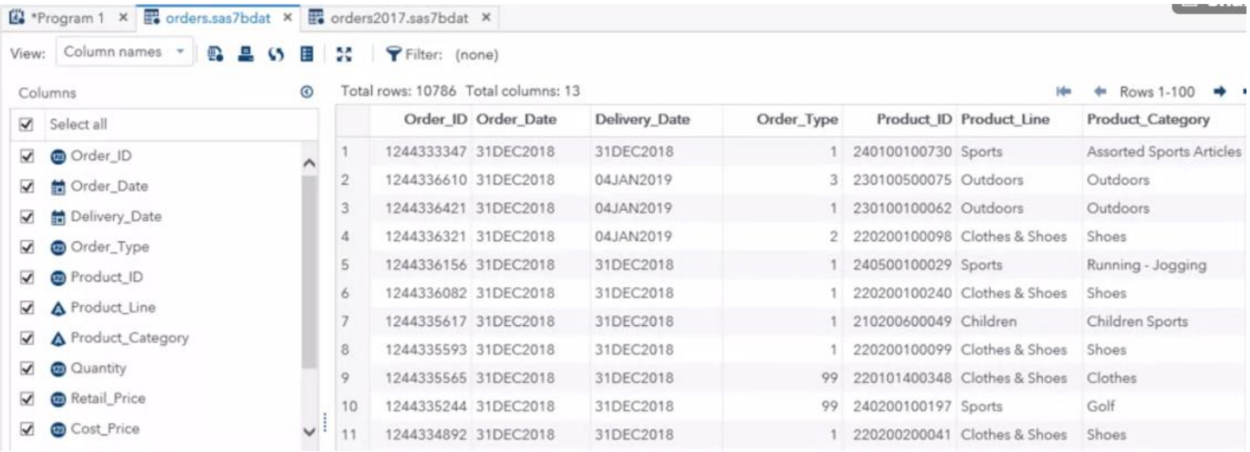
Preparing for the SAS Programming Certification

Week 4: Controlling DATA Step

1. **Combining Tables Review**
2. **Processing Repetitive Code Review**
3. **Restructuring Tables Review**

```
/**********************************************/

/* This code defines macro variables and the   */

/* library for this course. You must run       */

/* this code each time you start SAS OnDemand  */

/* for Academics to access your practice data. */

/**********************************************/
```



*Program 1 × | orders.sas7bdat × | orders2017.sas7bdat ×

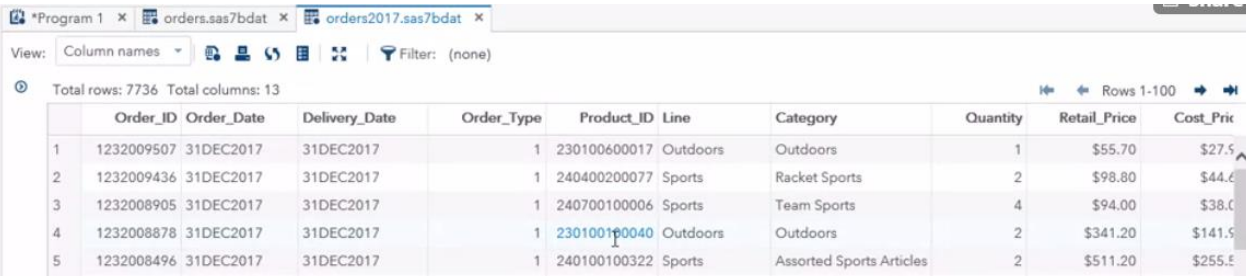
View: Column names | Filter: (none)

Columns

- ☑ Select all
- ☑ Order_ID
- ☑ Order_Date
- ☑ Delivery_Date
- ☑ Order_Type
- ☑ Product_ID
- ☑ Product_Line
- ☑ Product_Category
- ☑ Quantity
- ☑ Retail_Price
- ☑ Cost_Price

Total rows: 10786 Total columns: 13 Rows 1-100

	Order_ID	Order_Date	Delivery_Date	Order_Type	Product_ID	Product_Line	Product_Category
1	1244333347	31DEC2018	31DEC2018	1	240100100730	Sports	Assorted Sports Articles
2	1244336610	31DEC2018	04JAN2019	3	230100500075	Outdoors	Outdoors
3	1244336421	31DEC2018	04JAN2019	1	230100100062	Outdoors	Outdoors
4	1244336321	31DEC2018	04JAN2019	2	220200100098	Clothes & Shoes	Shoes
5	1244336156	31DEC2018	31DEC2018	1	240500100029	Sports	Running - Jogging
6	1244336082	31DEC2018	31DEC2018	1	220200100240	Clothes & Shoes	Shoes
7	1244335617	31DEC2018	31DEC2018	1	210200600049	Children	Children Sports
8	1244335593	31DEC2018	31DEC2018	1	220200100099	Clothes & Shoes	Shoes
9	1244335565	31DEC2018	31DEC2018	99	220101400348	Clothes & Shoes	Clothes
10	1244335244	31DEC2018	31DEC2018	99	240200100197	Sports	Golf
11	1244334892	31DEC2018	31DEC2018	1	220200200041	Clothes & Shoes	Shoes



*Program 1 × | orders.sas7bdat × | orders2017.sas7bdat ×

View: Column names | Filter: (none)

Total rows: 7736 Total columns: 13 Rows 1-100

	Order_ID	Order_Date	Delivery_Date	Order_Type	Product_ID	Line	Category	Quantity	Retail_Price	Cost_Pric
1	1232009507	31DEC2017	31DEC2017	1	230100600017	Outdoors	Outdoors	1	$55.70	$27.5
2	1232009436	31DEC2017	31DEC2017	1	240400200077	Sports	Racket Sports	2	$98.80	$44.6
3	1232008905	31DEC2017	31DEC2017	1	240700100006	Sports	Team Sports	4	$94.00	$38.0
4	1232008878	31DEC2017	31DEC2017	1	230100100040	Outdoors	Outdoors	2	$341.20	$141.9
5	1232008496	31DEC2017	31DEC2017	1	240100100322	Sports	Assorted Sports Articles	2	$511.20	$255.5

```
/* Concatenating Tables */

data profit;

	length Customer_Continent $ 20;

	set cr.orders cr.orders2017(rename=(Line=Product_Line Category=Product_Category));

	length Order_Source $ 8;
```

```
        where Delivery_Date>=Order_Date;

        Customer_Country=upcase(Customer_Country);

        If Quantity<0 then Quantity=.;

        Profit=(Retail_Price-Cost_Price)*Quantity;

        format Profit dollar12.2;

        ShipDays=Delivery_Date-Order_Date;

        Age_Range=substr(Customer_Age_Group, 1, 5);

        if Order_Type=1 then Order_Source="Retail";

        else if Order_Type=2 then Order_Source="Phone";

        else if Order_Type=3 then Order_Source="Internet";

        else Order_Source="Unknown";

        drop Retail_Price Cost_Price Customer_Age_Group Order_Type;

run;
```

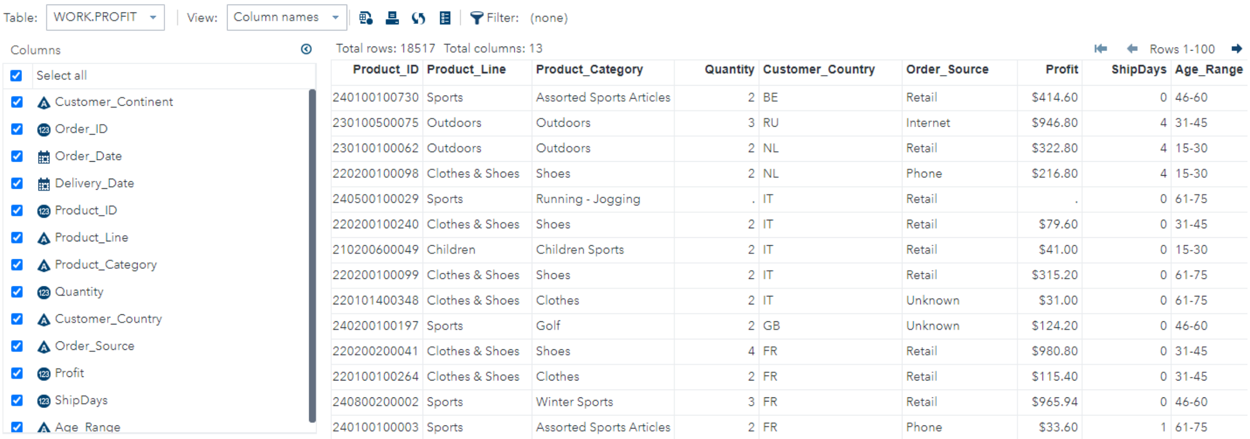


Table: WORK.PROFIT | View: Column names | Filter: (none)

Columns

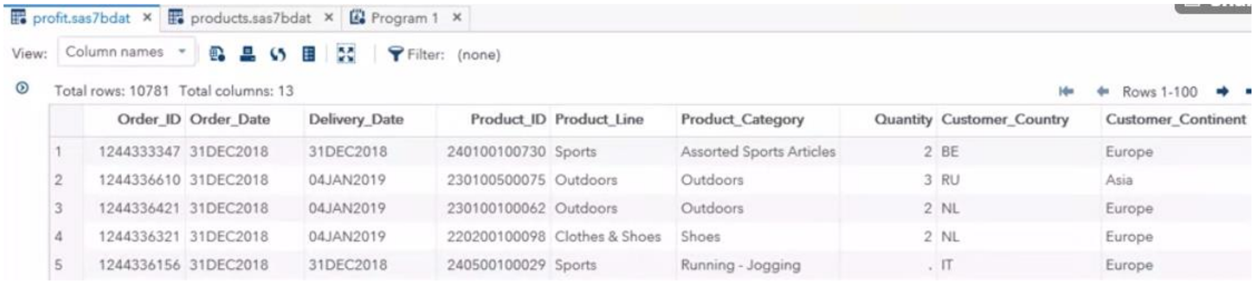
☑ Select all
☑ Customer_Continent
☑ Order_ID
☑ Order_Date
☑ Delivery_Date
☑ Product_ID
☑ Product_Line
☑ Product_Category
☑ Quantity
☑ Customer_Country
☑ Order_Source
☑ Profit
☑ ShipDays
☑ Age_Range

Total rows: 18517 Total columns: 13

Product_ID	Product_Line	Product_Category	Quantity	Customer_Country	Order_Source	Profit	ShipDays	Age_Range
240100100730	Sports	Assorted Sports Articles	2	BE	Retail	$414.60	0	46-60
230100500075	Outdoors	Outdoors	3	RU	Internet	$946.80	4	31-45
230100100062	Outdoors	Outdoors	2	NL	Retail	$322.80	4	15-30
220200100098	Clothes & Shoes	Shoes	2	NL	Phone	$216.80	4	15-30
240500100029	Sports	Running - Jogging	.	IT	Retail	.	0	61-75
220200100240	Clothes & Shoes	Shoes	2	IT	Retail	$79.60	0	31-45
210200600049	Children	Children Sports	2	IT	Retail	$41.00	0	15-30
220200100099	Clothes & Shoes	Shoes	2	IT	Retail	$315.20	0	61-75
220101400348	Clothes & Shoes	Clothes	2	IT	Unknown	$31.00	0	61-75
240200100197	Sports	Golf	2	GB	Unknown	$124.20	0	46-60
220200200041	Clothes & Shoes	Shoes	4	FR	Retail	$980.80	0	31-45
220100100264	Clothes & Shoes	Clothes	2	FR	Retail	$115.40	0	31-45
240800200002	Sports	Winter Sports	3	FR	Retail	$965.94	0	46-60
240100100003	Sports	Assorted Sports Articles	2	FR	Phone	$33.60	1	61-75

```
proc sort data=cr.profit out=profit_sort;

        by Product_ID;

run;


proc sort data=cr.products out=products_sort;

        by Product_ID;

run;
```

