ORACLE

Oracle Fusion Data Intelligence Workbook Performance Optimization Guide for Report Authors

A practical guide for report authors building Data Visualization workbooks in Fusion Data Intelligence bringing together best practices and recommendations to help design efficient, responsive workbooks right from the start.

May 2026, version 1.0

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Revision History

The following revisions have been made to this document since its initial publication.

DATE	REVISION
May 2026	Initial publication

Table of Contents

Purpose

This guide is intended for report authors - developers and business users working in a self-service model - who create DV workbooks in Fusion Data Intelligence (FDI). It provides performance optimization best practices and recommendations to help you build efficient, responsive workbooks. The content consolidates institutional knowledge, expert tips, and lessons learned from working with a wide range of customers. The goal of the guide is to ensure performance is built in from the start and common pitfalls are avoided.

Prerequisites

It is assumed that the following critical prerequisites to workbook development have been completed and best practices have been noted:

- Reporting requirements have been reviewed, and, if needed, the semantic model has been extended by your data modeling team following best practices.
- The subject area that the workbook will be based on has been identified.
- The prebuilt metrics provided in FDI have been reviewed. *As best practice, use or modify the prebuilt metrics. Create new metrics only if business requirements cannot be met with any of the prebuilt metrics or modifications of the prebuilt metrics.* See Chapter 5 Data Lineage for links to metric calculation spreadsheets, Chapter 8 Subject Area > Subject Area Metric Details and Chapter 9 Prebuilt > Prebuilt Business Metrics in the following Pillar Reference Guides:

  ERP Analytics | SCM Analytics | HCM Analytics | CX Analytics | *Note: Always use the latest available documentation*

- Appendix A: Report Authoring Tips in the above noted Reference Guides has been reviewed. *The filters that are required and recommended for the subject area your workbook is based on have been noted*. If using multiple subject areas, the section titled 'Cross-Subject Area Report Authoring Tips' has been reviewed.
- The use of Workbook Templates that are provided in FDI has been considered as starting point for building new workbooks. These templates have the necessary and critical filters already applied. Additional filters can be included based on requirements.

As you develop your workbook, consider the guidelines and best practices described in the sections below to optimize performance. Apply these practices during development by integrating them into your workbook design.

Workbook Performance Optimization

Number of Canvases and Visualizations

The visualizations and canvases in your workbook are the tools you use for data storytelling. As you decide how to present information - selecting the right visualizations and organizing them into canvases - keep performance considerations in mind. To achieve industry-standard performance for workbooks, it is recommended that a workbook contain:

- no more than 5 or 6 canvases per workbook and up to 8 visualizations per canvas

These recommended limits ensure that workbooks open quickly and don't send too many queries to the database per user.

If the performance of a canvas in a workbook is an issue, as a first step, cut down on the number of visualizations on that canvas, and determine if there is a performance improvement.

Use of Background Images

Images are frequently added to workbooks as backgrounds to create a corporate look and feel or to enhance visual impact and style. To ensure faster rendering and quicker workbook loading times, use compressed images with file sizes in the kilobyte range. Large image files can cause performance overhead and slow down the rendering of workbooks.

Follow these guidelines when incorporating images to your workbook:

- Use compressed images with sizes in the kilobyte range
- Use images that provide information and context and enhance storytelling. Avoid the use of unnecessary images.
- Determine the image size needed and reduce size *before* adding to a visualization. Avoid adding a large image to a visualization and then scaling it down in the visualization.

Use of Brushing

Brushing is an Oracle Analytics feature that dynamically highlights related data points across multiple visualizations in a canvas when a user selects data points in one visualization. Brushing actions can trigger queries across multiple visualizations and may impact the performance of workbooks.

The default system setting for Brushing in Oracle Fusion Data Intelligence is 'Off'. However, report builders can override the setting for an individual workbook and turn Brushing on from Workbook > Properties**.** It is recommended that this setting be turned "On" only after careful consideration. For large datasets or workbooks with many linked visualizations, brushing actions can be resource intensive and result in slow responsiveness.

NOTE: If Brushing has been enabled at the system level in your organization, then evaluate whether the Brushing feature is impacting workbook performance. For workbooks that are impacted, turn brushing off from Workbook > Properties.

Filters

Filters are used to narrow down data to focus on information relevant for analysis. They are also a valuable tool to optimize performance of analyses, canvases and workbooks by limiting the volume of data that is processed and rendered. Filters can be applied at various levels (workbook, canvas, visualization) and can be static or dynamic.

Consider these guidelines when creating filters in your workbook:

Required and Recommended Filters

As a first step, include the filters that are required and recommended for the subject area your report is based on. For the list of required and recommended filters for a given subject area, see the Report Authoring Tips section in the following Analytics Reference Guides: ERP Analytics | SCM Analytics | HCM Analytics | CX Analytics. *Note: Always use the latest available documentation*

An asterisk (*) against a filter denotes that it is required. Note that for the majority of the subject areas a Time filter is required. Other filters listed as recommended should be included based on your data shape. Test the filter recommendations for optimal performance against your data.

The Workbook Templates provided in FDI are a good way to begin your workbook development as they include the necessary and critical filters for the subject areas that they are based on. Additional filters may be added to meet specific business or reporting requirements.

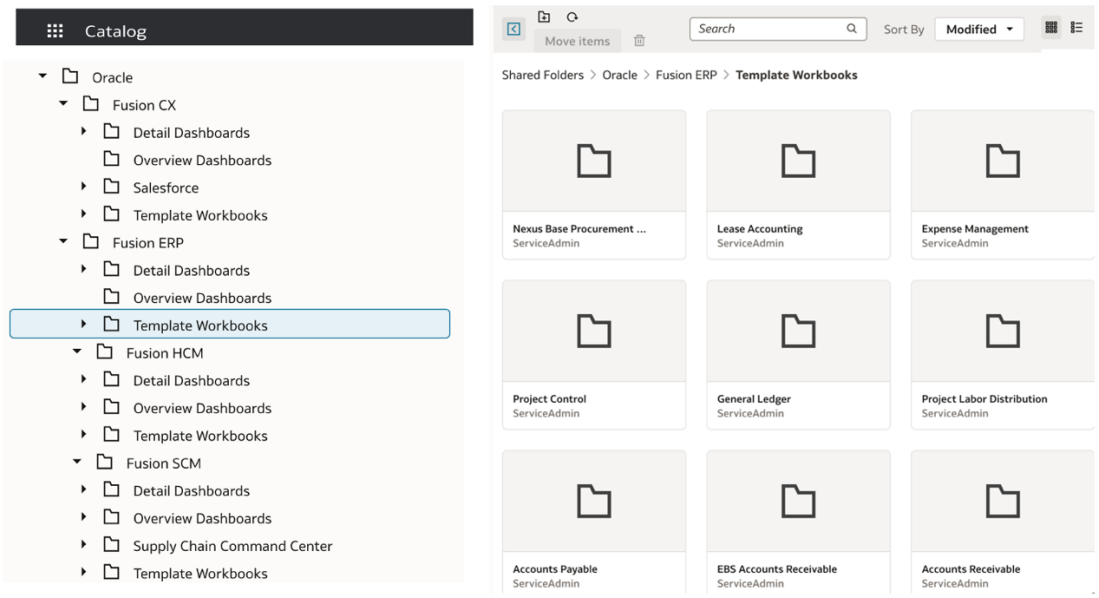

Image 1 Template Workbooks for each Fusion Analytics pillar are available under the Oracle shared folder in Catalog

