Welcome to the topic on variables and input parameters.
At the end of this course, you will be able to:

- Describe how to create Variables and Input Parameters
For query views, you can variables and input parameters to get data from report users.

Variables are bound to attributes and are used for filtering. As such, they can only contain the values available in the attribute they relate to.

Input parameters can contain any value the reporting user want to enter. Therefore, a data type for the input parameter must be specified.
Concepts for Variables

- You use variables to filter data at runtime.
- You assign values to these variables by entering the value manually, or by selecting it from the drop-down list.
- You do not need to decide on the restriction on the value of attributes at the design time.

- You use variables to filter data at runtime. You assign values to these variables by entering the value manually, or by selecting it from the drop-down list.
- Using variables means that you do not need to decide on the restriction on the value of attributes at the design time.
- You can apply variables in the analytic and calculation views.
- If a calculation view is created using an analytic view with variables, those variables are also available in the calculation view but cannot be edited.
The following types of variables are supported:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Value</td>
<td>Use this to apply a filter to a single value</td>
</tr>
<tr>
<td>Interval</td>
<td>Use this where you want the user to specify a set start and end to a selected interval.</td>
</tr>
<tr>
<td>Range</td>
<td>Use this when you want the end user to be able to use operators such as “Greater Than” or “Lesser Than”</td>
</tr>
</tbody>
</table>

Range Variable Example:
A variable restricts the results in the view for the selected attribute. To create a variable, select the attribute in the view that you want to filter on, then define the following values:

**Selection Type:** You can choose whether selections should be based on intervals, ranges or single values.

**Multiple Entries:** You control whether multiple values of the selection types should be allowed.

**Default Value:** You have the option to define a default value for the variable.

**Mandatory:** The variable can be set as mandatory.
Sometimes using a variable is not enough. You might want to do more than restrict the value of an attribute in a view.

You might want to take input from the user and process it, returning dynamic data based on the user selection.

Using Input Parameters makes this possible.
The following types of Input variables are supported:

**Direct**: Use this to specify an input parameter as currency and date during currency conversion, and target unit during unit of measure conversion.

**Column**: Use this when the value of a parameter comes from an attribute or table column.

**Static List**: Use this when the value of a parameter comes from a user-defined list of values.

**Derived from Table**: Use this when the value of a parameter comes from a table column based on some filter conditions and you do not need to provide any input at runtime.
Imagine that you have an Aging Query.

You would like to use the same query to run aging for accounts receivable or vendor liabilities.

You can use an input parameter for Business Partner Type to allow the report user to make this decision when running the report.

When running the report the user could choose Customer or Vendor.
When creating an input parameter definition, first add a name and give it a descriptive label.

Choose a suitable parameter type, for example a StaticList type. We used this type for the input parameter for our Aging Query because the value of this parameter will come from a user-defined list of values.

You have the option to define whether a parameter is mandatory. You also can set a default value. In this case, we have chosen to set a constant value of C.

Input parameters must have a data type defined because they are not directly associated with an attribute unlike variables.

In this example we chose to have a static list with values that the user can choose, therefore we entered the values “Customer” or “Vendor” in a list of values.
Create a Variable to filter the NetSales by year

- Create a Variable in our Query View:
  - Link the Variable to the YEAR attribute

- Run the Data Preview, you will be prompted to enter a YEAR before the view is executed.
We are going to calculate the commission charges based on an input parameter: % commission.

Add an Input Parameter to your Query view where the user will enter a % commission on sales:

- **Create the Input Parameter:**
  - Type – Decimal (21,6)

- **Create a calculated column**
  - Call the calculated column “Commission Total”
  - Multiply the NetSales with the % Discount Input Parameter and divide by 100.
Variables are bound to attributes and are used for filtering.

As such, they can only contain the values available in the attribute they relate to. You do not need to decide on the restriction on the value of attributes at the design time.

Input parameters are more flexible. They can contain any value the reporting user wants to enter.

Therefore, a data type for the input parameter must be specified.
Thank you

This concludes the topic on variables and input parameters.

Thank you for your time.

This concludes the topic on variables and input parameters in the SAP Business One, version for SAP HANA. Thank you for your time.