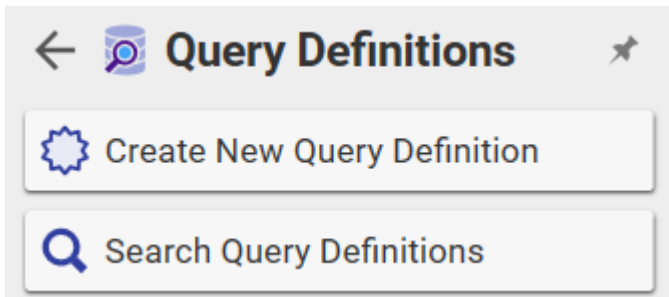


Developing Query Definitions

Reusable queries are defined by the Query Definition item and found as follows: **Contents --> Administration --> Configuration --> Query Definitions.**



When you click **Create New Query Definition**, the dialog shown in Figure 3 appears.

This image shows the 'Query Definition 2' dialog box. The title bar says 'Query Definition...' with a close button. The main title is 'Query Definition 2' with a database icon. Below the title are three buttons: 'Save' with a download icon, 'Done' with a checkmark icon, and 'Delete'. A sidebar on the left has a list icon and a 'Query Definition' header with an expand/collapse arrow. The main area contains two input fields: 'Name' and 'Context Item Type' (which has a dropdown arrow), and a larger 'Description' text area below them.

Each Query Definition starts with an initial Context Item before building it out to a more extensive structure. This specifies the *root* of the query, from where query of structures and property values will begin.

The basic steps for creating a Query Definition are as follows:

1. Define the Context Item.
2. Build the Relationship Structure.
3. Apply Search Conditions.

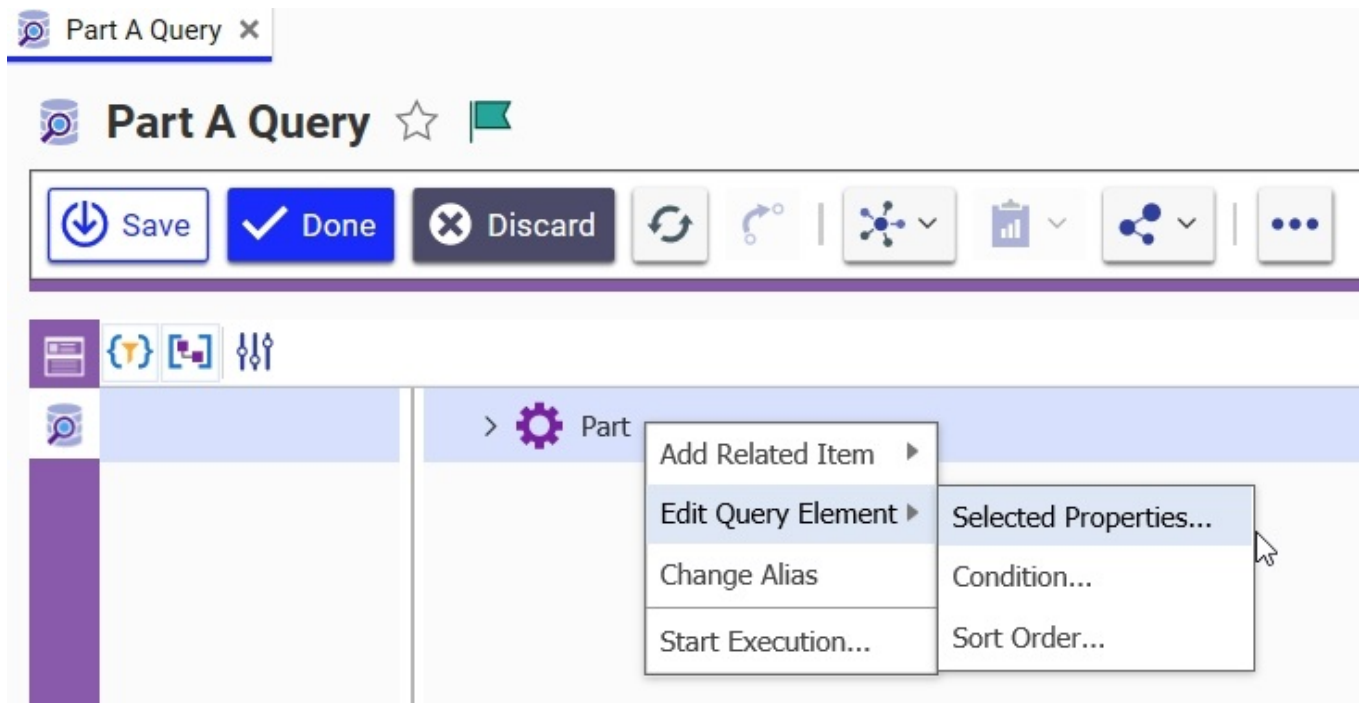
Defining a Context Item

The following example creates a **Query Definition** Item for the **Part** ItemType as a Context Item. Create a new query definition and specify a unique **Name** and a **Context Item Type**.

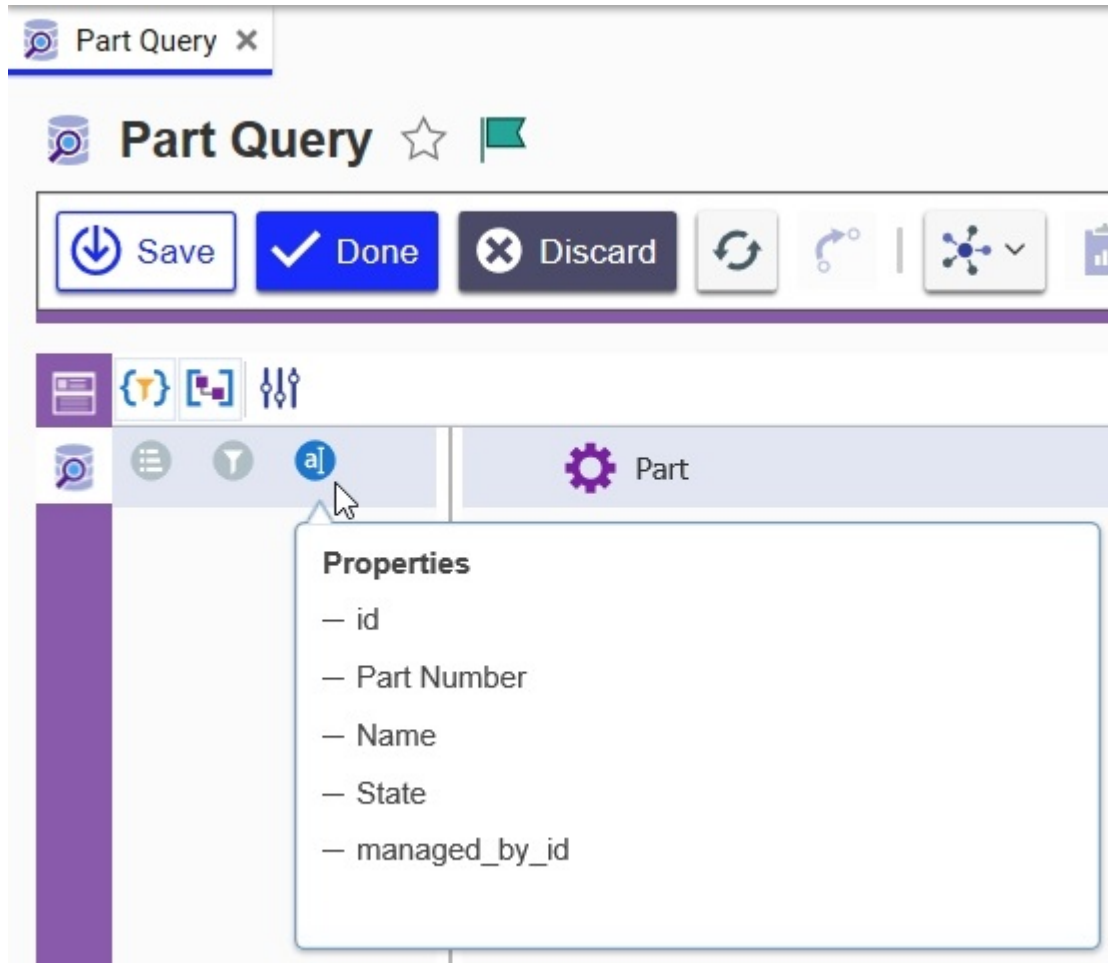
The screenshot shows a software interface for defining a query. At the top, there's a tab labeled 'Part Query' with a close button. Below the tab is a header area with the text 'Part Query' and some icons. A ribbon contains several icons: an 'Edit' button, a refresh icon, a circular arrow icon, a network diagram icon, a bar chart icon, a flow diagram icon, and a more options icon. On the left side, there's a sidebar with a 'Show Editor' icon. The main area is titled 'Query Definition' and contains three fields: 'Name' with the value 'Part Query', 'Context Item Type' with the value 'Part', and a large 'Description' text area.

1. Click null . The Show Editor icon appears in the ribbon.
2. Click the **Show Editor** icon on the left sidebar to go into Query-Editing Mode.