Use Workbook Filters to Limit Data Volume

It is essential to limit the data that is processed in your workbook to only what is actionable and relevant for user interaction. Restricting the number of records by excluding unnecessary data results in performance gains especially in production environments where data volumes are typically large.

Use Workbook filters to define the scope of data for the entire workbook:

1. Add the required filters and other desired filters to the Filter bar on a canvas and select appropriate filter values.
2. Pin the filters to create Workbook filters which are then applied to all canvases in the workbook.
3. Hide the filter controls to prevent user interaction with these filters by navigating to the Present tab and unchecking the Workbook filters in the Workbook tab of the property panel.

The Workbook filters created in the manner described above restrict the volume of data that the workbook will process without exposing filter controls to users. Users interact with only the filtered data.

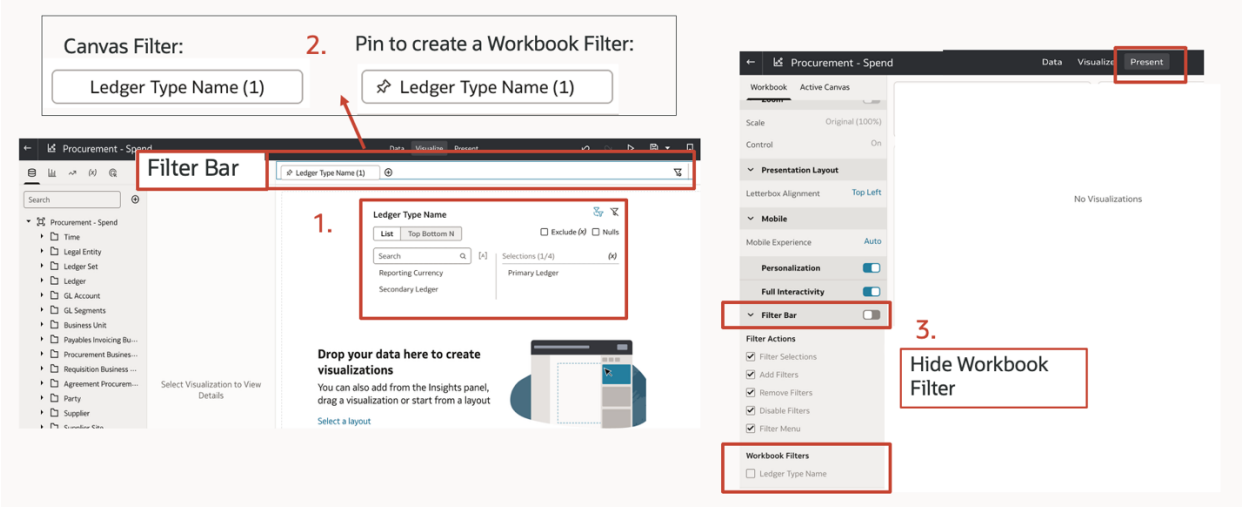

Image 2 Creating a Hidden Workbook Filter to limit Workbook data

Canvas filters apply to an individual canvas and may also be hidden. When user interaction with filters is needed, use Dashboard filters to allow end-user selection of filter values.

For more information on Workbook and Canvas filters, see Filter Scopes. *Note: Always use the latest available documentation*

Dashboard Filters

Filter controls within a Dashboard filter allow users to make filter selections. Dashboard filters have a rich set of configurable properties and can be applied to visualizations on the entire canvas (default) or selectively to some of the visualizations in the canvas (Dashboard Filter Properties > Filters > Apply This Filter To).

Dashboard filters can be cascaded to multiple canvases in a workbook with the use of the dashboard filter visual and parameters that capture user selections. Users do not have to make filter value selections again when navigating from the canvas where filter selections have been made to another canvas in the workbook. The canvas is loaded with the user selections already applied avoiding the situation where default filters are first applied and then user selections are applied.

Set Default Values for Filter Controls

As best practice when creating filters, set default values for filters that users will interact with. This is essential for attributes that have a large number of values.

Default values for Dashboard filter controls can be set from the Dashboard Filter Properties tab > Filter Control tab. Select the filter control and set the Default Value property to either “First in list”, “Parameter”, or “Custom”. Set defaults to values that will be commonly used or that are the most relevant for analysis.

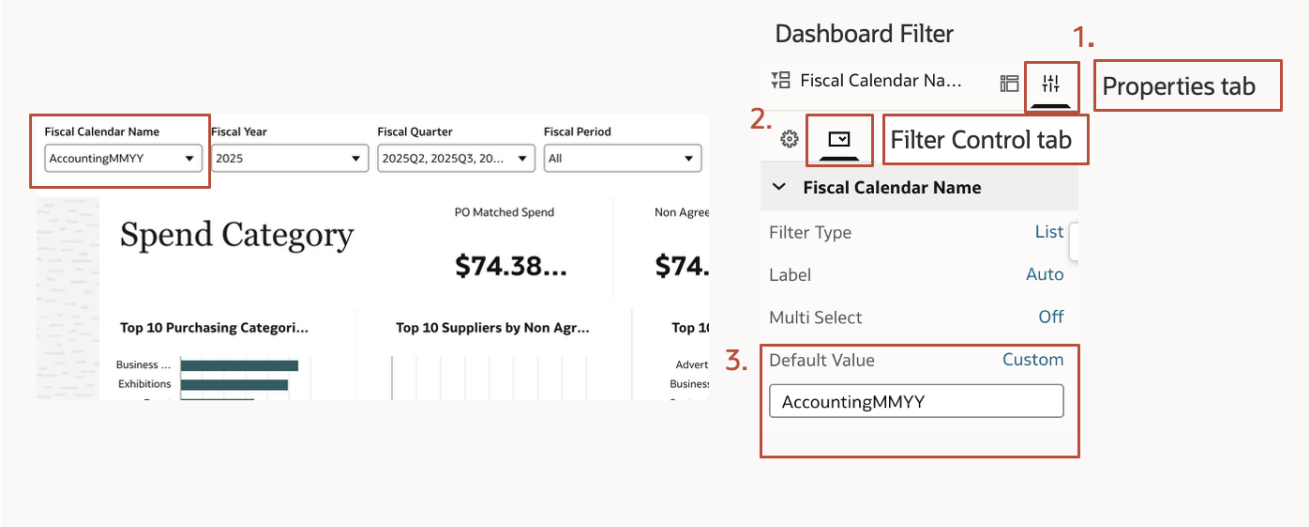

Image 3 Set Default Values for a Dashboard Filter

See Oracle Documentation
Dashboard Filters
Note: Always use the latest available documentation

Determine How Filter Selections Get Applied

For filters that users will interact with, determine whether filter selections should be applied automatically or require manual user action.

