Perceptive Business Insight

Advanced Design and Setup Guide

Version: 7.0.x

Written by: Product Knowledge, R&D
Date: August 2016

perceptivesoftware
from Lexmark

Table of Contents

Add Custom Property information to reports

As you create custom reports using the provided published packages (Business Logic View and Data Source View), it may be helpful to have a custom property included in the result set. There are multiple ways to add custom properties to a report to provide more information stored for the Perceptive Content data. Custom properties used in a report can then be aggregated like any other field to provide further insight into the data. This document provides several scenarios that can deliver that data in a meaningful way.

For each of the examples below, you must be logged in as a report author with appropriate privileges. All data for these examples is contained within a drawer named “CP Demo” to assist with data generation and report creation.

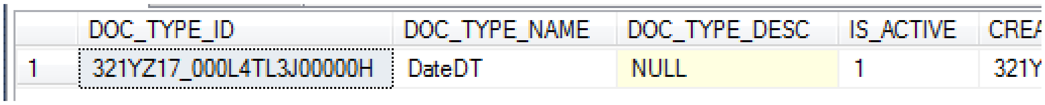
How Custom Properties are stored in the database

When a document type or folder type is assigned one or more custom properties, any document or folder created with that type then inherits those custom properties. Those values are then stored in the database in various tables, which are then retrieved later in the viewer or for reporting purposes. Typically, most custom properties are recorded in the same way. The following SQL exercises demonstrate how to find a custom property within the data.

Exercise 1

In the following example, review the document type table.

```
SELECT *
FROM IN_DOC_TYPE
```

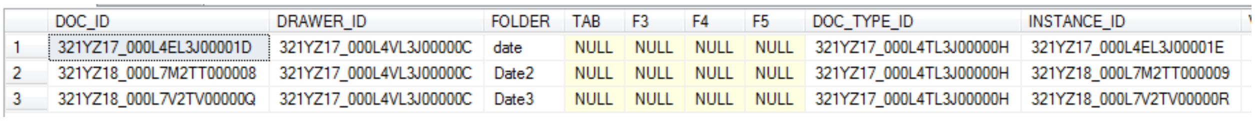


	DOC_TYPE_ID	DOC_TYPE_NAME	DOC_TYPE_DESC	IS_ACTIVE	CREA
1	321YZ17_000L4TL3J00000H	DateDT	NULL	1	321Y

From the output of this query, locate a document type that is known to have custom properties associated with it. Copy the DOC_TYPE_ID for the next query.

As with most queries, continue to build on the existing information from above and add the IN_DOC table.

```
SELECT IN_DOC.*
FROM IN_DOC_TYPE
INNER JOIN IN_DOC
       ON IN_DOC_TYPE.DOC_TYPE_ID = IN_DOC.DOC_TYPE_ID
WHERE IN_DOC_TYPE.DOC_TYPE_ID = ‘321YZ17_000L4TL3J00000H’
```

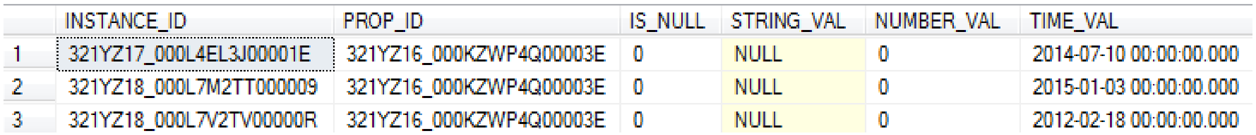


	DOC_ID	DRAWER_ID	FOLDER	TAB	F3	F4	F5	DOC_TYPE_ID	INSTANCE_ID
1	321YZ17_000L4EL3J00001D	321YZ17_000L4VL3J00000C	date	NULL	NULL	NULL	NULL	321YZ17_000L4TL3J00000H	321YZ17_000L4EL3J00001E
2	321YZ18_000L7M2TT000008	321YZ17_000L4VL3J00000C	Date2	NULL	NULL	NULL	NULL	321YZ17_000L4TL3J00000H	321YZ18_000L7M2TT000009
3	321YZ18_000L7V2TV00000Q	321YZ17_000L4VL3J00000C	Date3	NULL	NULL	NULL	NULL	321YZ17_000L4TL3J00000H	321YZ18_000L7V2TV00000R

Note The INSTANCE_NAME displays, in this case, the document name. For a folder, it displays the folder name, and so on.

You can then derive the property value of the custom property by including the IN_INSTANCE_PROP table.

```
SELECT IN_INSTANCE_PROP.*
FROM IN_DOC_TYPE
INNER JOIN IN_DOC
       ON IN_DOC_TYPE.DOC_TYPE_ID = IN_DOC.DOC_TYPE_ID
INNER JOIN IN_INSTANCE
       ON IN_DOC.INSTANCE_ID = IN_INSTANCE.INSTANCE_ID
INNER JOIN IN_INSTANCE_PROP
       ON IN_INSTANCE.INSTANCE_ID = IN_INSTANCE_PROP.INSTANCE_ID
WHERE IN_DOC_TYPE.DOC_TYPE_ID = '321YZ17_000L4TL3J00000H'
```



	INSTANCE_ID	PROP_ID	IS_NULL	STRING_VAL	NUMBER_VAL	TIME_VAL
1	321YZ17_000L4EL3J00001E	321YZ16_000KZWP4Q00003E	0	NULL	0	2014-07-10 00:00:00.000
2	321YZ18_000L7M2TT000009	321YZ16_000KZWP4Q00003E	0	NULL	0	2015-01-03 00:00:00.000
3	321YZ18_000L7V2TV00000R	321YZ16_000KZWP4Q00003E	0	NULL	0	2012-02-18 00:00:00.000

This exercise began by viewing a document type with a Date custom property. This information can now be seen in the TIME_VAL column. Below is a table that indicates where each custom property type is located within IN_INSTANCE_PROP.

