

## **SAS Data set for Q1 to Q5: WORK.Collectionrate**

**Q1.** Create a SAS data set 'WORK work.Collectionrate\_new' by applying DATA STEP to the table 'WORK.Collectionrate'. It is required to conduct the following programming (a) create a variable 'tran\_id' with the value being concatenation of the variable 'agency\_symbol', 'charge\_date', the character '\_' and table's row sequential number \_n\_ (b) create a new variable 'rate\_change\_ratio' by the formula  $100 * (\text{rate}_{2006} - \text{rate}) / \text{rate}$  and save the variable as character type (c) if errors occur, or the rate's value is missing then assign the 'rate\_change\_ratio' as the character 'N/A'.

**Q2.** Create a table 'WORK work.Collectionrate\_nomiss' by using DATA STEP to the data set 'WORK.Collectionrate\_new' generated in Q1. It is required to do the following SAS programming

(a) For an observation, if the value of variable 'rate' is missing or zero then delete this row

(b) Rename the column 'rate' to 'charge\_ratio' using the 'RENAME' statement inside the data step

(c) Assign the variable 'agency\_name' label 'Collection Agency'

(d) Use the 'keep' statement inside data step to only keep the columns 'agency\_name' 'charge\_date' and 'rate' (the resulting data set should contain the renamed variable, i.e. 'charge\_ratio').

**Q3.** Finish the same questions as Q2, except for (d), now you put the 'keep' statement in the data set definition statement. The resulting table should have the same contents as those of Q2.

**Q4.** Solve the same questions as Q2, except for (b), now you put the 'rename' statement in the 'set' statement when setting data set 'WORK.Collectionrate'. The resulting data set should have the same contents as those of Q1 to Q3.

**Q5.** Solve the same questions as Q3, except for (b), now you put the 'rename' statement in the 'set' statement when setting the data set 'WORK.Collectionrate'. The resulting data set should have the same contents as those of Q1 to Q4.

**SAS Data set for Q6: WORK.temperature\_city**

**Q6.** Observing the following SAS table 'WORK.temperature\_city':

	city	Spring	Summer	Fall	Winter
1	Toronto	20	37	19	-12
2	Beijing	25	39	16	0
3	london	17	29	13	1
4	Newyork	20	30	11	-9
5	Mexico	25	35	18	8

Write a DATA STEP program to transform the data into a new data set 'WORK.temperature\_city', shown below:

	city_name	season	Temperature
1	Toronto	spring	20
2	Toronto	summer	37
3	Toronto	fall	19
4	Toronto	winter	-12
5	Beijing	spring	25
6	Beijing	summer	39
7	Beijing	fall	16
8	Beijing	winter	0
9	london	spring	17
10	london	summer	29
11	london	fall	13
12	london	winter	1
13	Newyork	spring	20
14	Newyork	summer	30
15	Newyork	fall	11
16	Newyork	winter	-9
17	Mexico	spring	25
18	Mexico	summer	35
19	Mexico	fall	18
20	Mexico	winter	8

### **SAS Data set for Q7: WORK.Sales\_records**

**Q7.** Observing the following SAS table 'WORK.WORK.Sales\_records'. You first run the following SAS codes:

```
proc sort data=Sales_records out=temp;
    by sales_id transaction_date;
Run;
```

Then you are asked to create a new table 'Sales\_records\_cum' using the table 'temp' created from the data step above. The resulting table should contain the following columns (a) 'sales\_id' (b) 'cum\_sale', which is the newly created variable containing the total cumulative 'sale\_amount' for each sales employee. (Hint: use 'retain' statement).

### **SAS Data set for Q8: WORK.Retail\_sales**

**Q8.** Checking the following SAS table 'WORK.Retail\_sales'. You first run the following SAS codes:

```
proc sort data=Retail_sales out=temp;
    by Store_num date;
run;
```

Then you are required to create a new table 'Retail\_sales\_new' using the table 'temp' created from the data step above. The resulting table should include the following columns (a) 'Store\_num' (b) 'firstdate', which is the newly created variable containing the first transaction date for each store. (c) 'lastdate', which is the newly created variable containing the last transaction date for each store (d) 'numdays', which saves the number of days between 'firstdate' and 'lastdate' (e) 'avg\_sales', which contains the average value of variable 'Weekly\_Sales' for each store (hint: use 'retain', 'if first.' and 'if last.' statement in data step).