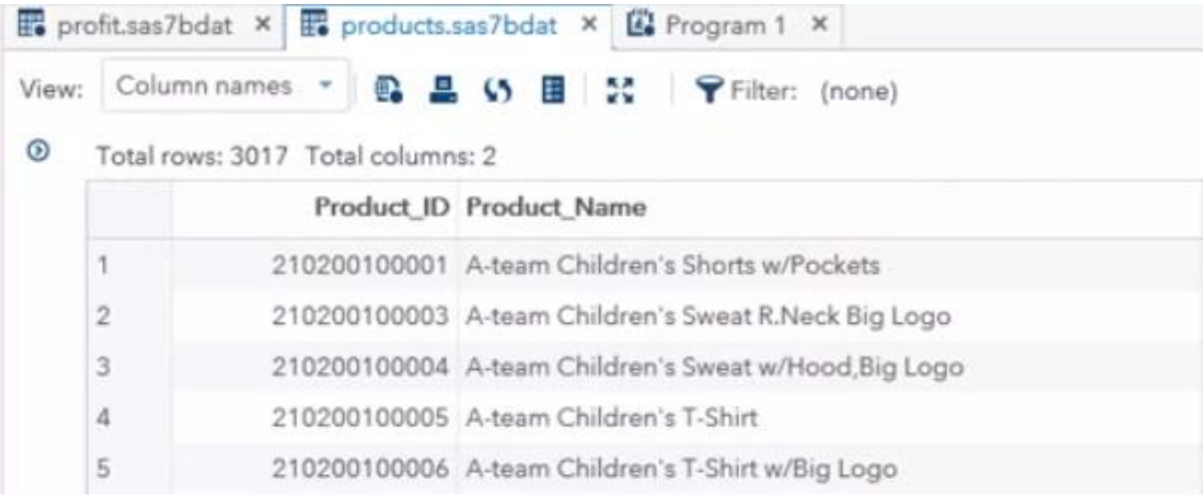
```
data product_detail product_nosales(keep=Product_ID Product_Name);
        merge profit_sort(in=inprof) products_sort(in=inprod);
        by Product_ID;
        if inprof=1 and inprod=1 then output product_detail;
        if inprof=0 and inprod=1 then output product_nosales;
run;
```



Total rows: 10781 Total columns: 13

Rows 1-100

	Order_ID	Order_Date	Delivery_Date	Product_ID	Product_Line	Product_Category	Quantity	Customer_Country	Customer_Continent
1	1244333347	31DEC2018	31DEC2018	240100100730	Sports	Assorted Sports Articles	2	BE	Europe
2	1244336610	31DEC2018	04JAN2019	230100500075	Outdoors	Outdoors	3	RU	Asia
3	1244336421	31DEC2018	04JAN2019	230100100062	Outdoors	Outdoors	2	NL	Europe
4	1244336321	31DEC2018	04JAN2019	220200100098	Clothes & Shoes	Shoes	2	NL	Europe
5	1244336156	31DEC2018	31DEC2018	240500100029	Sports	Running - Jogging	.	IT	Europe



Total rows: 3017 Total columns: 2

	Product_ID	Product_Name
1	210200100001	A-team Children's Shorts w/Pockets
2	210200100003	A-team Children's Sweat R.Neck Big Logo
3	210200100004	A-team Children's Sweat w/Hood,Big Logo
4	210200100005	A-team Children's T-Shirt
5	210200100006	A-team Children's T-Shirt w/Big Logo

/* Programming Question 5.01

If necessary, start SAS Studio and submit libname.sas. Write a new program to do the following:

Concatenate the cr.m7_sales, cr.m8_sales, and cr.m9_sales tables to create a new table named q3_sales.

Ensure that values in similar columns with different names are in a single column in the new table.

Create a frequency report that counts the number of orders for each value of Order_Type.

How many rows are in the q3_sales table?

How many columns are in the q3_sales table?

How many orders have Order_Type equal to 1?

*/

```
data q3_sales;
        set cr.m7_sales cr.m8_sales cr.m9_sales(rename=(EmpID=Employee_ID));
run;


proc freq data=q3_sales;
        table Order_Type;
run;
```

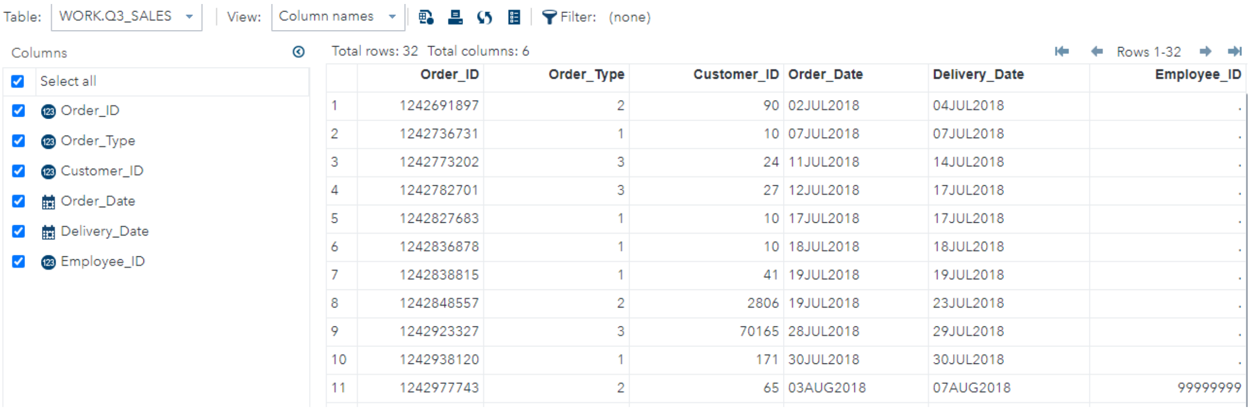


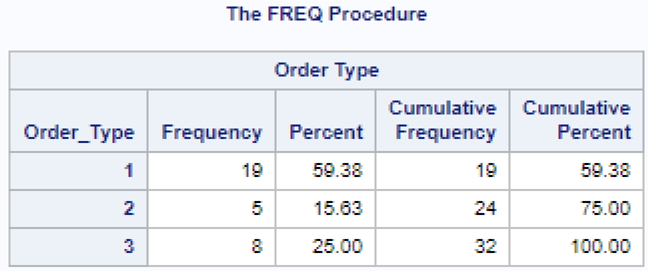
Table: WORK.Q3_SALES View: Column names Filter: (none)

Columns

☑ Select all
☑ Order_ID
☑ Order_Type
☑ Customer_ID
☑ Order_Date
☑ Delivery_Date
☑ Employee_ID

Total rows: 32 Total columns: 6 Rows 1-32

	Order_ID	Order_Type	Customer_ID	Order_Date	Delivery_Date	Employee_ID
1	1242691897	2	90	02JUL2018	04JUL2018	.
2	1242736731	1	10	07JUL2018	07JUL2018	.
3	1242773202	3	24	11JUL2018	14JUL2018	.
4	1242782701	3	27	12JUL2018	17JUL2018	.
5	1242827683	1	10	17JUL2018	17JUL2018	.
6	1242836878	1	10	18JUL2018	18JUL2018	.
7	1242838815	1	41	19JUL2018	19JUL2018	.
8	1242848557	2	2806	19JUL2018	23JUL2018	.
9	1242923327	3	70165	28JUL2018	29JUL2018	.
10	1242938120	1	171	30JUL2018	30JUL2018	.
11	1242977743	2	65	03AUG2018	07AUG2018	99999999



The FREQ Procedure

Order Type				
Order_Type	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	19	59.38	19	59.38
2	5	15.63	24	75.00
3	8	25.00	32	100.00

```
/*Programming Question 5.02

If necessary, start SAS Studio and submit libname.sas. Write a new program to do the following:

Merge the cr.employee and cr.employee_addresses tables to create a new table named emp_full.

Include all rows from the cr.employee table in the emp_full table.

Ensure that the emp_full table is ordered by EmpID.

How many rows are in the emp_full table?

What is the value of Emp_ID for row 27?

What is the value of Street_Number for row 20?

*/

proc sort data=cr.employee;
```

```
            by EmpID;
run;


proc sort data=cr.employee_addresses;
            by Employee_ID;
run;


data emp_full;
            merge cr.employee cr.employee_addresses(rename=(Employee_ID=EmpID));
            by EmpID;
run;
*Solution;
proc sort data=cr.employee_addresses(rename=(Employee_ID=EmpID))
        out=address_sort;
   by EmpID;
run;


data emp_full;
   merge cr.employee(in=e) address_sort;
   by EmpID;
   if e;
run;
```

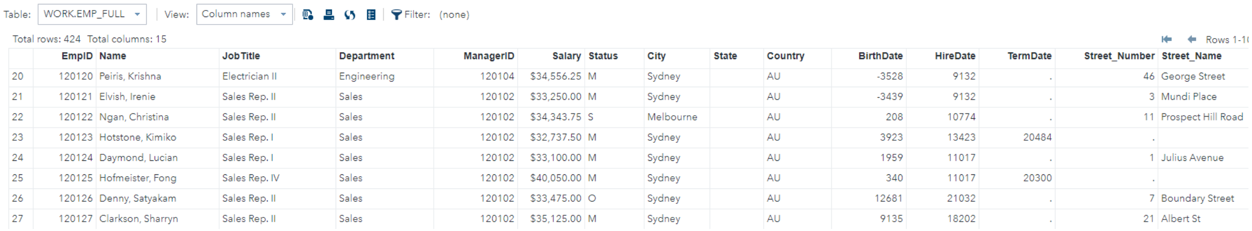


Table: WORK.EMP_FULL | View: Column names | Filter: (none)

Total rows: 424 Total columns: 15

Rows 1-1

	EmpID	Name	JobTitle	Department	ManagerID	Salary	Status	City	State	Country	BirthDate	HireDate	TermDate	Street_Number	Street_Name
20	120120	Peiris, Krishna	Electrician II	Engineering	120104	$34,556.25	M	Sydney		AU	-3528	9132	.	46	George Street
21	120121	Elvish, Irenie	Sales Rep. II	Sales	120102	$33,250.00	M	Sydney		AU	-3439	9132	.	3	Mundi Place
22	120122	Ngan, Christina	Sales Rep. II	Sales	120102	$34,343.75	S	Melbourne		AU	208	10774	.	11	Prospect Hill Road
23	120123	Hotstone, Kimiko	Sales Rep. I	Sales	120102	$32,737.50	M	Sydney		AU	3923	13423	20484	.	
24	120124	Daymond, Lucian	Sales Rep. I	Sales	120102	$33,100.00	M	Sydney		AU	1959	11017	.	1	Julius Avenue
25	120125	Hofmeister, Fong	Sales Rep. IV	Sales	120102	$40,050.00	M	Sydney		AU	340	11017	20300	.	
26	120126	Denny, Satyakam	Sales Rep. II	Sales	120102	$33,475.00	O	Sydney		AU	12681	21032	.	7	Boundary Street
27	120127	Clarkson, Sharryn	Sales Rep. II	Sales	120102	$35,125.00	M	Sydney		AU	9135	18202	.	21	Albert St

/*Programming Question 5.03

If necessary, start SAS Studio and submit libname.sas. Open p205q3.sas from the programs folder.

Fix the program to ensure that the following actions occur:

The donation table contains all EmpID values that are in both the cr.employee and cr.employee_donations tables.

TotalDonation should be calculated in the donation table.

Create an additional output table named nodonation that includes EmpID values that are in the cr.employee table

but not in the cr.employee_donations table.

How many rows are in the donation table?

What is the TotalDonation value for Hill, Ronald?

How many rows are in the nodonation table?

```
*/
proc sort data=cr.employee(keep=EmpID Name Department) out=emp_sort;
    by EmpID;
run;


proc sort data=cr.employee_donations out=donate_sort;
    by EmpID;
run;


data donation nodonation(keep=Name Department);
    merge emp_sort(in=in_emp) donate_sort(in=in_don);
    by EmpID;
    if in_don=1 and in_emp=1 then do;
        TotalDonation=sum(of Qtr1-Qtr4);
        output donation;
    end;
    if in_don=0 and in_emp=1 then output nodonation;
run;
```

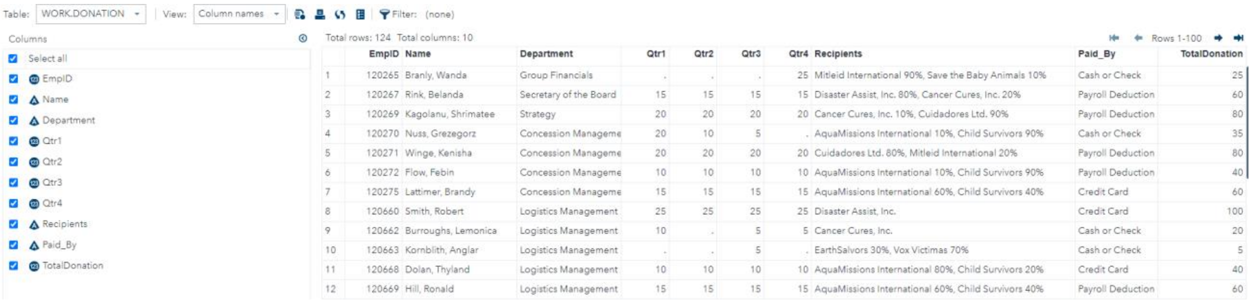


Table: WORK.DONATION View: Column names Filter: (none)

Columns

☑ Select all
☑ EmpID
☑ Name
☑ Department
☑ Qtr1
☑ Qtr2
☑ Qtr3
☑ Qtr4
☑ Recipients
☑ Paid_By
☑ TotalDonation