The automatic application of filter values can cause multiple queries to run in quick succession as users adjust filter values. Whereas, when users select all desired filters first and then explicitly apply the filter values by clicking the

Apply button, only one query execution per set of filter selections is generated. When data volumes are large, consider requiring users to manually apply filters using the Apply button.

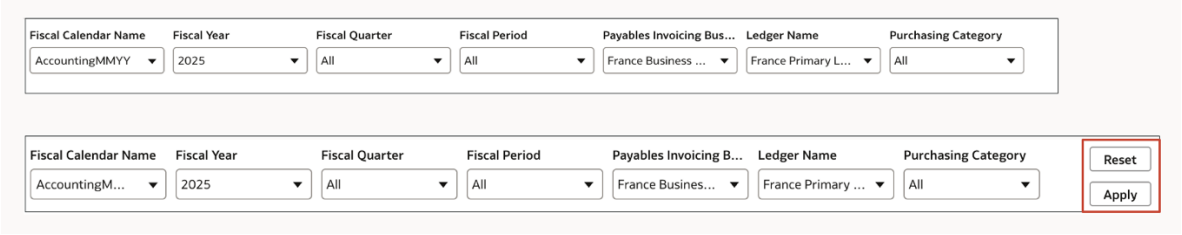

Image 4 Dashboard Filters with and with the Apply and Reset buttons

The Auto-Apply property on filters lets you control when filter values get applied. When the Auto-Apply Filters property in the Filter Bar Menu is set to “On”, Workbook, Canvas and Visualization filters are automatically applied every time users change filter values. For Dashboard filters, the automatic application of filter values is set from Dashboard filter Properties > General tab. For the Buttons property, select “None”.

When Auto-Apply is set to “Off” in the Filter bar properties panel, an Apply button appears on the filter. For Dashboard filters, this is set from Dashboard filter Properties > General tab > Buttons property. Select “Apply” and, optionally, “Reset”. Filter values are applied only when a user manually clicks the Apply button.

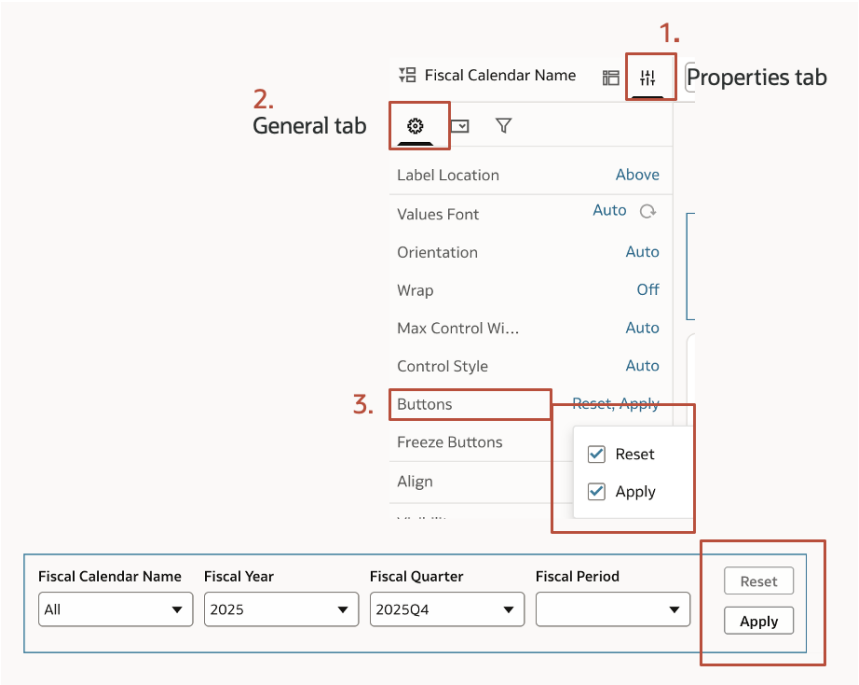

Image 5 Configure a Dashboard Filter to display Apply and Reset buttons for manual user action to apply filter values

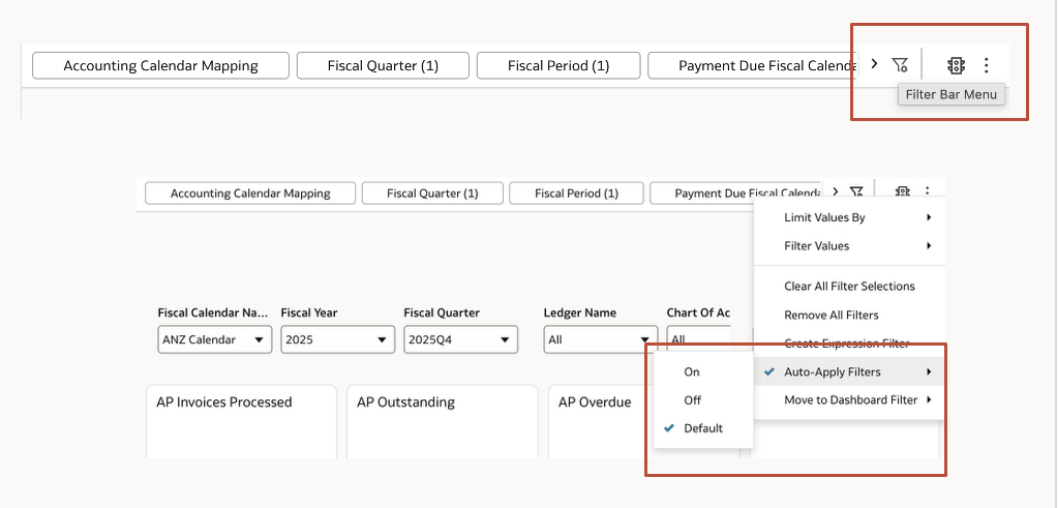

Image 6 Configure Workbook or Canvas Filters to display Apply button for manual user action to apply filter values

Optimize Use of the Limit Values By Property (Independent and Cascading Filters)

Filters can be defined as either independent filters or cascading filters. Cascading filters are filters with hierarchical relationships where a selection in one filter automatically restricts the available options in subsequent filters. For example, selecting a value in a Country filter automatically displays only relevant cities in a City filter. An independent filter, on the other hand, is not constrained by a value selected in another filter.

