Case Study: Guided Version

This suggested guide uses the steps of the SAS programming process to help you solve the business problem. This is only one of many ways that you could solve this problem.

Begin the case study by opening and running the **StarterProgram.sas**.

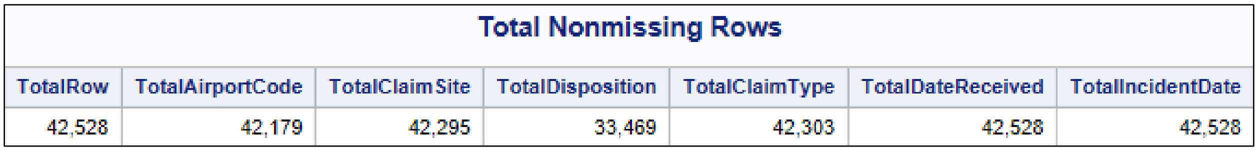
Access Data

1. The tables are created in the **Sq** library after you run the **casestudy_createdata.sas** program.

Explore Data

2. Preview the first ***10 rows*** and the ***descriptor portion*** of the following tables:
   a. **sq.claimsraw** table
   b. **sq.enplanement2017** and **sq.boarding2013_2016** tables
      1) What type is the **Year** column in each table?
      2) What is the column name that holds the value of the number of passengers that boarded a plane in each table?
3. Count the number of nonmissing values in the ***entire table*** and in the following columns:
   a. **Airport_Code**
   b. **Claim_Site**
   c. **Disposition**
   d. **Claim_Type**
   e. **Date_Received**
   f. **Incident_Date**

Results

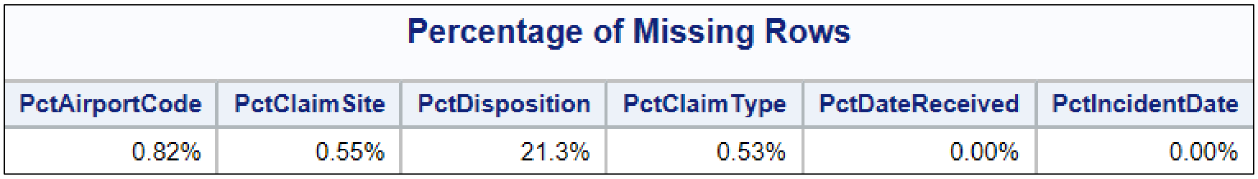


Total Nonmissing Rows

TotalRow	TotalAirportCode	TotalClaim Site	TotalDisposition	TotalClaimType	TotalDateReceived	TotalIncidentDate
42,528	42,179	42,295	33,469	42,303	42,528	42,528

4. In one query, find the percentage of missing values in the following columns:
   a. **Airport_Code**
   b. **Claim_Site**
   c. **Disposition**
   d. **Claim_Type**
   e. **Date_Received**
   f. **Incident_Date**

Results



Percentage of Missing Rows					
PctAirportCode	PctClaimSite	PctDisposition	PctClaimType	PctDateReceived	PctIncidentDate
0.82%	0.55%	21.3%	0.53%	0.00%	0.00%

5. Explore the distinct values of the following columns to determine whether any adjustments are needed. Use the required column values in the **Case Study Data Layout** PDF.

   a. **Claim_Site**

   b. **Disposition**

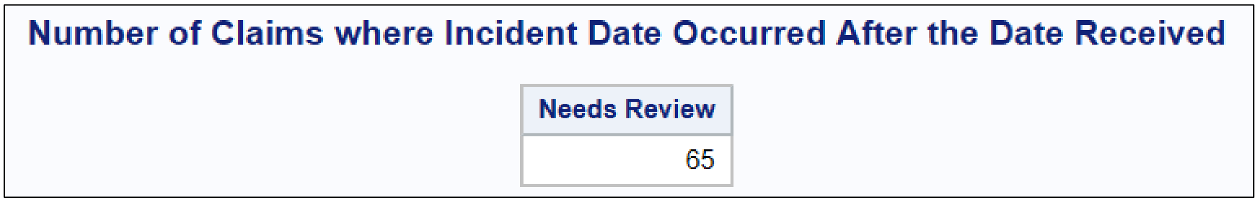
   c. **Claim_Type**

   d. The year from **Date_Received**
   (Hint: Use the PUT function.)

   e. The year from **Incident_Date**
   (Hint: Use the PUT function.)

