



PUBLIC

SAP SuccessFactors People Analytics: Story Report Design & Implementation Considerations

Document Details

Name	Objective	Audience
SAP SuccessFactors People Analytics: Story Report Design & Implementation Considerations	This document provides product guidance on how to successfully help customers migrate to the new Story reports. It also provides necessary guardrails to effectively create and manage Story reports, that are important to you as a report administrator.	SAP SuccessFactors Customers: IT and HR professionals; SAP SuccessFactors Implementation Partners: Consultants, solution architects and project managers

Change Log

Version	Date	Description
1.0	07.05.2020	First Published Version
2.0	04.05.2021	Updated Performance Best Practice
3.0	11.02.2022	Removed section containing Story Report Permissions (previously section 5.2) as SAP Help Portal's People Analytics Story guide contains latest information. In addition, updated performance guardrails in section 6 and also aligned the SAP branding of Story report in the document.
4.0	30 th March 2022	Updated additional performance guidance in Section
5.0	27 th June 2022	Updated section 6.1.3 on number point chart type performance and widget selection based on unit weight per page
7.0	02.09.2022	Broken links fixed
8.0	7 th December 2022	Added additional Trouble Shooting tips Section 5.6.2.7 and a few filter tips in Section 6.1.6
9.1	23.05.2023	Removed section regarding the Explorer Mode.
9.2	23 rd August 2023	Added section 5.3 Handling Currency Conversion

Supported Releases

Product	Release - From	Release-Valid till
SAP SuccessFactors People Analytics	2111	

Contribution

Role	Name	Organization
Author / Owner	SAP SuccessFactors Product Management	SAP SE
Author	Jeroen Allonsius	SAP PS
Author	Bryan Finn	SAP
Reviewer	Erin Roberts	SAP

The recommendations in this document are based on the functionality available up to the SuccessFactors release mentioned above. Future functionality can impact recommendations provided by this document. We strive to keep these recommendations up-to-date; however, in case you find that recent new functionality has not yet been considered in the latest version of this document, please reach out to your Customer Success Manager / Partner Delivery Manager or send an e-mail to: SAPSuccessFactorsIDPDoc@sap.com.

SAP delivers Implementation Design Principles (IDPs) for SuccessFactors solutions to help customers and partners choose the most appropriate strategy and solution architecture for SuccessFactors implementations. IDPs are compiled considering the experience of many implementation projects and addressing frequent business requirements as well as real-life implementation challenges. They are continuously reviewed and updated as product functionality evolves. In addition, the reader is advised to read and familiarise with essential and additional product-related documentation, which includes Implementation Guides, SAP Notes, SAP Knowledge Base Articles, and other assets as referenced in this document; see chapter 7.

TABLE OF CONTENTS

1	TERMINOLOGY	4
2	ABSTRACT	4
3	INTRODUCTION	4
4	BUSINESS REQUIREMENTS	4
4.1	FUNCTIONAL REQUIREMENTS	4
4.2	TECHNICAL REQUIREMENTS.....	5
5	DETAILED SOLUTION	5
5.1	UNDERSTANDING REPORT CENTER USAGE FOR TRANSITION.....	5
5.1.1	<i>Table Reports</i>	<i>8</i>
5.1.2	<i>Tiles and Dashboards</i>	<i>12</i>
5.1.3	<i>Canvas Reports.....</i>	<i>14</i>
5.2	QUERY DESIGNER & ADVANCED FILTERS	28
5.2.1	<i>Selecting Data from Left Outer Join and Inner Join.....</i>	<i>28</i>
5.2.2	<i>Advanced Filters at Table Level and Query Level.....</i>	<i>29</i>
5.2.3	<i>Using Input Parameter in Advanced Filters.....</i>	<i>29</i>
5.2.4	<i>Aggregation Types in Query Designer.....</i>	<i>32</i>
5.3	HANDLING CURRENCY CONVERSION	33
5.4	LOCALIZATION CONSIDERATIONS	36
5.5	GUARDRAILS	38
5.6	DATA ANALYZER	39
5.7	DO'S & TROUBLESHOOTING TIPS OF REPORT CENTER	39
5.7.1	<i>Do's.....</i>	<i>39</i>
5.7.2	<i>Troubleshooting Tips.....</i>	<i>43</i>
6	BEST PRACTICES FOR PERFORMANCE	47
6.1	GENERAL PURPOSE GUIDELINES	48
6.1.1	<i>General.....</i>	<i>48</i>
6.1.2	<i>Modelling and Data.....</i>	<i>48</i>
6.1.3	<i>Story Design.....</i>	<i>48</i>
6.1.4	<i>Charts and Tables.....</i>	<i>49</i>
6.1.5	<i>Filters.....</i>	<i>49</i>
6.1.6	<i>Story Maintenance</i>	<i>50</i>
6.2	COMPLEX SCENARIO GUIDELINES	50
6.2.1	<i>Data Source Design for Complex Scenarios.....</i>	<i>51</i>
7	APPENDIX.....	55
8	REFERENCES	56

1 TERMINOLOGY

The following table explains some abbreviations used in this document.