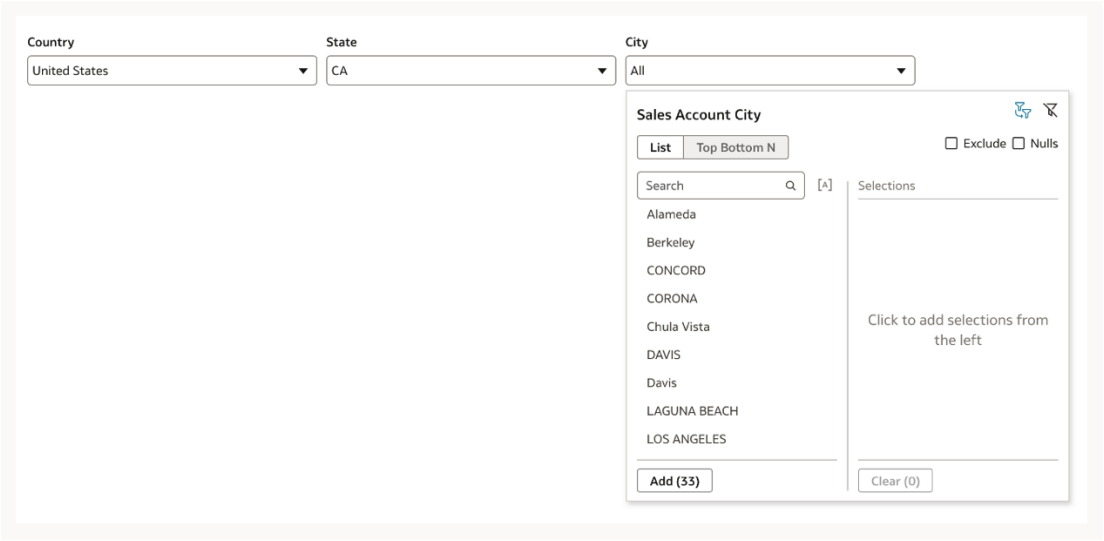

Image 7 Cascading filters 'Country', 'State' and 'City'. The city name values appearing in the 'City' filter are constrained by selections in 'Country' and 'State' filters

The Limit Values By property setting controls how selections in one filter affect the available options in other filters. The Limit Values By setting of "Auto" lets the system automatically determine whether there are any hierarchical relationships between the filter control and the other filter controls in that location (filter bar or Dashboard filter). Using "Auto" for a filter control results in potentially expensive queries because when a value is selected for the filter the system must determine all relationships with the other filter controls in that location, even if some or all filter controls are independent or unrelated.

Consider the following approaches to optimize performance when using the Limit Values By property:

- Defining Hierarchical Relationships for Filter Controls in a Dashboard Filter: Dashboard filters often need to have multiple filter controls to allow users to make desired filter selections across multiple attributes. To define hierarchical relationships for filter controls within the Dashboard filter, it is recommended to explicitly set the filter control that will constrain a given filter. This can be configured by going to Dashboard Filter Properties > Filter Controls and selecting the filter control. For this filter control, set Limit Values By to the filter control that will constrain the filter by choosing the appropriate filter control from the list. See Image 8 below. While the "Auto" setting will also create cascading filters, it is not recommended for the reasons stated above.
- Limit Values By setting for filter controls in the Filter bar: The Limit Values By property setting applies to all filter controls in the Filter bar. It does not provide the option for constraining a filter control by explicitly setting the filter control that constrains it. If cascading filters are a requirement, consider creating Dashboard filters instead.
- Defining Independent filters: Explicitly set the Limit Values By property to “None” to define a filter control as independent.

NOTE: The Limit Values By property has an additional setting of “Default”, which is driven by a system-wide setting configured by your administrator. When the system setting is “None”, the Limit Values By setting of “Default” for filters in the Filter bar or in Dashboard filters is the same as the setting “None”. When the system setting is “Auto”, then the Default setting for filter controls acts as “Auto”, which is to be avoided. If in doubt, set Limit Values By for independent filter controls to “None” rather than “Default”.