6. Count the number of rows in which **Incident_Date** occurs **after Date_Received**.

Results



Number of Claims where Incident Date Occurred After the Date Received

Needs Review
65

7. Run a query to view the **Claim_Number**, **Date_Received**, and **Incident_Date** columns in the **sq.claimsraw** table in which **Incident_Date** occurs ***after*** **Date_Received**.

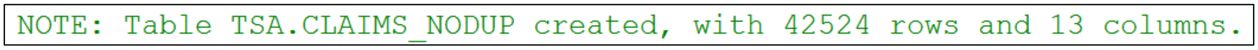
   a. What assumption can you make about the **Date_Received** column values in your results?

Prepare Data

Use the information from the Explore stage to begin preparing the data for analysis.

8. Create a new table named **Claims_NoDup** that removes entirely duplicated rows. A duplicate claim exists if ***every value*** is duplicated.

Log



```
NOTE: Table TSA.CLAIMS_NODUP created, with 42524 rows and 13 columns.
```

9. Using the **Claims_NoDup** table, create a table named **sq.Claims_Cleaned** by doing the following:

   a. Select the **Claim_Number** and **Incident Date** columns.

   b. Fix the 65 date issues that you identified earlier by replacing the year 2017 with 2018 in the **Date_Received** column. (Hint: One method is by using the INTNX function.)

   c. Select the **Airport_Name** column.

   d. Replace missing values in the **Airport_Code** column with the value *Unknown*.

e. Clean the following columns by applying the requirements for the values in the **Case Study Data Layout** PDF:

   1) **Claim_Type**

   2) **Claim_Site**

   3) **Disposition**

f. Select the **Close_Amount** column and format it with a dollar sign. Include two decimal places (for example, $130.28).

g. Select the **State** column and convert all values to uppercase.

h. Select the **StateName**, **County**, and **City** columns. Convert all values to proper case (for example, *Raleigh*).

i. Include only those rows where **Incident_Date** is between 2013 and 2017.

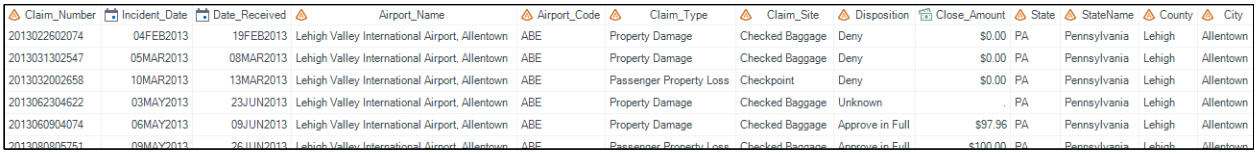
j. Order the results by **Airport_Code** and **Incident_Date**.

k. Assign permanent labels for columns by adding a space between words (for example, Close Amount).

Log

```
NOTE: Table TSA.CLAIMS_CLEANED created, with 42522 rows and 13 columns.
```