IN_PROP.PROP_TYPE Value	Custom Property Type	Table/Field Location
3	Date	IN_INSTANCE_PROP.TIME_VAL
4	Flag	IN_INSTANCE_PROP.NUMBER_VAL*
5	List	IN_INSTANCE_PROP.STRING_VAL**
2	Number	IN_INSTANCE_PROP.NUMBER_VAL
1	String	IN_INSTANCE_PROP.STRING_VAL
6 or 7	User	IN_INSTANCE_PROP.STRING_VAL***

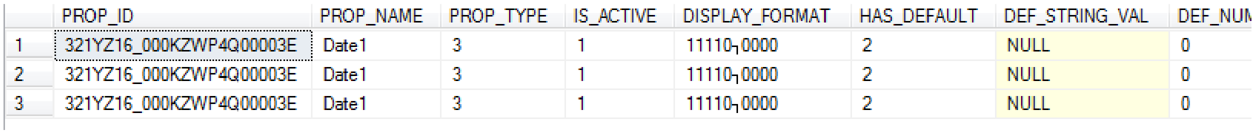
* The Flag custom property is a binary (0 or 1) indicator that displays if the client displays the positive or negative flag result from IN_PROP.DISPLAY_FORMAT.

** The List custom property is a unique ID that references IN_PROP_VAL.PROP_VAL_ID for the list value. Refer to **Appendix A** for an example.

*** The User custom property is a unique ID that references IN_SC_USR.USR_ID. Refer to **Appendix A** for an example.

The IN_PROP table provides basic information about the custom property type itself, much like the document type table. If there are default values or if the custom property is a flag, the information is found in the following example.

```
SELECT IN_PROP.*
FROM IN_DOC_TYPE
INNER JOIN IN_DOC
       ON IN_DOC_TYPE.DOC_TYPE_ID = IN_DOC.DOC_TYPE_ID
INNER JOIN IN_INSTANCE
       ON IN_DOC.INSTANCE_ID = IN_INSTANCE.INSTANCE_ID
INNER JOIN IN_INSTANCE_PROP
       ON IN_INSTANCE.INSTANCE_ID = IN_INSTANCE_PROP.INSTANCE_ID
INNER JOIN IN_PROP
ON IN_INSTANCE_PROP.PROP_ID = IN_PROP.PROP_ID
WHERE IN_DOC_TYPE.DOC_TYPE_ID = '321YZ17_000L4TL3J00000H'
```



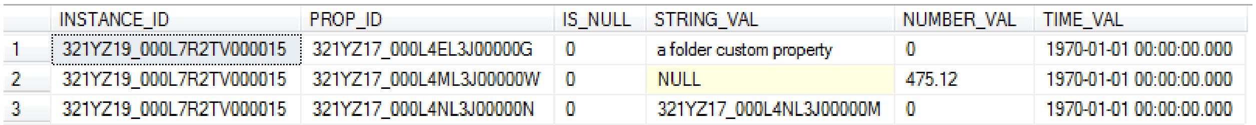
	PROP_ID	PROP_NAME	PROP_TYPE	IS_ACTIVE	DISPLAY_FORMAT	HAS_DEFAULT	DEF_STRING_VAL	DEF_NUN
1	321YZ16_000KZWP4Q00003E	Date1	3	1	11110┐0000	2	NULL	0
2	321YZ16_000KZWP4Q00003E	Date1	3	1	11110┐0000	2	NULL	0
3	321YZ16_000KZWP4Q00003E	Date1	3	1	11110┐0000	2	NULL	0

Note The PROP_NAME displays the name of the custom property.

For the last exercise, examine the results that display if a folder has three different custom properties. This folder type consists of a list, number, and string. Below is an example of the query.

```
SELECT IN_INSTANCE_PROP.*
FROM IN_PROJ_TYPE
INNER JOIN IN_PROJ
       ON IN_PROJ_TYPE.PROJ_TYPE_ID = IN_PROJ.PROJ_TYPE_ID
INNER JOIN IN_INSTANCE
       ON IN_PROJ.INSTANCE_ID = IN_INSTANCE.INSTANCE_ID
INNER JOIN IN_INSTANCE_PROP
       ON IN_INSTANCE.INSTANCE_ID = IN_INSTANCE_PROP.INSTANCE_ID
INNER JOIN IN_PROP
       ON IN_INSTANCE_PROP.PROP_ID = IN_PROP.PROP_ID
WHERE IN_PROJ_TYPE.PROJ_TYPE_ID = '321YZ19_000L7X2TV00002E'
```

The only modifications that occur to get folder information are the IN_DOC and IN_DOC_TYPE changes to IN_PROJ and IN_PROJ_TYPE. Otherwise, everything else remains the same and the results are shown in the following example.



	INSTANCE_ID	PROP_ID	IS_NULL	STRING_VAL	NUMBER_VAL	TIME_VAL
1	321YZ19_000L7R2TV000015	321YZ17_000L4EL3J00000G	0	a folder custom property	0	1970-01-01 00:00:00.000
2	321YZ19_000L7R2TV000015	321YZ17_000L4ML3J00000W	0	NULL	475.12	1970-01-01 00:00:00.000
3	321YZ19_000L7R2TV000015	321YZ17_000L4NL3J00000N	0	321YZ17_000L4NL3J00000M	0	1970-01-01 00:00:00.000

The INSTANCE_ID for the three rows are all the same, indicating that this all points to one unique instance (one folder). The PROP_ID values are unique and indicate different custom property values in the STRING_VAL and NUMBER_VAL fields. The TIME_VAL field is not used in this example.

Create a new report from view using multiple custom properties

A simple and effective way to bring custom property data into your report is to use the Report from View feature in Perceptive Content. A view is defined as a subset of all content that is tailored for you, meaning the view can hide dozens, hundreds, or even thousands of results that might otherwise clutter your search results list. Views display only the columns you need to see or work with, including custom property columns. In addition, your view is customized so that the appearance, name, and width of columns fit your needs. A Report from View takes advantage of the pre-built and custom views in Perceptive Content that allow you to add several types of custom properties to a view screen. After the view is created, the report is created.

1. In **Management Console**, navigate to the report section.
2. Click **New > Report**.
3. In the **Create Report from View** dialog box, choose the view type, and then select the view and the location where you want to create the report.
4. Rename the report if desired.
5. Click **OK**.

Report Studio then launches with a shell of how the view is laid out. The report can be further modified to add charts or include aggregations, or simply run the report as it exists.

There are multiple benefits to using this method versus other methods listed later in this document.

- Simplicity. This is the easiest way to include multiple custom properties in one report.
- A report is automatically created to mimic the view, but other query fields are available to enhance the report if necessary.
- The query generated in the report from view is highly optimized and should perform very well for your report.
- Add multiple custom properties to a view without issues.
- Statistics added to the view are automatically added to the report.

As with the view itself, composite properties are not available in the report from view.