Total rows: 124 Total columns: 10

Rows 1-100

	EmpID	Name	Department	Qtr1	Qtr2	Qtr3	Qtr4	Recipients	Paid_By	TotalDonation
1	120265	Branly, Wanda	Group Financials	.	.	.	25	Mitleid International 90%, Save the Baby Animals 10%	Cash or Check	25
2	120267	Rink, Belanda	Secretary of the Board	15	15	15	15	Disaster Assist, Inc. 80%, Cancer Cures, Inc. 20%	Payroll Deduction	60
3	120269	Kagolanu, Shrimatee	Strategy	20	20	20	20	Cancer Cures, Inc. 10%, Cuidadores Ltd. 90%	Payroll Deduction	80
4	120270	Nuss, Grezegorz	Concession Manageme	20	10	5	.	AquaMissions International 10%, Child Survivors 90%	Cash or Check	35
5	120271	Winge, Kenisha	Concession Manageme	20	20	20	20	Cuidadores Ltd. 80%, Mitleid International 20%	Payroll Deduction	80
6	120272	Flow, Febin	Concession Manageme	10	10	10	10	AquaMissions International 10%, Child Survivors 90%	Payroll Deduction	40
7	120275	Lattimer, Brandy	Concession Manageme	15	15	15	15	AquaMissions International 60%, Child Survivors 40%	Credit Card	60
8	120660	Smith, Robert	Logistics Management	25	25	25	25	Disaster Assist, Inc.	Credit Card	100
9	120662	Burroughs, Lemonica	Logistics Management	10	.	5	5	Cancer Cures, Inc.	Cash or Check	20
10	120663	Kornblith, Anglar	Logistics Management	.	.	5	.	EarthSalvors 30%, Vox Victimas 70%	Cash or Check	5
11	120668	Dolan, Thyland	Logistics Management	10	10	10	10	AquaMissions International 80%, Child Survivors 20%	Credit Card	40
12	120669	Hill, Ronald	Logistics Management	15	15	15	15	AquaMissions International 60%, Child Survivors 40%	Payroll Deduction	60

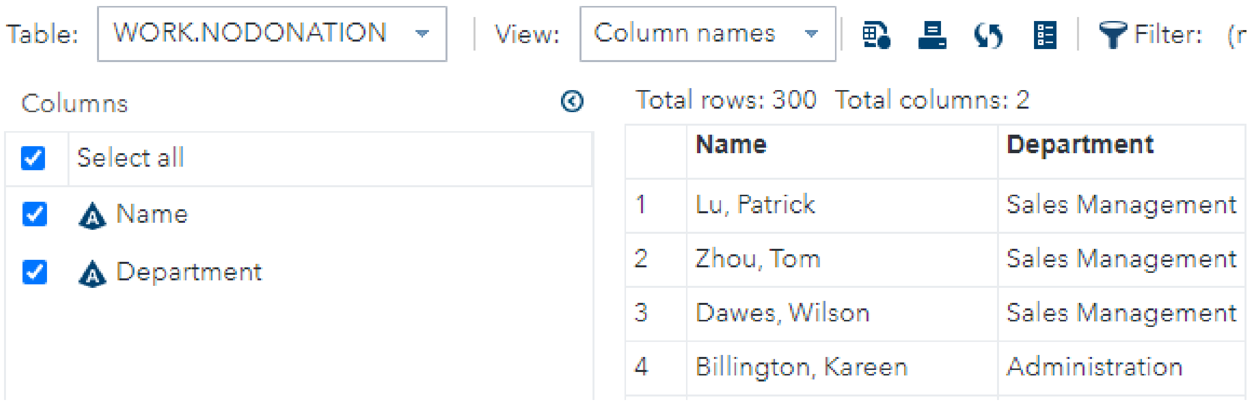


Table: WORK.NODONATION View: Column names Filter: (r

Columns

☑ Select all
☑ Name
☑ Department

Total rows: 300 Total columns: 2

	Name	Department
1	Lu, Patrick	Sales Management
2	Zhou, Tom	Sales Management
3	Dawes, Wilson	Sales Management
4	Billington, Kareen	Administration

```
/* Using Iterative and Conditional DO Loops */
data profit_forecast;
        set cr.profit_summary;
        do Year=1 to 3;
        *do Year=1, 2, 3;
                TotalProfit=TotalProfit*1.05;
                Output;
        end;
run;

*Original;
data profit_forecast;
        set cr.profit_summary;
        Year=1;
        TotalProfit=TotalProfit*1.05;
        Output;
        Year=2;
        TotalProfit=TotalProfit*1.05;
        Output;
        Year=3;
        TotalProfit=TotalProfit*1.05;
        Output;
run;
```

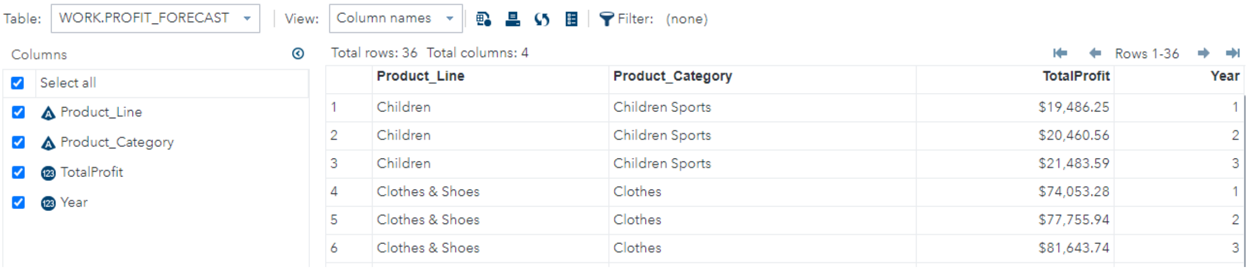


Table: WORK.PROFIT_FORECAST | View: Column names | Filter: (none)

Columns

☑ Select all
☑ Product_Line
☑ Product_Category
☑ TotalProfit
☑ Year

Total rows: 36 Total columns: 4

Rows 1-36

	Product_Line	Product_Category	TotalProfit	Year
1	Children	Children Sports	$19,486.25	1
2	Children	Children Sports	$20,460.56	2
3	Children	Children Sports	$21,483.59	3
4	Clothes & Shoes	Clothes	$74,053.28	1
5	Clothes & Shoes	Clothes	$77,755.94	2
6	Clothes & Shoes	Clothes	$81,643.74	3

```
data rewards;
        Reserve=1000000;
        do Year=1 to 10;
                Reserve+25000;
                Reserve+(Reserve*0.03);
                format Reserve dollar12.;
                output;
        end;
run;
```

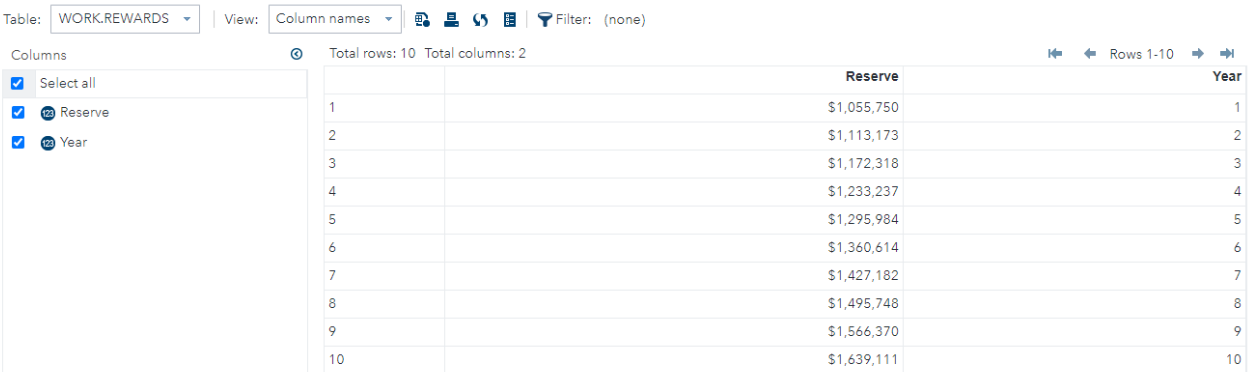


Table: WORK.REWARDS View: Column names Filter: (none)

Columns

☑ Select all

☑ Reserve

☑ Year

Total rows: 10 Total columns: 2 Rows 1-10

	Reserve	Year
1	$1,055,750	1
2	$1,113,173	2
3	$1,172,318	3
4	$1,233,237	4
5	$1,295,984	5
6	$1,360,614	6
7	$1,427,182	7
8	$1,495,748	8
9	$1,566,370	9
10	$1,639,111	10

```
*Practice2;
data rewards;
        Reserve=1000000;
        Rewards=-100000;
        do Year=1 to 10;
                Reserve+25000;
                Reserve+(Reserve*0.03);
                Reserve+Rewards;
                format Reserve Rewards dollar12.;
                output;
        end;
run;
```

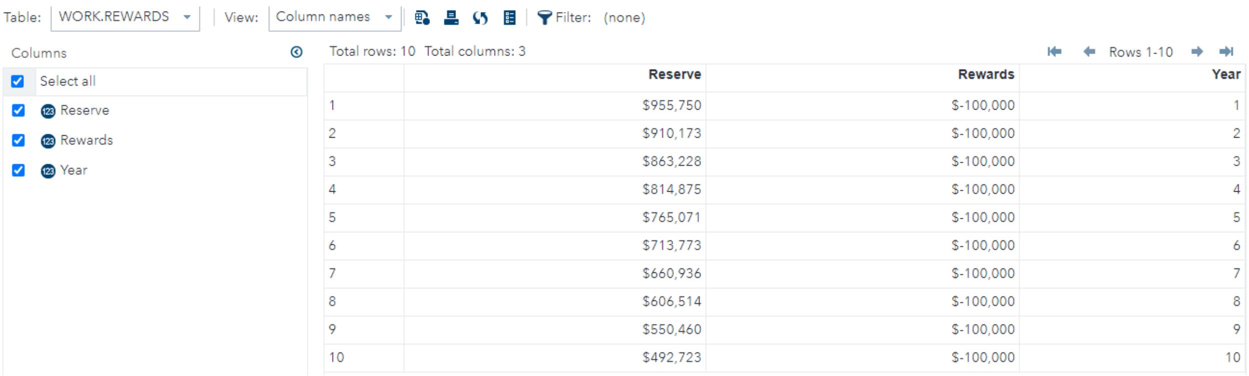


Table: WORK.REWARDS View: Column names Filter: (none)

Columns

☑ Select all

☑ Reserve

☑ Rewards

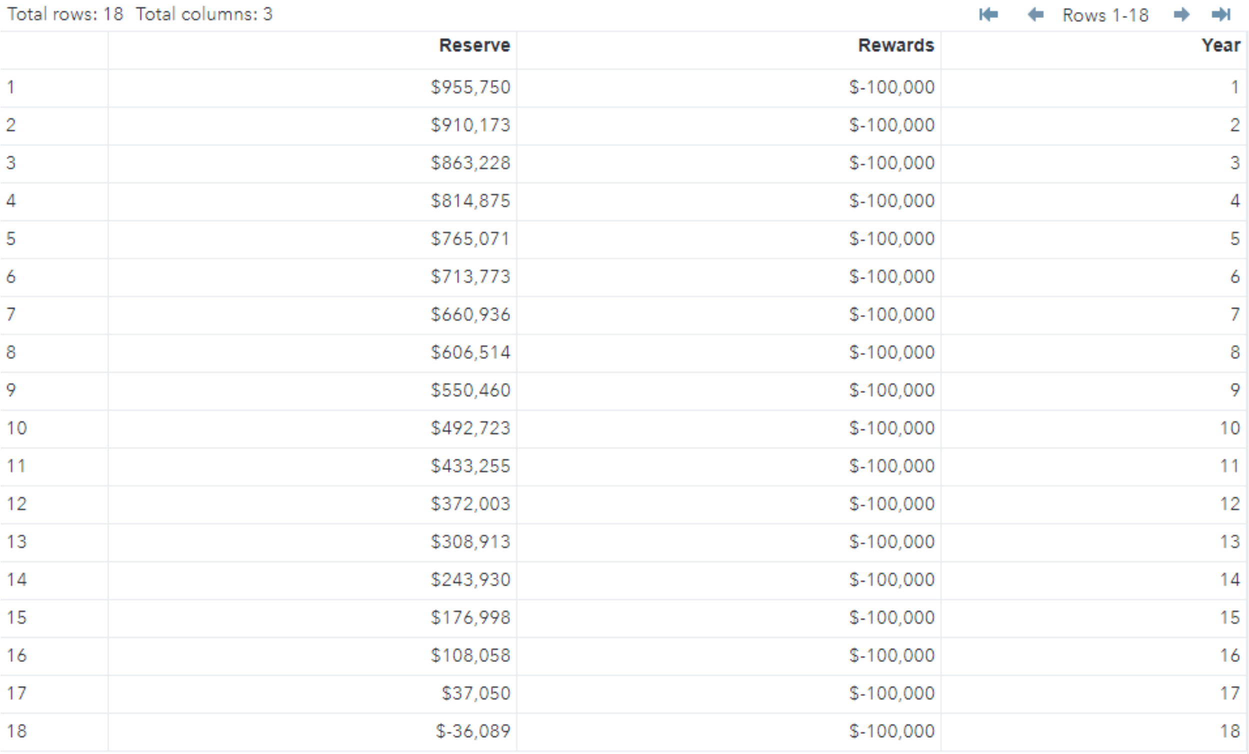
☑ Year

Total rows: 10 Total columns: 3

Rows 1-10

	Reserve	Rewards	Year
1	$955,750	$-100,000	1
2	$910,173	$-100,000	2
3	$863,228	$-100,000	3
4	$814,875	$-100,000	4
5	$765,071	$-100,000	5
6	$713,773	$-100,000	6
7	$660,936	$-100,000	7
8	$606,514	$-100,000	8
9	$550,460	$-100,000	9
10	$492,723	$-100,000	10

```
*Practice3;

data rewards;

        Reserve=1000000;

        Rewards=-100000;

        do while(Reserve>0);

                Year+1;

                Reserve+25000;

                Reserve+(Reserve*0.03);

                Reserve+Rewards;

                format Reserve Rewards dollar12.;

                output;

        end;

run;
```