3. Right-click the Context Item and select **Edit Query Element --> Selected Properties**.

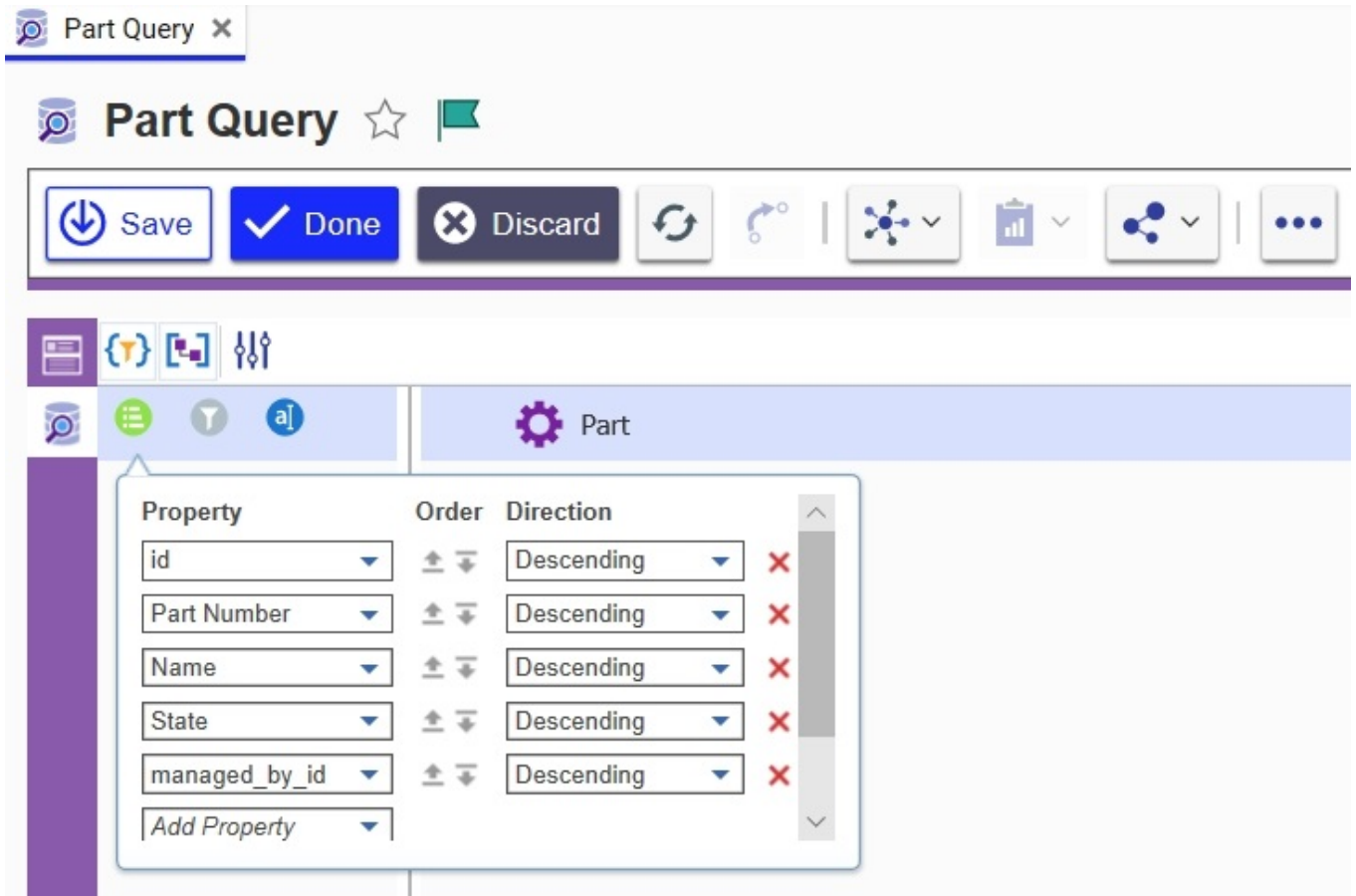


Select the desired properties that the query should return. In this example they are: id, Part Number, Name, State, and managed_by_id.



Extended properties enable you to add values such as cost to an object in the database.

4. Right-click on the Context Item again and select **Edit Query Element --> Sort Order**.
5. Choose the properties that should be used for the Order By command and specify whether they should be in ascending or descending order.



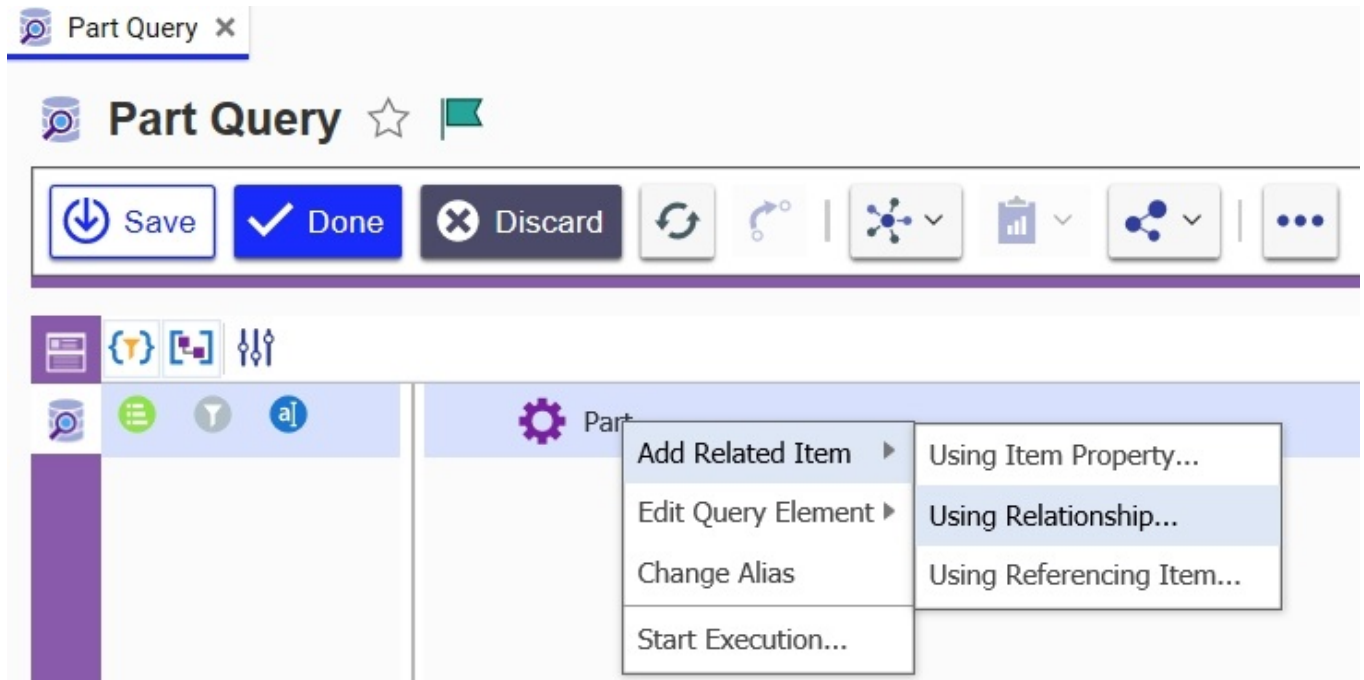
6. Save the Query.

Building a Relationship Structure

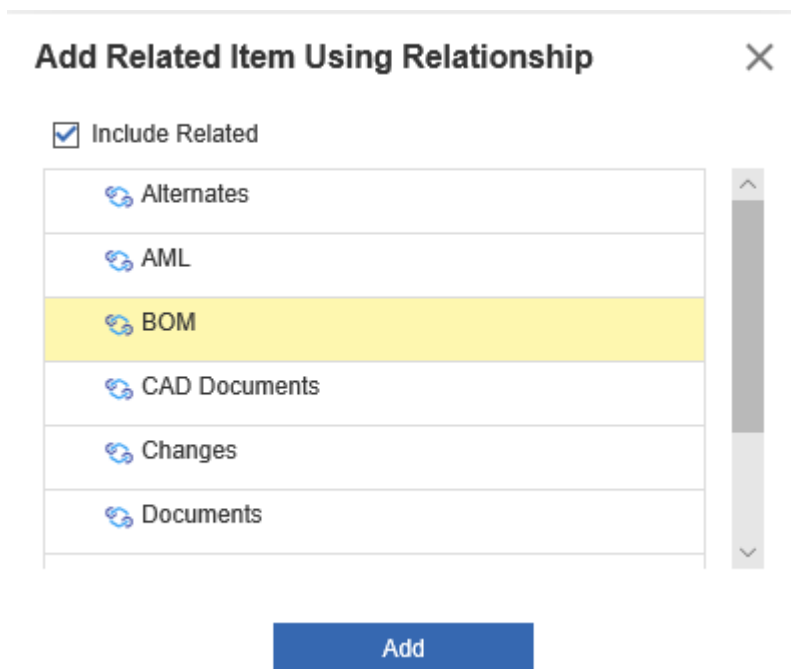
It is possible to define a Relationship from the context item that can go either up or down the data structure. 'Down' refers to a relationship direction starting from the Item representing the *source* side of the relationship and following the related content to the *related* Item. 'Up' by contrast refers to a relationship direction starting from the Item representing the *related* side of the relationship and following the related content to the *source* Item. It is also possible to set up a recursive structure, such as the Part --> Part BOM --> Part, to reuse the same logic for multiple levels. The following procedures show how to expand the query both up and down for a Part Query and introduces recursion.

Building a Downward Relationship Structure

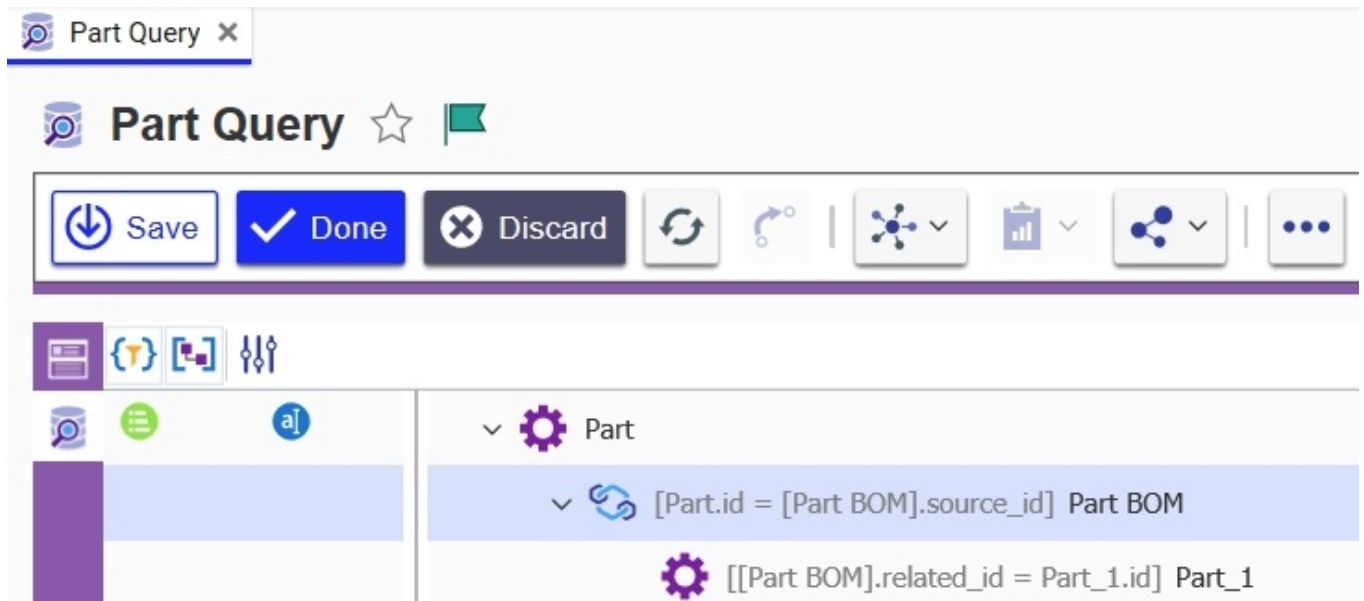
1. Right-click on the Part Context Item and select **Add Related Item --> Using Relationship**.



