

[Print](#)

# Essential Functions of SAS® Intelligent Decisioning

## Activities and Practices

---

Course code EBDID0D, prepared date: Tue, Feb 22 2022

Copyright © 2022 SAS Institute Inc., Cary, NC, USA. All rights reserved.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 01, Section 2 Practice: Working in SAS Intelligent Decisioning

In this practice, you log on to SAS Intelligent Decisioning and explore the user interface.

1. Open a browser (such as Google Chrome) in the virtual lab. From the home page, log on to SAS using the user ID lynn and password Student1.
  - a. Launch a browser (such as Google Chrome).
  - b. For user ID, enter **lynn**.
  - c. For Password, enter **Student1**. *The password is case-sensitive.*
  - d. Click **Sign In**.
2. Navigate to SAS Intelligent Decisioning and then to rule sets. Search for rule sets with the text example in the name.
  - a. At the top left of the interface, click **Show list of applications, Build Decisions**.
  - b. Click **Rule sets**.
  - c. In the **Search name** field at the top left of the list of decisions, enter **example** and click **Start search**.
  - d. Notice that a rule set named ExampleRuleSet is displayed.
3. Duplicate the rule set named ExampleRuleSet. Name the duplicate DuplicateRuleSet.
  - a. Enable the check box next to ExampleRuleSet.
  - b. At the top right of the interface, click **Actions, Duplicate**.
  - c. In the Name field, enter **DuplicateRuleSet**.
  - d. Click **Duplicate**.
4. Clear the search text. Sort the rule sets by descending date modified and confirm that DuplicateRuleSet appears at the top of the list.
  - a. Click **Clear search text**.
  - b. Click the **Date Modified** column heading in the list of rule sets to sort in ascending order.
  - c. Click the **Date Modified** column heading a second time to sort in descending order.
  - d. Confirm that DuplicateRuleSet appears at the top of the list.
5. Access the Help Center.
  - a. Click the **L** in the circle at the top right of the interface.
  - b. Click **Help Center**.
  - c. The SAS Intelligent Decisioning User's Guide opens in a new browser tab.
6. Sign out of SAS Intelligent Decisioning.
  - a. Click the SAS Intelligent Decisioning browser tab to return to the application.
  - b. Click the **L** in the circle at the top right of the interface.
  - c. Click **Sign out**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 01, Section 2 Practice: Loading Tables into CAS

In this practice, you load the tables needed for testing in Intelligent Decisioning into CAS.

**Reminder:** Sign in to SAS using the user ID **Lynn** and password **Student1**.

1. Load the table **hmeqapps.sashdat** from **cas-shared-default, Public** into CAS.

**Hint:** At the top left of the interface, click **Show list of applications, Manage data**. Then click the **Data Sources** tab.

- a. At the top left of the interface, click **Show list of applications, Manage Data**.
- b. Click **Data Sources**.
- c. Navigate to **cas-shared-default, Public**.
- d. Click **hmeqapps.sashdat**.
- e. At the top right of the interface, click **Load into memory**.

**Note:** Alternatively, you can right-click the SASHDAT file and select **Load**.

- f. Notice that **HMEQAPPS** appears as a loaded table in the list on the left.

2. Repeat the previous steps to load the tables Investments.sashdat, TSAClaims.sashdat, SalesReps.sashdat, and VendorList.sashdat into CAS.

Refer to the details in step 1.

3. On the **Available** tab, confirm that all five tables are listed.

- a. Click the **Available** tab.
- b. Confirm that the following tables are listed:
  - HMEQAPPS
  - INVESTMENTS
  - SALESREPS
  - TSACLAIMS
  - VENDORLIST

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 02, Section 1 Practice: Managing and Mapping Variables in a Decision

In this practice, you create, manage, and map variables in a decision.

**Reminder:** Sign in to SAS using the user ID **Lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Create a decision named Practice1 in My Folder/Decisioning/Practices.

- a. If necessary, click **Decisions**.
- b. Click **New Decision**.
- c. For **Name**, enter **Practice1**.
- d. Next to **Location**, click **Choose a location**. Click **My Folder**, **Decisioning**, **Practices**, **OK**.
- e. Click **Save**.

2. Import variables from the comma-delimited file D:\Workshop\BDID0D\Practices\ClaimVariables.csv. Notice that the variables listed below are created.

Variable	Data Type	Input	Output
Airport_Code	Character	✓	✓
Claim_Num	Character	✓	✓
Incident_Date	Date	✓	✓

- a. Click **Variables**.
- b. Click **Import, Comma-delimited (\*.csv)**.
- c. Click **Browse**.
- d. Navigate to **D:\Workshop\BDID0D\Practices\** and select **ClaimVariables.csv**.
- e. Click **Open**.
- f. For **Encoding**, confirm that **UTF-8 (Default)** is selected.
- g. Click **Import**.
- h. Click **Add**.
- i. Confirm that the **Input** and **Output** properties are enabled for both variables.

3. View the properties of the rule set named Practice\_VariableMapping. Notice that it includes the variables listed below.

Variable	Data Type	Input	Output
Airport_Code	Character	✓	✓
Claim_Number	Character	✓	✓
Elapsed_Days	Integer		✓
Incident_Date	Date	✓	✓

- a. Click **Rule Sets**
- b. Click **Practice\_VariableMapping** to open it.
- c. Click **Variables** and view the variable properties.

4. Close the rule set. Return to the decision and add the Practice\_VariableMapping rule set to follow the Start node.

- a. Click **Close** to close the rule set.
- b. Click **Decisions** to return to the decision.
- c. Click **Decision Flow**.
- d. Right-click **Start** and select **Add below, Rule set**.
- e. Click **My Folder**, **Decisioning**, **Practices**, **Practice\_VariableMapping**. Click **OK**.

**Note:** Depending on your screen resolution and browser zoom setting, you might not see the full list of rules. If you do not see **Practice\_VariableMapping** in the list, click the **Sort by** icon and select **Most recent**. Alternatively, click any rule in the list and then press the **down-arrow** key until you reach **Practice\_VariableMapping**.

5. View the decision variables. Notice that the variables **Claim\_Number** and **Elapsed\_Days** have been added to the decision.

Click **Variables** and view the variable properties.

6. Map the rule set input variable **Claim\_Number** to the decision input variable **Claim\_Num**. **Hint:** Click the rule set node and then click **Input Variables** at the right of the Properties pane for the rule set.
  - a. Click **Decision Flow**.
  - b. If necessary, click the **Practice\_VariableMapping** node.
  - c. Click **Input Variables** in the Properties pane to the right of the interface.
  - d. In the **Maps To** column for the input variable **Claim\_Number**, select **Claim\_Num**.
7. Save the decision and confirm removal of the **Claim\_Number** variable from the decision. This variable from the rule set maps to **Claim\_Num** in the decision and is not needed as a decision variable.
  - a. Click **Save**.
  - b. Click **Yes** in the window to confirm removal of the **Claim\_Number** variable from the decision.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 02, Section 2 Practice: Creating a Rule Set

In this practice, you create a rule set.

**Reminder:** Sign in to SAS using the user ID **Lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications, Build Decisions**.

1. Create an assignment rule set named **Practice\_ClaimStatus** in My Folder/Decisioning/Practices.

- a. Click **Rule Sets**.
- b. Click **New Rule Set**.
- c. For **Name**, enter **Practice\_ClaimStatus**.
- d. For **Type**, confirm that **Assignment** is selected.
- e. Next to **Location**, click **Choose a location**. Click **My Folder, Decisioning, Practices, OK**.
- f. Click **Save**.

2. Add the variables **Claim\_Amount** and **Claim\_Code** from the data table **TSAClaims**. Specify that they are both input and output variables. The variable properties should match those shown below.

Variable	Data Type	Input	Output
<b>Claim_Amount</b>	Decimal	✓	✓
<b>Claim_Code</b>	Character	✓	✓

**Hint:** if the **TSAClaims** table does not appear in the list of tables on the Available tab, click **Refresh**. If it still does not appear, navigate to **cas-shared-default, Public** and load it into memory.

- a. If necessary, click **Variables**.
- b. Click **Add variable, Data Table**.
- c. Select **TSAClaims**
- d. Press the Ctrl key and click **Claim\_Amount** and **Claim\_Code**.
- e. Click **Add**.
- f. Click **Add**.
- g. Confirm that the **Input** and **Output** properties are enabled for both variables.

3. Add a custom character output variable named **Claim\_Status**. Assign a length of 8 and the initial value **PROCESS** to the variable. The variable properties should match those shown below.

Variable	Data Type	Input	Output	Initial Value
<b>Claim_Status</b>	Character		✓	PROCESS

- a. Click **Add variable, Custom variable**.
- b. For **Name**, enter **Claim\_Status**.
- c. For **Data type**, confirm that **Character** is selected.
- d. Click **Add**.
- e. Disable the **Input** property.
- f. For **Length**, enter **8**.
- g. For **Initial Value**, enter **PROCESS**.
- h. Click **OK**.

4. Define a rule named *Reject missing claim codes* to set **Claim\_Status** to **REJECT** if the variable **Claim\_Code** does not have a value.