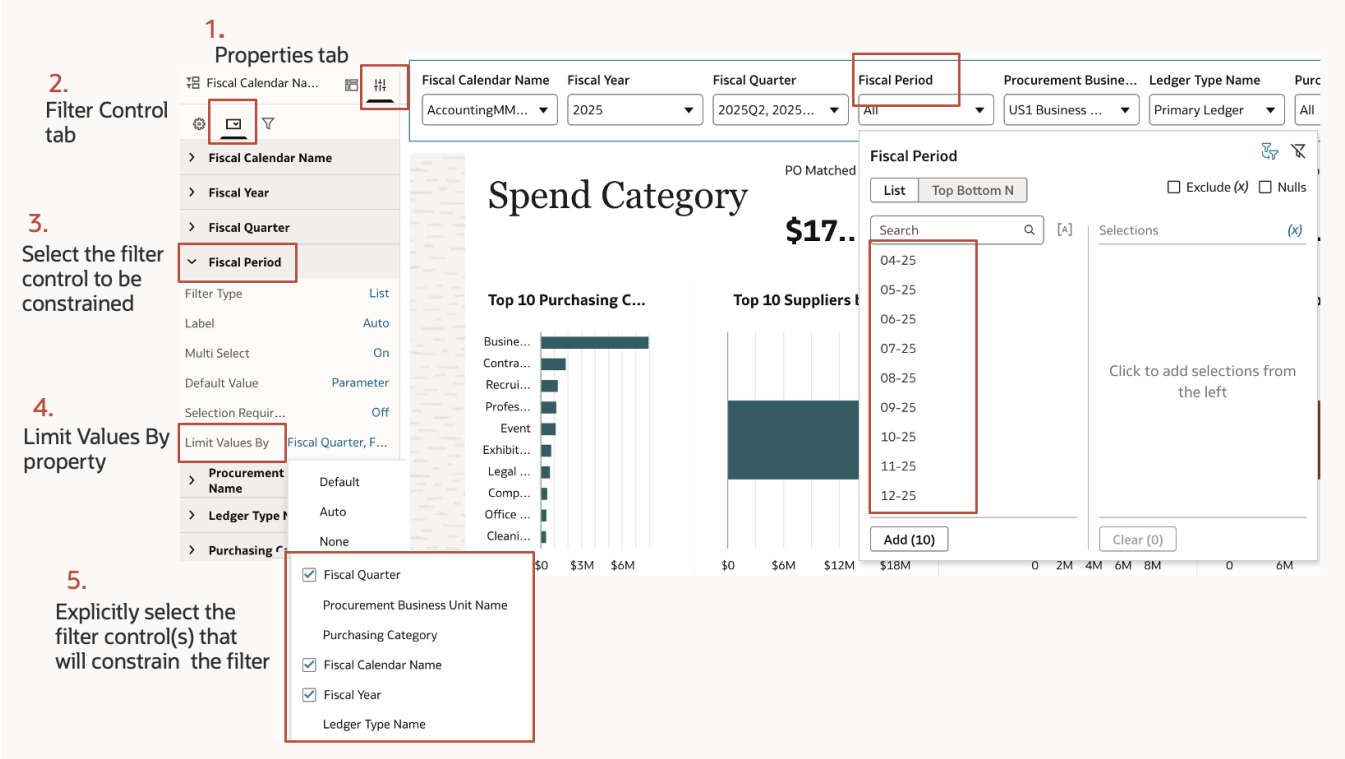

Image 8 Constrain the values of filter control ‘Fiscal Period’ by explicitly selecting the constraining filter controls - ‘Fiscal Calendar Name’, ‘Fiscal Year’ and ‘Fiscal Quarter’ for the Limit Values By property

See Oracle Documentation:

How Do Filters Limit Each Other?

Note: Always use the latest available documentation

Calculated Columns, Expression Filters and Functions

Fusion Data Intelligence provides a large library of metrics. Modify prebuilt metrics to meet business requirements and create custom metrics only when necessary. For complex calculations and custom metrics that will be reused across multiple workbooks, it is recommended that you work with your Data Modeling and Data Steward teams to push these into the semantic layer or into the data model to be stored in tables.

Follow the guidelines below when creating calculations and expression filters in workbooks:

Use Measures for Aggregation

Measures (Facts) are numeric data. Use measures to aggregate/perform mathematical operations like count, sum, average, etc. Attributes (Dimensions) are descriptive and categorize data, and should be used for grouping, filtering and slicing data.

Don't count attributes, for example, attribute Customer ID

`COUNT(Customer ID)` will return a number value, but the value may be incorrect.

`COUNT(DISTINCT Customer ID)` may or may not return a correct number; however, this may result in performance issues particularly with large datasets.

If performance issues are encountered with the use of Count Distinct then work with your Data Modeling team to create a measure in the semantic layer of Fusion Data Intelligence. This will allow you to use the created measure in your calculations.

NOTE: If Count Distinct returns an incorrect value, then your team may need to work with the Data Stewards in your organization to create a Fact table in the database, which will then allow for an aggregation such as sum to obtain the correct counts.

Use Code Columns in Expression Filters and Calculations

To optimize performance of Expression Filters and Calculations, use Code columns in formulas instead of the corresponding Name/Description columns. The use of a Code column results in a consistently performant query whereas the use of the Name or Description column leads to a join to a translation table and potentially a less efficient query being generated.

To identify the values of the Code column that correspond to the value of the Name column being filtered, add a temporary Table visualization to the canvas with Code and the Name attribute columns.

For example, the Workforce Absence Management subject area has two columns: Assignment Type and the corresponding Assignment Type Code. Use the Assignment Type Code column in calculations and expression filters.

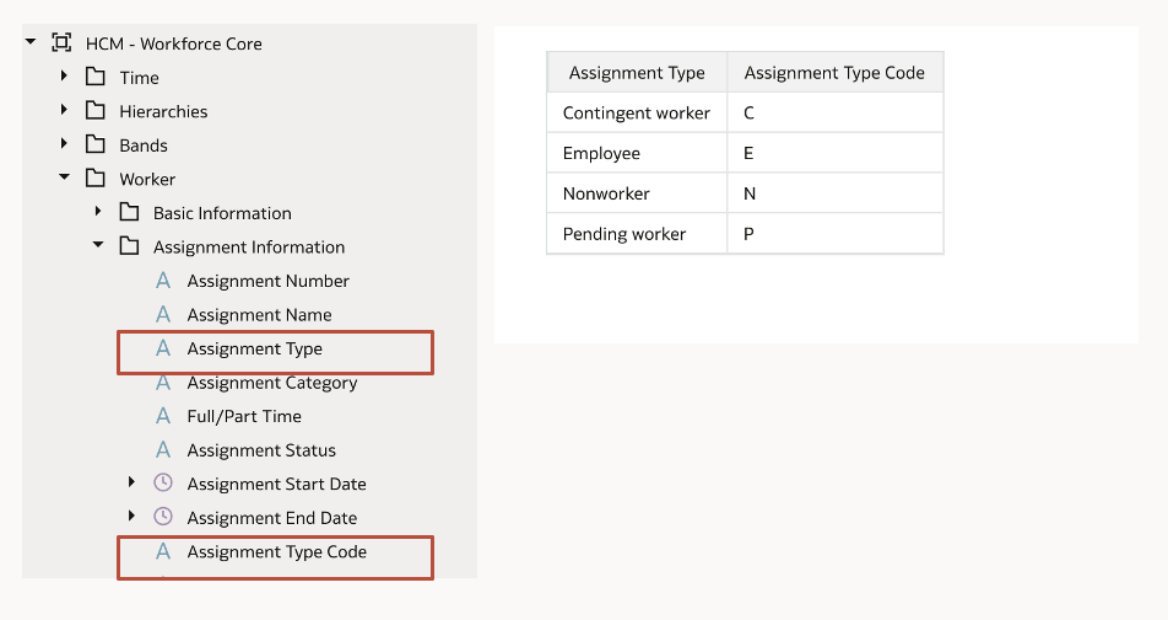

Image 9 Values of Assignment Type and corresponding Assignment Type Code columns displayed in a temporary visualization

Use INDEXCOL Function in Conjunction with CASE Statements

If your CASE statement references a parameter, consider using the INDEXCOL function in conjunction with the CASE statement to improve query performance. Query performance is optimized with the use of the INDEXCOL function because the INDEXCOL function is evaluated first, and only the required column or expression is pushed to the database. In contrast, calculations using CASE statements are sent in entirety to the database.