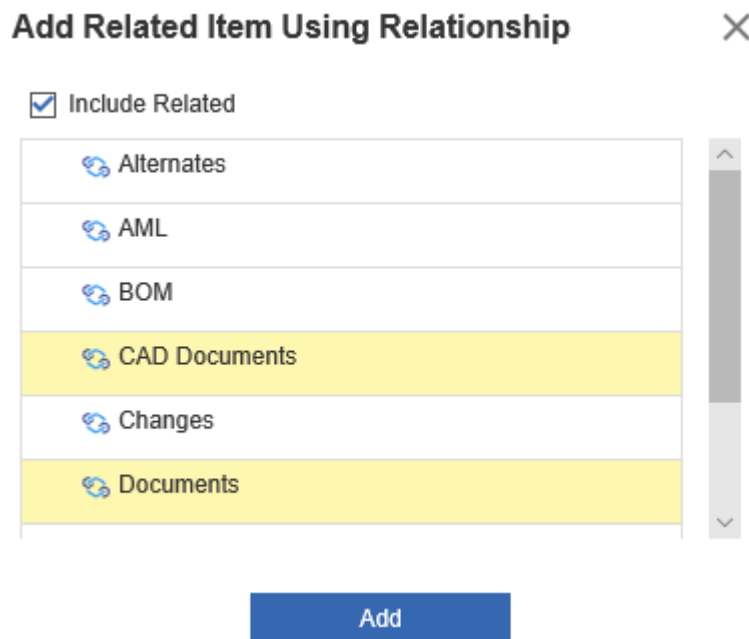
2. Select the Relationship to include in the structure. In this case, select **BOM**, check **Include Related** and click **Add**.



The query now looks like shown in figure 10. Note that the **Part_1** alias is assigned to the related Part Query Element in the list.



3. Right-click the **Part** Context Item and click **Add Related Item --> Using Relationship**. In this case, click while pressing Ctrl to select **Documents** and **CAD Documents**. Select **Include Related**.



4. Right-click on the top Context Item and select **Add Related Item --> Using Item Property**. Select an Item Property to add to the query structure. In this case, select **created_by_id**.

Add Related Item Using Item Property ×

Find property to add:

⚙️ config_id

👤 created_by_id

⦿ current_state

⚙️ id

👤 locked_by_id

🔗 managed_by_id

^

▼

Add

5. Use the following procedure to specify the desired Properties for each new element in the Query Definition:

Important

Optionally, you can also set Order By settings as well.

1. For [created_by_id] User, specify the following properties:
 - Item ID
 - First Name
 - Last Name
 - Login Name
2. For [Relationship] Part CAD, specify the following property:
 - Sequence
3. For [Related] CAD, specify the following properties:

- id
- Document Number
- Name
- State

4. For [Relationship] Part Document, specify the following property:

- Sequence

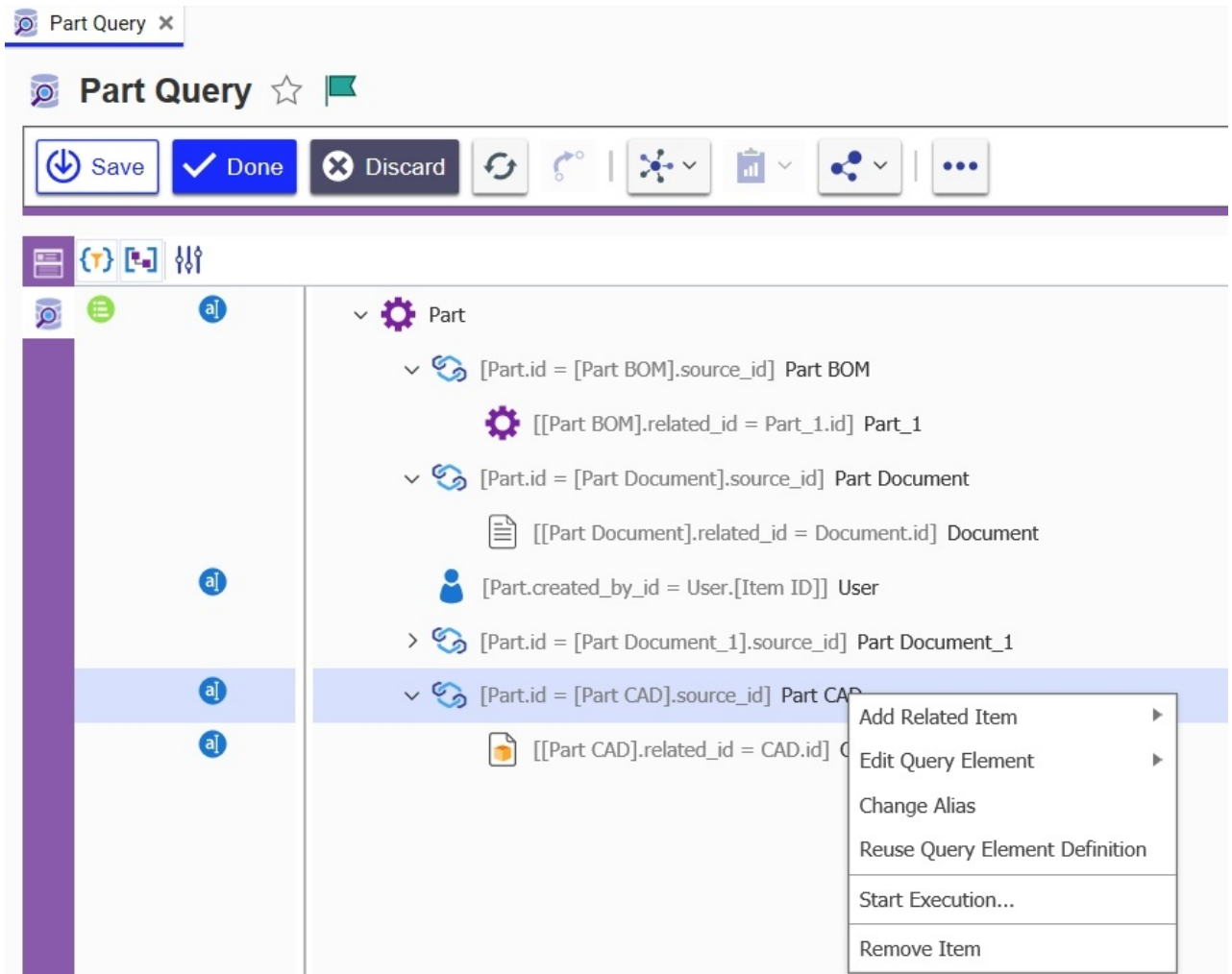
5. For [Related] Document, specify the following properties:

- id
- Document Number
- Name
- State

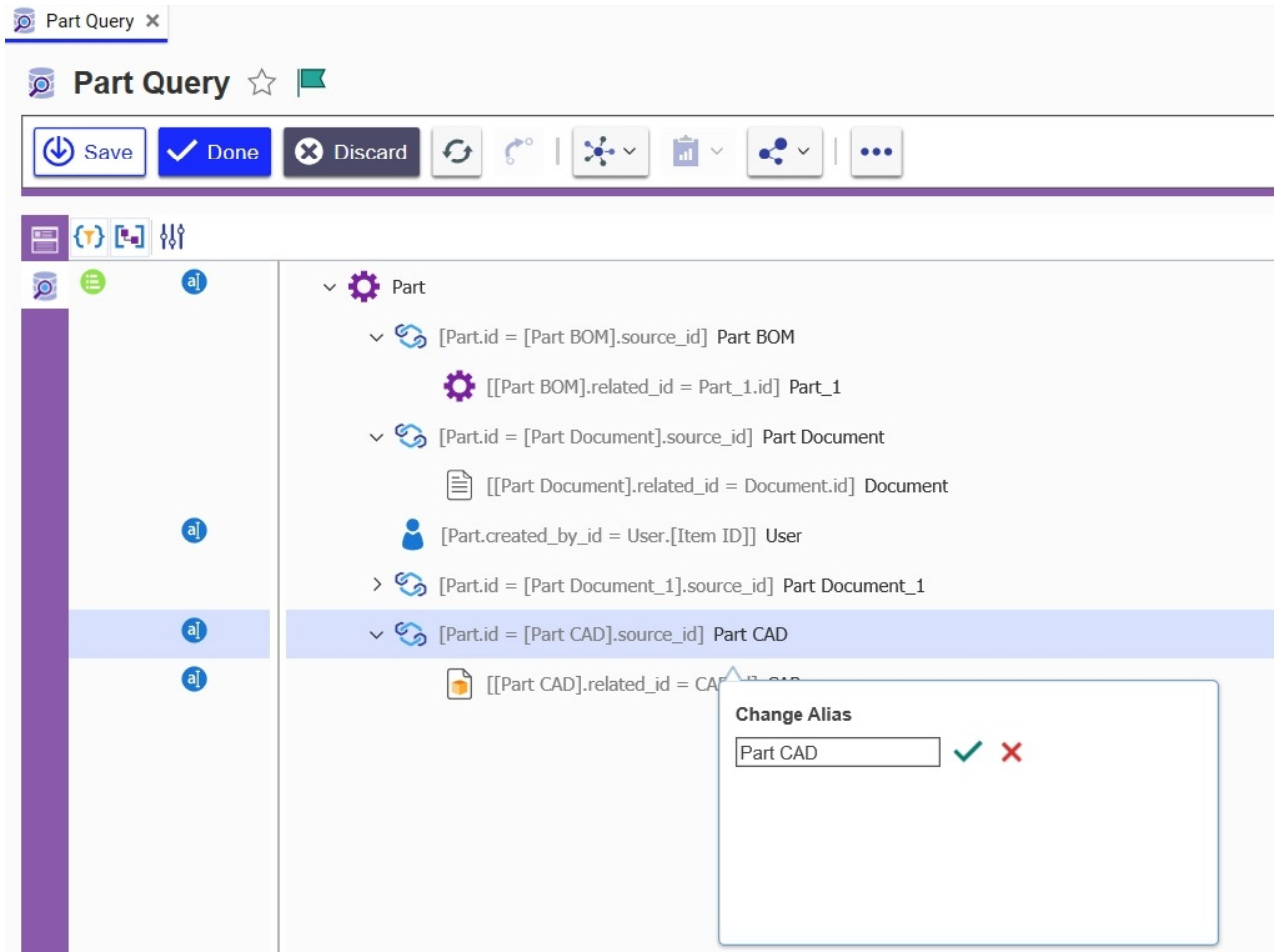
6. Change the Alias on the **CAD Documents** element to “Part CAD”:

1. Right-click **CAD Documents** and click **Change Alias**.





2. Enter **Part CAD** as the name and click the check mark.



7. Use the same procedure to change **Documents** Relationship to **Part Document**.
8. Use the same procedure to change **User** Relationship to **Part Created By**.