**Hint:** Use the expression editor to specify the **MISSING** function to test whether **Claim\_Code** has a value.

- a. Click **Rule Set**.
- b. Click **Add Rule**. Notice that an IF condition and an ASSIGN action are populated in the rule set editor.
- c. Rename the rule.
  - Click **Actions, Rename rule**.
  - Enter **Reject missing claim codes**.
- d. Specify the condition for **Claim\_Code**.
  - Move the mouse pointer over the condition and click **Open the expression editor**.
  - Click **Clear**.
  - Click the **Functions** tab.
  - In the **Filter** box, enter **missing**.
  - Notice that the **Data Grid** and **Special** folders are listed.
  - Expand the **Special** folder and double-click the **MISSING** function to add it to the expression.
  - Notice that the argument to the function is selected.
  - Click the **Variables** tab and double-click **Claim\_Code** to add as the function argument.
  - Click **Validate** to validate the expression.
  - Click **Save**.
- e. Next to **ASSIGN**, select **Claim\_Status**.
- f. For the value, enter '**REJECT**'.

5. Define an ELSE rule named *Review high claim amounts* to set **Claim\_Status** to *REVIEW* if **Claim\_Amount** exceeds 250.

**Hint:** Rename the rule before changing the IF to ELSE.

- a. Click **Add Rule**.
- b. Rename the rule.
  - Click **Actions, Rename rule**.
  - Enter **Review high claim amounts**.
- c. Click **IF** and select **ELSE**.
- d. For the ELSE condition, select **Claim\_Amount**.
- e. For the operator, select **> - (Is greater than)**.
- f. For the value, enter **250**.
- g. Next to **ASSIGN**, select **Claim\_Status**.
- h. For the value, enter '**REVIEW**'.

6. Import test scenarios from the file D:\Workshop\BDID0D\Practices\TSAClaimScenarioInput.csv.

**Hint:** You use the **Import Scenarios** button on the Scenarios tab to import scenario tests. There is also an Import button that appears at the top of the rule set that is used for a different purpose.

This file contains the following records:

```
Claim_Code,Claim_Amount,Claim_Status_expected
,125,REJECT
PD,300,REVIEW
PP,75,PROCESS
```

- a. Click **Scoring**.
- b. Click **Scenarios**.
- c. Click **Import Scenarios**.
- d. Click in the **Import from** field, navigate to **D:\Workshop\BDID0D\Practices**, and select **TSAClaimScenarioInput.csv**. Click **Open**.
- e. For **Encoding**, confirm that **UTF-8 (Default)** is selected.
- f. For **Scenario name prefix**, enter **TSAClaim**.
- g. For **Folder Location**, click **Choose a location**. Click **My Folder, Decisioning, Practices, OK**.
- h. Click **Output data library**. Click **cas-shared-default, Public, OK**.
- i. Click **Import**.
- j. Click **Close** in the Import Scenarios window.
- k. Notice that three scenarios have been imported.

7. Run the scenarios and confirm that all three scenarios ran successfully without warnings. Close the rule set.

- a. Click **Select all**.
- b. Click **Run**.
- c. Confirm that the Status column contains a green check mark without a warning icon for all three scenarios.
- d. Click **Close** to close the rule set.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 02, Section 2 Practice: Adding a Rule Set to a Decision

In this practice, you create a decision and add a rule set.

**Reminder:** Sign in to SAS using the user ID **Lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Create a decision named **Practice\_AddRuleSet** in My Folder/Decisioning/Practices.
  - a. If necessary, click **Decisions**.
  - b. Click **New Decision**.
  - c. For **Name**, enter **Practice\_AddRuleSet**.
  - d. Next to **Location**, click **Choose a location**. Click **My Folder**, **Decisioning**, **Practices**, **OK**.
  - e. Click **Save**.
2. Add the rule set named **Practice\_ClaimStatus** created in a previous practice to follow the Start node. If you did not complete the previous practice, use the rule set named **Practice\_ClaimStatusSolution1**.
  - a. Right-click **Start** and select **Add below, Rule set**.
  - b. Click **My Folder**, **Decisioning**, **Practices**, **Practice\_ClaimStatus**, **OK**.
3. On the Variables tab, confirm that **Claim\_Amount**, **Claim\_Code**, and **Claim\_Status** have been added to the decision.
  - a. Click **Variables**. Confirm that **Claim\_Amount**, **Claim\_Code**, and **Claim\_Status** have been added to the decision.
  - b. Click **Save**.
4. Define and run a basic test in the folder Users/lynn/My Folder/Decisioning/Practices using the data table named **TSAClaims**.
  - a. Click **Scoring**.
  - b. Click **New Test**.
  - c. Accept the default name for the test.
  - d. If **Location** is not populated with the desired path, click **Choose a location**. Click **My Folder**, **Decisioning**, **Practices**, **OK**.
  - e. If **Input table** is not populated with the desired table, click **Select a table** and select **TSAClaims**.
  - f. Click **Run**.
5. View the test results. Confirm that the values assigned to the variable **Claim\_Status** are as expected for the rule set.  
**Hint:** Recall that **Claim\_Status** should have the value *REJECT* when **Claim\_Code** is missing, the value *REVIEW* when **Claim\_Amount** is greater than 250, and the value *PROCESS* otherwise.
  - a. Click **Results**.
  - b. Confirm that **Claim\_Status** has the value *REJECT* when **Claim\_Code** has a missing value.
  - c. Confirm that **Claim\_Status** has the value *REVIEW* when **Claim\_Amount** is greater than 250.
  - d. Confirm that **Claim\_Status** has the value *PROCESS* for all other records.
6. Close the test results and the rule set.
  - a. Click **Close** to close the test results.
  - b. Click **Close** to close the rule set.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 02, Section 2 Practice: Creating a Lookup Table

In this practice, you create a lookup table containing claim type codes and descriptions.

**Reminder:** Sign in to SAS using the user ID **Lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Create a lookup table named **ClaimType** in My Folder/Decisioning/Practices.
  - a. Click **Lookup tables**.
  - b. Click **New Lookup Table**.
  - c. Specify table properties.
    - For **Name**, enter **ClaimType**.
    - Next to **Location**, click **Choose a location**. Click **My Folder**, **Decisioning**, **Practices**, **OK**.
    - Click **Save**.
2. Import key-value pairs from the file **D:\Workshop\BDID0D\Practices\ClaimTypeLookup.csv**.
  - a. Click **Import**.
  - b. In the Import Lookup Table window, click **Browse**.
  - c. Navigate to **D:\Workshop\BDID0D\Practices** and select **ClaimTypeLookup.csv**. Click **Open**.
  - d. For **Encoding**, confirm that **UTF-8 (Default)** is selected.
  - e. Click **Import**.
3. Activate and close the lookup table.
  - a. Click **Activate**.
  - b. Click **Yes**.
  - c. Click **Close** to close the lookup table.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 02, Section 2 Practice: Using Lookup Table Functions in a Rule Set

In this practice, you create a new version of a rule set and add an assignment using the LOOKUPVALUE lookup table function.

**Reminder:** Sign in to SAS using the user ID **Lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Open the rule set named **Practice\_ClaimStatus** and create a new minor version.

- a. Click **Rule Sets**.
- b. Click **Practice\_ClaimStatus**.
- c. Click **Versions**.
- d. Click **New Version**.
- e. For **Type**, maintain the default selection of **Minor**.
- f. Click **Save**.

2. Add a custom character output variable named **Claim\_Description** with a length of 24. The variable properties should match those shown below.

Variable	Data Type	Input	Output	Length
Claim_Description	Character		✓	24

- a. Click **Add variable**, **Custom variable**.
- b. For **Name**, enter **Claim\_Description**.
- c. For **Data type**, confirm that **Character** is selected.
- d. Click **Add**.
- e. Disable the **Input** property.
- f. For **Length**, enter **24**.
- g. Click **OK**.

3. Define an assignment to retrieve a value for **Claim\_Description** from the ClaimType lookup table using the key value **Claim\_Code**. If you did not create and activate the ClaimType lookup table in the previous practice, use the ClaimTypeSolution lookup table instead.

- a. Click **Rule Set**.
- b. Click **Add Rule**. Notice that an IF condition and an ASSIGN action are populated in the rule set editor.
- c. Place the mouse pointer over the IF condition and click **Delete the selected condition**.
- d. Click **ASSIGN** and select **LOOKUPVALUE**.
- e. For the variable, select **Claim\_Description**.
- f. Click **Select a lookup table**. Navigate to **My Folder**, **Decisioning**, **Practices**, and select **ClaimType**.
- g. For the lookup key, select **Claim\_Code**.

4. Save and close the rule set.

- a. Click **Save**.
- b. Click **Close**.

5. Update the **Practice\_AddRuleSet** decision to use the new version of the rule set.

- a. Click **Decisions**.
- b. Click **Practice\_AddRuleSet**.
- c. On the Decision Flow tab, click the **Practice\_ClaimStatus** node.
- d. In the Properties pane, select **1.1** for **Version**.
- e. Click **Save** to save the decision.

6. Run the previously defined test. Confirm that **Claim\_Description** is populated with values as expected.

- a. Click **Scoring**.
- b. Enable the check-box for the previously defined test.
- c. Click **Run**.
- d. After the test finishes running, confirm that the **Status** column contains a green check-mark.
- e. Click **Results**.
- f. Confirm that **Claim\_Description** has the value *Property Damage* when **Claim\_Code** has the value *PD*.
- g. Confirm that **Claim\_Description** has the value *Passenger Property Loss* when **Claim\_Code** has the value *PP*.