Total rows: 18 Total columns: 3

	Reserve	Rewards	Year
1	$955,750	$-100,000	1
2	$910,173	$-100,000	2
3	$863,228	$-100,000	3
4	$814,875	$-100,000	4
5	$765,071	$-100,000	5
6	$713,773	$-100,000	6
7	$660,936	$-100,000	7
8	$606,514	$-100,000	8
9	$550,460	$-100,000	9
10	$492,723	$-100,000	10
11	$433,255	$-100,000	11
12	$372,003	$-100,000	12
13	$308,913	$-100,000	13
14	$243,930	$-100,000	14
15	$176,998	$-100,000	15
16	$108,058	$-100,000	16
17	$37,050	$-100,000	17
18	$-36,089	$-100,000	18

```
*Practice4;

data rewards;

        Reserve=1000000;

        Rewards=-100000;

        do until(Reserve<0);

                Year+1;

                do Month=1 to 12;

                        Reserve+2500;

                        Reserve+(Reserve*0.03/12);

                end;

                Reserve+Rewards;

                output;

        end;

        format Reserve Rewards dollar12.;

run;
```

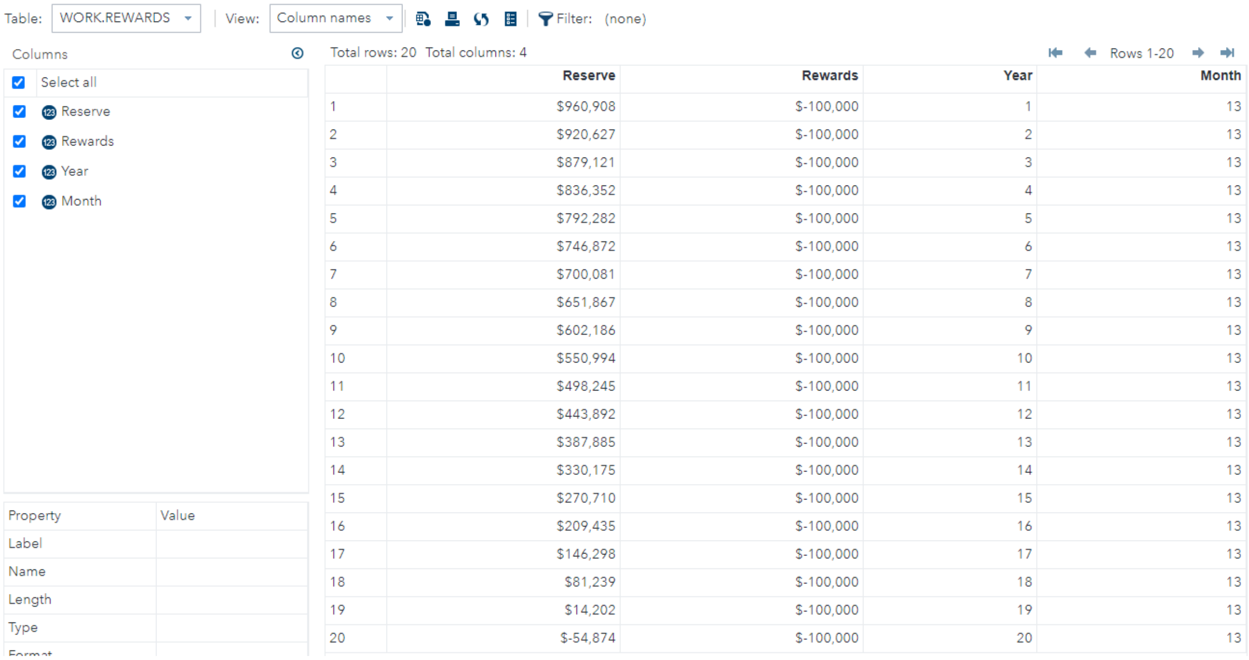


Table: WORK.REWARDS View: Column names Filter: (none)

Columns

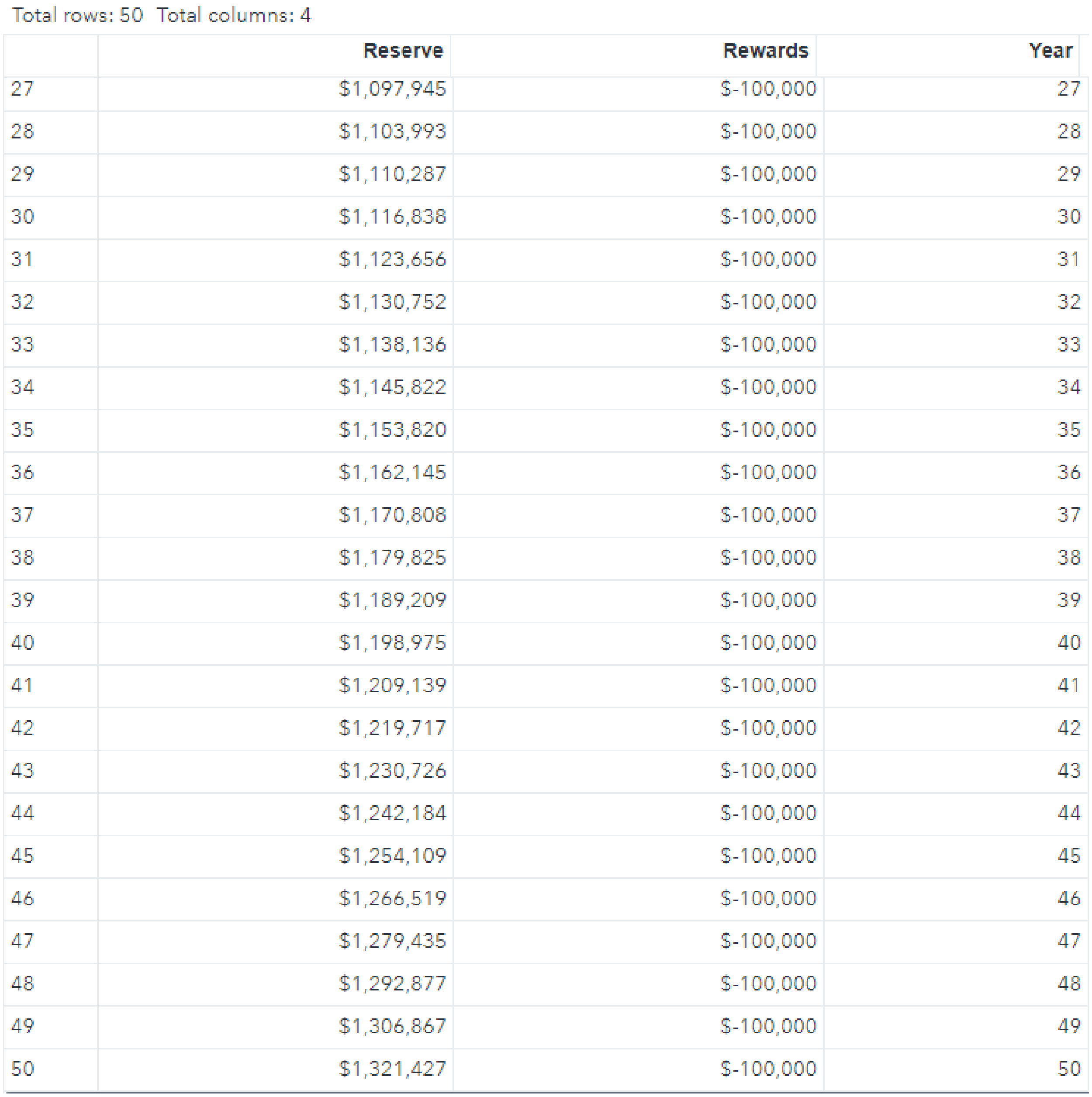
☑ Select all
☑ Reserve
☑ Rewards
☑ Year
☑ Month

Property	Value
Label	
Name	
Length	
Type	
Format	

Total rows: 20 Total columns: 4 Rows 1-20

	Reserve	Rewards	Year	Month
1	$960,908	$-100,000	1	13
2	$920,627	$-100,000	2	13
3	$879,121	$-100,000	3	13
4	$836,352	$-100,000	4	13
5	$792,282	$-100,000	5	13
6	$746,872	$-100,000	6	13
7	$700,081	$-100,000	7	13
8	$651,867	$-100,000	8	13
9	$602,186	$-100,000	9	13
10	$550,994	$-100,000	10	13
11	$498,245	$-100,000	11	13
12	$443,892	$-100,000	12	13
13	$387,885	$-100,000	13	13
14	$330,175	$-100,000	14	13
15	$270,710	$-100,000	15	13
16	$209,435	$-100,000	16	13
17	$146,298	$-100,000	17	13
18	$81,239	$-100,000	18	13
19	$14,202	$-100,000	19	13
20	$-54,874	$-100,000	20	13

```
*Practice5;
data rewards;
        Reserve=1000000;
        Rewards=-100000;
        do Year=1 to 50 until(Reserve<0);
                do Month=1 to 12;
                        Reserve+5000;
                        Reserve+(Reserve*0.04/12);
                end;
                Reserve+Rewards;
                output;
        end;
        format Reserve Rewards dollar12.;
run;
```

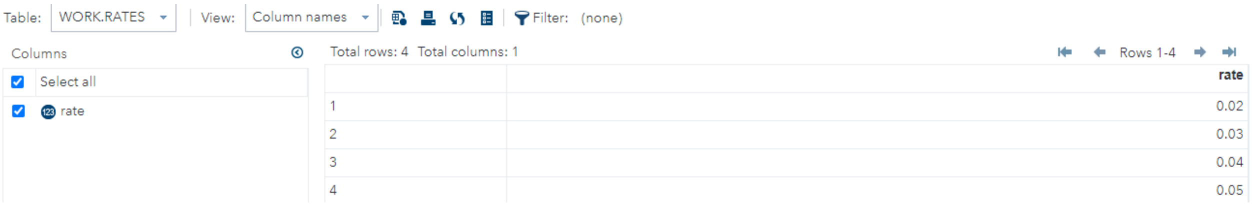


Total rows: 50 Total columns: 4

	Reserve	Rewards	Year
27	$1,097,945	$-100,000	27
28	$1,103,993	$-100,000	28
29	$1,110,287	$-100,000	29
30	$1,116,838	$-100,000	30
31	$1,123,656	$-100,000	31
32	$1,130,752	$-100,000	32
33	$1,138,136	$-100,000	33
34	$1,145,822	$-100,000	34
35	$1,153,820	$-100,000	35
36	$1,162,145	$-100,000	36
37	$1,170,808	$-100,000	37
38	$1,179,825	$-100,000	38
39	$1,189,209	$-100,000	39
40	$1,198,975	$-100,000	40
41	$1,209,139	$-100,000	41
42	$1,219,717	$-100,000	42
43	$1,230,726	$-100,000	43
44	$1,242,184	$-100,000	44
45	$1,254,109	$-100,000	45
46	$1,266,519	$-100,000	46
47	$1,279,435	$-100,000	47
48	$1,292,877	$-100,000	48
49	$1,306,867	$-100,000	49
50	$1,321,427	$-100,000	50

```
*Practice6;

data rates;

        input rate;

        datalines;

.02

.03

.04
```