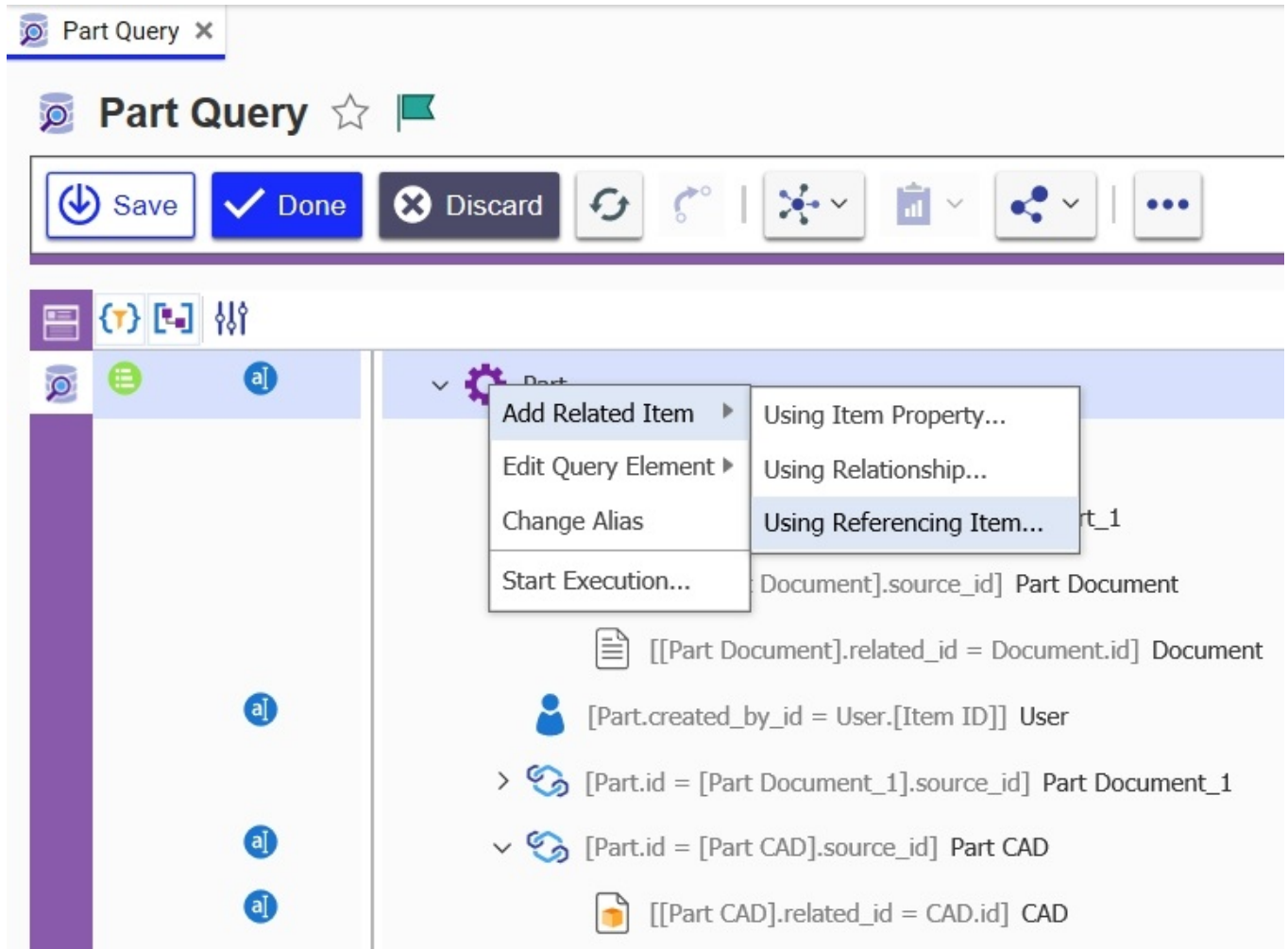
Important

Each element on a given level must have a unique alias.

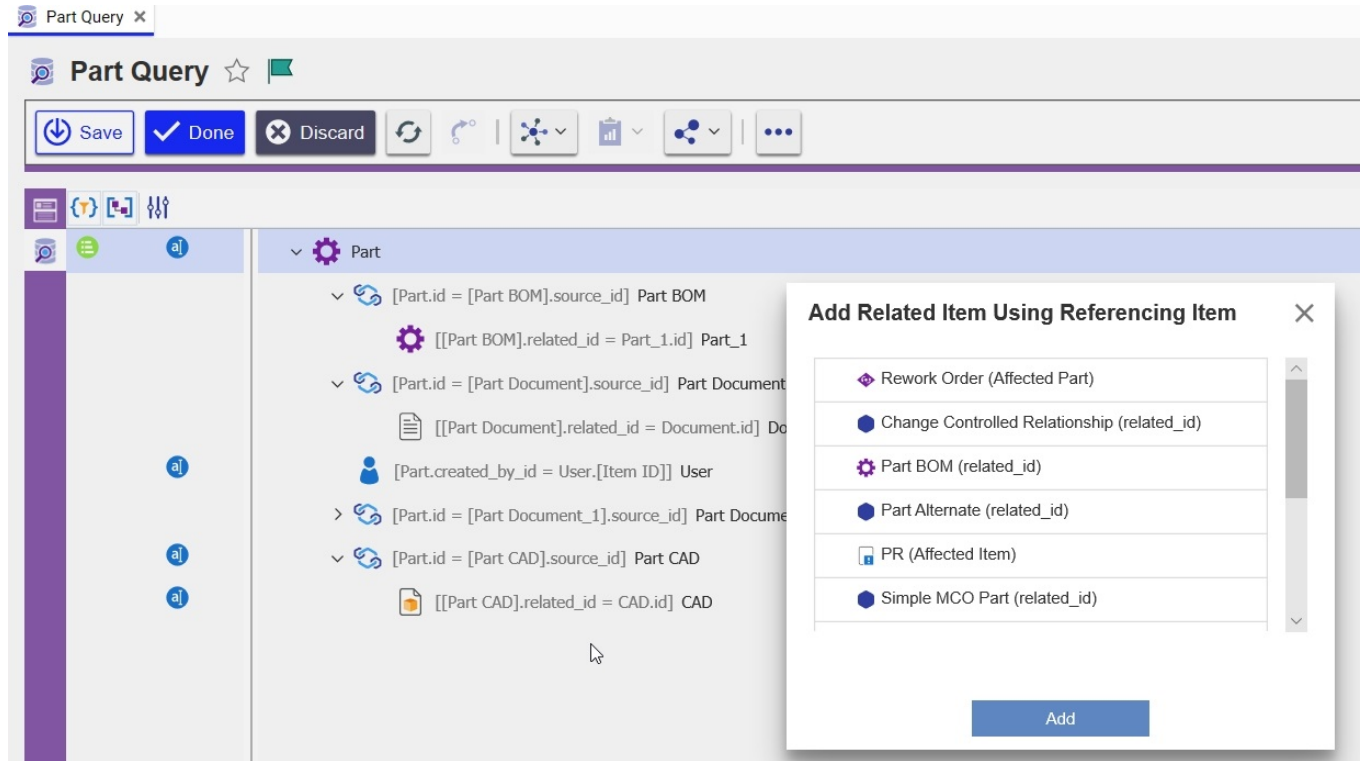
9. Save the Query Definition.

Building an Upward Relationship Structure

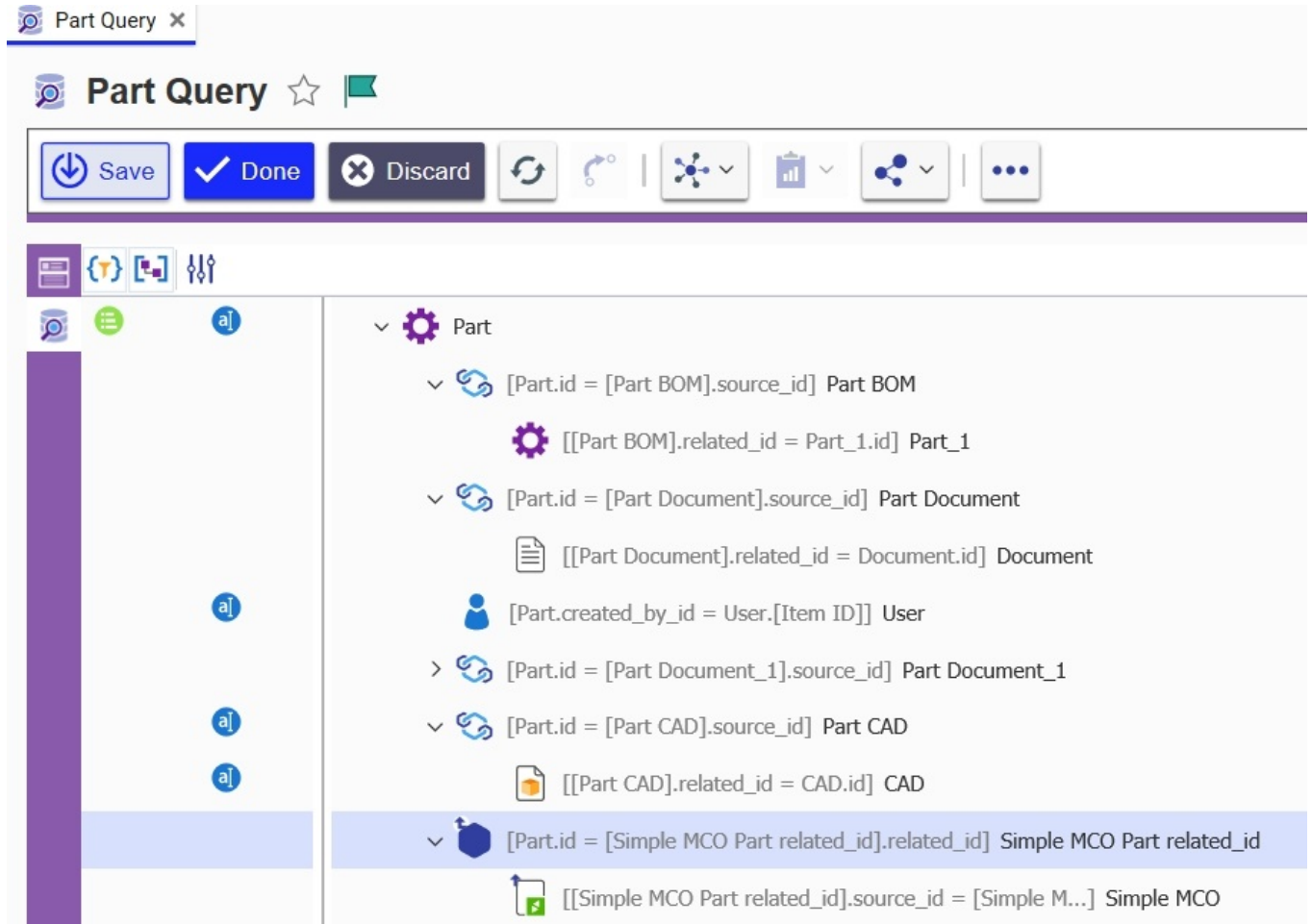
1. Right-click the top **Part** element and click **Add Related Item --> Using Referencing Item**.



