SQL Processing with SAS® Tip Sheet

This tip sheet is associated with the SAS® Certified Professional Prep Guide Advanced Programming Using SAS® 9.4. For more information, visit www.sas.com/certify

Basic Queries

PROC SQL <*options*>;
  SELECT *column-1* <, ...*column-n*>
    FROM *input-table*
    <**WHERE** *expression*>
    <**GROUP BY** *col-name*>
    <**HAVING** *expression*>
    <**ORDER BY** *col-name*> <**DESC**> <,...*col-name*>;

SQL Query Order of Execution:

Clause	Description
SELECT	Retrieve data from a table
FROM	Choose and join tables
WHERE	Filter the data
GROUP BY	Aggregate the data
HAVING	Filter the aggregate data
ORDER BY	Sort the final data

Managing Tables

CREATE TABLE	**CREATE TABLE** *table-name* (*column-specification-1*<, ...*column-specification-n*>);
DESCRIBE TABLE	**DESCRIBE TABLE** *table-name-1* <,...*table-name-n*>;
DROP TABLE	**DROP TABLE** *table-name-1* <,...*table-name-n*>;

Managing Views

CREATE VIEW	**CREATE VIEW** *table-name* **AS** *query*;
DESCRIBE VIEW	**DESCRIBE VIEW** *view-name-1* <,...*view-name-n*>;
DROP VIEW	**DROP VIEW** *view-name-1* <,...*view-name-n*>;

Modifying Columns

LABEL=	**SELECT** *col-name* **LABEL=***'column label'*
FORMAT=	**SELECT** *col-name* **FORMAT=***format.*
Creating a new column	**SELECT** *col-name* **AS** *new-col-name*
Filtering new columns	**WHERE CALCULATED** *new-col-name*

Modifying Rows

Inserting rows into tables	**INSERT INTO** *table* **SET** *column-name=value* <, ...*column-name=value*>; **INSERT INTO** *table* <(*column-list*)> **VALUES** (*value*<,...*value*>); **INSERT INTO** *table* <(*column-list*)> **SELECT** *column-1*<,...*column-n*> **FROM** *input-table*;
Eliminating duplicate rows	**SELECT DISTINCT** *col-name*<,...*col-name*>
Filtering rows	**WHERE** *col-name* **IN** (*value1*, *value2*, ...) **WHERE** *col-name* **LIKE** *"_string%"* **WHERE** *col-name* **BETWEEN** *value* **AND** *value* **WHERE** *col-name* **IS NULL** **WHERE** *date-value* *"<01JAN2019>"***d** **WHERE** *time-value* *"<14:45:35>"***t** **WHERE** *datetime-value* *"<01JAN201914:45:35>"***dt**

Remerging Summary Statistics

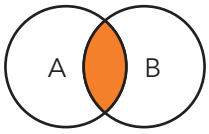
SELECT *col-name*, *summary function*(*argument*)
  FROM *input table*;

SQL Processing with SAS® Tip Sheet

This tip sheet is associated with the SAS® Certified Professional Prep Guide Advanced Programming Using SAS® 9.4. For more information, visit www.sas.com/certify

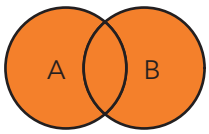
Joins Summary

Inner Join



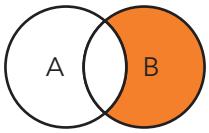
SELECT *<list>*
FROM *table-A* **INNER JOIN** *table-B*
ON A.*Key*=B.*Key*;

Full Join



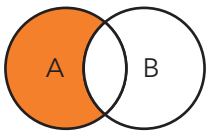
SELECT *<list>*
FROM *table-A* **FULL JOIN** *table-B*
ON A.*Key*=B.*Key*;

Right Join



SELECT *<list>*
FROM *table-A* **RIGHT JOIN** *table-B*
ON A.*Key*=B.*Key*;

Left Join



SELECT *<list>*
FROM *table-A* **LEFT JOIN** *table-B*
ON A.*Key*=B.*Key*;

Creating Macro Variables

Storing a value in a macro variable using SQL:

SELECT *col-name-1 <,...col-name-n>*
INTO:*macrvar_1<,...macvar-n>*
FROM *input-table*;

Storing a list of values in a macro variable using SQL:

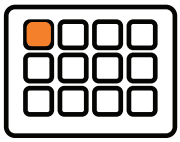
SELECT *col-name-1 <,...col-name-n>*
INTO:*macrvar_1* **SEPARATED BY** *'delimiter'*
FROM *input-table*;

Viewing the value of the macro variable in the SAS Log:

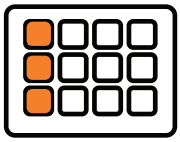
%PUT &=*macvar*;

Subqueries

SELECT *col-name*,
(**SELECT** *function*(*argument*)
FROM *input-table*)
FROM *input-table*;

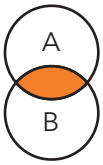


SELECT *col-name*, *<,...col-name>*
FROM *input-table*
WHERE *col-name*
(**SELECT** *function*(*argument*)
FROM *input-table*)



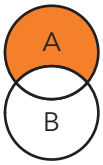
Set Operators

The INTERSECT operator selects unique rows that are common to both tables.



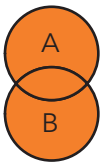
SELECT *<list>*
FROM *table-A* **INTERSECT**
SELECT *<list>*
FROM *table-B*;

The EXCEPT operator selects unique rows from table A that are not found in table B.



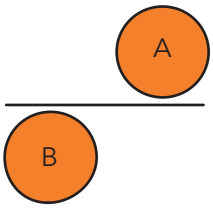
SELECT *<list>*
FROM *table-A* **EXCEPT**
SELECT *<list>*
FROM *table-B*;

The UNION operator selects unique rows from both tables.



SELECT *<list>*
FROM *table-A* **UNION**
SELECT *<list>*
FROM *table-B*;

The OUTER UNION operator selects all rows from both tables.



SELECT *<list>*
FROM *table-A* **OUTER UNION**
SELECT *<list>*
FROM *table-B*;

Accessing DBMS Data

The SQL pass-through facility enables you to code in the native DBMS SQL syntax and pass the query to the database.

PROC SQL;
CONNECT TO *DBMS-name* <**AS** *alias*>
(*DBMS-connection-options*);
SELECT *col-name*
FROM CONNECTION TO *DBMS-name|alias* (*dbms-query*);
DISCONNECT FROM *DBMS-name|alias*;
QUIT;