Package versus Direct SQL

Predefined Package

The predefined packages that are deployed with the Perceptive Business Intelligence suite are located under the Packages folder in Cognos. Currently there are two packages: Business Logic View and Data Source View. Only one package can be used in a report at any given time. Packages are like predefined query generators that provide different views of your data.

Note The Business Logic View segments the sections into areas similar to what you see in the client (Document, Workflow, and so on), whereas the Data Source View provides the entire contents in one view. The Data Source View is intended for more advanced development and the Business Logic View is intended for more ad hoc and light consumption.

Direct SQL

Using direct SQL is an advanced technique that allows the report developer to insert a query into the report. The developer can also utilize a direct SQL query in an existing report consisting of one of the published packages to display a custom or composite property. It requires the user to understand and build out an SQL query to provide additional data, but after that data is built and joined appropriately, the data can be displayed and aggregated as normal. It is critical when linking an existing query to a new direct SQL query to have a common key that joins the two. After that common key exists, the author can then join the queries.

Note Adding the extra query to the report ultimately decreases report performance. Use this method only after ensuring the direct SQL query has been optimized.

Create a base report structure

The base report structure will be reused for most of the remaining examples. Reference this section to recreate the base report, if necessary.

1. Open **Report Studio**.
2. Under **Recently used packages**, click **Business Logic View** (or if the **Recently used packages** is empty, click the **Packages** folder and select **Business Logic View**).
3. Click the **Create New** button.
4. In the **New** dialog box, click the **List** icon and then click **OK** to access the **Report Studio** template.
5. In the **Source** pane, navigate to and expand **Business Logic View > Document Star Schema > Document Fact.**
6. Click and drag the following items to the List.
    - **Name**
    - **Field1**
    - **Field2**
    - **Field3**
7. In the **Source** pane, navigate to and expand **Business Logic View > Document Star Schema > Drawer**.
8. Click and drag the following item to the **List**.
    - **Drawer Name**
9. To highlight the column, select the **Drawer Name** from the list.
10. In the toolbar, click the **Filters** button and select **Create Custom Filter**.
11. In the **Keywords** dialog box, type `CP Demo` and click **Search**.
12. In the **Values** section, double-click **CP Demo** and ensure that it moves to the **Selected values** section.
13. Click **OK**. This filters your data to bring back CP Demo Drawers only.

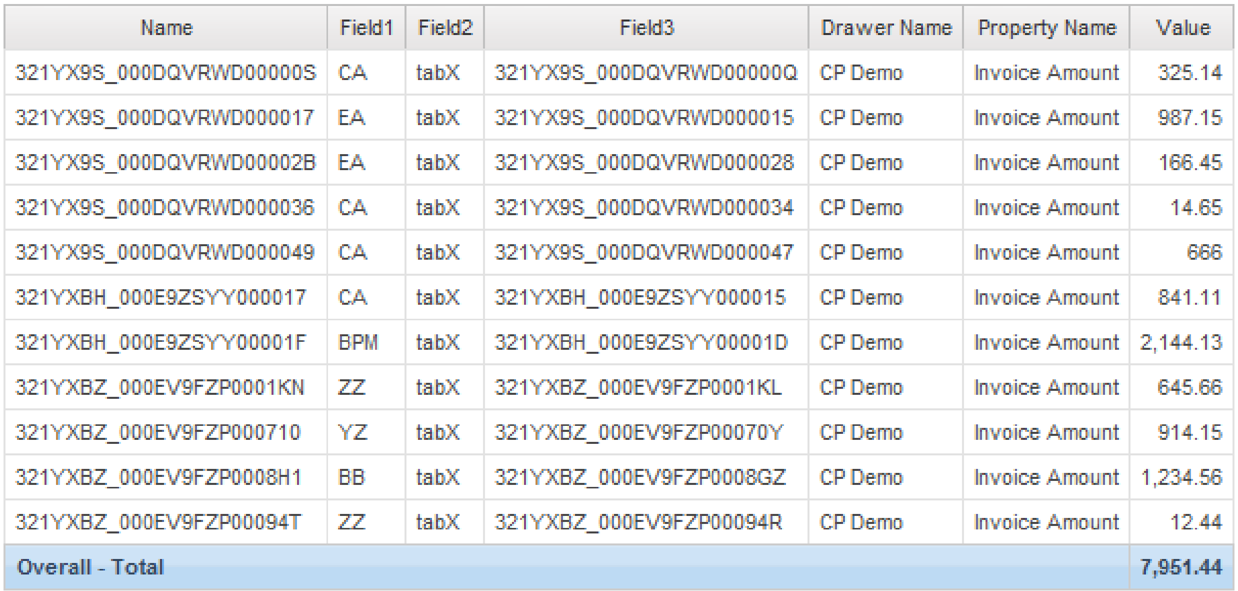
Create a new report with a number custom property using a package

For this exercise, create a report centered on an Invoice document type that consists of two number custom properties: Invoice Amount and Invoice Number. In this report, you want to display the total amount of remaining invoices in the system, so you will use the Invoice Amount custom property for this report and exclude the Invoice Number.

1. Create a base report.
2. In the **Source** pane, navigate to and expand **Business Logic View > Document Star Schema > Number Custom Property**.
3. Click and drag the following items to the **List**
   - **Property Name**
   - **Value**
4. To narrow the search, place a filter on the custom property to define which properties the report should display.
   1. To highlight the column, select the **Property Name** from the list.
   2. In the toolbar, click the **Filters** button, and select **Create Custom Filter**.
   3. In the **Keywords** dialog box, type `Invoice` and click **Search**.
   4. In the **Values** section, double-click **Invoice Amount** and ensure that it moves to the **Selected values** section.
   5. Click **OK**. This filters the data to return only the **Invoice Amount** custom property.
5. Run the report.

To apply a summary for the Invoice Amount value, complete the following steps.

1. In the list, click the **Value** field.
2. In the toolbar, click the **Summarize** button and select **Total**.
3. Run the report.



