SAS Data sets for Q1 to Q4: WORK.Sales_employees and WORK.Sales_records

Q1.Applying the ‘PROC SQL’ to extract observations from the SAS data set 'WORK.Sales_employees' using the restriction rules below and saving the results into a SAS table 'WORK.empres'.

The restriction rules
(a) Only keep 'sales_name'.
(b) The field 'sales_id' is between 20000 and 40000.
(c) All sales come from North or South areas.

Q2. Using the ‘PROC SQL’ to summarize observations from the SAS table 'WORK.Sales_records' and saving the results into a table 'WORK.salesres_1'. The resulting table contains the aggregated fields 'sale_amount' and ‘total transaction number’ for each ‘sales employee’ and ‘product’ group. Finally sort the resulting table by the descending order of the aggregated 'sale_amount'.

Q3.Conducting the same SQL query as in Q2 and saving the results into a table 'WORK.salesres_2', but adding the following restricted rules:

(a) The transaction time is between May and September.

(b) The aggregated 'sale_amount' is more than $40,000.

Q4.Using the ‘PROC SQL’ to solve the following questions

(a)Creating a data set 'WORK.salesarea' by inner joining the tables ‘WORK.Sales_employees’ and ‘WORK.Sales_records’ by the field 'sales_id'. Where only keeping the columns 'sales_name', 'sale_amount' and 'area'.

(b)Creating a table 'WORK.salesarea_sum' by summarizing the column 'sale_amount' in the table 'WORK.salesarea' (create above) for each sales employee. Listing the 'area', 'maximum sale_amount', 'minimum sale_amount' and 'mean sale_amount' in the resulting table. All the sales amount columns are required to be assigned with the dollar format. Finally, the resulting table only contains the records with the average sale_amount>$6000 and ordered by the average sale_amount.

SAS Data sets for Q5 to Q6: WORK.Travel_flight and WORK. Travel_delay

Q5. Applying the ‘PROC SQL’ to extract observations from the table 'WORK.Travel_flight' to create a resulting table which contains the following columns: 'flight', 'miles' and 'empty_seats' with the filter condition 'miles>1000'. Where the first two columns are the existing variables in the table 'WORK.Travel_flight' and the last column ('empty_seats') is a newly created variable using the following steps:

(a) Calculating a variable temp=capacity - boarded.
(b) If temp is between 0 and 20 then empty_seats='low';
    If temp is between 21 and 100 then empty_seats='medium';
    If temp is more than 100 then empty_seats='high'.

Finally please de-duplicate the result.

Q6. Applying 'PROC SQL' to extract observations from the table 'WORK.Travel_delay' based on the following requests:

(a) Finding the number of 'Domestic' flights for each destination type without delay.

(b) Finding all records with the destination being 'WAS' before MAR 5 and ordering the resulting table by variable 'delay'.

(c) Finding how many flights are 'normal (no delay)', 'delayed' and 'advanced' separately?

(d) Creating a SAS table 'des' which contains only unique destination and destination type.

(e) Updating the table by changing the values of destination type from string 'International' to 'INTER' and 'Domestic' to 'DOM' respectively.

(f) Calculating the average delay for those flights with 1-10 minutes delay.

(g) Deleting records from the table 'des' created in (d) with destination being 'PAR'.

(h) Comparing the total delay time on MAR and APR for each destination, and finding which destination has the largest time difference (for delay time)?

(I)Using 'PROC SQL' to create a SAS table by extracting observations from the table 'WORK.Travel_delay'. The resulting table has the following restriction rules (a) Only extract the columns 'flight', 'date', 'dest' and 'delay' (b) No duplicated records (c) All flight numbers must belong to those flight numbers in the SAS table 'WORK.Travel_flight' with the following conditions 'miles>3500 and capacity>220' (d) All the flight dates are March 1994.

SAS Data sets for Q7: WORK.Big_cities

Q7.Using the 'PROC SQL' to create a SAS data set by extracting all 'USA' city records from the table 'WORK.Big_cities'. Also listing the country name ('USA' here) in the resulting table.