Abbreviation	Description
SAC	SAP Analytics Cloud
RBP	Role Based Permissions
ORD	Online Report Designer now called Canvas Reporting
WFA	Workforce Analytics

2 ABSTRACT

This document provides various design and implementation considerations on leveraging SAP SuccessFactors People Analytics Story reports to create visually rich and critical reports for various business scenarios as well provides product guidance on how to successfully help customers migrate to the new Story reports from the existing reporting artefacts. It also provides necessary guidelines to effectively create and manage Story reports, that are important to you as a report administrator.

3 INTRODUCTION

SAP SuccessFactors People Analytics Story reports will consolidate all legacy SAP SuccessFactors transactional reporting tools into a single tool for reporting, insights, and dashboards which will enable customers to perform cross-suite reporting based on live transactional data across the SAP SuccessFactors HCM suite. This document will guide you on how to make use of the features of Story reports which are essential to create efficient and effective Stories and at the same time help you migrate to the new report tool by providing detailed design and implementation considerations.

4 BUSINESS REQUIREMENTS

Following business requirements are discussed in this document.

4.1 Functional Requirements

- **Transition to the New Reporting Tool**

People Analytics Story reports (using SAP analytics Cloud) will replace the current SuccessFactors reporting tools namely report Canvas, Table, Tiles & Dashboards in a phased manner. Customers need the skills in using the new tools and guidance on how to migrate the existing reports to Story reports as well as how they could introduce and rollout the new reports to their organization.

Due to the complexity of the tools and differences in the data model, report capabilities and customer specific SAP SuccessFactors configurations, it is not possible to provide a migration tool to convert the current reports to the new Story report type. With the new features, you can also reinvent the reporting experience to offer a better understanding of the data and HR processes with powerful data exploration and to position the Story report as a real alternative to Excel download. Therefore, it is important that you start to make an inventory of your reports and map them with business needs, become familiar with the new tool via the standard content, acquire the proper skills and start rebuilding the needed reports. This requirement is addressed in section 5.1 along with many tips and guidelines explaining how to achieve this.

4.2 Technical Requirements

This document also addresses various technical guidelines that affect the Story creation. Among other topics, it provides details on how joins are used in query designer, the different kinds of advanced filters available, and how to utilize explorer mode. The detailed guidelines are discussed from section [5.3](#) onwards.

5 DETAILED SOLUTION

5.1 Understanding Report Center Usage for Transition

When you plan to start using Story reports, it is advised to investigate the current use of Report Center and available reports in your organization. Some of the questions to investigate are as follows:

- Which type of reports are available in Report Center?
- Who are using these reports and which reports are shared by which users / roles / groups?
- What are the reports used for? Which process does a report support and in what way?
- Which reports are actually used?
- When are reports used (frequency) and how (online / offline / export)?

Report Center contains the full list of available reports, such as Canvas, Table, Tile, and Dashboard. As a Report Administrator you should be able to find the full list of all kinds of reports of your company in the “All Reports” tab in Report Center (Figure 1.):