Name	Field1	Field2	Field3	Drawer Name	Property Name	Value
321YX9S_000DQVRWD00000S	CA	tabX	321YX9S_000DQVRWD00000Q	CP Demo	Invoice Amount	325.14
321YX9S_000DQVRWD000017	EA	tabX	321YX9S_000DQVRWD000015	CP Demo	Invoice Amount	987.15
321YX9S_000DQVRWD00002B	EA	tabX	321YX9S_000DQVRWD000028	CP Demo	Invoice Amount	166.45
321YX9S_000DQVRWD000036	CA	tabX	321YX9S_000DQVRWD000034	CP Demo	Invoice Amount	14.65
321YX9S_000DQVRWD000049	CA	tabX	321YX9S_000DQVRWD000047	CP Demo	Invoice Amount	666
321YXBH_000E9ZSYY000017	CA	tabX	321YXBH_000E9ZSYY000015	CP Demo	Invoice Amount	841.11
321YXBH_000E9ZSYY00001F	BPM	tabX	321YXBH_000E9ZSYY00001D	CP Demo	Invoice Amount	2,144.13
321YXBZ_000EV9FZP0001KN	ZZ	tabX	321YXBZ_000EV9FZP0001KL	CP Demo	Invoice Amount	645.66
321YXBZ_000EV9FZP000710	YZ	tabX	321YXBZ_000EV9FZP00070Y	CP Demo	Invoice Amount	914.15
321YXBZ_000EV9FZP0008H1	BB	tabX	321YXBZ_000EV9FZP0008GZ	CP Demo	Invoice Amount	1,234.56
321YXBZ_000EV9FZP00094T	ZZ	tabX	321YXBZ_000EV9FZP00094R	CP Demo	Invoice Amount	12.44
Overall - Total						**7,951.44**

Create a new report with multiple number custom properties using a package

For another variation on this, consider there are two number custom properties on this document type and there is a need to view both of these in the report. Furthermore, rather than displaying the Value column header, you want to display the custom property name in the header bar.

1. Create a base report.
2. To select the **Toolbox,** in the **Source** pane, click the 🧰 tab.
3. Click and drag a **Query Calculation** into the **List**.
4. In the **Name** section, type `Invoice Amount`.
5. To select **Functions**, in the **Available Components** section, click the fourth tab.
6. To expand the section, click the plus sign next to **Constructs** and double-click the search case. This moves the search case function to the **Expression Definition** window.
7. Modify the **Expression Definition** so it resembles the following example.

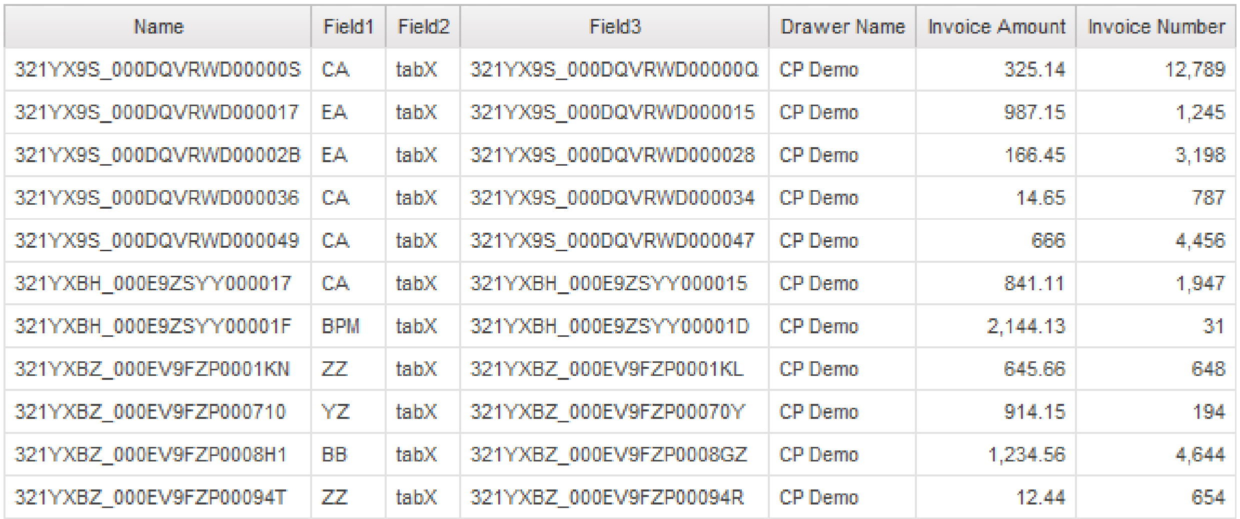
```
CASE
WHEN [Document Star Schema].[Number Custom Property].[Property Name] = 'Invoice
Amount' THEN [Document Star Schema].[Number Custom Property].[Value]
END
```

8. Click **OK**.

9. To create a second **Query Calculation** for **Invoice Number Repeat**, click and drag another **Query Calculation** into the **List** and repeat this process. The calculation should resemble the following example.

```
CASE
WHEN [Document Star Schema].[Number Custom Property].[Property Name] = 'Invoice
Number' THEN [Document Star Schema].[Number Custom Property].[Value]
END
```

10. Run the report.



Name	Field1	Field2	Field3	Drawer Name	Invoice Amount	Invoice Number
321YX9S_000DQVRWD00000S	CA	tabX	321YX9S_000DQVRWD00000Q	CP Demo	325.14	12,789
321YX9S_000DQVRWD000017	EA	tabX	321YX9S_000DQVRWD000015	CP Demo	987.15	1,245
321YX9S_000DQVRWD00002B	EA	tabX	321YX9S_000DQVRWD000028	CP Demo	166.45	3,198
321YX9S_000DQVRWD000036	CA	tabX	321YX9S_000DQVRWD000034	CP Demo	14.65	787
321YX9S_000DQVRWD000049	CA	tabX	321YX9S_000DQVRWD000047	CP Demo	666	4,456
321YXBH_000E9ZSYY000017	CA	tabX	321YXBH_000E9ZSYY000015	CP Demo	841.11	1,947
321YXBH_000E9ZSYY00001F	BPM	tabX	321YXBH_000E9ZSYY00001D	CP Demo	2,144.13	31
321YXBZ_000EV9FZP0001KN	ZZ	tabX	321YXBZ_000EV9FZP0001KL	CP Demo	645.66	648
321YXBZ_000EV9FZP000710	YZ	tabX	321YXBZ_000EV9FZP00070Y	CP Demo	914.15	194
321YXBZ_000EV9FZP0008H1	BB	tabX	321YXBZ_000EV9FZP0008GZ	CP Demo	1,234.56	4,644
321YXBZ_000EV9FZP00094T	ZZ	tabX	321YXBZ_000EV9FZP00094R	CP Demo	12.44	654

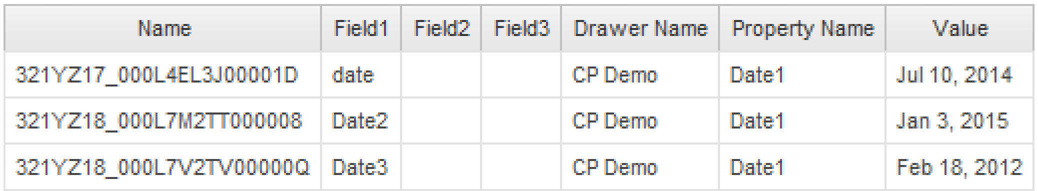
Note When mixing custom property types, such as date custom property and number custom property, it is best to do this with either direct SQL or the report from the view method.