2. Select an appropriate Referencing item from the structure. In this case, select **Simple MCO Part**.



3. Right-click the **Simple MCO Part** element and click **Add Related Item --> Using Item Property** to select **source_id**.



4. Set Selected Properties on the Simple MCO element to:

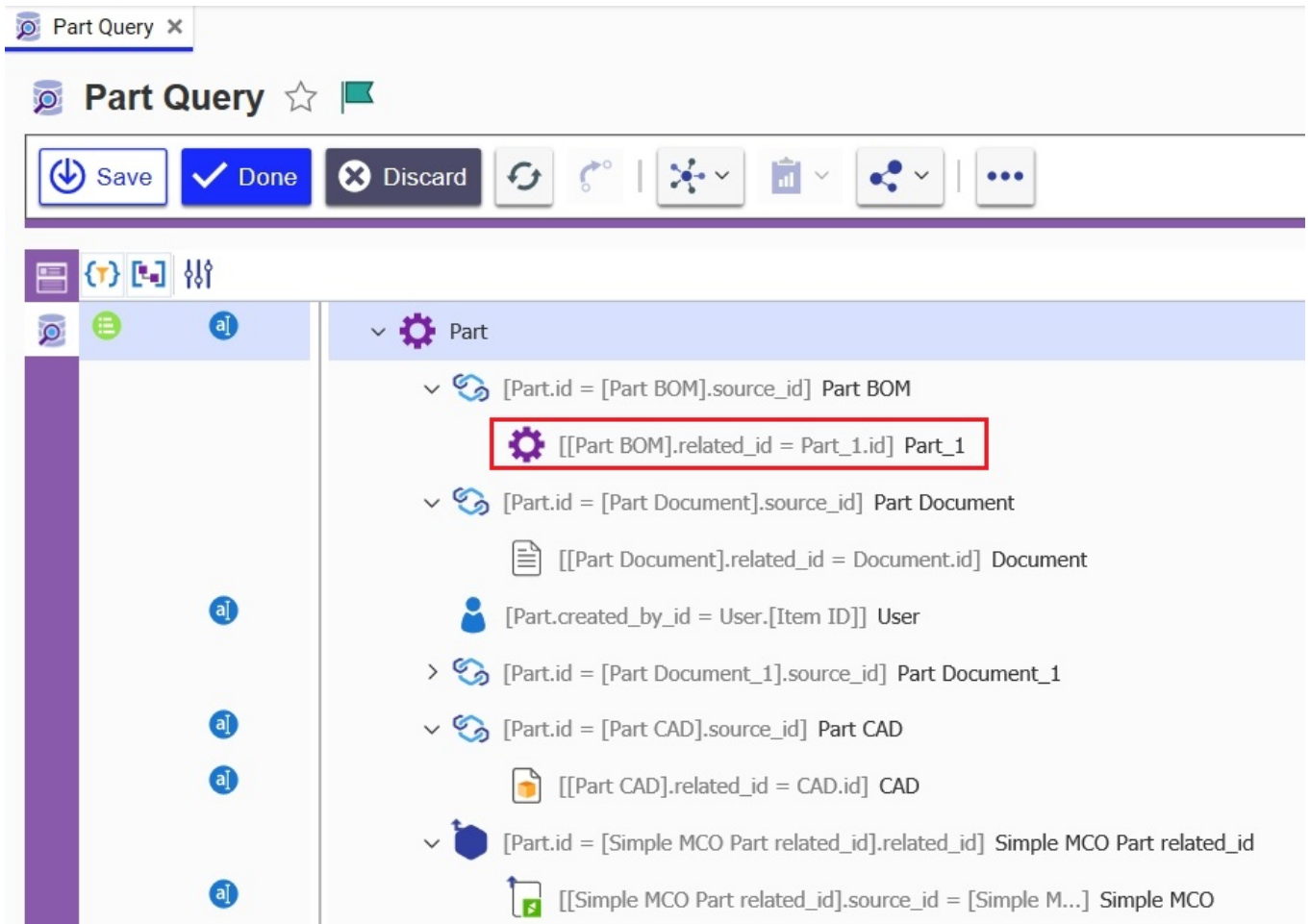
- id
- MCO Number
- Status
- Date Originated

5. Save the Query Definition.

Adding a Recursive Call Structure

Recursive structures can reuse sections of the already defined queries. The following example shows how to set up a recursive call for the **Part** --> **Part BOM** --> **Part** relationship structure.

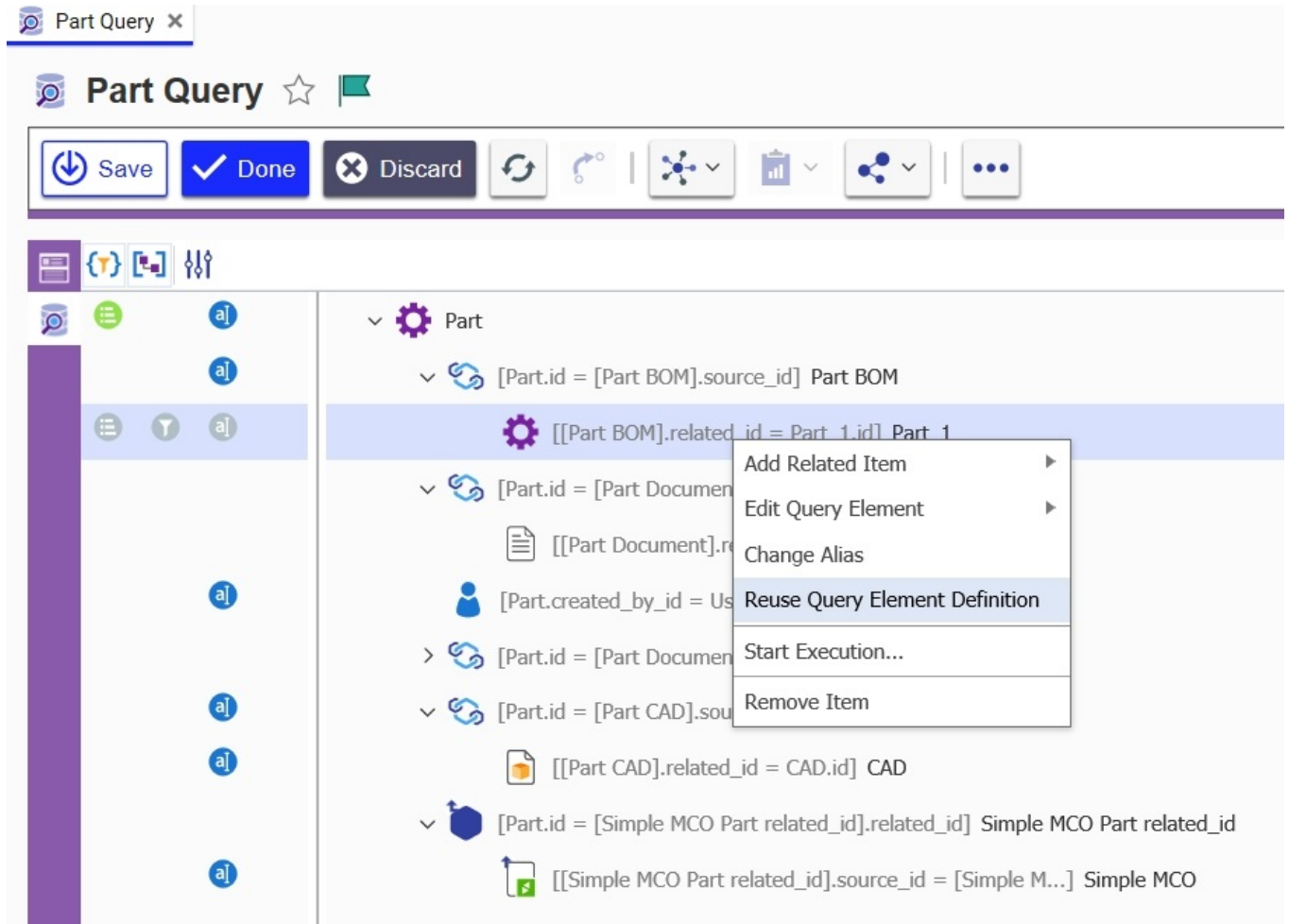
1. Right-click on the top-level **Part** element and add the **Part BOM** --> **Part** structure.