7. Close the test results and the decision.

- a. Click **Close** to close the test results.
- b. Click **Close** to close the decision.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 02, Section 3 Practice: Creating and Using a Global Variable

In this practice, you create and use a global variable.

**Reminder:** Sign in to SAS using the user ID **Lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Create a decimal global variable named **NYC\_Factor** with a value of **1.15**.
  - a. Click **Global Variables**.
  - b. Click **New Global Variable**.
  - c. For **Name**, enter **NYC\_Factor**.
  - d. For **Data Type**, select **Decimal**.
  - e. For **Value**, enter **1.15**.
  - f. Click **Save**.
2. Create a new minor version and activate the variable.
  - a. Click **NYC\_Factor** to edit the variable.
  - b. Click **Versions**.
  - c. Click **Activate**.
  - d. Click **Yes** to create a new minor version.
  - e. Click **Cancel**.
3. Open the rule set named **Practice\_UseGlobalVariable** and add the **NYC\_Factor** variable.
  - a. Click **Rule sets**.
  - b. Click **Practice\_UseGlobalVariable** to open the rule set.
  - c. Click **Variables**.
  - d. Click **Global Variables**.
  - e. Click **Select Variables**.
  - f. Enable the check-box for **NYC\_Factor**.
  - g. Click **OK**.
4. Modify the THEN assignment in the existing rule to use the global variable instead of a constant **1.2**. The final expression should be similar to the following: **Claim\_Amount = Claim\_Amount \* NYC\_Factor**.
  - a. Click **Rule Set**.
  - b. Move the mouse pointer over the existing THEN condition criteria and click **Open the expression editor**.
  - c. Select the constant value **1.2** in the expression and double-click **NYC\_Factor** in the list of variables.
  - d. Click **Validate** and make corrections if necessary.
  - e. Click **Save**.
  - f. Click **Save** to save the rule set.
  - g. Click **Close** to close the rule set.
5. Save and close the rule set.
  - a. Click **Save**.
  - b. Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 02, Section 4 Practice: Configuring a Branch Node

In this practice, you configure a branch node.

**Reminder:** Sign in to SAS using the user ID **Lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Open the decision named **Practice\_Branch**. Notice that the decision includes an input variable named **Claim\_Amount**.
  - a. If necessary, click **Decisions**.
  - b. Click **Practice\_Branch**.
  - c. Click **Variables**.
  - d. Notice that the decision includes an input variable named **Claim\_Amount**.
2. Add a Range branch node using the variable **Claim\_Amount** following the Start node.
  - a. Click **Decision Flow**.
  - b. Right-click **Start** and select **Add below, Branch**.
  - c. For **Branch type**, select **Range**.
  - d. Click **OK**.
  - e. In the Properties pane, confirm that **Claim\_Amount** is selected for the **Branch variable**.
3. Define a branch corresponding to values between 0 and 100.
  - a. In the Properties pane, click **Add branch path**.
  - b. In the **Minimum** field, enter **0**.
  - c. In the **Maximum** field, enter **100**.
4. Define a branch corresponding to values between 100 and 200.
  - a. Click **Add branch path**.
  - b. In the **Minimum** field, enter **100**.
  - c. In the **Maximum** field, enter **200**.
5. Define a branch corresponding to values greater than 200.
  - a. Click **Add branch path**.
  - b. In the **Minimum** field, enter **200**.
6. Notice the warning icon on the branch node. Move the mouse pointer over the icon and notice the message about overlapping values.

If there are overlapping values in a range branch definition, a value is assigned to the first path that it matches. You can change the path order by clicking **More**.
7. Change the label for the branch labeled *200 - No maximum to 200 or more*. **Hint:** Click the **More** link at the bottom of the branch Properties pane to change branch labels.
  - a. If necessary, click the **Branch** node.
  - b. In the Properties pane, click **More**.
  - c. In the **Alternate Label** field for the branch, enter **200 or more**.
  - d. Click **Close**.
8. Save and close the decision.
  - a. click **Save**.
  - b. Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 02, Section 4 Practice: Configuring a Cross-Branch Link

In this practice, you configure a cross-branch link.

**Reminder:** Sign in to SAS using the user ID **Lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Open the decision named **Practice\_CrossBranchLink**. Notice that the decision includes a Range branch with four paths.
  - a. If necessary, click **Decisions**.
  - b. Click **Practice\_CrossBranchLink** to open the decision.
2. Add the rule set named **Practice\_CrossBranchRuleSet** to follow the existing rule set in the first branch.
  - a. On the Decision Flow tab, click **0-100** and select **Add , Rule set**.
  - b. Click **My Folder, Decisioning, Practices, Practice\_CrossBranchRuleSet, OK**.
3. Add a cross-branch link to the newly added rule set from the second branch of the node.
  - a. Right-click **100-200** and select **Add, Cross-branch link**.
  - b. For **Target node**, select **Practice\_CrossBranchRuleSet (1.0)**.
  - c. Click **OK**.
4. Save and close the decision.
  - a. Click **Save**.
  - b. Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 02, Section 5 Practice: Configuring a Record Contacts Node

In this practice, you configure a Record Contacts node.

**Reminder:** Sign in to SAS using the user ID **Lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Open the decision named **Practice\_RecordContacts**. Notice that the decision includes a Range branch with four paths.
  - a. If necessary, click **Decisions**.
  - b. Click **Practice\_RecordContacts** to open the decision.
2. Add a Record Contacts node to follow the existing rule set in the first branch. Track the variables **Claim\_Number**, **ElapsedDays**, and **Item\_Category**.
  - a. On the Decision Flow tab, click **Practice\_CrossBranchRuleSet** and select **Add, Record contacts**.
  - b. In the Properties pane, click **Select one or more variables**.
  - c. Hold down the Ctrl key and click **Claim\_Number**, **ElapsedDays**, and **Item\_Category**.
  - d. Click **Add**.
  - e. Click **OK**.
3. Save and close the decision.
  - a. Click **Save**.
  - b. Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 02, Section 6 Practice: Comparing Decision Versions

In this practice, you compare versions of a decision.

**Reminder:** Sign in to SAS using the user ID **Lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Open the decision named **Practice\_Versions**.
  - a. If necessary, click **Decisions**.
  - b. Click **Practice\_Versions** to open the decision.
2. Compare versions 1.0 and 1.1 of the decision.
  - a. Click **Versions**.
  - b. Click **Actions, Compare object contents**.
  - c. Maintain the default selections for base and comparison versions.
  - d. Click **Compare**.
3. View the differences on the Decision Flow tab. Notice that the two versions of the decision use different versions of the same rule set.
  - a. Confirm that the Decision Flow tab is active.
  - b. Click **Show Differences**.
  - c. Notice that version 1.0 uses version 1.0 of the rule set **Practice\_AssignClaimStatus**, while version 1.1 uses version 1.1 of the same rule set.
4. View the differences on the Decision tab. Notice that both versions of the decision have a branch node based on the variable **Claim\_Amount**, but that version 1.0 has four output paths while version 1.1 has three.
  - a. Click **Decision** and view the path definitions.
5. View the differences on the Variables tab. Notice that version 1.1 includes a variable that version 1.0 does not.
  - a. Click **Variables**
  - b. Notice that version 1.1 includes the variable **ClaimDescription** while version 1.0 does not.
6. Close the comparison results and the decision.
  - a. Click **Close** to close the comparison results.
  - b. Click **Close** to close the decision.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 03, Section 1 Activity

In the virtual lab, sign in to SAS as Lynn and navigate to SAS Intelligent Decisioning. Click the L at the top right of the interface and select **Help Center**. Locate the documentation for developing SQL code for data query files. How do you specify decision variables in a query? How do you specify input variables?

**Hint:** Search for the text **sql code** and click the link to the documentation for data query files.

You enclose decision variables in braces and specify input variables with a question mark, such as in the following example:  
SELECT debtinc AS {:debtRatio:decimal}, reason AS {:cause:string:8} FROM hmeq\_test WHERE bad = {?:badloan:decimal}

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 03, Section 1 Practice: Creating a Data Query for Use in a Decision

In this practice, you create a data query using the SQL editor.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Create a data query named Practice\_DataQuery using the SQL editor in My Folder/Decisioning/Practices.

- a. Click **Code files**.
- b. Click **New Code File**.
- c. For **Name**, enter **Practice\_DataQuery**
- d. For **Type**, select **Data Query**.
- e. For **Editor**, confirm that **SQL editor** is selected.
- f. Next to **Location**, click **Choose a location**. Click **My Folder**, **Decisioning**, **Practices**, **OK**.
- g. Click **Save**.

2. Modify the query properties to return scalar variables.

- a. Click the **Properties** tab.
- b. For **Output Type**, select **Scalar**.

3. Use Notepad to open the file named Practice\_QueryCode.txt in the folder D:\Workshop\BDID0D\Practices. Copy the code and paste it into the Code tab following the comment:  
/\* include sqlReturnInfo \*/.