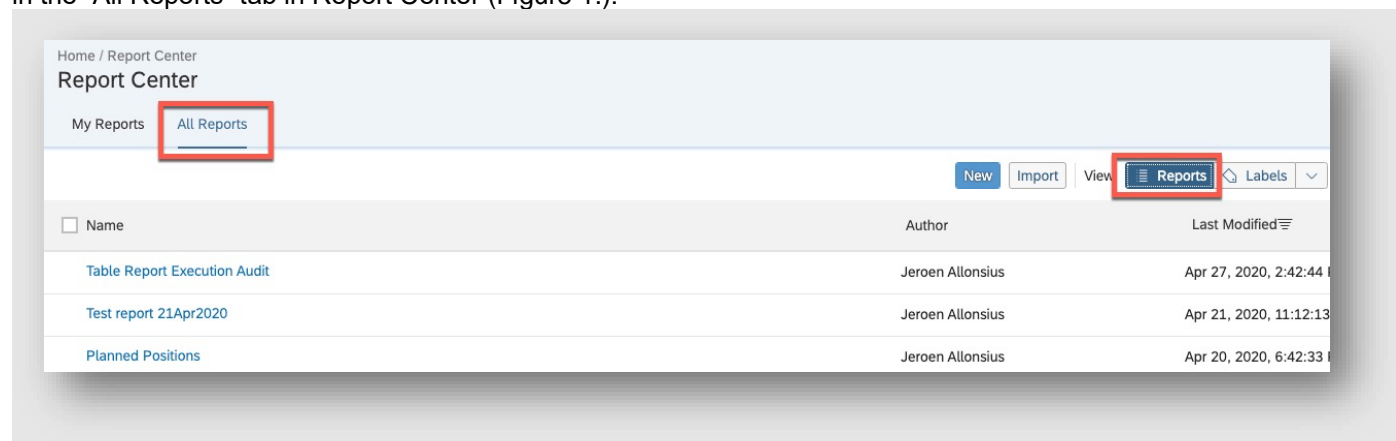


Figure 1: Report Center showing *All Reports*

Make sure you select the “All Reports” tab and select the “Reports” button as illustrated above (Figure 1). This will provide you with the full list of reports created. You can identify the report author and check if the report has been shared with other Users, Groups or Roles as shown in Figure 2. Knowing the report owners and consumers of the reports gives an opportunity to ask if the reports are still in use, are valid and what purpose they serve. Also, they should be able to answer which HR process does each report support.

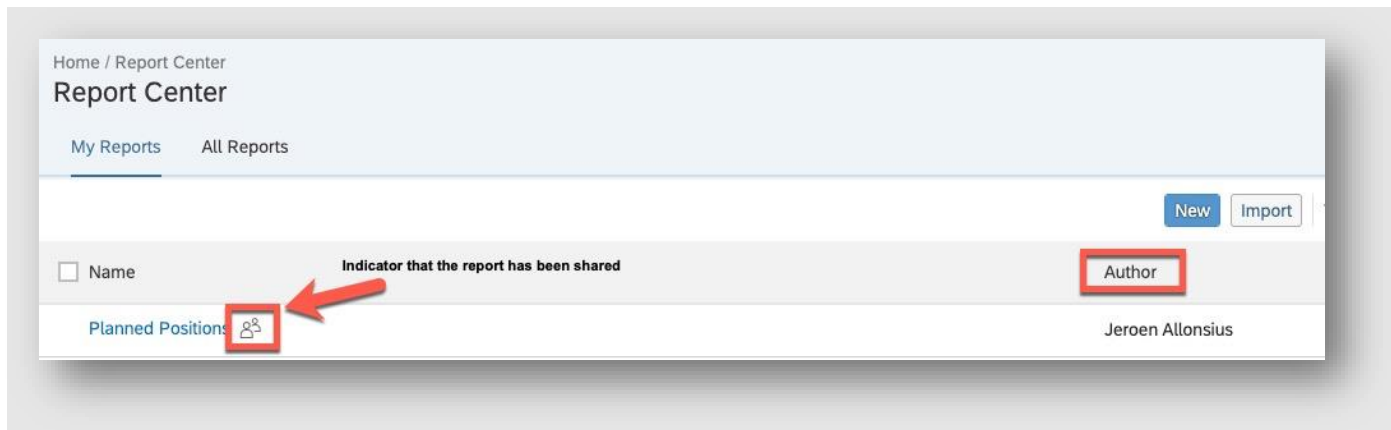


Figure 2: Report Center showing a report which is shared

A report can be shared with users using their username, a group, or a role. Using the action button in Report Center you can find out who has permissions to run each report as shown in Figure 3.

