morales	249
Peter Mooney	254
Jake Williams	209
Jackie Bradley Jr.	162

Example – Select and From Clauses

```
proc sql;
  select player,
   hits/atbats as
   avg
  from bbstats;
quit;
```

player	avg
Christian Walker	.358
Scott Wingo	.338
Brady Thomas	.316
Evan Marzilli	.291
Robert Beary	.289
Adrian Morales	.281
Peter Mooney	.280
Jake Williams	.268
Jackie Bradley Jr.	.247

WHERE Clause

- The WHERE clause subsets the data from the table(s) based on one or more conditions
- The WHERE clause can be any valid SAS expression
- The columns specified in the WHERE clause do not have to appear in the SELECT clause

Example – Where Clause

```
proc sql;
  select player,
   atbats
  from bbstats
  where atbats >
   230;
quit;
```

player	atbats
Christian Walker	271
Scott Wingo	240
Brady Thomas	231
Adrian Morales	249
Peter Mooney	254

HAVING Clause

- Works with the GROUP BY clause to restrict groups that are displayed in the output, based on one or more conditions
- Discussed in more detail in Chapter 2

ORDER BY Clause

- ORDER BY sorts rows by the values of a specified column or specified columns. Each column must be separated by a comma
- Descending sort – specify the keyword DESC following the column name
- In the ORDER BY clause, you can reference the column by its name or the position in the SELECT clause. Use an integer to indicate the column's position.

Example – Order By Clause

```
proc sql;
  select player,
   atbats
  from bbstats
  order by atbats
   desc;
quit;
```

player	atbats
Christian Walker	271
Peter Mooney	254
Adrian Morales	249
Scott Wingo	240
Brady Thomas	231
Evan Marzilli	220
Robert Beary	211
Jake Williams	209
Jackie Bradley Jr.	162

Example – Order By Clause

```
proc sql;
  select player,
   atbats
  from bbstats
  order by 2 desc;
quit;
```

This example uses the column's position in the SELECT clause in the ORDER by clause

player	atbats
Christian Walker	271
Peter Mooney	254
Adrian Morales	249
Scott Wingo	240
Brady Thomas	231
Evan Marzilli	220
Robert Beary	211
Jake Williams	209
Jackie Bradley Jr.	162

CREATE TABLE Statement

- Use the CREATE TABLE statement to create a new table from the results of a query

PROC SQL Step for Creating a Table from a Query Result

```
proc sql;
  create table table-name as
  select column1, column2,…, columnn
  from table1 (or view1)
  <where expression>
<group by column1,…,columnn>
<order by column1,…,columnn>;
```