- a. Click **Start**, **File Explorer**.
- b. Navigate to **D:\Workshop\BDID0D\Practices**.
- c. Double-click **Practice\_QueryCode.txt** to open the file.
- d. Hold down the Ctrl key and click **A** to select all the code in the file.
- e. Hold down the Ctrl key and click **C** to copy the selected code.
- f. Click the browser tab in the system tray to return to Intelligent Decisioning.
- g. Click the **Code** tab.
- h. Click in the code window below the comment in the first line.
- i. Hold down the Ctrl key and click **V** to paste the copied code.

This code returns the columns **State**, **Class**, and **ClaimsPaid** from the **airportdata** table when the **AirportCode** column matches the input variable value.

4. Validate the code and make corrections if necessary.

- a. Click **Validate**.
- b. Notice the message indicating the statement must be run. Click **Run Validation**.
- c. Click **Close** to close the window showing the validation results.
- d. If validation was not successful, make corrections and repeat the validation.

5. Synchronize and view the variables.

- a. Click **Sync Variables**. Notice the message indicating that the Variables tab has been updated.
- b. Click the **Variables** tab.
- c. Notice that the query includes the input variable **Airport\_Code** and output variables **ClaimsPaid**, **Class**, **ReturnCode**, **RowCount**, and **State**.

6. Save the code file. Create a test for the query in My Folder/Decisioning/Practices using the input table **TSAClaims**. Accept the default name for the test.

- a. Click **Save** to save the code file.
- b. Click the **Scoring** tab.
- c. Click **New Test**.
- d. For **Name**, accept the default value.
- e. For **Location**, confirm that **/Users/lynn/My Folder/Decisioning/Practices** is selected.
- f. For **Input table**, click **Select an input table**. Click **TSAClaims** and click **OK**.
- g. Click **Run**.

7. View the test results and confirm that the results include the variables specified on the query SELECT clause. Close the code file.

- a. When the test completes, confirm that the **Status** column contains a green check indicating that the test ran successfully.
- b. Click **Results**.
- c. Notice that the output table includes the three variables specified in the SELECT clause along with **ReturnCode** and **RowCount**.
- d. Click **Close** to close the test results.
- e. Click **Close** to close the code file.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 03, Section 1 Practice: Adding a Data Query to a Decision

In this practice, you create a decision and add a data query.

**Reminder:** Sign in to SAS using the user ID **Lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Create a decision named **Practice\_AddDataQuery** in My Folder/Decisioning/Practices.
  - a. If necessary, click **Decisions**.
  - b. Click **New Decision**.
  - c. For **Name**, enter **Practice\_AddDataQuery**.
  - d. Next to **Location**, click **Choose a location**. Click **My Folder**, **Decisioning**, **Practices**, **OK**.
  - e. Click **Save**.
2. Add the data query named **Practice\_DataQuery** created in a previous practice to follow the Start node. If you did not complete the previous practice, use the rule set named **Practice\_DataQuerySolution**.
  - a. Right-click **Start** and select **Add below**, **Data query**.
  - b. Click **My Folder**, **Decisioning**, **Practices**, **Practice\_DataQuery**, **OK**.
3. Notice the settings in the Properties pane that enable you to change the output type and open the query code file.

In the Properties pane, notice the **Select output type** button and the **Open** button under Code file editor.

4. Save and close the decision.

- a. Click **Save**.
  - b. Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 03, Section 1 Activity

In the virtual lab, sign in to SAS as Lynn and navigate to SAS Intelligent Decisioning. Click the **L** at the top right of the interface and select **Help Center**. Locate the documentation for DS2 code files. What DS2 data types are supported for Intelligent Decisioning?

**Hint:** Search for the text **ds2 code** and click the link to the documentation for DS2 code files.

The supported data types are double, varchar, and package datagrid.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 03, Section 1 Practice: Defining DS2 Code for Use in a Decision

In this practice, you define a DS2 code file for use in a decision.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Create a new DS2 code file named Practice\_DS2Code in My Folder/Decisioning/Practices.
  - a. Click **Code files**.
  - b. Click **New Code File**.
  - c. For **Name**, enter **Practice\_DS2Code**.
  - d. For **Type**, confirm that **DS2 Code File** is selected.
  - e. Next to **Location**, click **Choose a location**. Click **My Folder, Decisioning, Practices, OK**.
  - f. Click **Save**.

2. Notice that the Code tab is populated with starter code.

The following code appears on the Code tab:

```
package "${PACKAGE_NAME}" /inline;  
method execute();  
end;  
endpackage;
```

3. On the Variables tab, notice that there are no variables defined.

- a. Click the **Variables** tab.
  - b. Notice that no variables are defined.

4. Use Notepad to open the file named Practice\_DS2Code.txt in the folder D:\Workshop\BDID0D\Practices. Replace the starter code on the Code tab with the code from the file.

- a. Click **Start, File Explorer**.
  - b. Navigate to **D:\Workshop\BDID0D\Practices**.
  - c. Double-click **Practice\_DS2Code.txt** to open the file.
  - d. Hold down the Ctrl key and click **A** to select all the code in the file.
  - e. Hold down the Ctrl key and click **C** to copy the selected code.
  - f. Click the browser tab in the system tray to return to Intelligent Decisioning.
  - g. Click the **Code** tab.
  - h. Click in the code window.
  - i. Hold down the Ctrl key and click **A** to select all code in the window.
  - j. Hold down the Ctrl key and click **V** to paste the copied code.

This code determines a value for the output variable **Status** based on the input variables **Airport\_Code**, **Claim\_Code**, **Incident\_Date**, and **Receipt\_Date**. The logic in this code could be reproduced in a rule set.

5. Synchronize the variables and confirm that variables are defined on the Variables tab. Define all variables except for **Status** as input variables.

- a. Click **Sync Variables**.
  - b. Click the **Variables** tab.
  - c. Enable the Input property for **Airport\_Code**, **Claim\_Code**, **Incident\_Date**, and **Receipt\_Date**.

6. Save the code file.

- a. Click **Save** to save the code file.
  - b. If prompted, click **Yes** to sync the variables.

7. Test the decision using the data table **tsaclaims**. Confirm that the test runs successfully and the output includes the output variable **Status**.

- a. Click **New Test**.
- b. Accept the default name for the test.
- c. Confirm that **/Users/lynn/My Folder/Decisioning/Practices** is selected for **Location**.
- d. For **Input table**, click **Choose a location** and select **tsaclaims**.
- e. Click **Run**.
- f. When the test completes, confirm that the **Status** column indicates that the test completed successfully.
- g. Click **Results**.
- h. Notice the results include the column **Status**, which was assigned a value in the code.
- i. Click **Close** to close the test results.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 03, Section 2 Practice: Adding a Model to a Decision

In this practice, you add a model to a decision.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Open the decision Practice\_AddModel. Notice that it includes a data query named GetHMEQData.
  - a. If necessary, click **Decisions**.
  - b. Click **Practice\_AddModel** to open the decision.
  - c. View the decision flow.
  - d. On the Decision Flow tab, notice that the flow includes a data query named GetHMEQData.
2. View the decision variables. Notice that the variable **Account** is both an input and output variable, and that **Loan** is an input variable. All other variables are temporary. **Hint:** Click the **Input** column heading twice to sort the variables by the presence of the Input property.
  - a. Click the **Variables** tab.
  - b. Click the **Input** column heading twice to sort the variables by the presence of the Input property.
  - c. Notice that the variable **Account** has both the Input and Output properties enabled, and **Loan** has the Input property enabled. The remaining variables have neither property enabled.
3. In the decision flow following the data query, add the model QS\_Tree1 from the model repository named Public.
  - a. Click the **Decision Flow** tab.
  - b. Click the node labeled **GetHMEQdata (1.0)** tab.
  - c. Right-click and select **Add, Model**.
  - d. Navigate to **Public**, **QS\_Tree1** and select **QS\_Tree1**. Click **OK**.
  - e. This model uses a decision tree to predict the probability of defaulting on a loan.
4. View the decision variables. Notice that the decision includes the additional input variable **Value**. This is an input variable to the model that is not defined in the decision.
  - a. Click the **Variables** tab.
  - b. Click the **Input** column heading twice to sort the variables by the presence of the Input property.
  - c. Notice that **Value** now appears with the Input property enabled.
5. Map the model input variable **Value** to the decision variable **Loan**.
  - a. Click the **Decision Flow** tab.
  - b. If necessary, click the node labeled **QA\_Tree1**.
  - c. Click **Input Variables**.
  - d. Click in the **Maps To** column next to **Value** and select **Loan**.
6. Save the decision. Confirm the removal of the variable **Value** from the decision. This variable from the model maps to the decision variable **Loan** and is not needed as a decision variable.
  - a. Click **Save** to save the decision.
  - b. Click **Yes** to confirm removal of the variable **Value** the decision.
7. Test the decision using the previously defined test. Confirm that the test runs successfully and the output includes model results such as the variable **EM\_PROBABILITY**.
  - a. Click **Scoring**.
  - b. Click the name of the existing test to open it.
  - c. Click **Run**.
  - d. When the test completes, confirm that the **Status** column indicates that the test completed successfully.
  - e. Click **Results**.
  - f. Notice the results include the variable **EM\_PROBABILITY**, which is the predicted probability of default.
  - g. Click **Close** to close the test results.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 03, Section 3 Practice: Configuring a Data Query to Return a Data Grid

In this practice, you configure a data query to return a data grid.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Open the data query named **Practice\_QueryReturnDataGrid**. Notice that the query returns the variables **State**, **Class**, and **ClaimsPaid**.