Example: Display Credit Amount in either Analytics Currency, Document Currency or Ledger Currency based on user selection

Users select whether Credit and Debit Amounts are to be shown in Analytics Currency, Document Currency or Ledger Currency via a Dashboard Filter (Filter Type: Checkbox/Radio Button). A parameter captures the user selection. Calculated columns for Credit Amount and Debit Amount are created which respectively display the Credit and Debit amounts dynamically based on the parameter value (i.e., the currency selected by the user).

The calculated column Credit Amount using CASE Statement:

```
case
when @parameter("Choose Currency")('AC') = 'AC' then
"Procurement - Receipt Accounting"."Facts - Analytics Currency"."Credit Amount"
when @parameter("Choose Currency")('DC') = 'DC' then
"Procurement - Receipt Accounting"."Facts - Document Currency"."Credit Amount"
when @parameter("Choose Currency")('LC') = 'LC' then
"Procurement - Receipt Accounting"."Facts - Ledger Currency"."Credit Amount"
End
```

The use of Case Statements above results in a complex query. The calculated column Credit Amount using INDEXCOL in conjunction with the CASE statement results in a simpler and more efficient query:

```
indexcol(case @parameter("Choose Currency")('AC')
         when 'AC' then 0
         when 'DC' then 1
         when 'LC' then 2 else 0 end,
```

```
"Procurement - Receipt Accounting"."Facts - Analytics Currency"."Credit Amount",
"Procurement - Receipt Accounting"."Facts - Document Currency"."Credit Amount",
"Procurement - Receipt Accounting"."Facts - Ledger Currency"."Credit Amount")
```

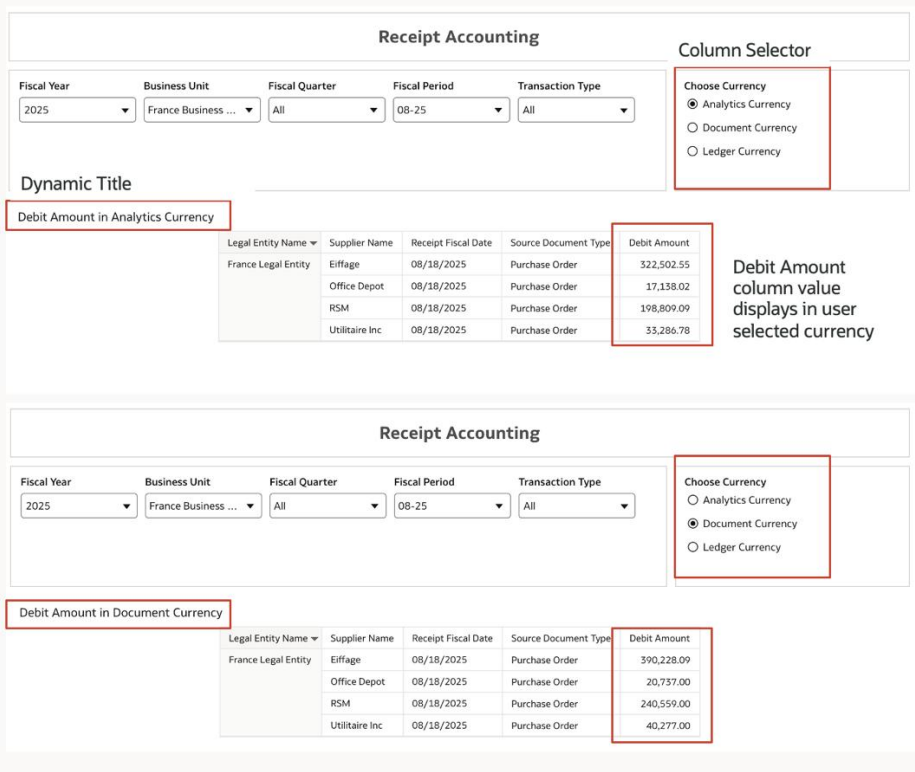

Image 10 Debit Amount displayed in either Analytics Currency or Document Currency based on user selection in the Column Selector 'Choose Currency'

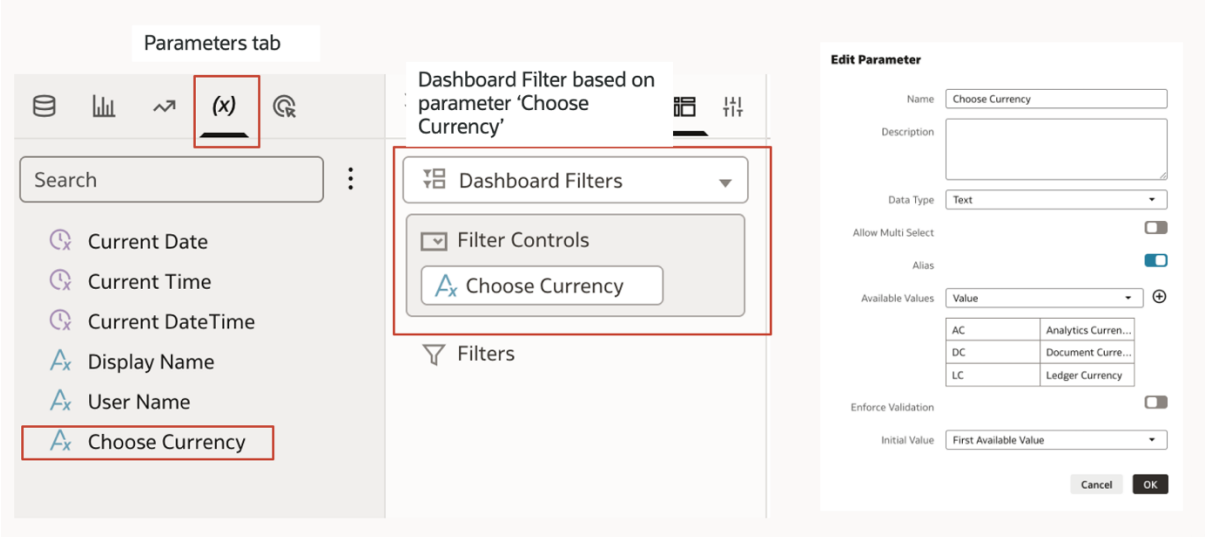

Image 11 Parameter 'Choose Currency' used to create the 'Choose Currency Column Selector by creating a Dashboard Filter based on the parameter. The Edit Parameter screen shows how the parameter was defined

See:

Best Practices for Case Statements in Oracle Analytics

Oracle Analytics Best Practices for IndexCol

Indexcol Part II – More Best Practices for IndexCol

Use of Filter Functions Instead of Time Series Functions

The Time Series functions provide for easy time-based calculations. Consider the use of filter functions if calculations using time series are non-performant. Filter functions can be used to create time-based measures through explicit filtering to achieve an equivalent result of a time series function such as the Ago function.