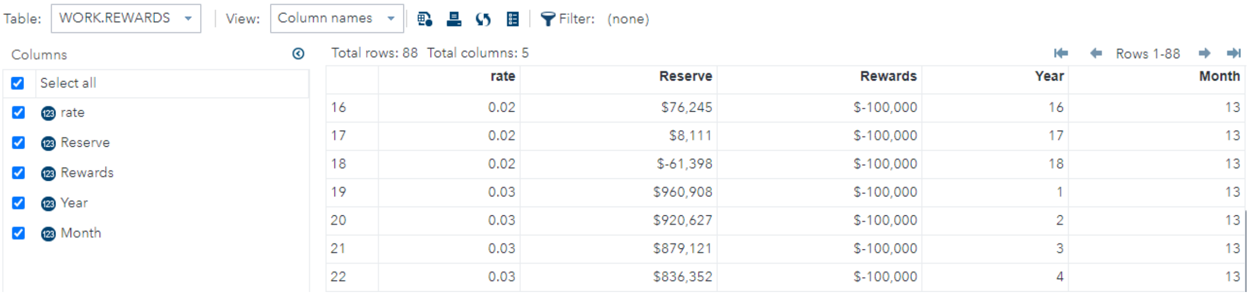
```
.05

;

run;
```



Total rows: 4 Total columns: 1

	rate
1	0.02
2	0.03
3	0.04
4	0.05

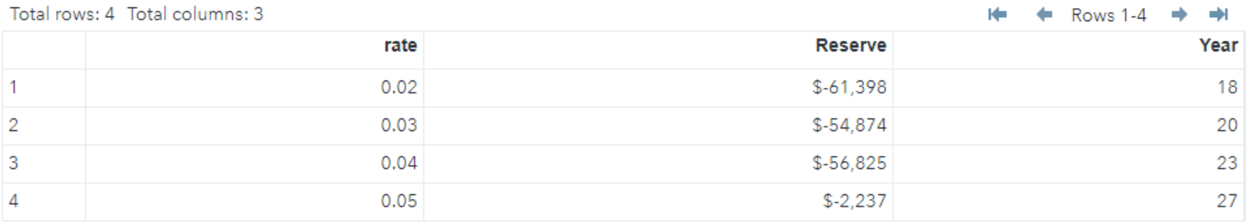
```
data rewards;

        set rates;

        Reserve=1000000;

        Rewards=-100000;

        do Year=1 to 50 until(Reserve<0);

                do Month=1 to 12;

                        Reserve+2500;

                        Reserve+(Reserve*rate/12);

                end;

                Reserve+Rewards;

                output;

        end;

        format Reserve Rewards dollar12.;

run;
```



Total rows: 88 Total columns: 5

	rate	Reserve	Rewards	Year	Month
16	0.02	$76,245	$-100,000	16	13
17	0.02	$8,111	$-100,000	17	13
18	0.02	$-61,398	$-100,000	18	13
19	0.03	$960,908	$-100,000	1	13
20	0.03	$920,627	$-100,000	2	13
21	0.03	$879,121	$-100,000	3	13
22	0.03	$836,352	$-100,000	4	13

```
*Practice7;

data rewards;

        set rates;
        Reserve=1000000;

        Rewards=-100000;

        do Year=1 to 50 until(Reserve<0);

                do Month=1 to 12;

                        Reserve+2500;

                        Reserve+(Reserve*rate/12);

                end;

                Reserve+Rewards;

        end;

        format Reserve Rewards dollar12.;

        drop Rewards Month;

run;
```

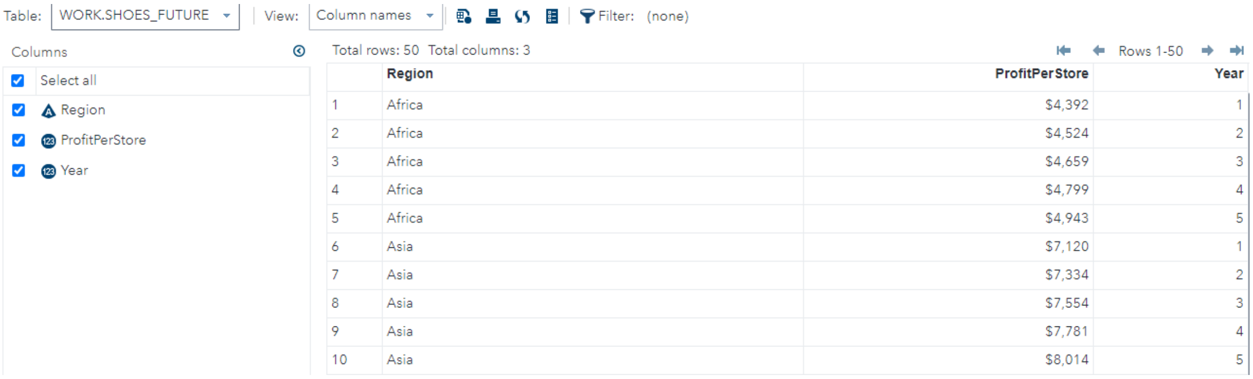


Total rows: 4 Total columns: 3

	rate	Reserve	Year
1	0.02	$-61,398	18
2	0.03	$-54,874	20
3	0.04	$-56,825	23
4	0.05	$-2,237	27

```
/* Programming Question 5.04

If necessary, start SAS Studio and submit libname.sas. Write a new program to do the following:

Read the cr.shoes_summary table and create a new table named shoes_future.

For every row read from the input table, write five rows to the output table.

Create a column named Year that will be 1 through 5 for the five rows.

For each year, increase ProfitPerStore by 3%.

Drop the TotalStores and TotalProfit columns.

How many rows are in the shoes_future table?

What is the value of ProfitPerStore for Asia in year 5? Note: Type only numbers in your answer.

*/

data shoes_future;

        set cr.shoes_summary;

        do Year=1 to 5;
```

```
                ProfitPerStore+(ProfitPerStore*0.03);
                output;
        end;
        drop TotalStores TotalProfit;
run;


*Solution;
data shoes_future;
  set cr.shoes_summary;
  do year=1 to 5;
    ProfitPerStore=ProfitPerStore*1.03;
    output;
  end;
  drop Total:;
run;
```



	Region	ProfitPerStore	Year
1	Africa	$4,392	1
2	Africa	$4,524	2
3	Africa	$4,659	3
4	Africa	$4,799	4
5	Africa	$4,943	5
6	Asia	$7,120	1
7	Asia	$7,334	2
8	Asia	$7,554	3
9	Asia	$7,781	4
10	Asia	$8,014	5

/* Programming Question 5.05

If necessary, start SAS Studio. Open p206q2.sas from the programs folder. Modify the program to do the following:

Insert a DO loop containing a statement to calculate the estimated values of Wages, Retire, and Medical for 10 years.

Assume the estimated annual increase shown in the table below.

Column	Estimated Annual Increase
Wages	6.0%

Retire 1.4%

Medical9.5%

Create a new column name TotalCost as the sum of each year's Wage, Retire, and Medical values. Apply a comma format to Wage, Retire, Medical, and TotalCost that rounds all values to the nearest whole number.

Output one row for each year.

What is the value of Medical when Year equals 5? Note: Type only numbers for your answer.

What is the value of TotalCost when Year equals 10?

*/

```
data future_expenses;
  Wages=12874000;
  Retire=1765000;
  Medical=649000;
 /* insert a DO loop here */
 do Year=1 to 10;
        Wages+(Wages*0.06);
        Retire+(Retire*0.014);
        Medical+(Medical*0.095);
        TotalCost=sum(Wages, Retire, Medical);
        format Wages Retire Medical TotalCost comma16.;
        output;
 end;
run;
```

```
*Solution;
data future_expenses;
   Wages=12874000;
   Retire=1765000;
   Medical=649000;
   do Year=1 to 10;

      Wages=Wages*1.06;

      Retire=Retire*1.014;

      Medical=Medical *1.095;

      TotalCost=sum(Wages,Retire,Medical);

      output;

   end;

   format Wages Retire Medical TotalCost comma12.;

run;
```

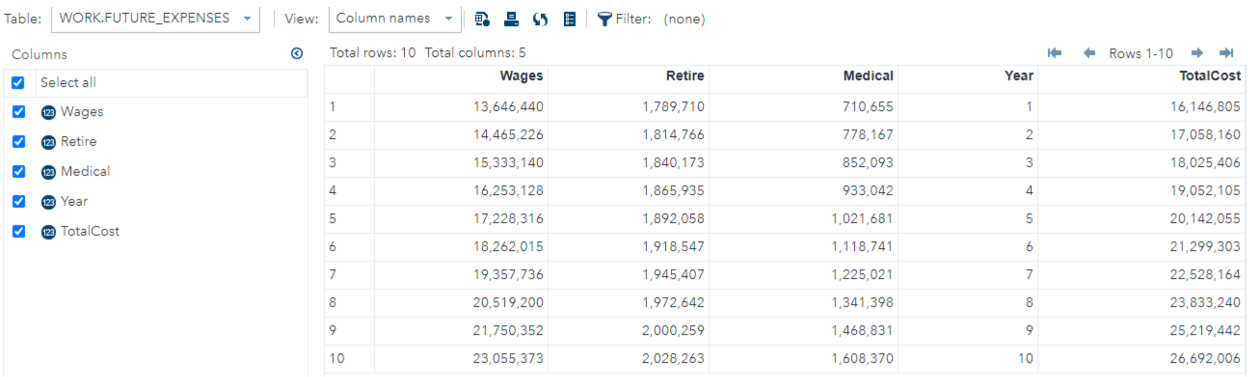


Table: WORK.FUTURE_EXPENSES | View: Column names | Filter: (none)

Columns

☑ Select all
☑ Wages
☑ Retire
☑ Medical
☑ Year
☑ TotalCost

Total rows: 10 Total columns: 5 | Rows 1-10

	Wages	Retire	Medical	Year	TotalCost
1	13,646,440	1,789,710	710,655	1	16,146,805
2	14,465,226	1,814,766	778,167	2	17,058,160
3	15,333,140	1,840,173	852,093	3	18,025,406
4	16,253,128	1,865,935	933,042	4	19,052,105
5	17,228,316	1,892,058	1,021,681	5	20,142,055
6	18,262,015	1,918,547	1,118,741	6	21,299,303
7	19,357,736	1,945,407	1,225,021	7	22,528,164
8	20,519,200	1,972,642	1,341,398	8	23,833,240
9	21,750,352	2,000,259	1,468,831	9	25,219,442
10	23,055,373	2,028,263	1,608,370	10	26,692,006

```
/* Programming Question 5.06

If necessary, start SAS Studio. Open p206q3.sas from the programs folder. Modify the program to do the following:

Income begins as 50,000,000. Increase Income by 1% each year.

Add a column named Year that will store the DO loop iteration number.

Change the DO loop so that it stops when TotalCost exceeds Income.

How many rows are in the income_expenses table?

What is the value of TotalCost when TotalCost exceeds Income? Note: Type only numbers in your answer.

*/

data income_expenses;

          Wages=12874000;

          Retire=1765000;

          Medical=649000;
```

```
       Income=50000000;
       do Year=1 to 100 until(TotalCost>Income);
              Income=Income*1.01;
              Wages=Wages*1.06;
              Retire=Retire*1.014;
              Medical=Medical *1.095;
              TotalCost=sum(Wages, Retire, Medical);
              output;
       end;
       keep Year TotalCost;
       format TotalCost comma12.;
run;

*Solution;
data income_expenses;
  Wages=12874000;
  Retire=1765000;
  Medical=649000;
  Income=50000000;

  do until (TotalCost > Income);
    year+1;
    Wages=Wages * 1.06;
    Retire=Retire*1.014;
    Medical=Medical *1.095;
    TotalCost=sum(Wages, Retire, Medical);
    income=Income *1.01;
    output;
  end;
```

```
   keep Year Income TotalCost;
   format Income TotalCost comma12.;
run;
```

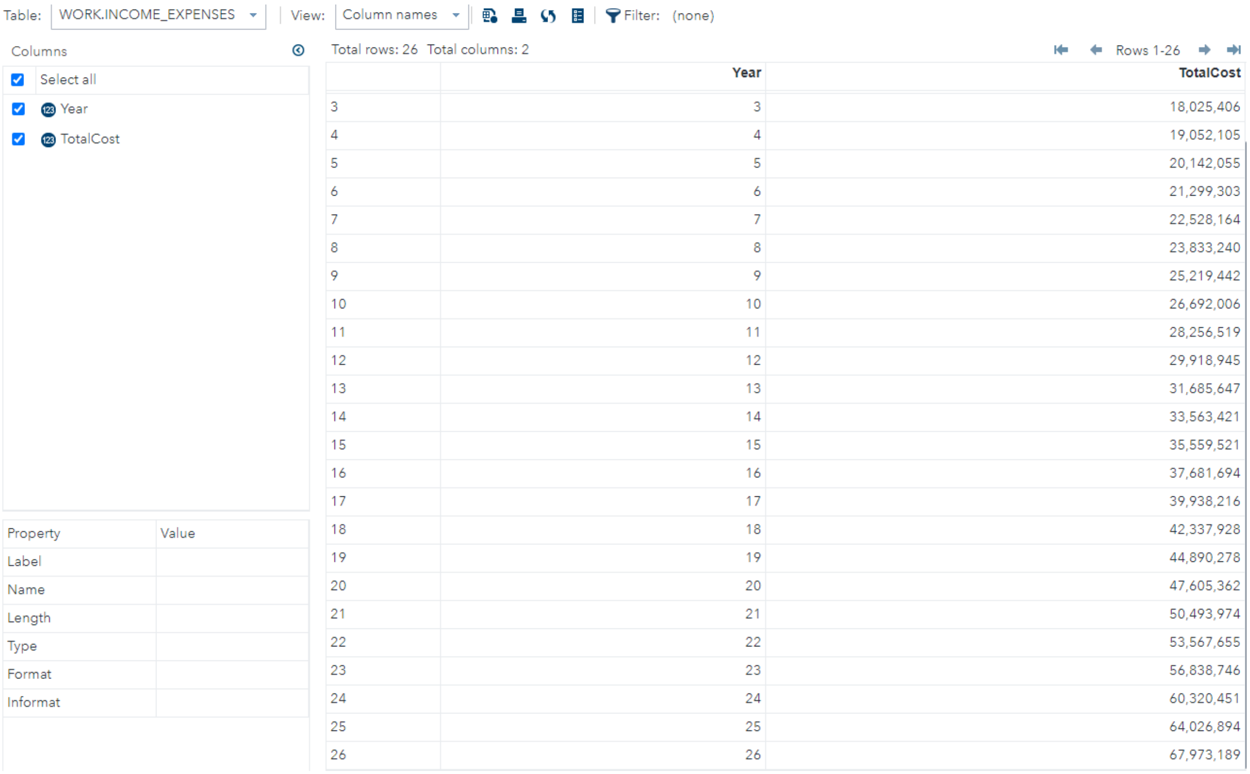


Table: WORK.INCOME_EXPENSES View: Column names Filter: (none)