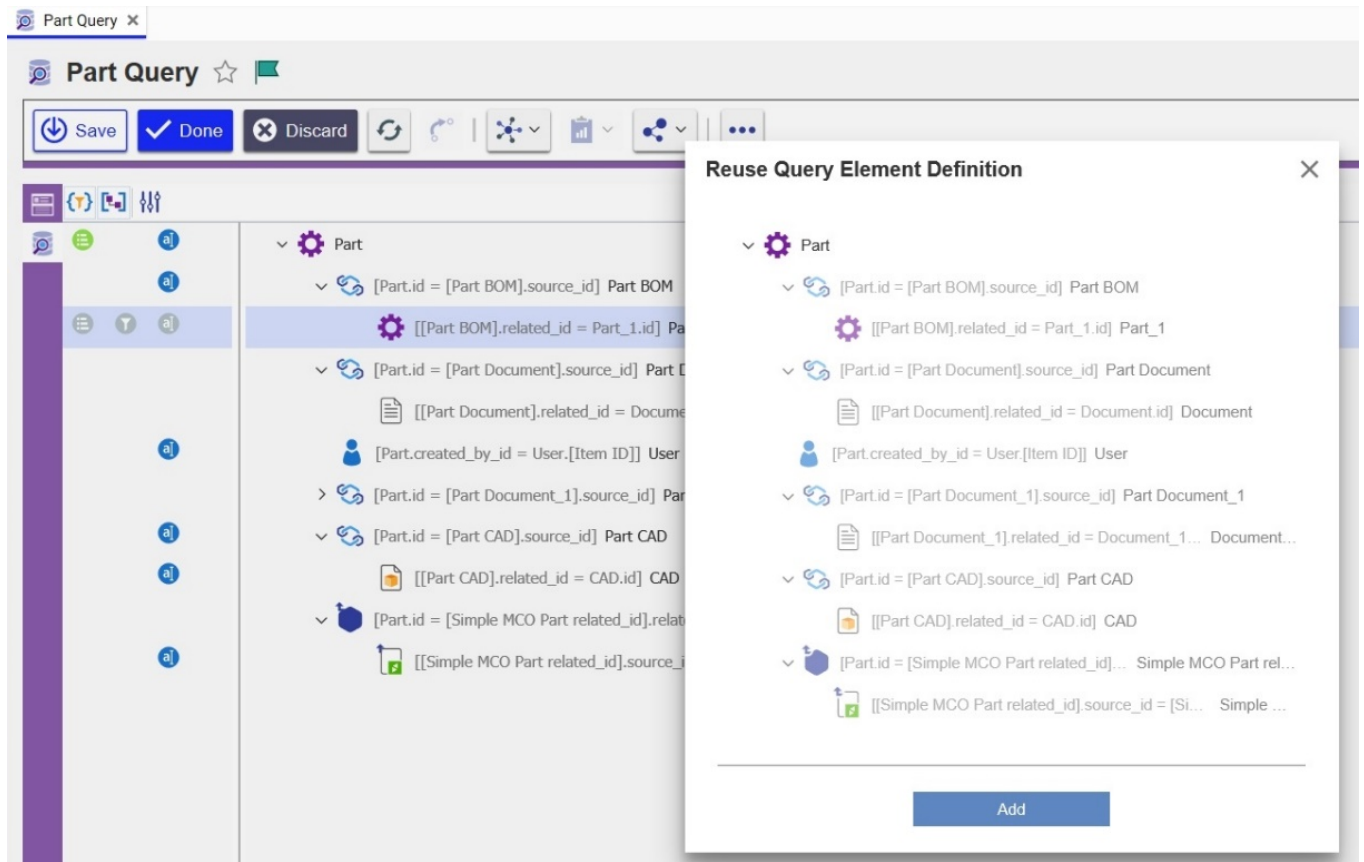
2. For the BOM element, select the following properties:

- Sequence
- Quantity
- related_id

3. Right-click on the child Part element and click **Reuse Query Element Definition**.



4. Select the top-level **Part** element and click **Add**.



5. Save the Query Definition item.

The child Part item now includes a search for the same elements as the top-level Part, including CADs, Documents, other Parts, etc. It also includes the same Properties and Order By logic as specified above.

Adding Search Conditions

You can specify Search conditions for every level of the Query Definition Structure. These search conditions can refine the set of data the Query returns. The query logic can include the components shown in Table 2. Query elements

Comparison Operators Boolean Operators Aggregate Functions

=!>>>=<<=<likeis null ANDORNOT Count()

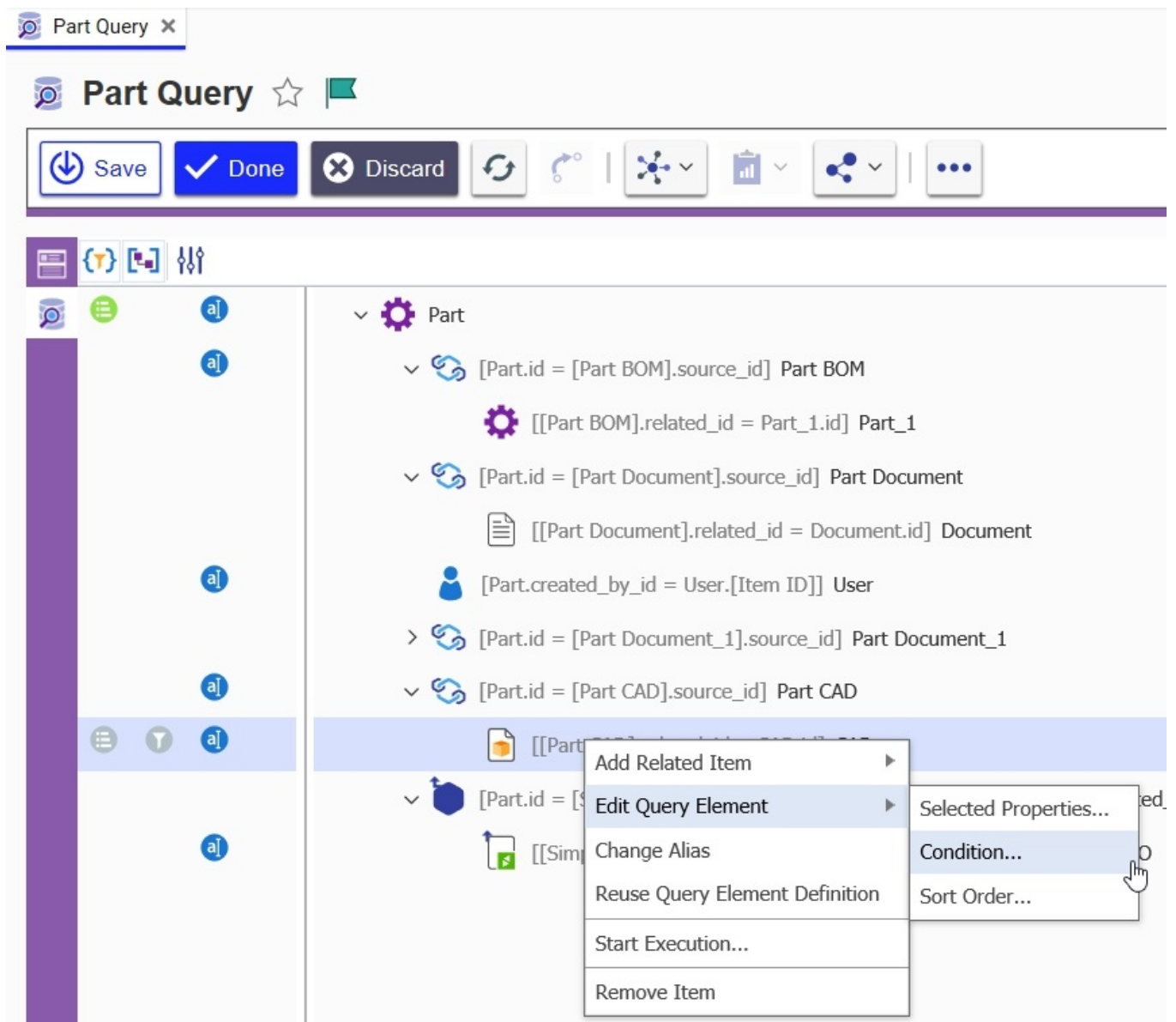
Important



Query Builder does not support comparison of string and integer values. For example, you can use logic like `Count ( ... ) = 3` , but you cannot use `Count ( ... ) = " 3 "` .

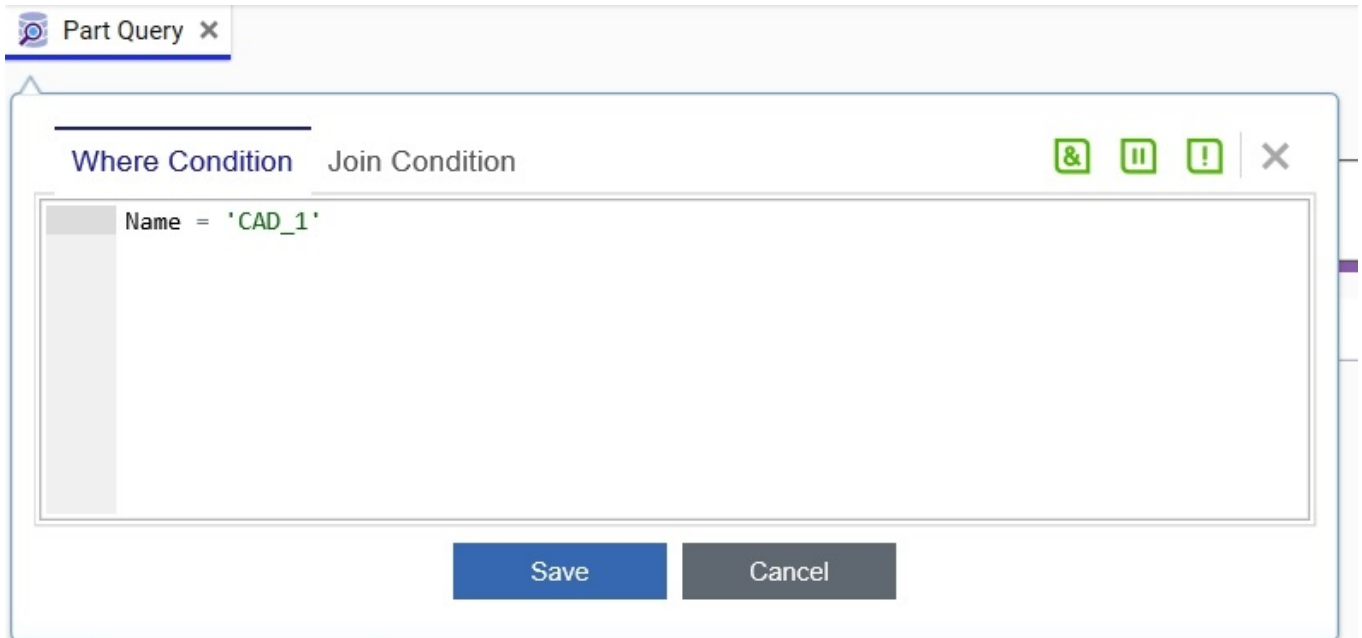
Use the following process to add Conditions to the **Query Definition** item:

