Proc SQL: An Overview

Carlo R. Cosenza

carlo@carlocosenza.com

Department of Statistics

California State University, East Bay

Hayward, CA, USA

Goals

Demonstrate the practical use of Proc SQL.

- Setup Database Schema
- Add Constraints
- Add Data
- Validate Data
- Extract Some Information

Database Schema

In a relational database, the schema defines the tables, the fields in each table, and the relationships between fields and tables.

- Department Table
- Employee Table
- Composite Manager Table

Creating Tables

Listing 1: Creating the Employee Table

```
/* Create Employee Table
   Employee Record Includes
       Unique ID (Primary Key)
       Last Name
       First Name
       Dept Number
*/
PROC SQL;
CREATE TABLE EMP_TABLE
    (EMP_ID NUM LABEL='Employee ID' UNIQUE,
         LAST_NAME CHAR(30) LABEL='Last Name' NOT NULL,
         FIRST_NAME CHAR(20) LABEL='First Name',
         DEPT_ID NUM LABEL='Department ID');
QUIT;
```

Creating Tables (cont)

Listing 2: Creating the Department Table

```
/* Create Department Table
   Department Record Includes
       Unique ID (Primary Key)
       Department Name
*/
PROC SQL;
CREATE TABLE DEPT_TABLE
    (DEPT_ID NUM LABEL='Department ID' UNIQUE,
     DEPT_NAME CHAR(20) LABEL='Department Name');
QUIT;
```

Creating Tables (cont)

Listing 3: Creating Management Table

```
/* Create Manager Table
   Department Manager Table Includes
       Unique Index (Primary Key)
       Dept Number (Foreign Key -- dept_table)
       Emp Number (Foreign Key -- emp_table)
*/
PROC SQL;
CREATE TABLE MGR_TABLE
        (EMP_ID NUM LABEL='Manager ID' NOT NULL,
         DEPT_ID NUM LABEL='Department ID' NOT NULL);
QUIT;

PROC SQL;
    CREATE INDEX MGR_INDEX ON MGR_TABLE(EMP_ID,DEPT_ID);
QUIT;
```

Take a Peek

Listing 4: Employee Table

```
/* take a peek at the employee table
   notice: constraints.
   keep the unique constraints in the back of your head.
*/

proc contents data=emp_table;
run;
```

Define Primary Keys I

Listing 5: ID Uniquely Identifies Employee

```
/*
   Define a Primary Key to index Employee Table
   Note: the Primary Key constraint overrides the Unique Constraint
         for a given field.
*/
PROC SQL;
   ALTER TABLE EMP_TABLE
        ADD CONSTRAINT PRIM_EMP_KEY PRIMARY KEY (EMP_ID);
QUIT;
```

Define Primary Keys II

Listing 6: ID Uniquely Identifies Employee

```
/* Establish DEPT_ID as Primary Key */
/* Notice that Primary Key overrides Unique in Constraints */
PROC SQL;
  ALTER TABLE DEPT_TABLE
      ADD CONSTRAINT PRIM_DEPT_KEY PRIMARY KEY (DEPT_ID);
QUIT;
```

Define Foreign Keys I

Listing 7: All Managers are Employees

```
/* Define Foreign Keys
   Note: Need to remove constraints before deleting tables.
         Keep an eye on cascade. */
PROC SQL;
  ALTER TABLE MGR_TABLE
       ADD CONSTRAINT FOREIGN_EMP_KEY FOREIGN KEY (EMP_ID)
         REFERENCES EMP_TABLE
           ON DELETE RESTRICT
           ON UPDATE CASCADE;
QUIT;
```

Define Foreign Keys II

Listing 8: Managers Manage Departments

```
/* Define Foreign Keys
   Note: Need to remove constraints before deleting tables.
         Keep an eye on cascade. */
PROC SQL;
  ALTER TABLE MGR_TABLE
       ADD CONSTRAINT FOREIGN_DEPT_KEY FOREIGN KEY (DEPT_ID)
         REFERENCES DEPT_TABLE
           ON DELETE RESTRICT
           ON UPDATE CASCADE;
QUIT;
```

