SAS System Options

```
OPTIONS SQLREMERGE|NOSQLREMERGE;
OPTIONS MSGLEVEL=N|I;
OPTIONS NOFULLSTIMER|FULLSTIMER;
OPTIONS NOSYMBOLGEN|SYMBOLGEN;
OPTIONS NOSTIMER|STIMER;
```

LIBNAME Statement

```
LIBNAME libref 'SAS-data-library' <options>;
```

Querying Tables

```
SELECT column-1<, ...column-n>
    FROM table-1|view-1
        <, ...table-n|view-n>
    <WHERE expression>
    <GROUP BY column-1<, …column-n>
    <HAVING expression>
    <ORDER BY column-1<DESC><,...column-2>>;
```

Formatting Data and Variable Names

```
SELECT column-1 label='text'|'text'
        format=format.
        <,column-2 'label text' format=format., ...>
        <other clauses>;
```

PROC SQL Options

```
PROC SQL EXEC|NOEXEC INOBS=n OUTOBS=n
         FEEDBACK PRINT|NOPRINT
         NOSTIMER|STIMER NOREMERGE
         NOERRORSTOP|ERRORSTOP
         NONUMBER|NUMBER
         FLOW<=n <m>>|NOFLOW
         NODOUBLE|DOUBLE;
```

RESET Statement

```
RESET option(s);
```

Displaying Table Information

```
DESCRIBE TABLE table-name <, …table-name>;
```

Conditional Processing in the SQL Step

```
SELECT column-1<, ...column-n >
  CASE <case-operand>
    WHEN when-condition THEN result-expression
   <WHEN when-condition THEN result-expression>
   <ELSE result-expression>
  END <as column>
```

Subsetting Data

```
WHERE expression
```

Sorting Data

```
PROC SQL <options>;
SELECT column-1<, ...column-2>
    FROM table|view
    <other clauses>
    ORDER BY column-1<DESC><,...column-2>;
QUIT;
```

SAS Date Functions

```
MONTH(date)
TODAY( )
```

SAS Character Functions

```
CATX(delimiter,argument-1,argument-2<, ..argument-n>)
UPCASE(argument)
PROPCASE(argument<,delimiters>)
SCAN(string,n<,charlist><,modifier(s)>)
STRIP(argument)
FIND(string, substring<, modifier(s)> <, start position>)
PUT(source,format)
```

ANSI Summary Functions

```
AVG(argument-1<, …argument-n >)
COUNT(*|argument)
COALESCE(argument-1,argument-2<, …argument-n >)
```

SAS Numeric Functions

```
FREQ(argument)
INT(argument)
MAX(argument-1<, …argument-n>)
MEAN(argument-1<, …argument-n>)
MIN(argument-1<, …argument-n>)
N(argrument)
NMISS(argument)
SUM(argument-1<, …argument-n>)
STD(argument-1<, …argument-n>)
VAR(argument-1<, …argument-n>)
```

Inner Joins

```
SELECT column-1<, ...column-n>
    FROM table-1|view-1 <, ... table-n|view-n>
    WHERE join-condition(s)
          <AND other subsetting conditions>
          <other clauses>;
SELECT column-1<,  ...column-n>
    FROM table-1
    INNER JOIN
          table-2
    ON join-condition(s)
        <other clauses>;
QUIT;
```

Outer Joins

```
SELECT column-1<, ...column-n>
      FROM table-1
      LEFT|RIGHT|FULL JOIN
            table-2
      ON join-condition(s)
          <other clauses>;
```

SQL Set Operators

```
SELECT *
      FROM table-1
EXCEPT|INTERSECT|UNION <CORR> <ALL>
SELECT *
      FROM table-2
 <other clauses>;

SELECT *
      FROM table-1
OUTER UNION <CORR>
SELECT *
      FROM table-2
<other clauses>;
```

Creating Tables

```
CREATE TABLE table-name AS
      query-expression;

CREATE TABLE table-name
      (column-name type(length)
      <, ...column-name type(length) >);

CREATE TABLE table-name-2
      LIKE table-name-1;
```

Creating Views

```
CREATE VIEW view-name AS
     query-expression
     <USING LIBNAME-clause<,…LIBNAME-clause>>;
QUIT;
```

Displaying View Information

```
DESCRIBE VIEW proc-sql-view<,…proc-sql-view>;
```

Creating Indexes

```
CREATE <UNIQUE> INDEX index-name
      ON table-name (column-name<, …column-name>);
```

SAS Data Set Options

```
SAS-data-set(IDXWHERE=YES|NO)
SAS-data-set(IDXNAME=<name>)
```

Adding Data to a Table

```
PROC SQL;
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-2>
            FROM table;
QUIT;
```

Modifying Rows in an Existing Table

```
UPDATE table-name
      SET column-name=expression
            <, ...column-name=expression>
      WHERE expression;
```

Deleting Rows from a Table or View

```
DELETE FROM table|view
      WHERE expression;
```

Adding, Dropping, and Modifying Columns in Tables

```
ALTER TABLE table-name
      ADD column-definition <, ...column-definition>
      DROP column-1<, ...column-2>
      MODIFY column-definition <, ...column-
definition>;
```

Deleting Tables, Indexes, and Views

```
DROP TABLE table-name<, ...table-name>;
DROP VIEW view-name<, ...view-name>;
DROP INDEX index-name<, ...index-name>
      FROM table-name;
```

Creating Macro Variables

```
SELECT column-1<, ...column-n>
      INTO :macvar_1<, ... :macvar_n>
      FROM table|view
      <other clauses>;
SELECT column-1<,... column-2>
      INTO : macvar_a1-: macvar_an
          <, : macvar_b1-: macvar_bn>
      FROM table|view
      <other clauses>;
SELECT column-1<, ...column-2 >
      INTO :macvar_1 SEPARATED BY 'delimiter'
          <,:macvar_2 SEPARATED BY 'delimiter'>
      FROM table|view
      <other clauses>;
```