Create a new report with a date custom property using a package

Date custom properties capture the date and not the time value. When pulled into a report, they initially display both the date and the time. You can truncate the time portion from the display with formatting by completing the following steps.

1. Create a base report.
2. In the **Source** pane, navigate to and expand **Business Logic View > Document Star Schema > Date Custom Property**.
3. Click and drag the following items to the List.
    - **Property Name**
    - **Value**
4. To narrow the search, it is best to to place a filter on the custom property to define which properties the report should display. To do this, complete the following substeps.
    1. To highlight the column, select the **Property Name** from the list.
    2. In the toolbar, click the **Filters** button, and select **Create Custom Filter**.
    3. In the **Keywords** dialog box, type `Date` and click **Search**.

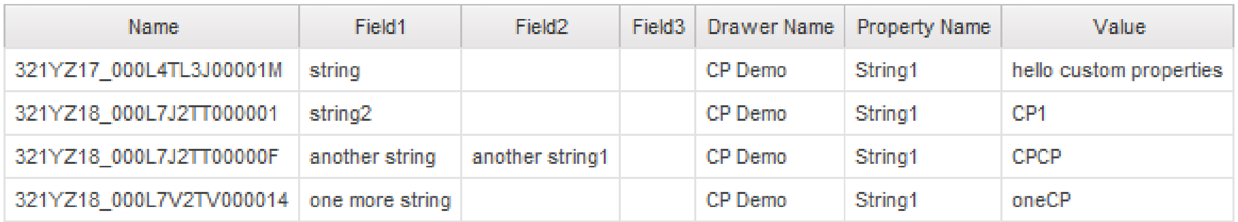
    4. In the **Values** section, double-click **Date1** and ensure that it moves to the **Selected values** section.
    5. Click **OK**. This filters the data to return only the **Date1** custom property.
5. Run the report. The date custom property is followed with a timestamp of 12:00:00 AM.
6. Click the **Value** field in the list.
7. In the **Properties** pane, double-click the **Data Format** dialog box.
8. Click the drop-down list for **Format** type and select **Date**. You can make additional modifications to the date format in this dialog box if required. Click **OK**.
9. Run the report.



Name	Field1	Field2	Field3	Drawer Name	Property Name	Value
321YZ17_000L4EL3J00001D	date			CP Demo	Date1	Jul 10, 2014
321YZ18_000L7M2TT000008	Date2			CP Demo	Date1	Jan 3, 2015
321YZ18_000L7V2TV00000Q	Date3			CP Demo	Date1	Feb 18, 2012

Create a new report with a string custom property using the package

1. Create a base report.
2. In the Source pane, navigate to and expand Business Logic View > Document Star Schema > String Custom Property.
3. Click and drag the following items to the List.
    - Property Name
    - Value
4. To narrow the search, it is best to to place a filter on the custom property to define which properties the report should display. To do this, complete the following substeps.
    1. To highlight a column, select the **Property Name** from the list.
    2. In the toolbar, click the **Filters** button, and select **Create Custom Filter**.
    3. In the **Keywords** dialog box, type `String` and click **Search**.
    4. In the **Values** section, double-click **String1** and ensure that it moves to the **Selected values** section.
    5. Click **OK**. This filters the data to return only the **String1** custom property.
5. Run the report.

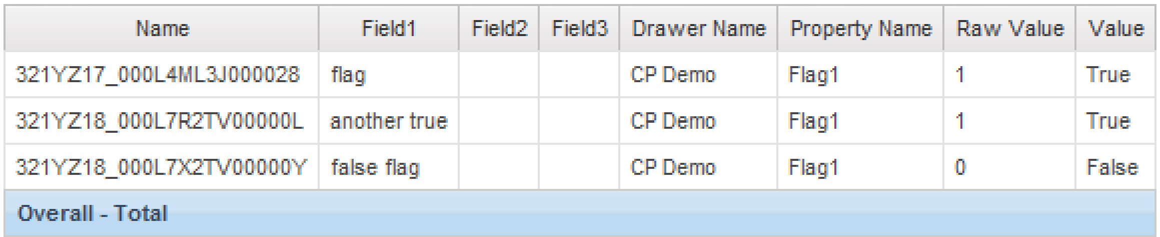


Name	Field1	Field2	Field3	Drawer Name	Property Name	Value
321YZ17_000L4TL3J00001M	string			CP Demo	String1	hello custom properties
321YZ18_000L7J2TT000001	string2			CP Demo	String1	CP1
321YZ18_000L7J2TT00000F	another string	another string1		CP Demo	String1	CPCP
321YZ18_000L7V2TV000014	one more string			CP Demo	String1	oneCP

Create a new report with a flag custom property using a package

The flag custom property is unique in that the database stores both values of the flag in the system as a delimited value and indicates which of the values to present. The published package does the heavy lifting by parsing that information but you can also display the raw data.:

1. Create a base report.
2. In the **Source** pane, navigate to and expand **Business Logic View > Document Star Schema > Flag Custom Property**.
3. Click and drag the following items to the **List**.
   - **Property Name**
   - **Raw Value**
   - **Value**
4. To narrow the search, it is best to to place a filter on the custom property to define which properties the report should display. To do this, complete the following substeps.
   1. To highlight the column, select **Property Name** from the list.
   2. In the toolbar, click the **Filters** button, and select **Create Custom Filter**.
   3. In the **Keywords** dialog box, type `Flag` and click **Search**.
   4. In the **Values** section, double-click **Flag1** and ensure that it moves to the **Selected values** section.
   5. Click **OK**. This filters the data to return only the **Flag1** custom property
5. Run the report.