- a. Click **Code files**.
- b. Click **Practice\_QueryReturnDataGrid**.
- c. On the Code tab, notice that the query code includes a SELECT clause with the variables **State**, **Class**, and **ClaimsPaid**.

2. View the code variables and notice that the variables **State**, **Class**, and **ClaimsPaid** are defined as output variables.

- a. Click the **Variables** tab.
- b. Notice that the variables **State**, **Class**, and **ClaimsPaid** are defined as output variables.

3. Modify the query properties to return a data grid instead of scalar variables.

- a. Click the **Properties** tab.
- b. For **Output Type**, select **Data grid**.

4. Synchronize the variables. View the code variables and notice that the variables **State**, **Class**, and **ClaimsPaid** no longer appear and a data grid variable has been added.

- a. Click the **Code** tab.
- b. Click **Sync Variables**.
- c. Click the **Variables** tab.
- d. Notice that the variables **State**, **Class**, and **ClaimsPaid** no longer appear and the variable **dgo** has been added.  
The variable **dgo** is a data grid with columns corresponding to the three columns that were specified on the SELECT clause.

5. Save the code and run the test that has been previously defined.

- a. Click **Save** to save the code.
- b. Click the **Scoring** tab.
- c. Click the name of the existing test to open it.
- d. Click **Run**.

6. View the test results. Confirm that the output data grid contains the columns **State**, **Class**, and **ClaimsPaid**.

**Note:** **RowCount** has a value of 0 or 1 for each row in the input table. For this test data, at most one row matches the query for each row in the input data. Therefore, the corresponding data grids all have at most a single row. However, it is possible for a data grid to include multiple rows.

- a. When the test completes, confirm that the **Status** column contains a green check indicating that the test ran successfully.
- b. Click **Results**.
- c. Notice that the output table includes the data grid along with **ReturnCode** and **RowCount**.
- d. Click in the data grid column for the first row where **rowCount=1** to view the contents of the data grid. Notice that the data grid includes columns correspond to the three variables specified on the SELECT clause.
- e. Click **Close** to close the data grid.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 03, Section 3 Activity

In the virtual lab, sign in to SAS as Lynn and navigate to SAS Intelligent Decisioning. Click the **L** at the top right of the interface and select **Help Center**. Locate the documentation for the DATAGRID\_SORT function. What value do you specify for the sort order argument to sort a data grid in descending order?

**Hint:** Search for the text **datagrid\_sort** and click the link to the documentation for the function.

You specify **D** for the sort order argument to sort in descending order.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 03, Section 3 Practice: Using Data Grid Functions

In this practice, you use data grid functions to calculate the maximum value of a data grid column and create a sorted copy of a data grid.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications, Build Decisions**.

1. Open the decision Practice\_DataGridFunctions. Notice that it includes a data query named OrderQuery that returns a data grid.

- a. If necessary, click **Decisions**.
- b. Click **Practice\_DataGridFunctions** to open the decision.
- c. View the decision flow.
  - On the Decision Flow tab, notice that the flow includes a data query named OrderQuery.
  - Click the **OrderQuery** node and notice that the data query is configured to return a data grid.

2. View the properties of the decision variables to identify the columns in the data grid.

- a. Click the **Variables** tab.
- b. Click **OrderQuery\_out** to view the properties of the data grid.
- c. Notice that the data grid includes the columns **Profit**, **Total\_Retail\_Price**, **Employee\_ID**, **Order\_ID**, **Order\_Date** and **Quantity**.
- d. Click **Cancel**.

3. Create an assignment rule set named Practice\_DGF in the folder My Folder/Decisioning/Practices.

- a. Click **Rule sets**.
- b. Click **New Rule Set**.
- c. For **Name**, enter **Practice\_DGF**.
- d. For **Type**, confirm that **Assignment** is selected.
- e. For **Location**, click **Choose a location** and navigate to **My Folder, Decisioning, Practices**. Click **OK**.
- f. Click **Save**.

4. In the rule set, add the **OrderQuery\_out** data grid variable from the Practice\_DataGridFunctions decision and enable the **Input** property. The variable properties should match those shown below.

Variable	Data Type	Input	Output
OrderQuery_out	Data grid	✓	✓

- a. Click **Add variable, Decision**.
- b. Navigate to **My Folder, Decisioning, Practices**.
- c. Click **Practice\_DataGridFunctions**.
- d. Click **OK**.
- e. Click **OrderQuery\_out**.
- f. Click **Add**.
- g. Click **Add**.
- h. Enable the **Input** Property.

5. In the rule set, duplicate the **OrderQuery\_out** data grid variable to create a variable named **OrderQuery\_Desc\_Profit**. The variable properties should match those shown below.

Variable	Data Type	Input	Output
OrderQuery_Desc_Profit	Data grid		✓

- a. Enable the check box next to **OrderQuery\_out**.
- b. Click **Actions, Duplicate**.
- c. For **Name**, enter **OrderQuery\_Desc\_Profit**.
- d. Click **Duplicate**.
- e. Disable the **Input** property.

6. In the rule set, add a custom decimal output variable named **MaximumProfit**. The variable properties should match those shown below.

Variable	Data Type	Input	Output
<b>MaximumProfit</b>	Decimal		✓

- a. Click **Add variable**, **Custom variable**.
- b. For **Name**, enter **MaximumProfit**.
- c. For **Data type**, select **Decimal**.
- d. Click **Add**.
- e. Disable the **Input** property.
- f. Click **OK**.

7. Create an assignment to populate the **MaximumProfit** variable with the maximum value of the **Profit** column of the **OrderQuery\_out** data grid. **Hint:** Use the DATAGRID\_MAX function.

- a. Click **Rule Set**.
- b. Click **Add Assignment**.
- c. Next to **ASSIGN**, select **MaximumProfit**.
- d. Click **Open the expression editor**.
- e. Click **Functions**.
- f. Expand the **Data Grid** folder.
- g. Double-click **DATAGRID\_MAX** to add it to the expression.
- h. Click **Variables**.
- i. Double-click **OrderQuery\_out** to add it as the first argument to the function.
- j. Select **colName** in the second argument and enter **Profit**. The expression should appear as shown below:

```
MaximumProfit = DATAGRID_MAX(OrderQuery_out,'Profit')
```

- k. Click **Validate** to validate the expression. Make corrections if necessary.
- l. Click **Save**.

8. Create an assignment to populate the **OrderQuery\_Desc\_Profit** data grid with the values from the **OrderQuery\_out** data grid in descending order of profit. **Hint:** Use the DATAGRID\_SORT function.

- a. Click **Add, Add assignment**.
- b. Hover over the newly added assignment and click **Open the expression editor**.
- c. Click **Clear** to clear the expression.
- d. Click **Functions**.
- e. Expand the **Data Grid** folder.
- f. Double-click **DATAGRID\_SORT** to add it to the expression.
- g. Click **Variables**.
- h. Double-click **OrderQuery\_out** to add it as the first argument to the function.
- i. Select **sortCol** in the next argument and enter **Profit**.
- j. Select **sortOrder** in the next argument and enter **d**.
- k. Select **tgtGrid** in the final argument. Click **Variables** and double-click **OrderQuery\_Desc\_Profit**. The expression should appear as shown below:

```
DATAGRID_SORT(OrderQuery_out,'Profit','d',OrderQuery_Desc_Profit)
```

- l. Click **Validate** to validate the expression. Make corrections if necessary.
- m. Click **Save**.

9. Save and close the rule set. Add the rule set to the **Practice\_DataGridFunctions** decision following the data query. Save the decision.

- a. Click **Save**.
- b. Click **Close**.
- c. Click **Decisions** to return to the **Practice\_DataGridFunctions** decision.
- d. Click **Decision Flow**.
- e. Right-click the **OrderQuery** node and select **Add, Rule set**.
- f. Navigate to **My Folder, Decisioning, Practices**.
- g. Click **Practice\_DGF**.
- h. Click **OK**.
- i. Click **Save** to save the decision.

10. Test the decision using the previously defined test named Practice\_DataGridFunctions\_Test\_1. Confirm that the **MaximumProfit** variable has values and that the **OrderQuery\_Desc\_Profit** data grid is sorted by descending profit.

- a. Click **Scoring**.
- b. Click the test named **Practice\_DataGridFunctions\_Test\_1** to open it.
- c. Click **Run**.
- d. When the test completes, confirm that the **Status** column indicates that the test completed successfully.
- e. Click **Results**.
- f. Confirm that the **MaximumProfit** column is populated with values.
- g. Click **OrderQuery\_Desc\_Profit** to open the data grid for the first row in the output table. Confirm that the data grid rows are sorted by descending profit. Click **Close**.
- h. Click **Close** to close the test results.

11. Close the decision.

Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 03, Section 3 Practice: Scoring Rows in a Data Grid

In this practice, you create a rule set to score rows in a data grid to calculate a commission for products with a price of more than 200.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications, Build Decisions**.