Partial Table



Claim_Number	Incident_Date	Date_Received	Airport_Name	Airport_Code	Claim_Type	Claim_Site	Disposition	Close_Amount	State	StateName	County	City
2013022602074	04FEB2013	19FEB2013	Lehigh Valley International Airport, Allentown	ABE	Property Damage	Checked Baggage	Deny	$0.00	PA	Pennsylvania	Lehigh	Allentown
2013031302547	05MAR2013	08MAR2013	Lehigh Valley International Airport, Allentown	ABE	Property Damage	Checked Baggage	Deny	$0.00	PA	Pennsylvania	Lehigh	Allentown
2013032002658	10MAR2013	13MAR2013	Lehigh Valley International Airport, Allentown	ABE	Passenger Property Loss	Checkpoint	Deny	$0.00	PA	Pennsylvania	Lehigh	Allentown
2013062304622	03MAY2013	23JUN2013	Lehigh Valley International Airport, Allentown	ABE	Property Damage	Checked Baggage	Unknown	.	PA	Pennsylvania	Lehigh	Allentown
2013060904074	06MAY2013	09JUN2013	Lehigh Valley International Airport, Allentown	ABE	Property Damage	Checked Baggage	Approve in Full	$97.96	PA	Pennsylvania	Lehigh	Allentown
2013080805751	09MAY2013	26JUN2013	Lehigh Valley International Airport, Allentown	ABE	Passenger Property Loss	Checked Baggage	Approve in Full	$100.00	PA	Pennsylvania	Lehigh	Allentown

10. Use the **sq.Claims_Cleaned** table to create a view named **TotalClaims** to count the number of claims for each value of **Airport_Code** and **Year**.

    a. Include **Airport_Code**, **Airport_Name**, **City**, **State**, and the year from **Incident_Date**. Name the new column **Year**.

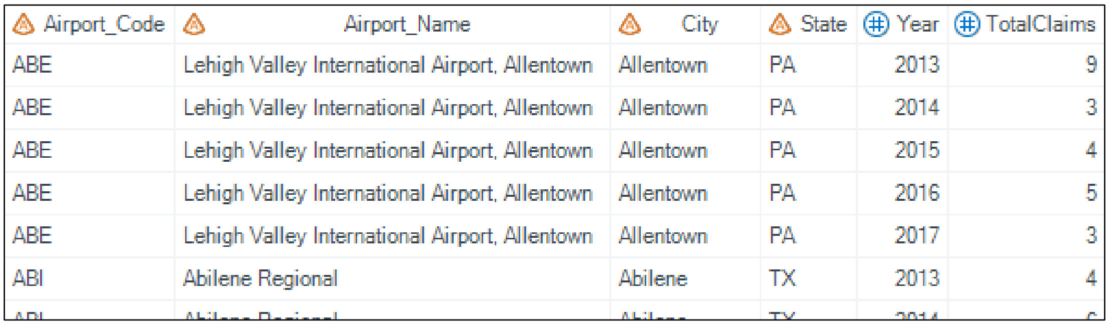
    b. Count the number of claims for each group using the COUNT function. Name the new column **TotalClaims**.

    c. Group by the correct columns.

    d. Order the table by **Airport_Code** and **Year**.

       Note: Typically, you do not use an ORDER BY clause when creating a view. For the purpose of this case study, it is used to produce a similar result image for validation.

Partial View



Airport_Code	Airport_Name	City	State	Year	TotalClaims
ABE	Lehigh Valley International Airport, Allentown	Allentown	PA	2013	9
ABE	Lehigh Valley International Airport, Allentown	Allentown	PA	2014	3
ABE	Lehigh Valley International Airport, Allentown	Allentown	PA	2015	4
ABE	Lehigh Valley International Airport, Allentown	Allentown	PA	2016	5
ABE	Lehigh Valley International Airport, Allentown	Allentown	PA	2017	3
ABI	Abilene Regional	Abilene	TX	2013	4
ABI	Abilene Regional	Abilene	TX	2014	6

11. Create a view named **TotalEnplanements** by using the OUTER UNION set operator to concatenate the **enplanement2017** and **boarding2013_2016** tables.