Name	Field1	Field2	Field3	Drawer Name	Property Name	Raw Value	Value
321YZ17_000L4ML3J000028	flag			CP Demo	Flag1	1	True
321YZ18_000L7R2TV00000L	another true			CP Demo	Flag1	1	True
321YZ18_000L7X2TV00000Y	false flag			CP Demo	Flag1	0	False
Overall - Total							

The value of 1 under the Raw Value column indicates the flag's positive label value. The 0 indicates the flag's negative label raw value. Those are then transformed to the corresponding Value column.

Create a new report with a user custom property using direct SQL

For the user custom property, use a direct SQL approach to pull the data into the report.

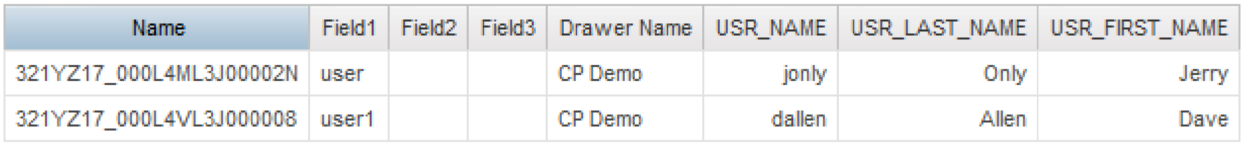
1. Create a base report.
2. Right-click the list and select **Go to Query**.
3. In the **Source** pane, navigate to and expand **Business Logic View > Document Star Schema > Document Fact**.
4. Click and drag the following item to the **Query**.
   - **Instance ID**

5. On the **Menu** bar, click **View > Queries**. This view shows all queries available in the report. At this point, only one query should display.
6. In the **Toolbox** pane, click and drag **SQL** below **Query1**.
7. Click once on the **SQL** box to display the properties pane.
8. In the **Properties** pane, double-click the **Data Source** box.
9. From the list of **Data Sources**, select **INOW** and click **OK**.
10. In the appendix of this document, locate the user custom property query and copy the contents to the clipboard.
11. In **Report Studio**, double-click the box next to **SQL** in the **Properties** pane.
12. Paste the contents of the clipboard in the window, and click **Validate**. If there are any errors, consider the following information.
    - This query was generated in a Perceptive Content 7.0 environment and may not work on earlier versions of the product. You may need to generate a different query for your version.
    - This query was generated on SQL Server and may not work on your database.
    - Not all contents were copied to the clipboard. If possible, try the query in a query engine to validate.
13. After you validate the query, click **OK**.
14. In **Query Explorer**, from the **Toolbox** pane, click and drag a **Join** query below **Query2**.
15. To the top right of the **Query3** join, click and drag **Query1**.
16. To the bottom right of the **Query3** join, click and drag **Query2**.
17. Double-click the center join icon represented by the two yellow boxes.
18. Click the **New Link** button and ensure that the highlighted items on both sides are **Instance ID** and **INSTANCE_ID**.
19. Change the cardinality to reflect what you expect to see. For this exercise, the proper cardinality should be 1..1 on **Query1** and 1..n on **Query2**.
20. Double-click **Query3**.
21. From the **Source** pane, click and drag items from **Query1** and **Query2** to the **Data Items** section. These are the usable items for the report. Only grab the items you need for higher efficiency. For this exercise, select the following information.
    - **Name**
    - **Field1**
    - **Field2**
    - **Field3**
    - **Drawer Name**
    - **USR_NAME**
    - **USR_LAST_NAME**
    - **USR_FIRST_NAME**

22. Navigate back to the report page (**Page Explorer > Report Pages > Page1**).
23. Select the list object from the ancestry tree. An easy way to do this is to click on the list, then in the **Properties** pane, click the UP arrow next to **Properties** and select **List**.
24. In the **Properties** pane, change the **Query** from **Query1** to **Query3**.

    Note if anything in the list displays a badge icon next to the name, it indicates that the link is broken. This is likely due to the object not having been included in the query. To resolve this, either include it in the new query or remove it from the list.

25. In the **Source** pane, click the second tab for **Data Items**. This displays items that are already defined at the Query level and are available for use in the report.
26. Scroll down to **Query3** and click and drag the following items to the **List**.
    - **USR_NAME**
    - **USR_LAST_NAME**
    - **USR_FIRST_NAME**
27. Run the report.



Name	Field1	Field2	Field3	Drawer Name	USR_NAME	USR_LAST_NAME	USR_FIRST_NAME
321YZ17_000L4ML3J00002N	user			CP Demo	jonly	Only	Jerry
321YZ17_000L4VL3J000008	user1			CP Demo	dallen	Allen	Dave

Create a new report with a list custom property using direct SQL

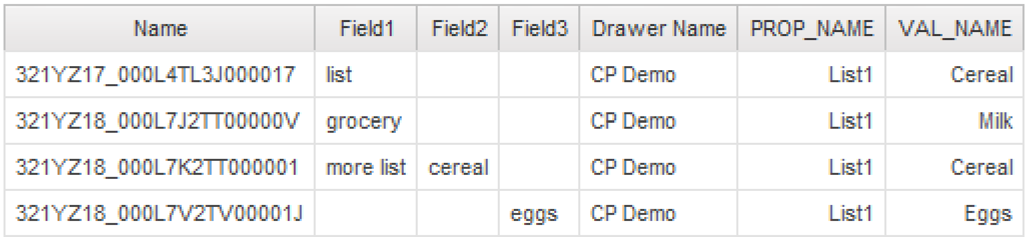
In this scenario for the list custom property, you will use a direct SQL approach to pull the data into the report.

1. Create a base report.
2. Right-click the list and select **Go to Query**.
3. In the **Source** pane, navigate to and expand **Business Logic View > Document Star Schema > Document Fact**.
4. Click and drag the following item to the **Query**.
    - **Instance ID**
5. On the **Menu** bar, click **View > Queries**. This view shows all queries available in the report. At this point, you should only have one query.
6. In the **Toolbox** pane, click and drag **SQL** below **Query1**.
7. Click once on the box titled **SQL** so that the properties pane becomes available.
8. In the **Properties** pane, double-click the **Data Source** box.
9. From the list of **Data Sources**, select **INOW** and click **OK**.
10. In the appendix of this document, locate the custom property query list and copy the contents to the clipboard.
11. In **Report Studio**, double-click the box next to **SQL** in the **Properties** pane.