Columns

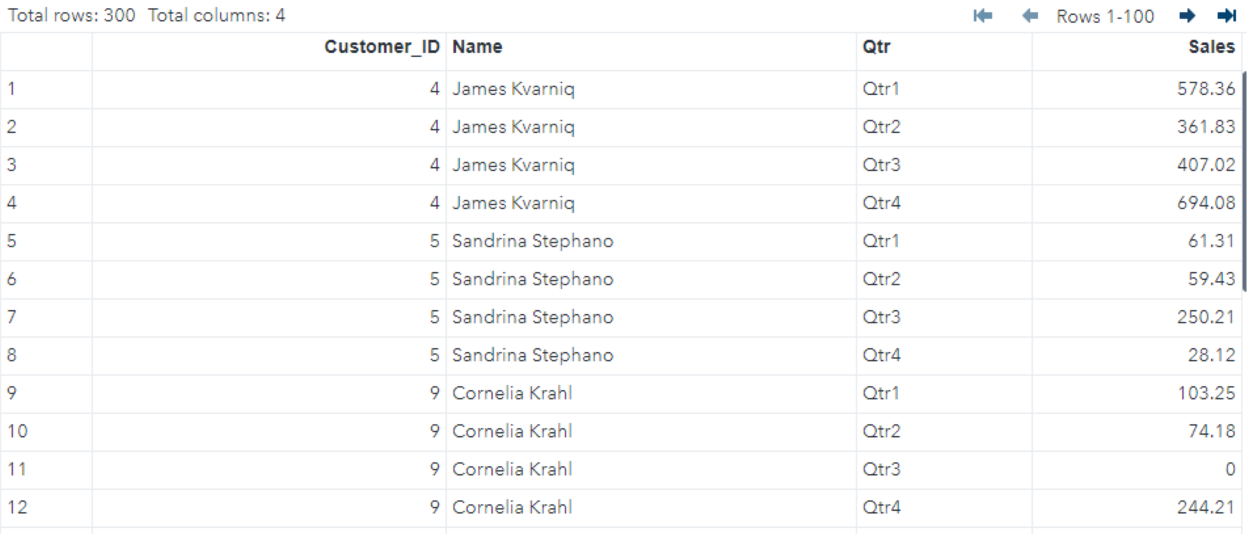
☑ Select all
☑ Year
☑ TotalCost

Property	Value
Label	
Name	
Length	
Type	
Format	
Informat	

Total rows: 26 Total columns: 2 Rows 1-26

	Year	TotalCost
3	3	18,025,406
4	4	19,052,105
5	5	20,142,055
6	6	21,299,303
7	7	22,528,164
8	8	23,833,240
9	9	25,219,442
10	10	26,692,006
11	11	28,256,519
12	12	29,918,945
13	13	31,685,647
14	14	33,563,421
15	15	35,559,521
16	16	37,681,694
17	17	39,938,216
18	18	42,337,928
19	19	44,890,278
20	20	47,605,362
21	21	50,493,974
22	22	53,567,655
23	23	56,838,746
24	24	60,320,451
25	25	64,026,894
26	26	67,973,189

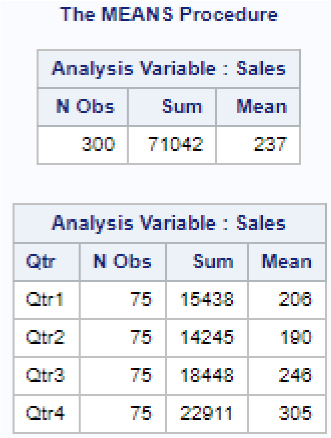
```
/* Restructuring tables with DATA Steps */
data sales_n;
        set cr.qtr_sales;
        Qtr="Qtr1";
        Sales=Qtr1;
        output;
        Qtr="Qtr2";
        Sales=Qtr2;
        output;
        Qtr="Qtr3";
        Sales=Qtr3;
        output;
        Qtr="Qtr4";
        Sales=Qtr4;
        output;
        keep Customer_ID Name Qtr Sales;
run;
```



Total rows: 300 Total columns: 4

	Customer_ID	Name	Qtr	Sales
1	4	James Kvarniq	Qtr1	578.36
2	4	James Kvarniq	Qtr2	361.83
3	4	James Kvarniq	Qtr3	407.02
4	4	James Kvarniq	Qtr4	694.08
5	5	Sandrina Stephano	Qtr1	61.31
6	5	Sandrina Stephano	Qtr2	59.43
7	5	Sandrina Stephano	Qtr3	250.21
8	5	Sandrina Stephano	Qtr4	28.12
9	9	Cornelia Krahl	Qtr1	103.25
10	9	Cornelia Krahl	Qtr2	74.18
11	9	Cornelia Krahl	Qtr3	0
12	9	Cornelia Krahl	Qtr4	244.21

```
proc means data=sales_n sum mean maxdec=0;
        var Sales;
        Class Qtr;
        ways 0 1;
run;
```



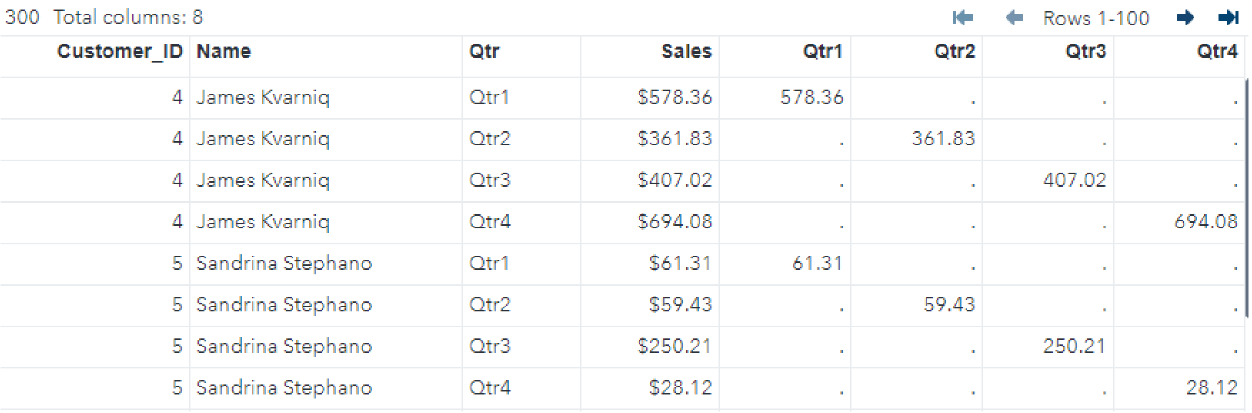
The MEANS Procedure

Analysis Variable : Sales		
N Obs	Sum	Mean
300	71042	237

Analysis Variable : Sales			
Qtr	N Obs	Sum	Mean
Qtr1	75	15438	206
Qtr2	75	14245	190
Qtr3	75	18448	246
Qtr4	75	22911	305

```
/* Restructuring tables narrow to wide with DATA Steps */
data sales_w;
```

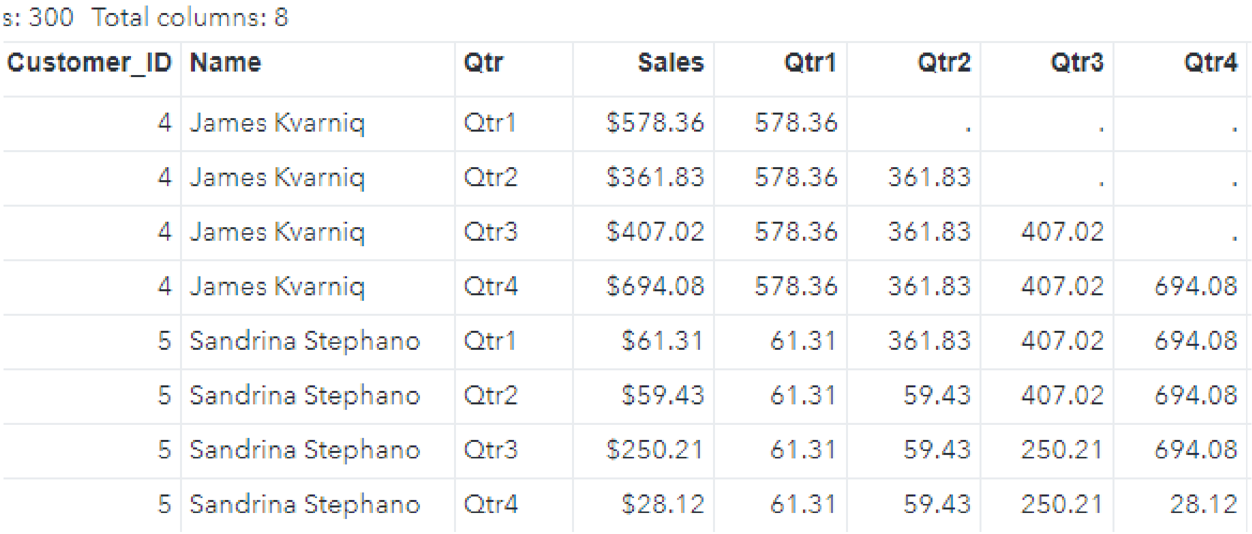
```
        set cr.sales;
        if Qtr="Qtr1" then Qtr1=Sales;
        else if Qtr="Qtr2" then Qtr2=Sales;
        else if Qtr="Qtr3" then Qtr3=Sales;
        else if Qtr="Qtr4" then Qtr4=Sales;
run;
```



300 Total columns: 8

Customer_ID	Name	Qtr	Sales	Qtr1	Qtr2	Qtr3	Qtr4
4	James Kvarniq	Qtr1	$578.36	578.36	.	.	.
4	James Kvarniq	Qtr2	$361.83	.	361.83	.	.
4	James Kvarniq	Qtr3	$407.02	.	.	407.02	.
4	James Kvarniq	Qtr4	$694.08	.	.	.	694.08
5	Sandrina Stephano	Qtr1	$61.31	61.31	.	.	.
5	Sandrina Stephano	Qtr2	$59.43	.	59.43	.	.
5	Sandrina Stephano	Qtr3	$250.21	.	.	250.21	.
5	Sandrina Stephano	Qtr4	$28.12	.	.	.	28.12

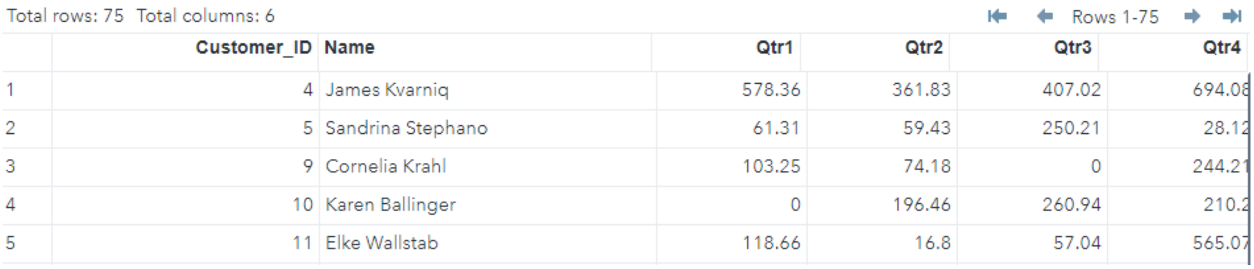
```
*Practice1;
data sales_w;
        set cr.sales;
        retain Qtr1-Qtr4;
        if Qtr="Qtr1" then Qtr1=Sales;
        else if Qtr="Qtr2" then Qtr2=Sales;
        else if Qtr="Qtr3" then Qtr3=Sales;
        else if Qtr="Qtr4" then Qtr4=Sales;
run;
```



s: 300 Total columns: 8

Customer_ID	Name	Qtr	Sales	Qtr1	Qtr2	Qtr3	Qtr4
4	James Kvarniq	Qtr1	$578.36	578.36	.	.	.
4	James Kvarniq	Qtr2	$361.83	578.36	361.83	.	.
4	James Kvarniq	Qtr3	$407.02	578.36	361.83	407.02	.
4	James Kvarniq	Qtr4	$694.08	578.36	361.83	407.02	694.08
5	Sandrina Stephano	Qtr1	$61.31	61.31	361.83	407.02	694.08
5	Sandrina Stephano	Qtr2	$59.43	61.31	59.43	407.02	694.08
5	Sandrina Stephano	Qtr3	$250.21	61.31	59.43	250.21	694.08
5	Sandrina Stephano	Qtr4	$28.12	61.31	59.43	250.21	28.12

