ALVAREZ
College of Business

DA 6223 Data Analytics Tools and Techniques

Week 6 Meeting Agenda

I. PROC SQL Fundamentals - Generating Simple Reports

Summarizing and Grouping Data

a) Eliminating Duplicate Rows

Activity 07: Orion Star would like to see a list of all unique products. First, import the products dataset, then use a proc SQL step to report columns of the products table in the WORK library. Next, report only the unique product lines (use the distinct keyword). Then, print all unique combinations of product lines and product categories.

b) Summary Functions: Down a Column
c) Summary Functions: Across a Row
d) Summarizing Data Using the COUNT Function

Activity 08: Summarizing Data Using the COUNT Function: Count the number of products in the Products table efficiently. Yes, you can use the number keyword right after proc SQL. However, think of a scenario where there are millions of rows. Will you scroll down to see how many records there are? Instead, use the COUNT() function. Count(*) will count all observations. You can also count unique occurrences. Use the distinct keyword within the COUNT() function. For example, count(distinct Product_Category) would tell you the number of unique product categories.

e) Grouping Data; Filtering Grouped Data

Activity 09: Grouping Data; Filtering Grouped Data: Orion wants to know its total profit each month. Extract the month of order date using the month() function and rename the new column as OrderMonth, aggregate profit data using sum() function and rename the new column as TotalProfit. You may format the TotalProfit column as dollar16.2 if you want. Present the table as grouped by OrderMonth.

Demonstration: Grouping Data by more than one variable; Filtering Grouped Data: Use the Transactions dataset to perform this task. The manager of store 10 wants to know the best-selling products in his store. Write a query that will show the total quantity sold for each Product where the store is 10. Also, the CEO wants to know the total quantity sold by product and store.

f) Advanced: Using Boolean Expressions

Week 6 Meeting Agenda

Creating and Managing Tables

i) Using Tables Created from a Query

(1) Creating a Table from a Query Result

Activity 10 – Creating a Table from a Query Result: Orion Star wants to identify the products with total profits of at least $500 and store the results by creating a table called **TOP_PRODUCTS**.

Examine and run the query below that creates a report (notice that it is NOT a table). View the results.

Remove the TITLE statements and add the CREATE TABLE statement and create a table named **Top_Products**. Run the query and confirm that the table was created successfully.

(2) Copying the Structure of an Existing Table

Activity 11– Copying the Structure of an Existing Table and Inserting Rows with the SET Clause: Orion Star wants to create a **MANAGERS** table. First, create an empty table using the structure of **EMPLOYEE_MASTER** table. Then, insert rows from the **EMPLOYEE_MASTER** table where the **Job_Title** contains ‘Manager’.

Complete the following CREATE TABLE statement using the structure of **EMPLOYEE_MASTER** table and keep only the following columns: **Employee_ID Employee_Name Employee_Hire_Date Department Job_Title**. Run the query and confirm that an empty table was created.

Enter the correct column names and a where clause (Job_Title should contain the word ‘Manager’) to complete the INSERT INTO statement. Run the query. How many rows were inserted into the table **Managers**?

Complete the INSERT INTO statement with the SET clause and insert yourself as a manager into the **Managers** table. Run the query. What does the note in the log say?

Week 6 Meeting Agenda

(3) Creating a Table by Defining Columns

ii) Inserting Rows into Tables with
 (1) a Query
 (2) the VALUES Clause
 (3) the SET Clause

iii) Dropping Tables in SQL

iv) Additional Statements
 (1) ALTER TABLE,
 (2) UPDATE,
 (3) DELETE

Activity 12– Additional statements

Rerun steps 1, 2, and 3 of the previous activity, i.e., do not drop the table.
Add two new empty columns to **work.managers** table: Twentieth_Anniversary, numeric column, represents a date; Upcoming_20th_Anniversary, character column, takes the value of “Yes” if a manager has an upcoming 20th anniversary.

Modify format of Employee_Hire_Date and Twentieth_Anniversary columns so that the date formats match.

Update the values of the newly created columns as follows: If it hasn’t been 20 years since an employee is hired, then we will update the values of Upcoming_20th_Anniversary column to ‘Yes’ and compute their Twentieth_Anniversary.

Remove the managers if their job title contains the word “Senior”.

II. In-class Exercise #5

III. Q&A

IV. Quiz #3