For example, the AGO time series function returns the Total Outstanding Amount for the previous quarter:

```
AGO("Financials – AP Aging"."Facts – Analytics Currency"."Total Outstanding
Amount","Financials – AP Aging"."Time"."Fiscal Calendar"."Fiscal Quarter",1)
```

The following Filter function is an equivalent and can be used if the Ago function is non-performant:

```
FILTER("Financials – AP Aging"."Facts – Analytics Currency"."Total Outstanding Amount"
USING "Financials – AP Aging"."Time"."Trailing Quarter Number"=1)
```

Returns the Total Outstanding Amount for the last quarter in a performant manner.

See Oracle Documentation:

Expression Editor Reference

Using Filter Functions instead of Time Series Functions

Note: Always use the latest available documentation

Optimize Filter Operators

When working with attributes that have a large number of data values, use selective operators (for example: list, equals, in) and avoid operators that are broad (for example, begins with, like, not in).

NOTE: Use of large IN lists will add to performance. If you are using an IN list that results in poor performance, consider the use of another field that identifies the desired group of values.

When filtering on Timestamp fields for comparison and range, use these recommended operators: >=, >, <=, <, BETWEEN.

Additional Performance Optimization Recommendations

- The CAST function is used to convert data from one data type to another data type - when data is stored as one type (for example, string/text) and you need to use it as another type (example, a number or date) for analysis, filtering, or calculations. Applying CAST or TO_DATETIME functions can introduce computation overheads, so use them judiciously, especially on large datasets. Work with the Data Stewards in your organization to ensure that data type conversions are performed in the database layer. This enables use of indexes and improves query performance.
- Applying functions such as CONCAT and SUBSTR directly to columns in filters can slow down your reports. For example, an expression filter with a concatenation of columns may result in slow performance:

  ```
  CONCAT("Procurement – Spend"."Cost Center"."Cost Center Code","Procurement – Spend"."Cost
  Center"."Cost Center Description") = '000Balance Sheet'
  ```

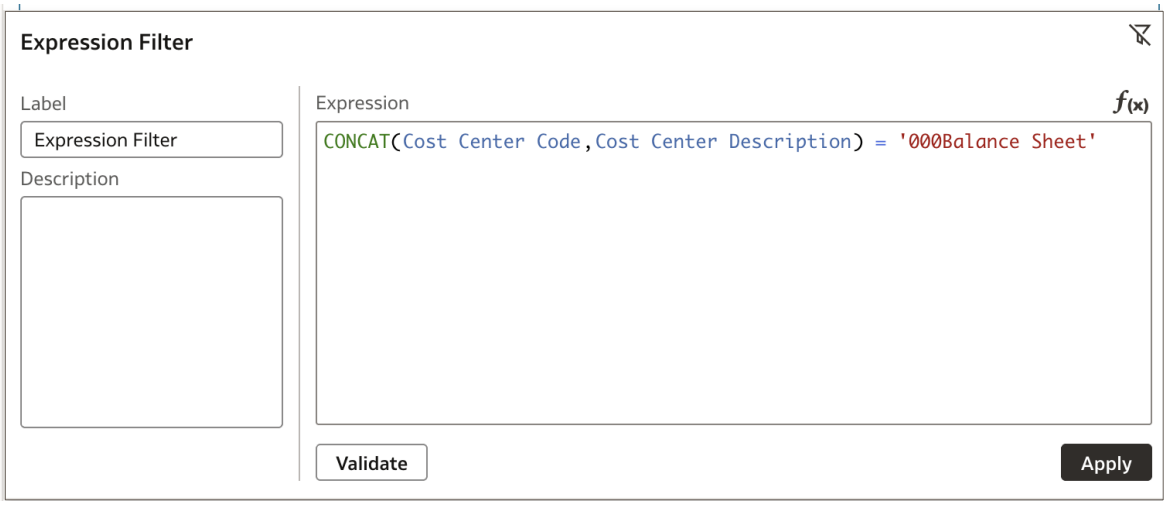

Image 12 Use of the CONCAT function in an expression filter may result in poor performance

Work with your Data Modeling and Data Steward teams to create a derived column in the semantic layer that combines the values you need. Then, use the new column added to the subject area directly in your filters. This allows indexes and statistics to work more efficiently, improving query speed.

- When using the TopN filter type to show the top measure values based on an attribute, set the By property to "All Attributes in Visual". If the By property is set to a single attribute, a separate query is created, and a second pass is made on the fact table to get the list of TopN which slow down the query.

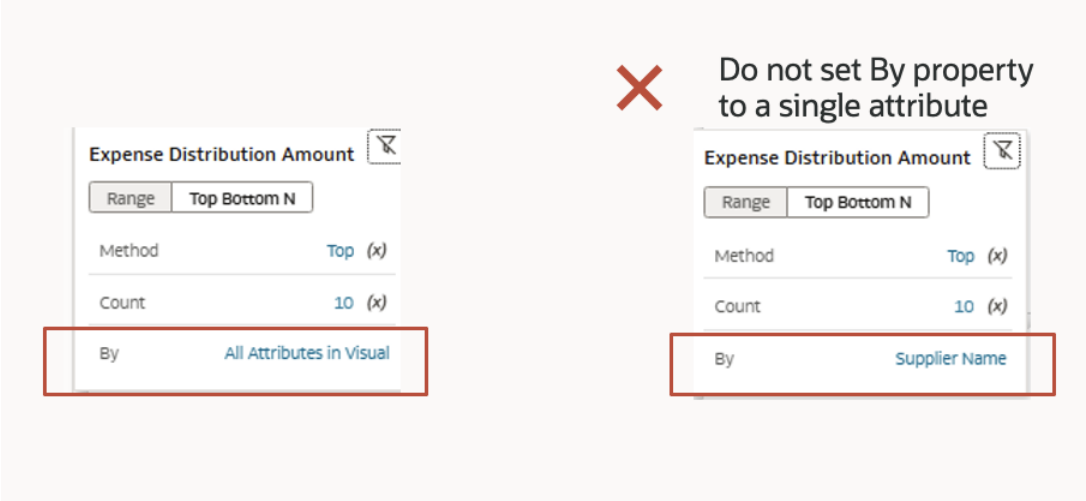

Image 13 Set the By property of a TopN filter type to 'All Attributes in Visual'. Do not set the By property to a single attribute

Also see the following section in the Oracle Documentation: Example – Use a Top N Filter to Show the Top Measure Values Based on an Attribute. *Note: Always use the latest available documentation*

- The use of AGGREGATE AT and AGGREGATE BY functions on large data volumes can result in slow performance. To achieve optimal performance, work with your Data Steward team to create a table or view that already contains the aggregated results.