1. Open the decision Practice\_GridScoreRows. In the data grid named **OrderQuery\_Desc\_Profit**, define a decimal column named Commission. Save the decision.

- If necessary, click **Decisions**.
- Click **Practice\_GridScoreRows** to open the decision.
- Click the **Variables** tab.
- Click **OrderQuery\_Desc\_Profit** to edit the data grid.
- Notice that the data grid includes the columns **Profit**, **Total\_Retail\_Price**, **Employee\_ID**, **Order\_ID**, **Order\_Date** and **Quantity**.
- Click **Edit**.
- Confirm that **Add a new column** is selected.
- Enter the name **Commission**.
- Select **Decimal**.
- Click **Add**.
- Click **OK** to close the Edit Columns window.
- Click **OK** to close the Edit Variable window.
- Click **Save** to save the decision.

2. Create an assignment rule set named Practice\_DGS in the folder My Folder/Decisioning/Practices.

- Click **Rule sets**.
- Click **New Rule Set**.
- For **Name**, enter **Practice\_DGS**.
- For **Type**, confirm that **Assignment** is selected.
- For **Location**, click **Choose a location** and navigate to **My Folder, Decisioning, Practices**. Click **OK**.
- Click **Save**.

3. In the rule set, add a custom decimal input variable named **Price**. Add a custom decimal output variable named **Commission**. The variable properties should match those shown below.

Variable	Data Type	Input	Output
Price	Decimal	✓	
Commission	Decimal		✓

- Define the **Price** variable.
  - Click **Add variable, Custom variable**.
  - For **Name**, enter **Price**.
  - For **Data type**, select **Decimal**.
  - Click **Add**.
  - Disable the **Output** property.
  - Keep the Add Variables window open.
- Define the **Commission** variable.
  - For **Name**, enter **Commission**.
  - For **Data type**, maintain the selection **Decimal**.
  - Click **Add**.
  - Disable the **Input** property.
  - Click **OK**.

4. Define a rule to calculate the value of **Commission**. If **Price** is greater than 200, then **Commission** is 5% of **Price**. Otherwise, **Commission** is 0.

- a. Click **Rule Set**.
- b. Define the IF rule.
  - Click **Add Rule**.
  - Next to IF, select **Price**.
  - For the operator, select **Is greater than**.
  - Enter the value **200**.
  - Next to THEN, confirm that **ASSIGN** is selected.
  - For the variable, select **Commission**.
  - Enter the expression **Price\*0.05**.
- c. Define the ELSE rule.
  - Click **Add, ELSE rule**.
  - Next to THEN, confirm that **ASSIGN** is selected.
  - For the variable, select **Commission**.
  - Enter the value **0**.

5. Save and close the rule set. Add the rule set to the Practice\_DataGridScoreRows decision following the rule set named Practice\_DGFSolution.

- a. Click **Save**.
- b. Click **Close**.
- c. Click **Decisions** to return to the Practice\_DataGridScoreRows decision.
- d. Click **Decision Flow**.
- e. Right-click the **Practice\_DGFSolution** node and select **Add, Rule set**.
- f. Navigate to **My Folder, Decisioning, Practices**.
- g. Click **Practice\_DGS**.
- h. Click **OK**.

6. Configure the rule set to score rows in the **OrderQuery\_Desc\_Profit** data grid. Map the input variable named **Price** to a data grid column named **Total\_Retail\_Price**. Map the scalar output variable named **Commission** to a data grid column of the same name. **Hint:** You perform these steps after clicking **Input Variables** in the Properties pane for the rule set in the decision flow.

- a. Click **Input variables**.
- b. Enable the **Score rows in this data grid** option.
- c. Confirm that the data grid named **OrderQuery\_Desc\_Profit** is selected.
- d. Configure variable mapping.
  - Click in the Maps To column next to **Price** and select the data grid column **Total\_Retail\_Price**.

**Note:** If the variable had been named **Total\_Retail\_Price**, this mapping would have been configured by default.

  - Click **Output variables**. Confirm that the output variable **Commission** maps to a data grid column of the same name.

7. Save the decision. Confirm the removal of any variables no longer referenced by objects in the decision.

- a. Click **Save**.
- b. Click **Yes**.

8. Test the decision using the previously defined test named Practice\_DataGridScoreRows\_Test\_1. Confirm that the **Commission** column of the **OrderQuery\_Desc\_Profit** data grid has a positive value for items with **Total\_Retail\_Price** greater than 200, and 0 otherwise.

- a. Click **Scoring**.
- b. Click the test named **Practice\_DataGridScoreRows\_Test\_1** to open it.
- c. Click **Run**.
- d. When the test completes, confirm that the **Status** column indicates that the test completed successfully.
- e. Click **Results**.
- f. Click **OrderQuery\_Desc\_Profit** to open the data grid for the first row in the output table. Confirm that the **Commission** column has a positive value for items with **Total\_Retail\_Price** greater than 200, and 0 otherwise.
- Click **Close**.
- g. Click **Close** to close the test results.

9. Close the decision.

Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 03, Section 3 Activity

In the virtual lab, sign in to SAS as Lynn and navigate to SAS Intelligent Decisioning. Click the L at the top right of the interface and select **Help Center**. Locate the documentation for the %DCM\_SERIALIZEGRID macro. What arguments are required?  
**Hint:** Search for the text **serialize** and click the link to the documentation for the macro.

The arguments GRIDCOLNAME, GRIDSOURCETABLE, and OUTPUTTABLE are required.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 04, Section 1 Practice: Creating and Locking a Treatment

In this practice, you create and lock a treatment.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Create a treatment named **PracticeTreatment** in My Folder/Decisioning/Practices.
  - a. Click **Treatments**.
  - b. Click **New Treatment**.
  - c. For **Name**, enter **PracticeTreatment**.
  - d. Next to **Location**, click **Choose a location**. Click **My Folder**, **Decisioning**, **Practices**, **OK**.
  - e. Click **Save**.
2. Add all attributes from the treatment named **StandardTreatment**.
  - a. Click **Add Attribute**, **Treatment**.
  - b. Navigate to **My Folder**, **Decisioning**, **Practices** and select **StandardTreatment**. Click **OK**.
  - c. Click **Add all**.
  - d. Click **Add**.
  - e. Notice that the attributes **Category** and **Priority** are added.
3. Change the value of the **Priority** attribute to 5.
  - a. Click **Priority**
  - b. In the **Value** field, enter **5**.
  - c. Click **OK**.
4. Add a dynamic URL custom attribute named **LinkURL**.
  - a. Click **Add Attribute**, **Custom attribute**.
  - b. For **Name**, enter **LinkURL**.
  - c. For **Data type**, select **URL** and **Dynamic**.
  - d. Click **Add**.
  - e. Click **OK**.
5. Add a fixed character custom attribute named **Description** with the value *Practice Treatment*.
  - a. Click **Add Attribute**, **Custom attribute**.
  - b. For **Name**, enter **Description**.
  - c. For **Data type**, select **Character** and **Fixed**.
  - d. For **Value**, enter **Practice Treatment**.
  - e. Click **Add**.
  - f. Click **OK**.
6. Specify that the treatment is effective until 12 pm the last day of the upcoming month.
  - a. Click the **Properties** tab.
  - b. Next to **End date**, click **Select a date and time**.
  - c. Select the last day of the upcoming month.
  - d. For time, select **12, 00, 00, PM**.
  - e. Click **OK**.
7. Create a locked version for use in a treatment group. Close the treatment.
  - a. Click the **Versions** tab.
  - b. Click **New Version**.
  - c. Maintain the default selection of **Minor**.
  - d. Click **Save**.
  - e. Notice that version 1.1 is created and version 1.0 is locked.
  - f. Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 04, Section 1 Practice: Creating a Treatment Eligibility Rule

In this practice, you create a filtering rule set and specify it as the eligibility rule for a treatment.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Create a filtering rule set named **Practice\_EligibilityRule** in My Folder/Decisioning/Practices.
  - a. Click **Rule sets**.
  - b. Click **New Rule Set**.
  - c. For **Name**, enter **Practice\_EligibilityRule**.
  - d. For **Type**, select **Filtering**.
  - e. For **Location**, click **Choose a location** and navigate to **My Folder, Decisioning, Practices**. Click **OK**.
  - f. Click **Save**.
2. Add the variable **Airport\_Code** from the data table **TSAClaims**.
  - a. Click **Add variable, Data table**.
  - b. If necessary, click **Select data table**, select **TSAClaims** and click **OK**.
  - c. Click **Airport\_Code**.
  - d. Click **Add**.
  - e. Click **Add**.
3. Define a rule that is true when **Airport\_Code** does not have the value *JFK*, *EWR*, or *LGA*.
  - a. Click the **Rule Set** tab.
  - b. Click **Add Rule**.
  - c. Notice that the variable **Airport\_Code** is populated in the IF condition.
  - d. For the operator, select **NOT IN**.
  - e. For the value, enter '**JFK','EWR','LGA**'.
4. Name the rule Not an NYC airport.
  - a. Click **Actions, Rename rule**.
  - b. Enter **Not an NYC airport**.
  - c. Click **Rename**.
5. Create a locked version of the rule set for use in a treatment. Close the rule set.
  - a. Click the **Versions** tab.
  - b. Click **New Version**.
  - c. Maintain the default selection of **Minor**.
  - d. Click **Save**.
  - e. Notice that version 1.1 is created and version 1.0 is locked.
  - f. Click **Close**.
6. Specify that the newly created rule set is the eligibility rule for the treatment named **Practice\_Treatment**. Save and close the treatment.
  - a. Click **Treatments**.
  - b. Click **Practice\_Treatment** to open it.
  - c. Click the **Eligibility Rule Set** tab.
  - d. Click **Add Rule Set**.
  - e. Navigate to **My Folder, Decisioning, Practices** and select **Practice\_EligibilityRule**.
  - f. Click **OK**.
  - g. Click **Save**.
  - h. Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 04, Section 1 Practice: Creating a Treatment Group