12. Paste the contents of the clipboard in the window, and click **Validate**. If there are any errors, consider the following information.
    - This query was generated in a Perceptive Content 7.0 environment and may not work on earlier versions of the product. You may need to generate a different query for your version.
    - This query was generated on SQL Server and may not work on your database
    - Not all contents were copied to the clipboard. If possible, try the query in a query engine to validate.
13. After you validate the query, click **OK**.
14. In **Query Explorer**, from the **Toolbox** pane, click and drag a **Join** query below **Query2**.
15. To the top right of the **Query3** join, click and drag **Query1**.
16. To the bottom right of the **Query3** join, click and drag **Query2**.
17. Double-click the center join icon represented by the two yellow boxes.
18. Click the **New Link** button and ensure that the highlighted items on both sides are **Instance ID** and **INSTANCE_ID**.
19. Change the cardinality to reflect what you expect to see. For this exercise, the proper cardinality should be 1..1 on **Query1** and 1..n on **Query2**.
20. Double-click **Query3**.
21. From the source pane, click and drag items from **Query1** and **Query2** to the **Data Items** section. These are the usable items for the report. Only grab the items you need for higher efficiency. For this exercise, select the following properties.
    - **Name**
    - **Field1**
    - **Field2**
    - **Field3**
    - **Drawer Name**
    - **PROP_NAME**
    - **VAL_NAME**
22. Navigate back to the report page (**Page Explorer > Report Pages > Page1**).
23. Select the list object from the ancestry tree. An easy way to do this is to click the list, then in the properties pane, click the UP arrow next to **Properties** and select **List**.
24. In the **Properties** pane, change the Query from **Query1** to **Query3**.

    Note If anything in the list displays a badge icon next to the name, it indicates that the link is broken. This is likely due to the object not having been included in the query. To resolve this, either include it in the new query or remove it from the list.
25. In the **Source** pane, click the second tab for **Data Items** to display items that are already defined at the **Query** level and are available for use in the report.

26. Scroll down to **Query3** and click and drag the following items to the **List**.
    - **PROP_NAME**
    - **VAL_NAME**
27. Run the report.



Name	Field1	Field2	Field3	Drawer Name	PROP_NAME	VAL_NAME
321YZ17_000L4TL3J000017	list			CP Demo	List1	Cereal
321YZ18_000L7J2TT00000V	grocery			CP Demo	List1	Milk
321YZ18_000L7K2TT000001	more list	cereal		CP Demo	List1	Cereal
321YZ18_000L7V2TV00001J			eggs	CP Demo	List1	Eggs

Add composite property to an existing report using direct SQL

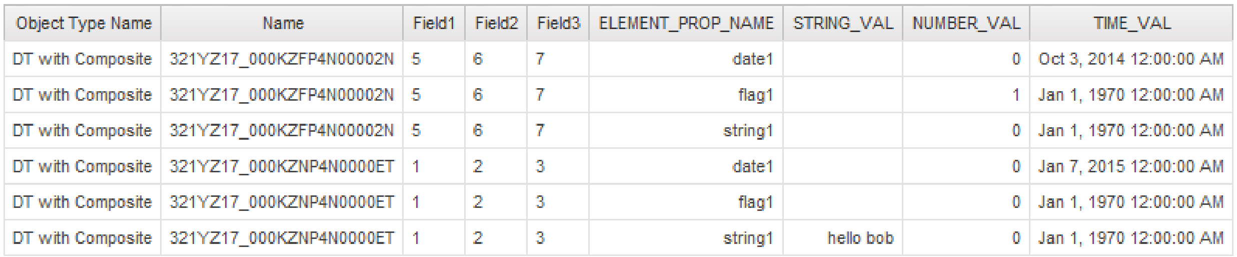
For simplicity sake, we will use the first report example to include this new composite property data. Open the report if it is saved (or recreate from the steps above) then continue with the following steps:

1. Right-click the list and select **Go to Query**.
2. In the **Source** pane, navigate to and expand **Business Logic View > Document Star Schema > Document Fact**.
3. Click and drag the following item to the Query.
    - Instance **ID**
4. On the **Menu** bar, click **View > Queries**. This view shows all queries available in the report. At this point, only one query should display.
5. In the **Toolbox** pane, click and drag **SQL** below **Query1**.
6. Click once on the **SQL** box to display the **Properties** pane.
7. In the **Properties** pane, double-click the **Data Source** box.
8. From the list of **Data Sources**, select **INOW** and click **OK**.
9. In the appendix of this document, locate the composite property query and copy the contents to the clipboard.
10. In **Report Studio**, double-click the box next to **SQL** in the **Properties** pane.
11. Paste the contents of the clipboard in the window, and click **Validate**. If there are any errors, consider the following information.
    - This query was generated in a Perceptive Content 7.0 environment and may not work on earlier versions of the product. You may need to generate a different query for your version.
    - This query was generated on SQL Server and may not work on your database.
    - Not all contents were copied to the clipboard. If possible, try the query in a query engine to validate.

12. After you validate the query, click **OK**.
13. In **Query Explorer**, from the **Toolbox** pane, click and drag a **Join** query below **Query2**.
14. To the top right of the **Query3** join, click and drag **Query1**.
15. To the bottom right of the **Query3** join, click and drag **Query2**.
16. Double-click the center join icon represented by the two yellow boxes.
17. Click the **New Link** button and ensure that the highlighted items on both sides are **Instance ID** and **INSTANCE_ID**.
18. Change the cardinality to reflect what you expect to view. For this exercise, the proper cardinality should be 1..1 on **Query1** and 1..n on **Query2**.
19. Double-click **Query3**.
20. From the **Source** pane, click and drag items from **Query1** and **Query2** to the **Data Items** section. These are the usable items for the report. Only grab the items you need for higher efficiency. For this exercise, select the following properties.
    - **Name**
    - **Field1**
    - **Field2**
    - **Field3**
    - **Drawer Name**
    - **ELEMENT_PROP_NAME**
    - **STRING_VAL**
    - **NUMBER_VAL**
    - **TIME_VAL**
21. Navigate back to the Report page (Page Explorer > Report Pages > Page1).
22. Select the list object from the ancestry tree. An easy way to do this is to click on the list, then in the Properties pane, click the UP arrow next to Properties and select List.
23. In the Properties pane, change the Query from Query1 to Query3.

    Note If anything in the list displays a badge icon next to the name, it indicates that the link is broken. This is likely due to the object not having been included in the query. To resolve this, either include it in the new query or remove it from the list.
24. In the **Source** pane, click the second tab for **Data Items** to display items that are already defined at the **Query** level and are available for use in the report.