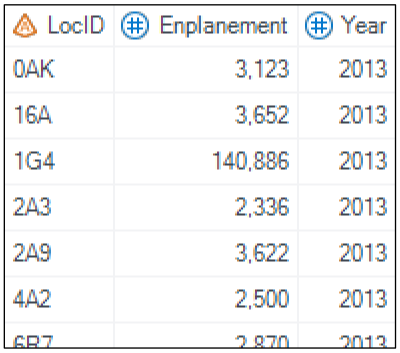
    a. From the **sq.enplanement2017** table, select the **LocID** and **Enplanement** columns. Create a new column named **Year** by converting the character **Year** column to numeric.

    b. Use the OUTER UNION set operator with the CORR modifier.

    c. From the **sq.boarding2013_2016** table, select the **LocID**, **Boarding**, and **Year** columns. Change the name of the **Boarding** column to **Enplanement**.

    d. Order the results by **Year** and **LocID**.

Partial View



LocID	Enplanement	Year
0AK	3,123	2013
16A	3,652	2013
1G4	140,886	2013
2A3	2,336	2013
2A9	3,622	2013
4A2	2,500	2013
6B7	2,870	2013

12. Create a table named **sq.ClaimsByAirport** by joining the **TotalClaims** and **TotalEnplanements** views.

    a. Select the **Airport_Code**, **Airport_Name**, **City**, **State**, **Year**, **TotalClaims**, and **Enplanement** columns.

    b. Create a new column to calculate the percentage of claims by enplanements by dividing **Enplanement** by **TotalClaims**. Name the column **PctClaims** and format it using PERCENT10.4.

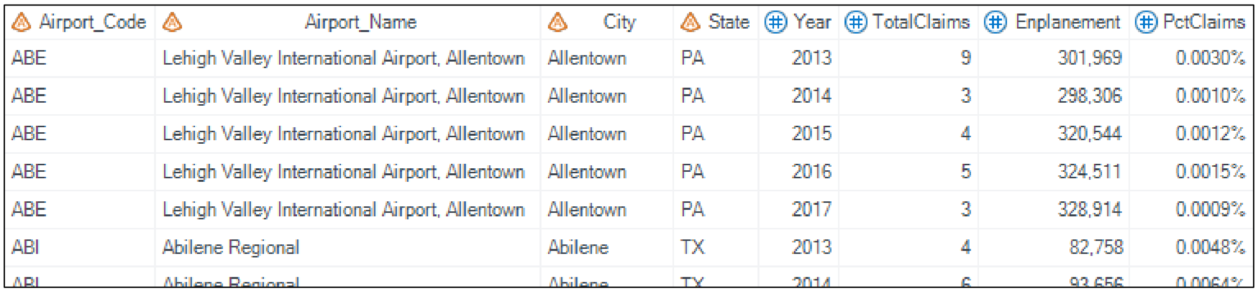
    c. Perform an inner join using the criterion **Airport_Code**=**LocID** and the **Year** columns.

    d. Order the results by **Airport_Code** and **Year**.

Log

```
NOTE: Table TSA.CLAIMSBYAIRPORT created, with 1438 rows and 8 columns.
```

Partial Table



Airport_Code	Airport_Name	City	State	Year	TotalClaims	Enplanement	PctClaims
ABE	Lehigh Valley International Airport, Allentown	Allentown	PA	2013	9	301,969	0.0030%
ABE	Lehigh Valley International Airport, Allentown	Allentown	PA	2014	3	298,306	0.0010%
ABE	Lehigh Valley International Airport, Allentown	Allentown	PA	2015	4	320,544	0.0012%
ABE	Lehigh Valley International Airport, Allentown	Allentown	PA	2016	5	324,511	0.0015%
ABE	Lehigh Valley International Airport, Allentown	Allentown	PA	2017	3	328,914	0.0009%
ABI	Abilene Regional	Abilene	TX	2013	4	82,758	0.0048%
ABI	Abilene Regional	Abilene	TX	2014	6	93,656	0.0064%

Hint: You can solve steps 10 through 12 in one query using inline views.

Analyze and Export Data

After you have prepared the data deliverables, open and run the **AnalysisProgram.sas** code to create **FinalReport.html**. **Note**: You must have the final tables in the **Sq** library for the program to run correctly. **FinalReport.html** is created in your course code folder. Use this report to answer the quiz questions.