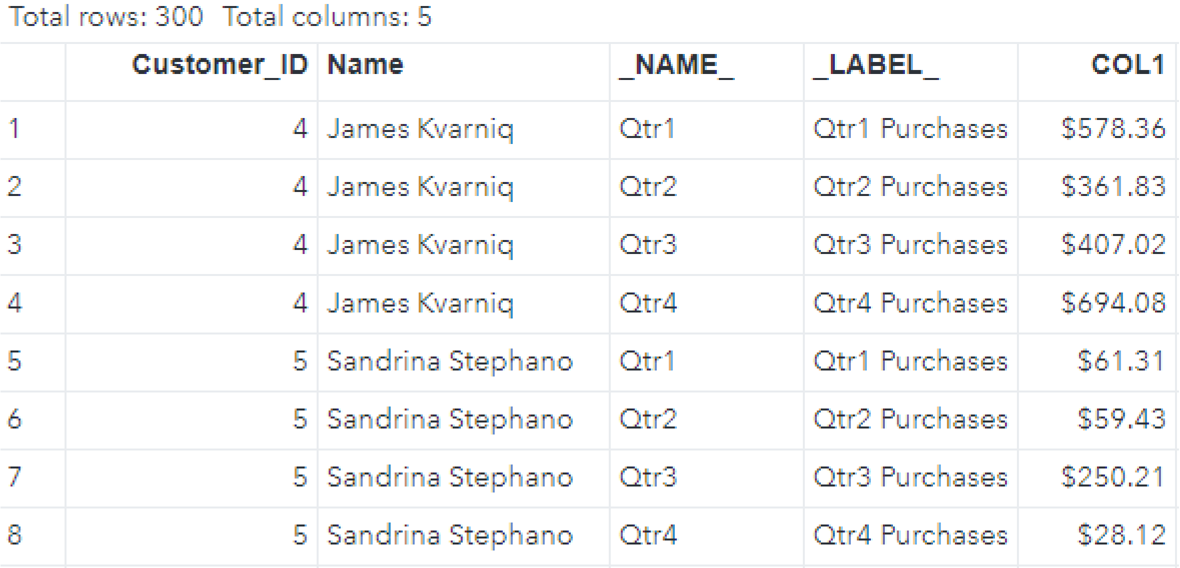
```
*Practice2;
data sales_w;
        set cr.sales;
        by Customer_ID;
        retain Qtr1-Qtr4;
        if Qtr="Qtr1" then Qtr1=Sales;
        else if Qtr="Qtr2" then Qtr2=Sales;
        else if Qtr="Qtr3" then Qtr3=Sales;
        else if Qtr="Qtr4" then Qtr4=Sales;
        if last.Customer_ID=1;
        drop Qtr Sales;
run;
```



Total rows: 75 Total columns: 6

	Customer_ID	Name	Qtr1	Qtr2	Qtr3	Qtr4
1	4	James Kvarniq	578.36	361.83	407.02	694.08
2	5	Sandrina Stephano	61.31	59.43	250.21	28.12
3	9	Cornelia Krahl	103.25	74.18	0	244.21
4	10	Karen Ballinger	0	196.46	260.94	210.2
5	11	Elke Wallstab	118.66	16.8	57.04	565.07

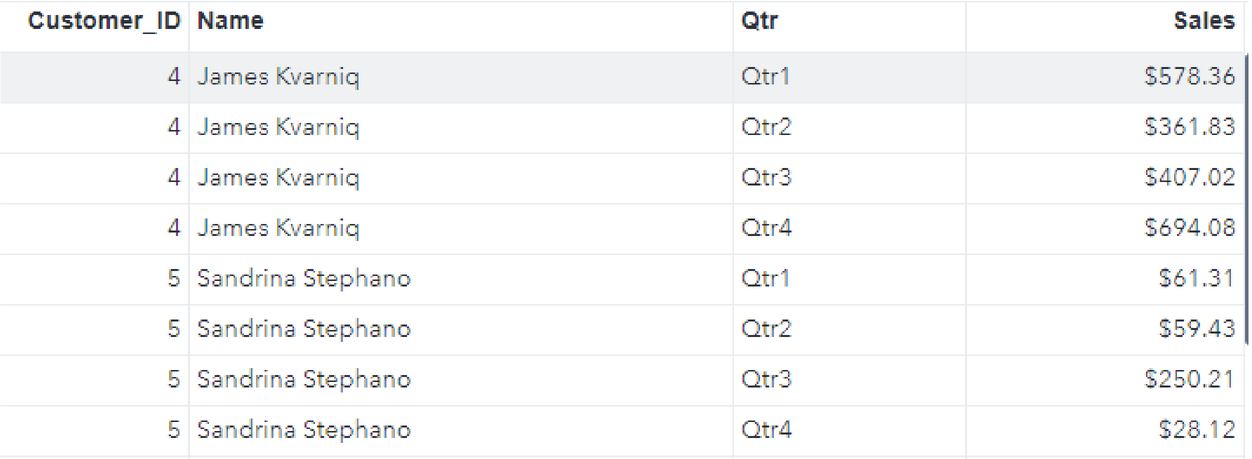
```
proc transpose data=cr.qtr_sales out=sales_n;
        var Qtr:;
        by Customer_ID Name;
run;
```



Total rows: 300 Total columns: 5

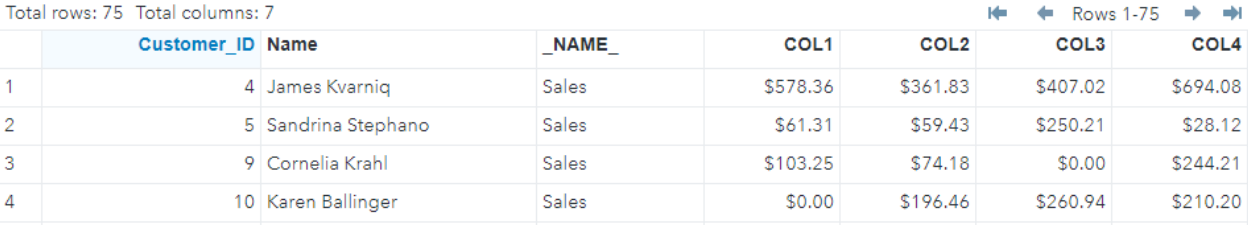
	Customer_ID	Name	_NAME_	_LABEL_	COL1
1	4	James Kvarniq	Qtr1	Qtr1 Purchases	$578.36
2	4	James Kvarniq	Qtr2	Qtr2 Purchases	$361.83
3	4	James Kvarniq	Qtr3	Qtr3 Purchases	$407.02
4	4	James Kvarniq	Qtr4	Qtr4 Purchases	$694.08
5	5	Sandrina Stephano	Qtr1	Qtr1 Purchases	$61.31
6	5	Sandrina Stephano	Qtr2	Qtr2 Purchases	$59.43
7	5	Sandrina Stephano	Qtr3	Qtr3 Purchases	$250.21
8	5	Sandrina Stephano	Qtr4	Qtr4 Purchases	$28.12

```
proc transpose data=cr.qtr_sales out=sales_n(rename=(col1=Sales) drop=_label_) name=Qtr;
        var Qtr:;
        by Customer_ID Name;
run;
```



Customer_ID	Name	Qtr	Sales
4	James Kvarniq	Qtr1	$578.36
4	James Kvarniq	Qtr2	$361.83
4	James Kvarniq	Qtr3	$407.02
4	James Kvarniq	Qtr4	$694.08
5	Sandrina Stephano	Qtr1	$61.31
5	Sandrina Stephano	Qtr2	$59.43
5	Sandrina Stephano	Qtr3	$250.21
5	Sandrina Stephano	Qtr4	$28.12

```
proc transpose data=cr.sales out=sales_w;
        var Sales;
        by Customer_ID Name;
run;
```

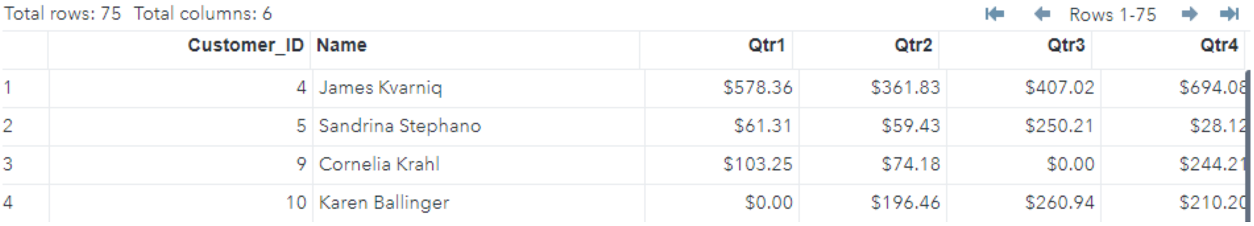


Total rows: 75 Total columns: 7

	Customer_ID	Name	_NAME_	COL1	COL2	COL3	COL4
1	4	James Kvarniq	Sales	$578.36	$361.83	$407.02	$694.08
2	5	Sandrina Stephano	Sales	$61.31	$59.43	$250.21	$28.12
3	9	Cornelia Krahl	Sales	$103.25	$74.18	$0.00	$244.21
4	10	Karen Ballinger	Sales	$0.00	$196.46	$260.94	$210.20

*Practice1;

```
proc transpose data=cr.sales out=sales_w(drop=_name_);
        var Sales;
        by Customer_ID Name;
        id Qtr;
run;
```



Total rows: 75 Total columns: 6

	Customer_ID	Name	Qtr1	Qtr2	Qtr3	Qtr4
1	4	James Kvarniq	$578.36	$361.83	$407.02	$694.08
2	5	Sandrina Stephano	$61.31	$59.43	$250.21	$28.12
3	9	Cornelia Krahl	$103.25	$74.18	$0.00	$244.21
4	10	Karen Ballinger	$0.00	$196.46	$260.94	$210.20

/*Programming Question 5.07

If necessary, start SAS Studio. Open p207q1.sas from the programs folder. Modify the program to do the following:

In the PROC SORT step, remove any rows with duplicate values of Region, Subsidiary, and Product.

Add statements to the PROC TRANSPOSE step to transpose values of Sales into columns based on the values of the Product column.

The resulting table should include one row for each unique combination of Region and Subsidiary.

Drop the _NAME_ and _LABEL_ columns.

How many rows were removed in the PROC SORT step?

How many rows are in the shoe_sales table?

How many columns are in the shoe_sales table?

What is the value of the Sandal column for the Toronto subsidiary in the Canada region?