25. Scroll down to **Query3** and click and drag the following items to the **List**.
    - ELEMENT_PROP_NAME
    - STRING_VAL
    - NUMBER_VAL
    - TIME_VAL
26. Run the report.



Object Type Name	Name	Field1	Field2	Field3	ELEMENT_PROP_NAME	STRING_VAL	NUMBER_VAL	TIME_VAL
DT with Composite	321YZ17_000KZFP4N00002N	5	6	7	date1		0	Oct 3, 2014 12:00:00 AM
DT with Composite	321YZ17_000KZFP4N00002N	5	6	7	flag1		1	Jan 1, 1970 12:00:00 AM
DT with Composite	321YZ17_000KZFP4N00002N	5	6	7	string1		0	Jan 1, 1970 12:00:00 AM
DT with Composite	321YZ17_000KZNP4N0000ET	1	2	3	date1		0	Jan 7, 2015 12:00:00 AM
DT with Composite	321YZ17_000KZNP4N0000ET	1	2	3	flag1		0	Jan 1, 1970 12:00:00 AM
DT with Composite	321YZ17_000KZNP4N0000ET	1	2	3	string1	hello bob	0	Jan 1, 1970 12:00:00 AM

You can use additional formatting and grouping features to make this report more visually appealing. For example, you can use new column headings to replace existing headings. You can also use case statements to display custom properties conditionally in a column similar to the “Create a new report with multiple number custom properties using a package” example.

Appendix A

This appendix contains examples of queries you can run.

User custom property query

```
SELECT II.INSTANCE_ID, U.USR_ID, USR_NAME, USR_LAST_NAME, USR_FIRST_NAME, USR_PREFIX
FROM IN_INSTANCE II
INNER JOIN IN_INSTANCE_PROP IP
       ON II.INSTANCE_ID = IP.INSTANCE_ID
INNER JOIN IN_SC_USR U
       ON IP.STRING_VAL = U.USR_ID

--List custom property query
SELECT II.INSTANCE_ID, P.PROP_NAME, PV.VAL_NAME
FROM IN_INSTANCE II
INNER JOIN IN_INSTANCE_PROP IP
       ON II.INSTANCE_ID = IP.INSTANCE_ID
INNER JOIN IN_PROP_VAL PV
       ON IP.STRING_VAL = PV.PROP_VAL_ID
INNER JOIN IN_PROP P
       ON IP.PROP_ID = P.PROP_ID
```

Composite Property query

```
SELECT i.INSTANCE_ID, p.PROP_NAME, ecp.SEQ_NUM AS ELEMENT_SEQ_NUM, ep.PROP_NAME AS
ELEMENT_PROP_NAME, ep.PROP_TYPE AS PROP_TYPE, ep.HAS_DEFAULT AS HAS_DEFAULT,
ep.DEF_STRING_VAL AS DEF_STRING_VAL, ep.DEF_NUMBER_VAL AS DEF_NUMBER_VAL,
ep.DEF_TIME_VAL AS DEF_TIME_VAL, ep.ELEMENT_TYPE_ID AS ELEMENT_TYPE_ID, ep.IS_ARRAY AS
IS_ARRAY, eip.STRING_VAL AS STRING_VAL, eip.NUMBER_VAL AS NUMBER_VAL, eip.TIME_VAL AS
TIME_VAL, ev.VAL_NAME AS VAL_NAME, ev.SEQ_NUM AS VAL_SEQ_NUM, eu.USR_NAME AS USR_NAME,
eu.USR_LAST_NAME AS USR_LAST_NAME, eu.USR_FIRST_NAME AS USR_FIRST_NAME, eu.USR_PREFIX
AS USR_PREFIX, eu.USR_SUFFIX AS USR_SUFFIX
FROM IN_INSTANCE i
INNER JOIN IN_CLASS_PROP cp
       ON i.CLASS_ID = cp.CLASS_ID
INNER JOIN IN_PROP p
       ON p.PROP_ID = cp.PROP_ID AND p.CATEGORY <> 2
LEFT JOIN IN_ELEMENT_TYPE et
       ON p.ELEMENT_TYPE_ID = et.ELEMENT_TYPE_ID
LEFT JOIN IN_CLASS_PROP ecp
       ON et.CLASS_ID = ecp.CLASS_ID
LEFT JOIN IN_PROP ep
       ON (ep.PROP_ID = ecp.PROP_ID OR (p.PROP_TYPE = 10 AND ep.PROP_ID = p.PROP_ID))
LEFT JOIN IN_INSTANCE_PROP ip
       ON ip.INSTANCE_ID = i.INSTANCE_ID AND ip.PROP_ID = p.PROP_ID
LEFT JOIN IN_PROP_VAL v
       ON v.prop_val_id = ip.STRING_VAL AND p.PROP_TYPE IN (5, 10)
LEFT JOIN IN_SC_USR u
       ON ip.STRING_VAL = u.USR_ID AND p.PROP_TYPE IN (6, 7)
LEFT JOIN (
       SELECT ELEMENT_ID, ELEMENT_TYPE_ID, ichild.PARENT_INSTANCE_ID,
ichild.INSTANCE_ID AS CHILD_INSTANCE_ID, seq.SEQ_NUM
       FROM IN_INSTANCE ichild
       INNER JOIN IN_ELEMENT e
              ON ichild.INSTANCE_ID = e.INSTANCE_ID
       INNER JOIN IN_INSTANCE_SEQ seq
              ON seq.INSTANCE_ID = ichild.INSTANCE_ID
       ) ir
       ON ir.ELEMENT_TYPE_ID = et.ELEMENT_TYPE_ID AND ir.PARENT_INSTANCE_ID = 
i.INSTANCE_ID
LEFT JOIN IN_INSTANCE_PROP eip
       ON eip.INSTANCE_ID = ir.CHILD_INSTANCE_ID AND eip.PROP_ID = ep.PROP_ID
LEFT JOIN IN_PROP_VAL ev
       ON ev.prop_val_id = eip.STRING_VAL AND ep.PROP_TYPE IN (5, 10)
LEFT JOIN IN_SC_USR eu
       ON eip.STRING_VAL = eu.USR_ID AND ep.PROP_TYPE IN (6, 7)
ORDER BY cp.CLASS_ID, cp.SEQ_NUM, ir.SEQ_NUM, ecp.CLASS_ID, ecp.SEQ_NUM
```