1. Right-click the **CAD** element and click **Edit Query Element --> Condition**.



2. Enter a search condition, such as `[Name] = 'CAD_1'Na`.

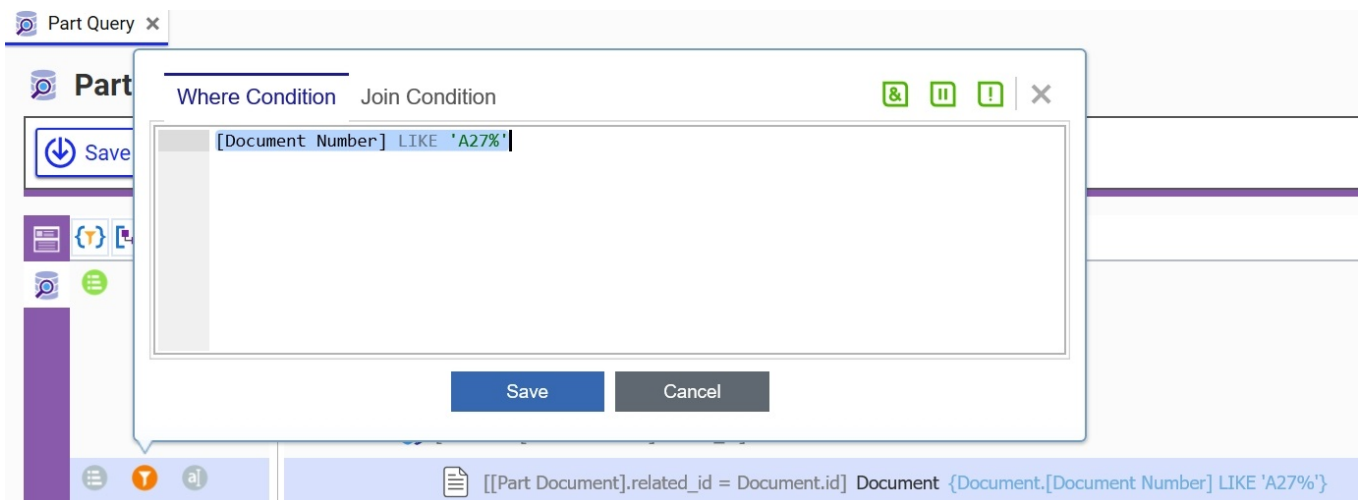




Important

Conditions written in Query Builder must be separated by white space. For example, you can use the condition `Cost >= 0` but not `Cost>=0`.

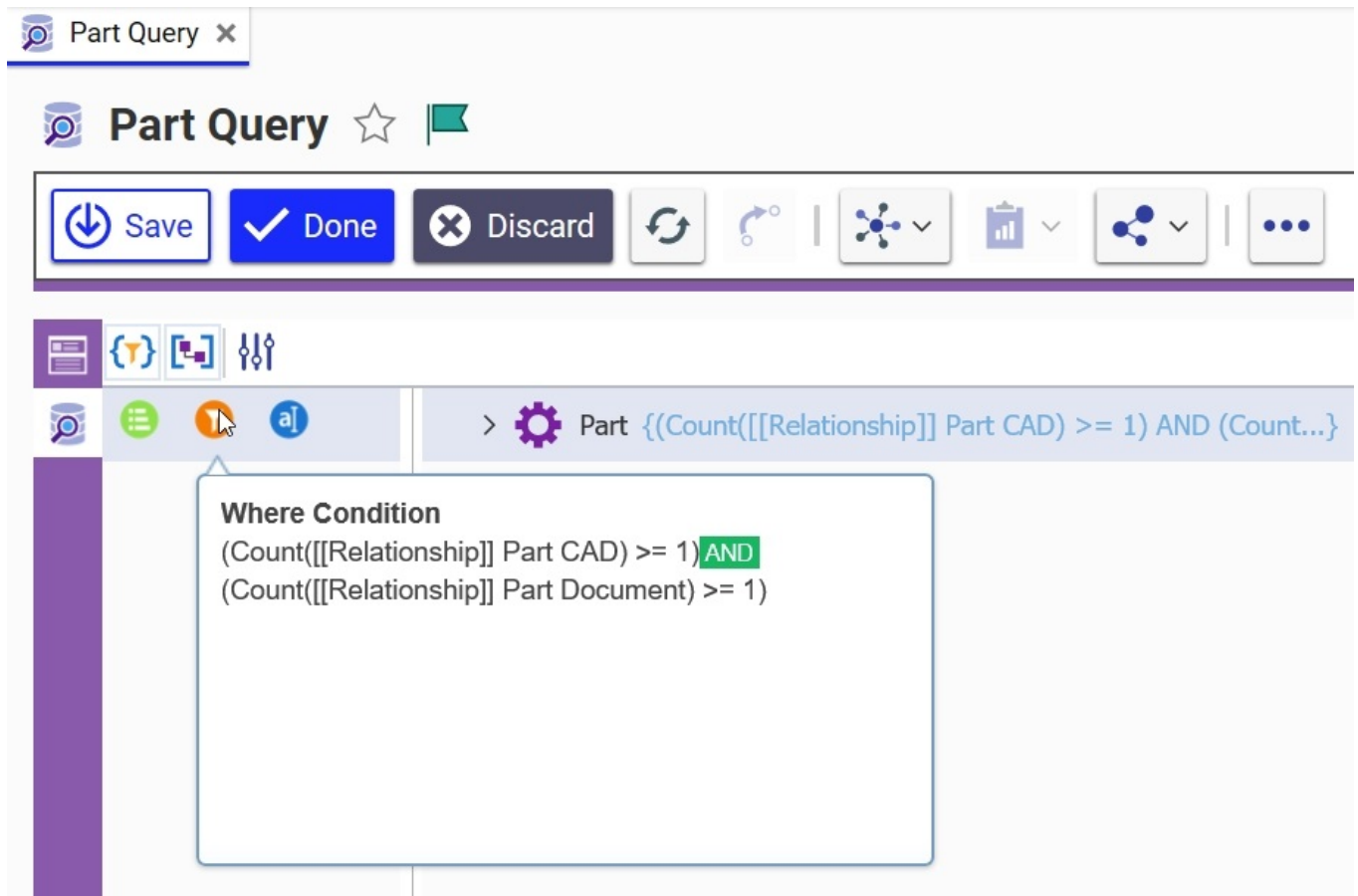
3. Right-click the **Document** element and click **Edit Query Element --> Condition**.
4. Enter a search condition such as `[Document Number] like 'A27%'`.



Important

Brackets are placed around multi-word properties in the condition window of Query Builder. You may notice that these brackets automatically disappear for single-word properties.

5. Right-click the top **Part** element and click **Edit Query Element --> Condition**.
6. Enter a search condition that enforces the CAD and Document Relationships to exist:
7. $(\text{Count}([[\text{Relationship}]] \text{ Part CAD}) \geq 1) \text{ AND } (\text{Count}([[\text{Relationship}]] \text{ Part Document}) \geq 1)$

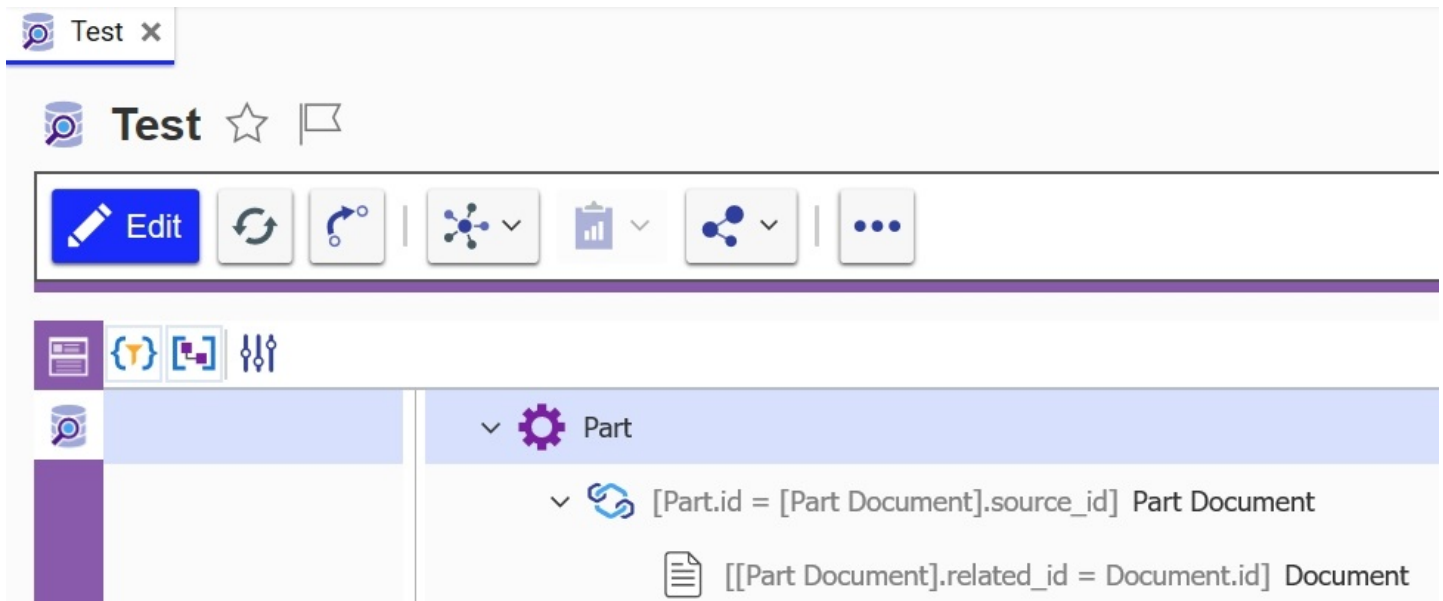


8. Save the Query Definition.

Adding/Editing Query Parameters



The following example walks you through the process of defining query parameters. It uses Test Query as an example.



1. Click the Edit Parameters button null . The Query Parameters dialog box appears.



Query Parameters

Actions

No Related

Name

Label

Default Value [...]

