Unit 1-10 Exercises

1. Appending Like-Structured Data Sets

a. Retrieve the starter program **p110e01**.

b. Submit the two PROC CONTENTS steps to compare the variables in the two data sets.

How many variables are in **orion.price_current**? _____

How many variables are in **orion.price_new**? _____

Does **orion.price_new** contain any variables that are not in **orion.price_current**?_____

c. Add a PROC APPEND step after the PROC CONTENTS steps to append **orion.price_new** to **orion.price_current**. The FORCE option is not needed.

Why is the FORCE option not needed? _____

d. Submit the program and confirm that 88 observations from **orion.price_new** were added to **orion.price_current**, which should now have 259 observations (171 original observations plus 88 appended observations).

2. Appending Unlike-Structured Data Sets

a. Write and submit two PROC CONTENTS steps to compare the variables in **orion.qtr1_2007** and **orion.qtr2_2007**.

How many variables are in **orion.qtr1_2007**? _____

How many variables are in **orion.qtr2_2007**? _____

Which variable is not in both data sets? _____

b. Write a PROC APPEND step to append **orion.qtr1_2007** to a non-existing data set called **Work.ytd**.

c. Submit the PROC APPEND step and confirm that 22 observations were copied to **Work.ytd**.

d. Write another PROC APPEND step to append **orion.qtr2_2007** to **Work.ytd**. The FORCE option is needed.

Why is the FORCE option needed? _____

e. Submit the second PROC APPEND step and confirm that 36 observations from **orion.qtr2_2007** were added to **Work.ytd**, which should now have 58 observations.

3. Using the Append Statement

a. Write and submit three PROC CONTENTS steps to compare the variables in **orion.shoes_eclipse**, **orion.shoes_tracker**, and **orion.shoes**.

b. Write a PROC DATASETS step with two APPEND statements to append **orion.shoes_eclipse** and **orion.shoes_tracker** to **orion.shoes**.

Documentation on the DATASETS procedure can be found in the SAS Help and Documentation from the Contents tab (**SAS Products** ⇨ **Base SAS** ⇨ **Base SAS 9.3 Procedures Guide** ⇨ **Procedures** ⇨ **The DATASETS Procedure**).

c. Submit the PROC DATASETS step and confirm that **orion.shoes** contains 34 observations (10 original observations plus 14 observations from **orion.shoes_eclipse** and 10 observations from **orion.shoes_tracker**).

4. Concatenating Like-Structured Data Sets

a. Write and submit a DATA step to concatenate **orion.mnth7_2007**, **orion.mnth8_2007**, and **orion.mnth9_2007** to create a new data set called **Work.thirdqtr**.

How many observations in **Work.thirdqtr** are from **orion.mnth7_2007**? _____

How many observations in **Work.thirdqtr** are from **orion.mnth8_2007**? _____

How many observations in **Work.thirdqtr** are from **orion.mnth9_2007**? _____

b. Write and submit a PROC PRINT step to create the following report:

Partial PROC PRINT Output (First 10 of 32 Observations)

Obs	Order_ID	Order_ Type	Employee_ID	Customer_ID	Order_ Date	Delivery_ Date
1	1242691897	2	99999999	90	02JUL2007	04JUL2007
2	1242736731	1	121107	10	07JUL2007	07JUL2007
3	1242773202	3	99999999	24	11JUL2007	14JUL2007
4	1242782701	3	99999999	27	12JUL2007	17JUL2007
5	1242827683	1	121105	10	17JUL2007	17JUL2007
6	1242836878	1	121027	10	18JUL2007	18JUL2007
7	1242838815	1	120195	41	19JUL2007	19JUL2007
8	1242848557	2	99999999	2806	19JUL2007	23JUL2007
9	1242923327	3	99999999	70165	28JUL2007	29JUL2007
10	1242938120	1	120124	171	30JUL2007	30JUL2007

5. Concatenating Unlike-Structured Data Sets

a. Retrieve the starter program **p110e05**.

b. Submit the two PROC CONTENTS steps to compare the variables in the two data sets.

What are the names of the two variables that are different in the two data sets?

orion.sales	orion.nonsales

c. Add a DATA step after the PROC CONTENTS steps to concatenate **orion.sales** and **orion.nonsales** to create a new data set called **Work.allemployees**.

Use a RENAME= data set option to change the names of the different variables in **orion.nonsales**.