*/

```
proc sort data=sashelp.shoes out=shoes_sort nodupkey dupout=out;
   by Region Subsidiary Product;
```

```
run;

proc transpose data=shoes_sort out=shoes_sales(drop=_name_ _label_);
   var Sales;
   by Region Subsidiary;
   id Product;
run;

*Original;
proc sort data=sashelp.shoes out=shoes_sort;
         by Region Subsidiary Product;
run;

proc transpose data=shoes_sort out=shoes_sales;
run;
```

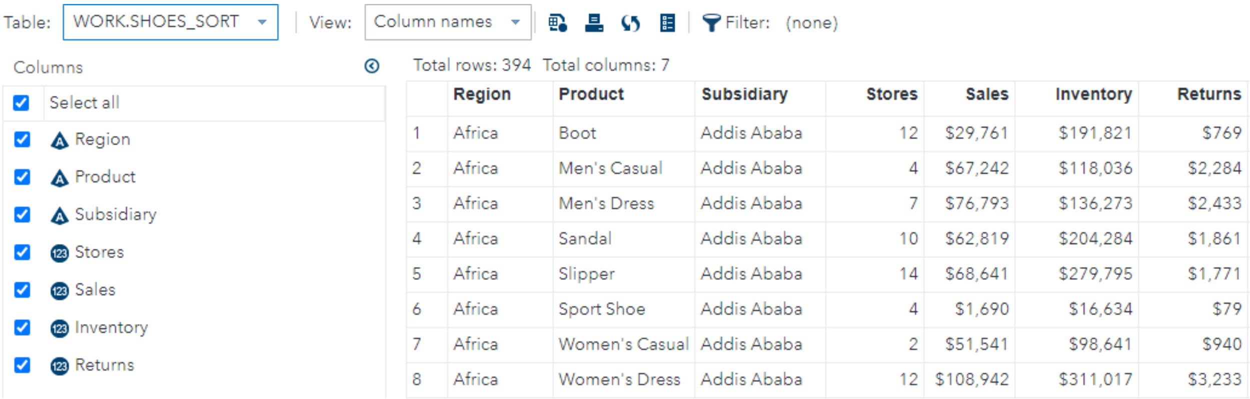


Table: WORK.SHOES_SORT View: Column names Filter: (none)

Columns

☑ Select all
☑ Region
☑ Product
☑ Subsidiary
☑ Stores
☑ Sales
☑ Inventory
☑ Returns

Total rows: 394 Total columns: 7

	Region	Product	Subsidiary	Stores	Sales	Inventory	Returns
1	Africa	Boot	Addis Ababa	12	$29,761	$191,821	$769
2	Africa	Men's Casual	Addis Ababa	4	$67,242	$118,036	$2,284
3	Africa	Men's Dress	Addis Ababa	7	$76,793	$136,273	$2,433
4	Africa	Sandal	Addis Ababa	10	$62,819	$204,284	$1,861
5	Africa	Slipper	Addis Ababa	14	$68,641	$279,795	$1,771
6	Africa	Sport Shoe	Addis Ababa	4	$1,690	$16,634	$79
7	Africa	Women's Casual	Addis Ababa	2	$51,541	$98,641	$940
8	Africa	Women's Dress	Addis Ababa	12	$108,942	$311,017	$3,233

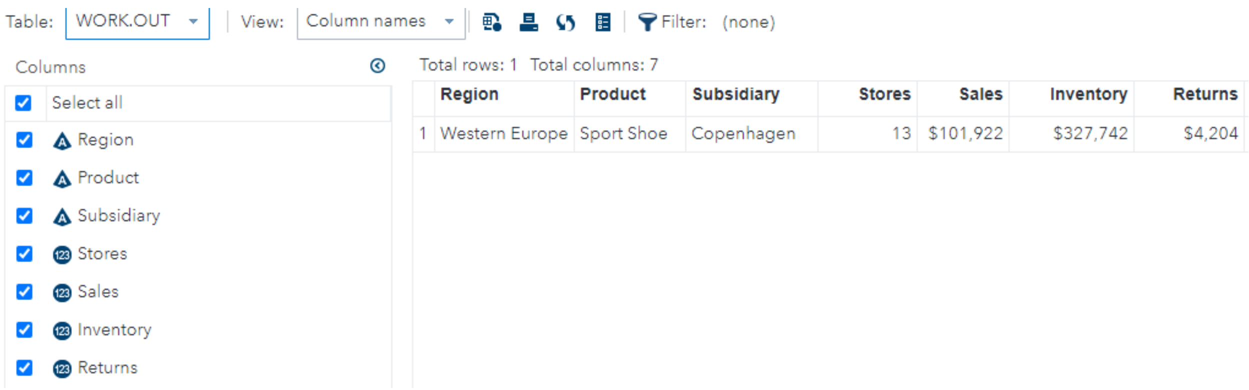


Table: WORK.OUT View: Column names Filter: (none)

Columns

☑ Select all
☑ Region
☑ Product
☑ Subsidiary
☑ Stores
☑ Sales
☑ Inventory
☑ Returns

Total rows: 1 Total columns: 7

	Region	Product	Subsidiary	Stores	Sales	Inventory	Returns
1	Western Europe	Sport Shoe	Copenhagen	13	$101,922	$327,742	$4,204

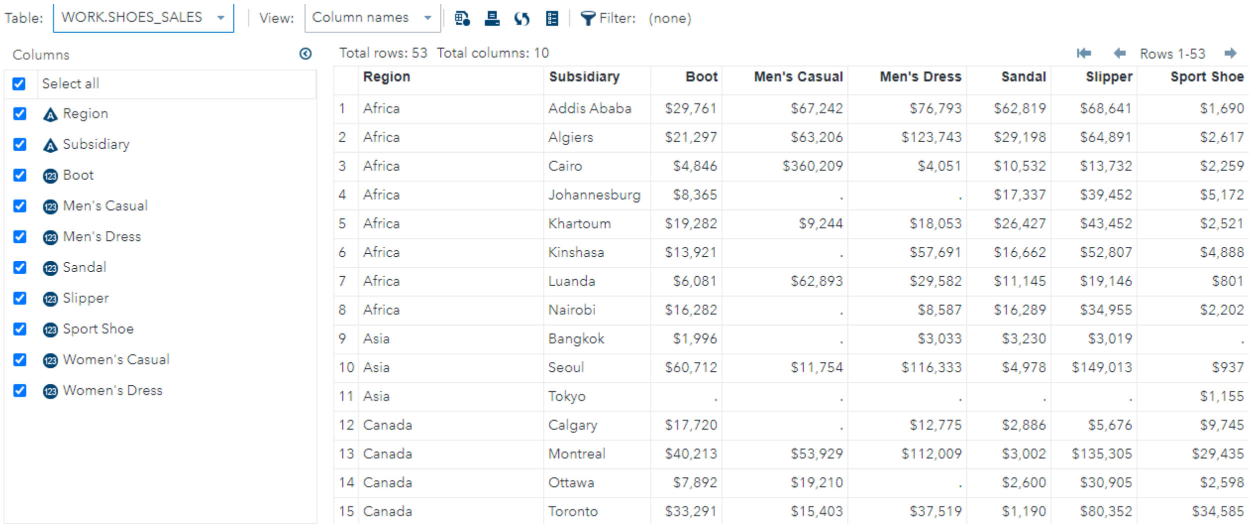


Table: WORK.SHOES_SALES | View: Column names | Filter: (none)

Columns

☑ Select all
☑ Region
☑ Subsidiary
☑ Boot
☑ Men's Casual
☑ Men's Dress
☑ Sandal
☑ Slipper
☑ Sport Shoe
☑ Women's Casual
☑ Women's Dress

Total rows: 53 Total columns: 10 Rows 1-53

	Region	Subsidiary	Boot	Men's Casual	Men's Dress	Sandal	Slipper	Sport Shoe
1	Africa	Addis Ababa	$29,761	$67,242	$76,793	$62,819	$68,641	$1,690
2	Africa	Algiers	$21,297	$63,206	$123,743	$29,198	$64,891	$2,617
3	Africa	Cairo	$4,846	$360,209	$4,051	$10,532	$13,732	$2,259
4	Africa	Johannesburg	$8,365	.	.	$17,337	$39,452	$5,172
5	Africa	Khartoum	$19,282	$9,244	$18,053	$26,427	$43,452	$2,521
6	Africa	Kinshasa	$13,921	.	$57,691	$16,662	$52,807	$4,888
7	Africa	Luanda	$6,081	$62,893	$29,582	$11,145	$19,146	$801
8	Africa	Nairobi	$16,282	.	$8,587	$16,289	$34,955	$2,202
9	Asia	Bangkok	$1,996	.	$3,033	$3,230	$3,019	.
10	Asia	Seoul	$60,712	$11,754	$116,333	$4,978	$149,013	$937
11	Asia	Tokyo	.	.	.	.	.	$1,155
12	Canada	Calgary	$17,720	.	$12,775	$2,886	$5,676	$9,745
13	Canada	Montreal	$40,213	$53,929	$112,009	$3,002	$135,305	$29,435
14	Canada	Ottawa	$7,892	$19,210	.	$2,600	$30,905	$2,598
15	Canada	Toronto	$33,291	$15,403	$37,519	$1,190	$80,352	$34,585

/* Programming Question 5.08

If necessary, start SAS Studio and submit libname.sas. Write a new program to do the following:

Read the cr.employee_training table and create a new table named training_narrow.

Convert the wide table to a narrow table so that there is one row for each date value.

Generate a report that counts the number of courses completed by month (Hint: Use the MONNAME. format.)

How many rows are in the training_narrow table?

How many courses were completed in April?

How many courses have a missing date value?

*/

```
proc sort data=cr.employee_training out=training_sort;

   by Name;

run;
```

/* Either PROC TRANSPOSE or the DATA step will create training_narrow */

```
proc transpose data=training_sort

            out=training_narrow(rename=(col1=Date)) name=Course;

   by Name;
   var Compliance_Training Corporate_Security On_the_Job_Safety;
run;

/* OR */

data training_narrow;
   set training_sort;
   Course="Compliance Training";
   Date=Compliance_Training;
   output;
   Course="Corporate_Security";
   Date=Corporate_Security;
   output;
   Course="On_the_Job_Safety";
   Date=On_the_Job_Safety;
   output;
   drop Compliance_Training Corporate_Security On_the_Job_Safety;
   format Date date9.;
run;

proc freq data=training_narrow;
   tables Date;
   format Date monname.;
run;
```

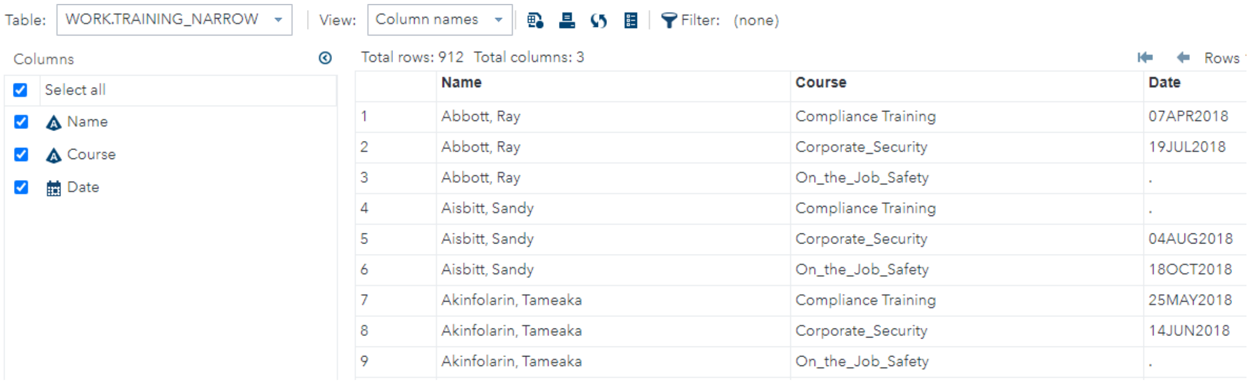
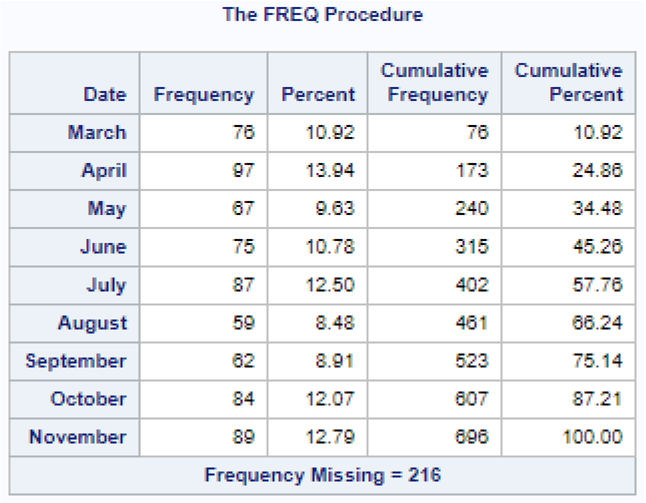


Table: WORK.TRAINING_NARROW View: Column names Filter: (none)

Total rows: 912 Total columns: 3

	Name	Course	Date
1	Abbott, Ray	Compliance Training	07APR2018
2	Abbott, Ray	Corporate_Security	19JUL2018
3	Abbott, Ray	On_the_Job_Safety	.
4	Aisbitt, Sandy	Compliance Training	.
5	Aisbitt, Sandy	Corporate_Security	04AUG2018
6	Aisbitt, Sandy	On_the_Job_Safety	18OCT2018
7	Akinfolarin, Tameaka	Compliance Training	25MAY2018
8	Akinfolarin, Tameaka	Corporate_Security	14JUN2018
9	Akinfolarin, Tameaka	On_the_Job_Safety	.



The FREQ Procedure

Date	Frequency	Percent	Cumulative Frequency	Cumulative Percent
March	76	10.92	76	10.92
April	97	13.94	173	24.86
May	67	9.63	240	34.48
June	75	10.78	315	45.26
July	87	12.50	402	57.76
August	59	8.48	461	66.24
September	62	8.91	523	75.14
October	84	12.07	607	87.21
November	89	12.79	696	100.00
Frequency Missing = 216				