< Prev

Next >

Page: 1 of 1

0 Results

2. Click the New button null to add a parameter. Enter the following information:

Name: doc_state

Label: Document State

Default Value: *

Query Parameters

Actions

No Related

Name	Label	Default Value [...]
doc_state	Document State	*

< Prev

Next >

Page: 1 of 1

| 0 Results

|

3. Close the dialog box to save the parameter information.
4. Select the **Document** level of the relationship structure and click the associated **Filter** button to filter using the parameter you just created.
5. Set any other filters and sort orders that you need.
6. Save the Query definition.

The parameters that you add to the Query definition appear in the **Map Parameters** dialog box in the Tree Grid View.



Map Parameters

✓

✗

Actions ▾

No Related ▾

+ New relationship

🔄 Pick/Replace Related Item

✗ Delete relationship ▾

Visible	Name	Label	Default Value	Data Type	Data Source [...]	Pattern
☒	doc_state	Document State	*			

Once mapped, the parameters become available in the Tree Grid View under the Modify Parameters null button.

^

BOM

BOM Structure

Alternates

AML

Documents

CAD Docu...

🖨

🔍

🔄

3

🔄

📄

Part Number	R...	State	Seque...	Quanti
Part1	A	Released	128	1
Part1_1	C	Released	128	1
Part1_3	A	Preliminary	256	2
Part2	C	Released	256	2
Part2_1	A	Released	128	2
Part4	A	Released	384	4
Part4_1	A	Released	128	1



This allows the entry or selection of the configured parameters from the Tree Grid View, setting query parameters in the underlying Query Definition from the Tree Grid User Interface at runtime. Parameters can also be used for previewing query results when Executing a Query in the Query Builder. See Section 3.0 “Executing Queries” for an example of this.

Structure Resolution Parameter

For structures of versionable Items, like **Part** and **Part BOM** Structures, it is useful to examine query results in a Tree Grid View by either the latest versions, or by released versions only, or a combination where if no released components exist then the latest (un-released) version is displayed in the overall structure. The default is to display the structure as it is stored in the database without any optional resolution. These Resolution Modes are provided to **Part BOM** Query Definitions (and Tree Grid Views) by a special Parameter which is pre-configured OOTB. Structure Resolution can be applied to any Item Relationship structure of versionable Items that follows the same rules as Part BOM structures: the **Released** Life Cycle State (**is_released** Boolean), incremental Revisions with interim versioned generations, etc.

Default Resolution Mode

The **Default** Resolution Mode uses the versions of related child Parts as stored on the BOM Relationships. This is the standard resolution in Aras Innovator.



Part Number	Revision	State
☐ ⚙ Child A	B	Preliminary
☐ ⚙ Child A.1	A	Preliminary
☐ ⚙ Child A.2	C	Released
☐ ⚙ Child A.3	C	Preliminary
☐ ⚙ Child B	B	Released
☐ ⚙ Child B.1	A	Released
☐ ⚙ Child B.2	B	Released
☐ ⚙ Child C	A	Preliminary
☐ ⚙ Child C.1	A	Preliminary
☐ ⚙ Child C.2	B	Released

The **Default** resolution mode

Latest Resolution Mode

When the **Latest** resolution mode is selected, the structure is resolved starting from the top using the latest versions of child items. When a child item is resolved to its latest version, then its child items are resolved using this resolution mode. This is done recursively. The **Latest** resolution mode does not consider the released state of the child Parts.



Part Number	Revision	State
☐ Child A	B	Preliminary
☐ Child A.1	A	Preliminary
☐ Child A.2	C	Released
☐ Child A.3	C	Preliminary
☐ Child B	B	Released
☐ Child B.1	A	Released
☐ Child B.2	D	Preliminary
☐ Child C	A	Preliminary
☐ Child C.1	A	Preliminary
☐ Child C.2	B	Released

Latest resolution mode

The latest revision **D** of **Child B.2** (Preliminary) is shown where this Part version is in the **Preliminary** state, although the default structure under revision **B** of **Child A** (this structure) refers to revision **B** of **Child B.2** (Released) as stored in the database. In other words, the Latest Resolution Mode dynamically provides a view to in-work changes to released parts that are not yet committed to the released structure.

Latest Released Resolution Mode

When the **Latest Released** resolution mode is selected, the structure is resolved starting from the top using the latest versions of child items that are in Released state. When the child item is resolved to its latest Released version, then its child items are resolved using this resolution mode. This is done recursively. If the related child item has no versions that are in Released state, then a blank row is displayed for that item. There is no further processing on that item.



Part Number	Revision	State
 Child A	A	Released
 Child A.2	C	Released
 Child A.3	B	Released
 Child B	B	Released
 Child B.1	A	Released
 Child B.2	C	Released
		

Latest Released resolution mode

The latest Released versions of child parts are resolved. A blank row is displayed for **Child C** because it has no version that is in the Released state.

Latest Released Or Latest Resolution Mode

When the **Latest Released or Latest** resolution mode is selected, the structure is resolved starting from the top using the latest versions of child items that are in Released state. If there are no versions that are in Released state, then the latest version is used. When the child item is resolved to a version, then its child items are resolved using this resolution mode. This is done recursively.



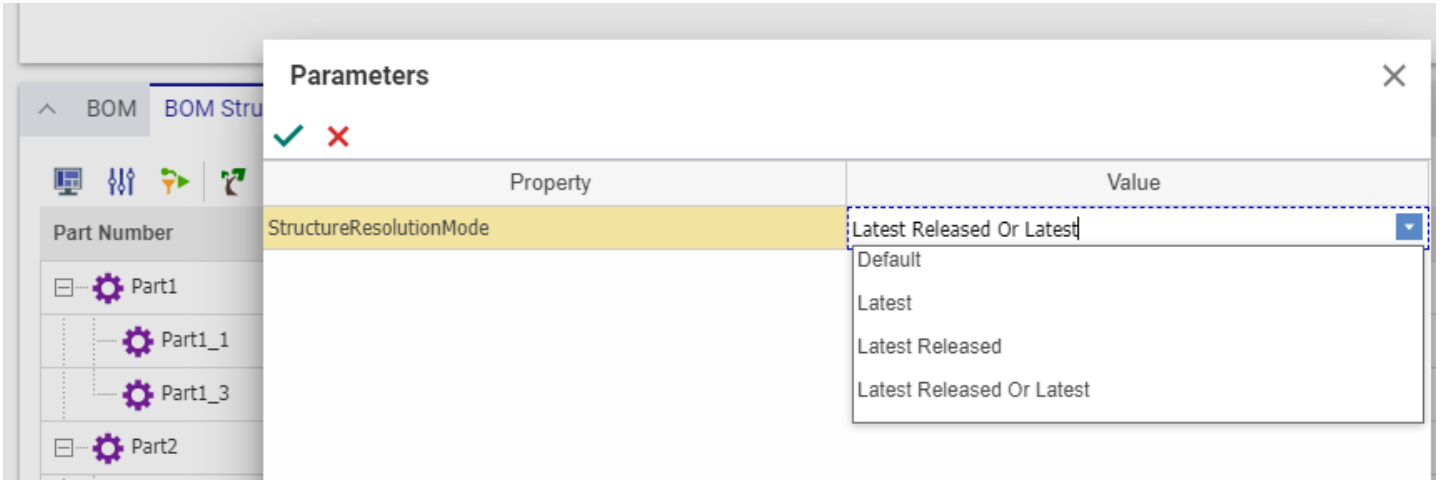
Part Number	Revision	State
☐ Child A	A	Released
☐ Child A.2	C	Released
☐ Child A.3	B	Released
☐ Child B	B	Released
☐ Child B.1	A	Released
☐ Child B.2	C	Released
☐ Child C	A	Preliminary
☐ Child C.1	A	Preliminary
☐ Child C.2	B	Released

Latest Released or Latest resolution mode

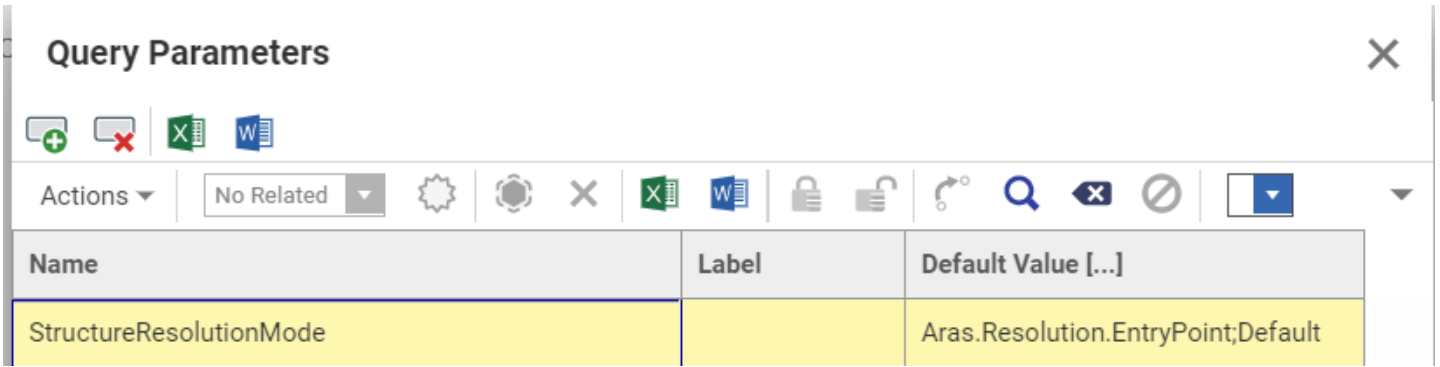
The latest Released versions of child parts are resolved. **Child C** does not have any version that is in the Released state. Therefore, its latest Preliminary version is resolved.

Structure Resolution Configuration for Part BOM Structures

In Aras Innovator, the BOM Structure grid (a TGV under the **BOM Structure** tab in the **Part** form) is pre-configured with a Structure Resolution Parameter, thus it supports the **Latest**, **Released**, and **Released or Latest** modes out of the box.



In the event you need to enable Structure Resolution for your Query Definition, you will need to add a Parameter that has a default value of **Aras.Resolution.EntryPoint;Default**, and then also follow the additional detailed steps for configuring Structure Resolution Parameters described in Section 7.



For more details and extensive examples of how Resolution Modes are applied in BOM Structures, Effectivity, and 3D Visualization please refer to the following documents:

- *Aras 3D Visualization - Administrator Guide*
- *Aras Innovator Product Engineering - User's Guide*