Include only the following five variables: **Employee_ID**, **First_Name**, **Last_Name**, **Job_Title**, and **Salary**.

d. Add a PROC PRINT step to create the following report:

Partial PROC PRINT Output (First 10 of 400 Observations)

Obs	Employee_ID	First_ Name	Last_Name	Salary	Job_Title
1	120102	Tom	Zhou	108255	Sales Manager
2	120103	Wilson	Dawes	87975	Sales Manager
3	120121	Irenie	Elvish	26600	Sales Rep. II
4	120122	Christina	Ngan	27475	Sales Rep. II
5	120123	Kimiko	Hotstone	26190	Sales Rep. I
6	120124	Lucian	Daymond	26480	Sales Rep. I
7	120125	Fong	Hofmeister	32040	Sales Rep. IV
8	120126	Satyakam	Denny	26780	Sales Rep. II
9	120127	Sharryn	Clarkson	28100	Sales Rep. II
10	120128	Monica	Kletschkus	30890	Sales Rep. IV

6. Interleaving Data Sets

a. Retrieve the starter program **p110e06**.

b. Add a PROC SORT step after the PROC SORT step in the starter program. The PROC SORT step needs to sort **orion.shoes_tracker** by **Product_Name** to create a new data set called **Work.trackersort**.

Documentation on the SORT procedure can be found in the SAS Help and Documentation from the Contents tab (**SAS Products** ⇨ **Base SAS** ⇨ **Base SAS 9.3 Procedures Guide** ⇨ **Procedures** ⇨ **The SORT Procedure**).

c. Add a DATA step after the two PROC SORT steps to interleave the two sorted data sets by **Product_Name** to create a new data set called **Work.e_t_shoes**.

Include only the following three variables: **Product_Group**, **Product_Name**, and **Supplier_ID**.

d. Add a PROC PRINT step to create the following report:

Partial PROC PRINT Output (First 10 of 24 Observations)

Obs	Product_Group	Product_Name	Supplier_ID
1	Eclipse Shoes	Atmosphere Imara Women's Running Shoes	1303
2	Eclipse Shoes	Atmosphere Shatter Mid Shoes	1303
3	Eclipse Shoes	Big Guy Men's Air Deschutz Viii Shoes	1303
4	Eclipse Shoes	Big Guy Men's Air Terra Reach Shoes	1303
5	Eclipse Shoes	Big Guy Men's Air Terra Sebec Shoes	1303
6	Eclipse Shoes	Big Guy Men's International Triax Shoes	1303
7	Eclipse Shoes	Big Guy Men's Multicourt Ii Shoes	1303
8	Eclipse Shoes	Cnv Plus Men's Off Court Tennis	1303
9	Tracker Shoes	Hardcore Junior/Women's Street Shoes Large	14682
10	Tracker Shoes	Hardcore Men's Street Shoes Large	14682

7. Merging orion.orders and orion.order_item in a One-to-Many Merge

a. Retrieve the starter program **p110e07**.

b. Submit the two PROC CONTENTS steps to determine the common variable among the two data sets.

c. Add a DATA step after the two PROC CONTENTS steps and prior to the PROC PRINT step to merge **orion.orders** and **orion.order_item** by the common variable to create a new data set called **Work.allorders**.

d. Submit the program and confirm that **Work.allorders** was created with 732 observations and 12 variables.

8. Merging orion.product_level and orion.product_list in a One-to-Many Merge

a. Write a PROC SORT step to sort **orion.product_list** by **Product_Level** to create a new data set called **Work.product_list**.

b. Write a DATA step to merge **orion.product_level** with the previous sorted data set by the appropriate common variable. Create a new data set called **Work.listlevel**.

c. Write a PROC PRINT step with a VAR statement to create the following report:

Partial PROC PRINT Output (First 10 of 556 Observations)

Obs	Product_ID	Product_Name	Product_ Level	Product_ Level_ Name
1	210200100009	Kids Sweat Round Neck,Large Logo	1	Product
2	210200100017	Sweatshirt Children's O-Neck	1	Product
3	210200200022	Sunfit Slow Swimming Trunks	1	Product
4	210200200023	Sunfit Stockton Swimming Trunks Jr.	1	Product
5	210200300006	Fleece Cuff Pant Kid'S	1	Product
6	210200300007	Hsc Dutch Player Shirt Junior	1	Product

7	210200300052	Tony's Cut & Sew T-Shirt	1	Product
8	210200400020	Kids Baby Edge Max Shoes	1	Product
9	210200400070	Tony's Children's Deschutz (Bg) Shoes	1	Product
10	210200500002	Children's Mitten	1	Product

9. Joining `orion.product_level` and `orion.product_list` in a One-to-Many Merge

a. Write a PROC SQL step to perform an inner join of **`orion.product_level`** and **`orion.product_list`** by **`Product_Level`** to create a new data set called **`Work.listlevelsql`**. The new data set should include only **`Product_ID`**, **`Product_Name`**, **`Product_Level`**, and **`Product_Level_Name`**.