Use Statistical Functions with Sufficient Data Points

Statistical functions like Trendline (which shows patterns or trends in data) and Cluster (which groups similar data points) can be resource intensive. While these functions can be applied to only a few data points (as few as two), the results are likely to be skewed and unreliable. Use statistical functions on larger amounts of data, ideally no less than 10-30 data points, so that analyses are accurate, and the performance costs are justified.

Optimize Parameters

Parameters are variables that you can define to hold dynamic values. They are used for interactive data manipulation to create column selectors and dynamic labeling. Parameters should be tuned to ensure they are efficient. For example:

```
select @parameter("<parameter name>")('') from subject_area
```

Is more performant than:

```
select column from subject_area where column = @parameter("<parameter name>")('')
```

Final Checks for your Workbook

As you near completion of your workbook, check for design-time remnants that are no longer relevant for a visualization or to the workbook. For example:

- Remove any unused or extra visuals and canvases from the workbook to reduce load times
- Remove any unused parameters as they will be initialized regardless of whether they are used in the workbook or not
- Remove any non-essential columns, redundant or hidden columns that are not serving a purpose from visualizations to reduce any extra joins

Use the Workbook Check-Up Tool to ensure that performance best practices are incorporated in your workbook. Run the Workbook Check-Up Tool as you near completion and before making the workbook available to users. The tool is available on the top bar when a Workbook is opened in edit mode: select the Custom Workbook Extension icon and then 'Workbook Check-up' from the menu. See Oracle Documentation: Analyze a Workbook for Performance Improvements. *Note: Always use the latest available documentation*

Test Your Workbook for Performance

It is critical to verify that your workbook achieves the required response times in the development environment. If your workbook is slow in development, it is unlikely to run faster in the production environment.

Test with a single non-admin user:

As a first step, make sure the workbook is fully optimized and performant for a *single user*. Be sure to test the workbook using a non-admin user account.

Test with users who have different security roles:

It is critical to test your workbook with user accounts that have different security roles to ensure the results of adding security filters can be evaluated. That is, your workbook should be tested with users with different application roles/different personas and combinations of security assignments. This testing is of particular importance as joins and queries may change depending on the security filters applied.

For example, for workbooks based on Fusion HCM Analytics subject areas, test for a Line Manager with AOR based security and a Line Manager with Security Context. Include security Exclusion Rules use cases in your testing scenarios.

Test with multiple, simultaneous users:

When you have confirmed optimal performance for individual users, test for desired concurrency with multiple, simultaneous users to validate that performance remains robust at scale before promoting your workbook to production.

Troubleshooting Your Workbook

The Developer Tool is useful in analyzing and troubleshooting query performance issues with DV. See the section "Getting logs for DV queries" in the blog titled "Oracle Analytics Best Practices – Gathering and Analyzing Query Logs" for information on how to enable and use the Developer Tool.

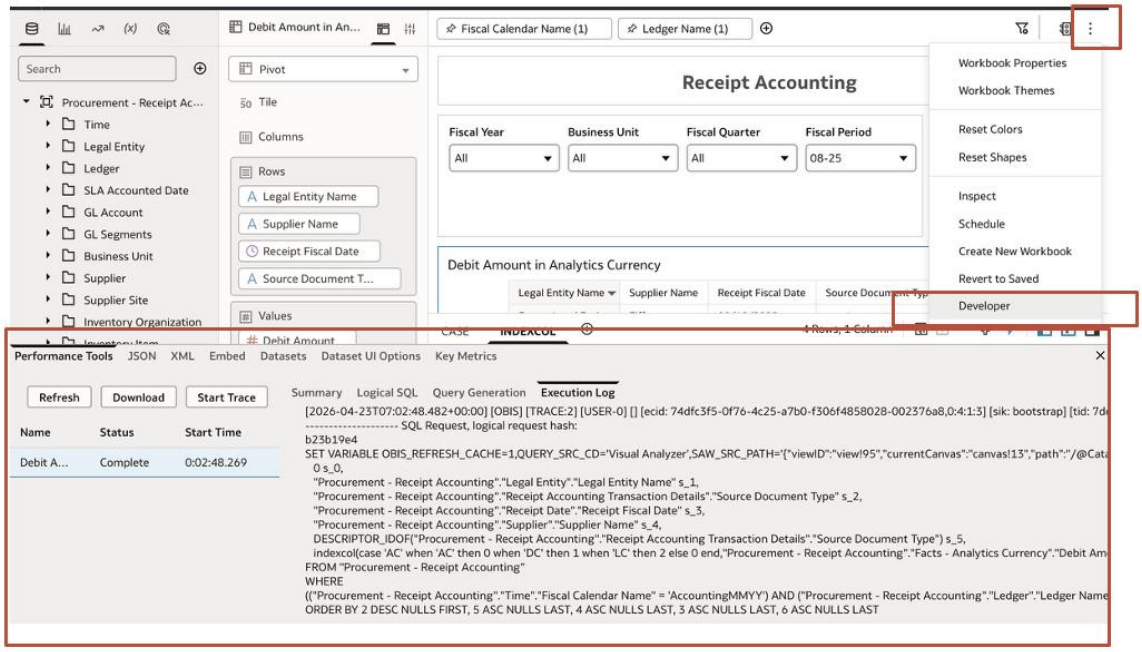

Image 14 The Developer Tool in use to analyze Execution Logs

Workbooks with Multiple Subject Areas (Cross-Subject Area Analyses)

The subject areas in Fusion Data Intelligence have been created to fit a wide range of business requirements and processes, and workbooks are typically based on a single subject area. In many cases, however, a holistic view of the business or deeper insight is required. FDI lets you build reports combining data from different functional areas by basing the workbook on more than one subject area. Consideration must be made to joins and potential performance impacts. These considerations are discussed in the FDI documentation section titled "Cross-Subject Area Report Authoring Tips". Review the information *before* building cross-subject area analyses:

ERP Analytics | SCM Analytics | HCM Analytics | CX Analytics

Note: Always use the latest available documentation

Workbooks Combining Subject Areas and Files

Data from external files (like Excel) may be combined with a subject area in a workbook to enrich analytics and create cross-functional insights.

Be aware that the joins created are between the data warehouse (ADW) and the datasets in Oracle Analytics, and as such are costly. With large volumes of data, the impact of these joins may be considerable, and workbooks may be slow to open or render.

Use external files judiciously. If performance is an issue when joining external files, consider working with Data Stewards in your organization to bring the data into the data warehouse.

Avoid joining subject areas to OTBI datasets unnecessarily; the fields in the OTBI dataset are likely to be already present in FDI.