Define Foreign Keys III

Listing 9: Every Employee Belongs to a Dept

```
/*
   Every Employee Belongs to a Dept
   Note: the Primary Key constraint overrides the Unique Constraint
         for a given field.
*/
PROC SQL;
   ALTER TABLE EMP_TABLE
        ADD CONSTRAINT FOREIGN_EMP_DEPT_KEY FOREIGN KEY (DEPT_ID)
          REFERENCES DEPT_TABLE
            ON DELETE RESTRICT
            ON UPDATE CASCADE;
QUIT;
```

Take Another Peek

Listing 10: Describe Manager Table

```
/* use the describe function.
   note: it goes straight to the log. */

proc sql;
    DESCRIBE TABLE mgr_table;
quit;
```

Add Some Data I

Listing 11: Add Departments

```
/* Add some Departments */
PROC SQL;
  INSERT INTO DEPT_TABLE
    (DEPT_ID, DEPT_NAME)
    VALUES(101, 'Engineering')
    VALUES(102, 'Maintenance')
    VALUES(103, 'Sales')
    VALUES(104, 'Microbiology')
    VALUES(105, 'Quality Assurance');

    SELECT *
        FROM DEPT
          ORDER BY DEPT_ID;

QUIT;
```

Add Some Data II

Listing 12: Add Employees

```
/* Add some Departments */
PROC SQL;
  INSERT INTO EMP_TABLE
    (EMP_ID, LAST_NAME, FIRST_NAME, DEPT_ID)
    VALUES(2001, 'Cosenza', 'Carlo', 101)
    VALUES(2002, 'Johnson', 'Bob', 102)
    VALUES(2003, 'Smith', 'Jack', 103)
    VALUES(2004, 'Lee', 'Mary', 104)
    VALUES(2004, 'Patel', 'Vijay', 105);
    SELECT *
        FROM EMP
          ORDER BY EMP_ID;
QUIT;
```

Add Some Data III

Listing 13: Everybody is a Manager

```
/* Add some Departments */
PROC SQL;
  INSERT INTO MGR_TABLE
    (EMP_ID, DEPT_ID)
    VALUES(2001, 101)
    VALUES(2002, 102)
    VALUES(2003, 103)
    VALUES(2004, 104)
    VALUES(2004, 105);
    SELECT *
        FROM MGR_TABLE
          ORDER BY EMP_ID;
QUIT;
```

Take One Last Peek

Listing 14: Describe Manager Table

```
/* use the describe function.
   note: it goes straight to the log. */

proc sql;
    DESCRIBE TABLE mgr_table;
quit;
```

Deleting Tables

Listing 15: Something is Wrong

```
/*
  Demonstrate that you need to remove constraints
  before deleting tables
*/

PROC SQL;
    DROP TABLE EMP_TABLE;
QUIT;

/* able to delete multiple tables */
/* notice inconsistent use of comma */
PROC SQL;
    DROP TABLE DEPT_TABLE, MGR_TABLE;
QUIT;
```

Deleting Tables (cont)

Listing 16: Need to Remove Foreign Key Constraints

```
/* constraints must be removed before you can delete the tables. */
PROC SQL;
     ALTER TABLE MGR_TABLE
          DROP CONSTRAINT FOREIGN_DEPT_KEY;
QUIT;
PROC SQL;
     ALTER TABLE MGR_TABLE
          DROP CONSTRAINT FOREIGN_EMP_KEY;
QUIT;
PROC SQL;
     ALTER TABLE EMP_TABLE
          DROP CONSTRAINT FOREIGN_EMP_DEPT_KEY;
QUIT;
```

Citations

- Cody, Ron. 2007. *Learning SAS® by Example: A Programmer's Guide*. Cary, NC: SAS Institute Inc.
- Lafler, Kirk Paul. 2004. *Proc SQL: Beyond the Basics Using SAS®*. Cary, NC: SAS Institute Inc.