Documentation on the SQL procedure can be found in the SAS Help and Documentation from the Contents tab (**SAS Products** ⇨ **Base SAS** ⇨ **Base SAS 9.3 Procedures Guide** ⇨ **Procedures** ⇨ **The SQL Procedure**).

b. Write a PROC PRINT step to create the following report:

Partial PROC PRINT Output (First 10 of 556 Observations)

Obs	Product_ID	Product_Name	Product_ Level	Product_Level_ Name
1	210000000000	Children	4	Product Line
2	210100000000	Children Outdoors	3	Product Category
3	210100100000	Outdoor things, Kids	2	Product Group
4	210200000000	Children Sports	3	Product Category
5	210200100000	A-Team, Kids	2	Product Group
6	210200100009	Kids Sweat Round Neck,Large Logo	1	Product
7	210200100017	Sweatshirt Children's O-Neck	1	Product
8	210200200000	Bathing Suits, Kids	2	Product Group
9	210200200022	Sunfit Slow Swimming Trunks	1	Product
10	210200200023	Sunfit Stockton Swimming Trunks Jr.	1	Product

10. Merging Using the IN= Option

a. Retrieve the starter program **p110e10**.

b. Add a DATA step after the PROC SORT step to merge **`Work.product`** and **`orion.supplier`** by **`Supplier_ID`** to create a new data set called **`Work.prodsup`**.

c. Submit the program and confirm that **`Work.prodsup`** was created with 556 observations and 10 variables.

d. Modify the DATA step to output only observations that are in **`Work.product`** but not **`orion.supplier`**. A subsetting IF statement that references IN= variables in the MERGE statement needs to be added.

e. Submit the program and confirm that **`Work.prodsup`** was created with 75 observations and 10 variables. The supplier information will be missing in the PROC PRINT output.

11. Merging Using the IN= and RENAME= Options

a. Write a PROC SORT step to sort **orion.customer** by **Country** to create a new data set called **Work.customer**.

b. Write a DATA step to merge the previous sorted data set with **orion.lookup_country** by **Country** to create a new data set called **Work.allcustomer**.

In the **orion.lookup_country** data set, **Start** needs to be renamed to **Country** and **Label** needs to be renamed to **Country_Name**.

Include only the following four variables: **Customer_ID**, **Country**, **Customer_Name**, and **Country_Name**.

c. Write a PROC PRINT step to create the following report:

Partial PROC PRINT Output (First 15 of 308 Observations)

Obs	Customer_ID	Country	Customer_Name	Country_Name
1	.	AD		Andorra
2	.	AE		United Arab Emirates
3	.	AF		Afghanistan
4	.	AG		Antigua/Barbuda
5	.	AI		Anguilla
6	.	AL		Albania
7	.	AM		Armenia
8	.	AN		Netherlands Antilles
9	.	AO		Angola
10	.	AQ		Antarctica
11	.	AR		Argentina
12	.	AS		American Samoa
13	.	AT		Austria
14	29	AU	Candy Kinsey	Australia
15	41	AU	Wendell Summersby	Australia

d. Modify the DATA step to store only the observations that contain both customer information and country information. A subsetting IF statement that references IN= variables in the MERGE statement needs to be added.

e. Submit the program to create the following report:

Partial PROC PRINT Output (First 7 of 77 Observations)

Obs	Customer_ID	Country	Customer_Name	Country_ Name
1	29	AU	Candy Kinsey	Australia
2	41	AU	Wendell Summersby	Australia
3	53	AU	Dericka Pockran	Australia
4	111	AU	Karolina Dokter	Australia
5	171	AU	Robert Bowerman	Australia
6	183	AU	Duncan Robertshawe	Australia
7	195	AU	Cosi Rimmington	Australia

12. Merging and Outputting to Multiple Data Sets

a. Write a PROC SORT step to sort **orion.orders** by **Employee_ID** to create a new data set called **Work.orders**.

b. Write a DATA step to merge **orion.staff** and **Work.orders** by **Employee_ID**.

Create two new data sets: **Work.allorders** and **Work.noorders**.

The data set **Work.allorders** should include all observations from **Work.orders**, regardless of matches or nonmatches from the **orion.staff** data set.

The data set **Work.noorders** should include the observations from **orion.staff** that do not have a match in **Work.orders**.

Include only the following six variables: **Employee_ID**, **Job_Title**, **Gender**, **Order_ID**, **Order_Type**, and **Order_Date**.

c. Using the new data sets, write two PROC PRINT steps to create two reports.

d. Submit the program and confirm that **Work.allorders** was created with 490 observations and 6 variables and **Work.noorders** was created with 324 observations and 6 variables.