In this practice, you create a treatment group and activate it.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Create a treatment group named **Practice\_TreatmentGroup** in My Folder/Decisioning/Practices.
  - a. Click **Treatment Groups**.
  - b. Click **New Treatment Group**.
  - c. For **Name**, enter **Practice\_TreatmentGroup**.
  - d. For **Location**, click **Choose a location** and navigate to **My Folder**, **Decisioning**, **Practices**. Click **OK**.
  - e. Click **Save**.
2. Add the treatments **AutoFinanceOffer**, **DebtConOffer**, and **HomeImpOffer**.
  - a. Click **Add Treatments**.
  - b. Enable the check boxes for the following treatments:
    - **AutoFinanceOffer**
    - **DebtConOffer**
    - **HomeImpOffer**
  - c. Click **OK**.
3. View the properties of the **AutoFinanceOffer** treatment. Identify the version number, attributes, and eligibility rule for the locked version. **Note:** The remaining two treatments have similar settings.
  - a. Click **AutoFinanceOffer**.
  - b. Click **Versions** and notice that version 1.0 is locked.
  - c. Click in the row for version 1.0 and click **Set Version**. Notice that version 1.0 is now the displayed version.
  - d. Click **Attributes**. Notice that the treatment has fixed attributes **Description** and **URL** and the dynamic attribute **Value**.
  - e. Click **Eligibility Rule Set** and notice that the eligibility rule is named **Purpose = Auto Finance** and includes the condition **If PURPOSE = 'AF'**.
  - f. Click **Close** to close the treatment.
4. In the treatment group, select version 1.0 of each treatment.
  - a. Click **Treatment Groups** to return to the treatment group.
  - b. In the **Version** column, select **1.0** for each treatment.
5. Click **Set Attributes** and verify the attribute settings for the treatments. **Important:** You must click each treatment and then click **OK** (not **Cancel**) to confirm the dynamic treatment attribute settings. If you do not do so, the dynamic attribute will not appear as an input variable for the treatment group.
  - a. Click **Set Attributes**.
  - b. Notice that the **AutoFinanceOffer** treatment is displayed. Notice that as seen previously the treatment has fixed attributes **Description** and **URL** and the dynamic attribute **Value**.
  - c. Click each of the two remaining treatments and notice that they all share the same three attributes.
  - d. Click **OK**.
6. Identify the eligibility variables.
  - a. Click **Eligibility Variables**.
  - b. Notice that there is one eligibility variable named **PURPOSE**. It applies to each of the three treatments.
7. Save the treatment group. Create a new minor version and activate version 1.0 of the treatment group. Close the treatment group.
  - a. Click **Save**.
  - b. Click **Versions**.
  - c. Click **New Version**.
  - d. Accept the default type of **Minor**. Click **Save**.
  - e. Notice that version 1.1 is created and version 1.0 is locked.
  - f. Click in the row for version 1.0.
  - g. Click **Activate**. Click **Yes** to confirm.
  - h. Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 04, Section 1 Practice: Adding a Treatment Group to a Decision

In this practice, you add a treatment group to a decision.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Open the decision named **Practice\_AddTreatmentGroup**. Notice that the decision includes a data query, a model, and a range branch.
  - a. If necessary, click **Decisions**.
  - b. Click **Practice\_AddTreatmentGroup**.
  - c. View the decision flow.
2. View the decision variables. Notice that the decision includes the input variables **Account**, **Loan** and **Purpose**. **Hint:** Click the **Input** column heading twice to sort the variables by the presence of the **Input** property.
  - a. Click the **Variables** tab.
  - b. Click the **Input** column heading twice to sort the variables by the presence of the **Input** property.
  - c. Notice that the variables **Account**, **Loan** and **Purpose** have the **Input** property enabled.
3. Add the **Practice\_TreatmentGroup** treatment group to the path of decision flow labeled *Low default probability*.
  - a. Click the **Decision Flow** tab.
  - b. Click the Range node and select **Add to branch path**, **Low default probability**, **Treatment group**.
  - c. Navigate to **My Folder**, **Decisioning**, **Practices** and select **Practice\_TreatmentGroup**.
  - d. Click **OK**.
4. View the input variables to the treatment group. What input variables appear, and why?
  - a. In the Properties pane, click **Input variables**.
  - b. Notice that the treatment group includes the input variables **Purpose**, and **Value**.
    - **Purpose** is an input to the treatment group because it used in treatment eligibility rules.
    - **Value** is an input to the treatment group because it is a dynamic treatment attribute.
5. Confirm that the treatment group input variable **Purpose** is mapped to a decision variable of the same name. Map the treatment group input variable **Value** to the decision variable **Loan**.
  - a. In the **Maps To** column for **Purpose**, confirm that the decision flow variable **Purpose** is selected.
  - b. Click in the **Maps To** column for **Value** and select **More**, **Loan**. Click **OK**.
6. Save the decision and confirm removal of the variable **Value**. This variable has been mapped to the decision variable **Loan** and is no longer needed at the decision level.
  - a. Click **Save**.
  - b. Click **Yes**.
7. View the output variables for the treatment group. What output variable appears?
  - a. If necessary, click the node for the treatment group.
  - b. In the Properties pane, click **Output Variables**.
  - c. Notice the treatment outcome maps to a data grid named **Practice\_TreatmentGroup\_out**.
  - d. Notice that the columns of the data grid include standard treatment properties such as **treatment\_definition\_name** and treatment attributes such as **Value**.
8. Close the decision.

Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 04, Section 2 Practice: Using an Assignment Rule Set for Treatment Arbitration

In this practice, you view a decision that has been configured to perform treatment arbitration.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Open the decision named **Practice\_TreatmentArbitration**. Notice that the decision includes a treatment group and a rule set at the end of one of the paths. The rule set performs arbitration of the treatments in the group.
  - a. If necessary, click **Decisions**.
  - b. Click **Practice\_TreatmentArbitration**.
  - c. View the decision flow.
2. Notice that the treatment group output data grid is named **Practice\_AccountOffers\_out** and that it contains a column named **Priority**. **Hint:** Click the treatment group node in the decision flow and then click **Output variables** in the Properties pane to view the output data grid and its columns.
  - a. Click the treatment group node named **Practice\_AccountOffers**.
  - b. In the Properties pane, click **Output variables**.
  - c. Notice that the output data grid is named **Practice\_AccountOffers\_out**.
  - d. Notice that the output data grid includes the column **Priority**.
3. Open the treatment group named **Practice\_AccountOffers**. Notice that **Priority** is a fixed numeric attribute of each of the three treatments in the group. Notice also that the treatment group includes eligibility rules. Close the treatment group.
  - a. In the Properties pane, click **Node properties**.
  - b. Click **Open** to open the treatment group editor.
  - c. Notice that the treatment group includes three treatments.
  - d. Click **Set Attributes**.
  - e. In the Set Attributes window, click each treatment. Notice that **Priority** is fixed and has a different value for each treatment.
  - f. Click **Cancel** to close the Set Attributes window.
  - g. Click the **Eligibility Variables** tab and notice that the treatment group includes eligibility variables.
  - h. Click **Close** to close the treatment group.
4. Return to the decision. Open the rule set named **Practice\_Arbitration**. Identify the input and output variables for the rule set and the assignment that it uses. Close the rule set.
  - a. Click **Decisions**.
  - b. Click the rule set node named **Practice\_Arbitration**.
  - c. In the Properties pane, click **Open**.
  - d. In the rule set, click the **Variables** tab. Notice that the rule set includes an input data grid variable named **Practice\_AccountOffers\_out** and an output data grid variable named **SortedOffers**.
  - e. Click the **Rule Set** tab. Notice that the rule set includes an assignment that uses the **DATAGRID\_SORT** function to assign the rows of the **Practice\_AccountOffers\_out** data grid in ascending sorted value of **Priority** to the **SortedOffers** data grid.
  - f. Click **Close** to close the rule set.
5. Return to the decision and run the test that was previously defined. Compare the **Practice\_AccountOffers\_out** and **SortedOffers** data grid variables for a row in the test data where they have values. **Note:** The data grids have values only for the rows in the test data that followed the path in the decision where the treatment group and arbitration rule set were added.
  - a. Click **Decisions**.
  - b. Click the **Scoring** tab.
  - c. Click the name of the existing test to open it.
  - d. Click **Run**.
  - e. When the test completes, confirm that the **Status** column indicates that the test completed successfully.
  - f. Click **Results**.
  - g. Click in the row for **Practice\_AccountOffers\_out** for a row where it is populated.
  - h. Note the number of rows in the data grid. This number corresponds to the number of treatments for which this row of data met the eligibility rules.
  - i. Scroll to the right. Take note of the values of **Priority** and whether they appear in sorted order. **Note:** Because **Priority** is a fixed treatment attribute, each treatment always has the same priority value. But because the treatments have eligibility rules, the list of treatments sorted by priority can be different for each row of test data.
  - j. Click in the same row for **SortedOffers**. Scroll to the right and confirm that the rows are sorted in order of **Priority**.
6. Close the test results and the decision.

Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 05, Section 1 Practice: Basic Testing of a Decision

In this practice, you test a decision using rule-fired analysis, decision path tracking, and advanced options.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Open the decision Practice\_BasicTesting. Notice that the decision includes a data query, a rule set, and a range branch with a rule set on each of three output paths.

- a. If necessary, click **Decisions**.
- b. Click **Practice\_BasicTesting** to open the decision.
- c. View the decision flow.

2. View the properties of the rule set named Practice\_AssignQueue. Notice that the rule set populates the output variable **Queue** based on the value of the input variable **MortDue** as depicted below.

MortDue Value	Queue Value
>120,000	A
>60,000 and <=120,000	B
>0 and <=60,000	C

- a. On the Decision Flow tab, click the node named **Practice\_AssignQueue**.
- b. In the Properties pane, click **Open**.
- c. Click the **Variables** tab. Notice that **MortDue** is an input variable and **Queue** is an output variable.
- d. Click the **Rule Set** tab. Notice that the rule set assigns one of three values to the variable **Queue** based on the value of the variable **MortDue**.

3. Return to the decision and run the test that has been previously defined.

- a. Click **Close** to close the rule set.
- b. Click **Decisions** to return to the decision.
- c. Click the **Scoring** tab.
- d. Enable the check box next to the test and click **Run**.

4. Open the test results and run rule-fired analysis. Confirm that the results match what you would expect for the Practice\_AssignQueue rule set.

- a. Click **Results**.
- b. Click **Rule-Fired Analysis**.
- c. Click **Run Rule-Fired Analysis**.
- d. For the first few rows, notice the value of **MortDue** and click the link in the **Rules Fired Count** column to view details of the rule that fired.

**Note:** The value of the column **Rule Order** listed in the Rule Fired Count window reflects the static ordering in the diagram and not necessarily the order that the rule fired at execution time.

5. Return to the decision flow. View the variable and ranges specified in the branch node.

- a. Click **Close** to close the test results.
- b. Click the **Decision Flow** tab.
- c. Click the node labeled **Range (DEBTINC)**.
- d. Notice that the branch variable is **DebtInc**.
- e. In the Properties pane, click **More**.
- f. Notice that there are two branches defined, along with an Other branch.

Alternate Label	Minimum	Maximum
Low	No minimum	36
Caution	36	43

- g. Click **Close** to close the More Branching Properties window.

6. View the properties of the rule set named Practice\_LowDebtInc. Notice that it includes an assignment for the variable **Message**. **Note:** This simple rule set is included only to produce path tracking results, and the rule sets on the other branch paths are similar.

- a. On the Decision Flow tab, click the node named **Practice\_LowDebtInc**.
- b. In the Properties pane, click **Open**.
- c. Notice that the rule set assigns the value *This is the Low branch* to the variable **Message**.

7. Return to the decision. Open the previously defined test and run path tracking. View the plot and the node counts.

- a. Click **Close** to close the rule set.
- b. Click **Decisions** to return to the decision.
- c. Click the **Scoring** tab.
- d. Click **Results** to open the results for the test that has been previously run.
- e. Click **Decision Path Tracking**.
- f. Click **Run Path Tracking**.
- g. In the plot, notice that 26 rows in the test data followed the Low path, 12 followed the Caution path, and 3 followed the Other path.
- h. Click **Node Count**. Notice that the results list the count of rows that passed through each node in the decision flow.

8. Duplicate the existing test. Enable the option Record variable values by node. Run the test and open the test results.

- a. Enable the check box next to the existing test.
- b. Click **Actions, Duplicate**.
- c. Accept the default name for the duplicate.
- d. Click **Duplicate**.
- e. Click the name of the duplicated test to open it.
- f. Click **Advanced**.
- g. Enable **Record variable values by node**
- h. Click **Run**.
- i. After the test runs, confirm that the Status column displays a green check to indicate that the test ran successfully.
- j. Click **Results**.

9. Click **nodeTraceDataGrid** to see variable changes by node. Notice that a row appears for the data query node and for each of two rule sets. Notice that most of the variable values do not change across the nodes, but that values for the variables **Queue** and **Message** are assigned in the rule sets. Close the data grid.

- a. Click a row in the column for the **nodeTraceDataGrid** to open it.
- b. Scroll right to view the variable values.
- c. Click **Close**.

10. Close the test results and the decision.

- a. Click **Close**.
- b. Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 05, Section 1 Practice: Scenario Testing

In this practice, you perform scenario testing of a rule set.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Open the rule set named **Practice\_ScenarioTesting**. Notice that the rule set assigns a value to the variable **Category** based on the variable **Price** as shown below.

Price	Category
<50	Low
≤50 and <100	Mid
≥100	High

- a. Click **Rule sets**.  
 b. Click **Practice\_ScenarioTesting** to open the rule set.  
 c. Notice that the rule set assigns one of three values to the variable **Category** based on **Price**.
2. Create a scenario test with cas-shared-default, Public as the output data library.
  - a. Click **Scoring**.
  - b. Click **Scenarios**.
  - c. Click **New Test**.
  - d. Notice that **Output table location** is required and does not have a value. Click **Output data library** and select **cas-shared-default, Public**. Click **OK**.
3. Specify that the input variable **Price** has the value **150**. Include the output variable **Category** with the value **High** as the expected output. Run the test, and confirm that it runs successfully and that the expected and actual values of **Category** match.
  - a. For **Price**, enter **150** in the **Value** field.
  - b. For **Category**, enable the **Include** check-box and enter **High** in the **Expected Output** field.
  - c. Click **Run**.
  - d. When the test completes, confirm that the **Status** column displays a green check mark indicating that the scenario test completed successfully.
  - e. Click **Results**.
  - f. Notice that the expected and actual values for **Category** match.
  - g. Close the test results.
4. Duplicate the scenario test. Change the expected output value for Category to **high** (all lowercase). Run the test and confirm that it runs with warnings and that the expected and actual values of **Category** do not match.
  - a. Enable the check-box next to the existing test.
  - b. Click **Actions, Duplicate**.
  - c. Accept the default name for the test and click **Duplicate**.
  - d. Click the name of the duplicated test to open it.
  - e. For **Category**, enter **high** (all lowercase) in the **Expected Output** field.
  - f. Click **Run**.
  - g. When the test completes, notice the warning icon over the green check-mark in the **Status** column. This icon indicates that the scenario test completed with warnings.
  - h. Click **Results**.
    - i. Notice the message *There are some actual output values that do not match the expected output values*.
    - j. Click **Show Differences**. Notice that the expected and actual values for **Category** do not match.
5. Close the test results and the rule set.
  - a. Click **Close**.
  - b. Click **Close**.

## Essential Functions of SAS® Intelligent Decisioning

### Lesson 05, Section 2 Practice: Publishing and Validating a Published Decision

In this practice, you publish and validate a decision.

**Reminder:** Sign in to SAS using the user ID **lynn** and password **Student1**. To navigate to SAS Intelligent Decisioning, click **Show list of applications**, **Build Decisions**.

1. Open the decision named **Practice\_PublishValidate**.
  - a. If necessary, click **Decisions**.
  - b. Click **Practice\_PublishValidate** to open the decision.
2. Lock the decision and publish to the maslocal destination. Confirm that publication is successful.
  - a. Click **Actions, Publish and Lock**.
  - b. Notice that you can change the value for **Published name**.
  - c. Notice that the destination **SAS Micro Analytic Service (maslocal)** is selected by default.
  - d. Click **Publish and Lock**.
  - e. When the process completes, confirm that the **Status** column indicates that the decision was published successfully.
  - f. Click **Close**.
3. Run the publishing validation test using the input table **HMEQApps**. Confirm that the test runs successfully.
  - a. Click the **Scoring** tab.
  - b. Click the **Publishing Validation** tab.
  - c. Notice that a publishing validation test has been created. Place your mouse pointer over the test name and notice that it is the name of the decision with a datetime stamp added.
  - d. Click the name of the test to open it.
  - e. Under **Input table**, click **Select a table**.
  - f. Click **HMEQApps**. Click **OK**.
  - g. Click **Run**.
  - h. When the test completes, confirm that the status column indicates that the test ran successfully.
4. Open the test results. Notice that the publishing validation test creates an output table similar to basic testing results. What optional analyses do not appear in the publishing validation test for the maslocal destination that did appear in basic tests on CAS?
  - a. Click **Results**.
  - b. Under Test Results, notice that **Rule-fired Analysis** and **Decision Path Tracking** do not appear.
5. Close the test results and the decision.
  - a. Click **Close**.
  - b. Click **Close**.