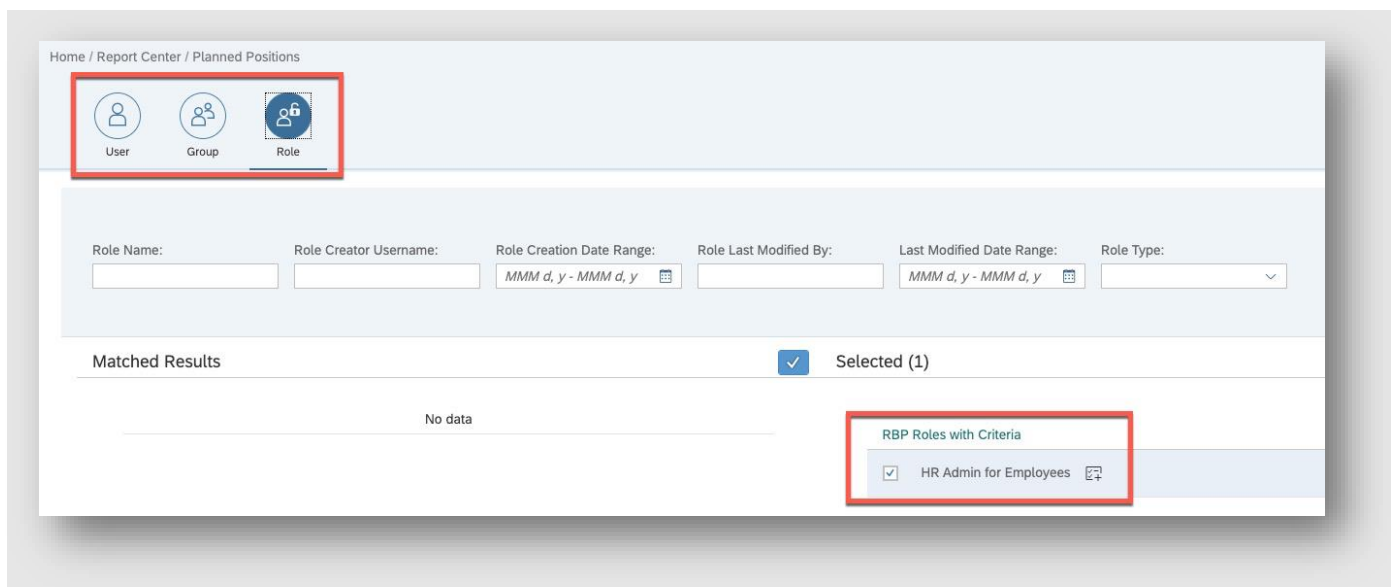


Figure 3: Permissions to run the report

Another way to identify if a report is used and how often it is run, is by using a Report Execution Audit report (for report Tables, Tiles, and Dashboards) or run Site Statistics (for Canvas).

Report Execution Audit Report

In order to track who has been running Tables, Tiles and Dashboards at which times you can create a Table Report using the *Report Execution Audit* Domain as shown in Figure 4:

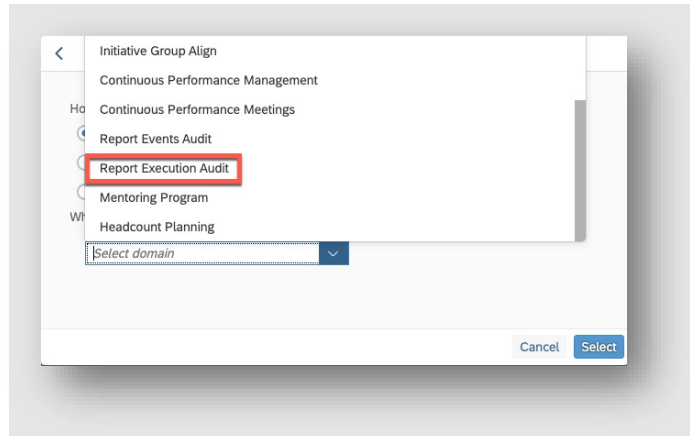


Figure 4: Report Execution Audit using table reports

The results can look like the screenshot below (Figure 5):

Table Report Execution Audit										
153272	0	admin_ja	4/27/2020	4/27/2020	Employee profile_Group_Report	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153273	0	admin_ja	4/27/2020	4/27/2020	Train Performance Tile_Group_Report	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153270	2080	admin_ja	4/27/2020	4/27/2020	Table Report Execution Audit	SINGLE_DOMAIN	Adhoc	run online	successful	Dashboard2.0
153269	0	admin_ja	4/27/2020	4/27/2020	Successors by Gender_Drill_Down_Report	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153268	1847	admin_ja	4/27/2020	4/27/2020	ListReport Succession Positions	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153267	2343	admin_ja	4/27/2020	4/27/2020	Readiness Scale	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153264	1846	admin_ja	4/27/2020	4/27/2020	TileReport Succession Maximum Successor Readiness	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153266	0	admin_ja	4/27/2020	4/27/2020	Successors by Gender_Group_Report	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153265	1848	admin_ja	4/27/2020	4/27/2020	TileReport Successors by Role and Readiness	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153263	1823	admin_ja	4/27/2020	4/27/2020	TileReport Succession Successor Readiness	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153262	2343	admin_ja	4/27/2020	4/27/2020	Readiness Scale	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153261	166	admin_ja	4/27/2020	4/27/2020	Dashboard	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153258	0	admin_ja	4/27/2020	4/27/2020	Train Performance Tile_Group_Report	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153259	0	admin_ja	4/27/2020	4/27/2020	employee profile Tile 1_Group_Report	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153260	0	admin_ja	4/27/2020	4/27/2020	Employee profile_Group_Report	SINGLE_DOMAIN	run online		successful	Dashboard2.0
153257	2061	admin_ja	4/27/2020	4/27/2020	test12	SINGLE_DOMAIN	Adhoc	EXPORT	xls	successful
153256	2061	admin_ja	4/27/2020	4/27/2020	test12	SINGLE_DOMAIN	Adhoc	download	csv	successful
153255	2061	admin_ja	4/27/2020	4/27/2020	test12	SINGLE_DOMAIN	Adhoc	run offline	xls	successful
153254	2061	admin_ja	4/27/2020	4/27/2020	test12	SINGLE_DOMAIN	Adhoc	run online		successful

Figure 5: Results of the Report Execution Audit

Site Statistics

In order to track who has been running Canvas reports at which times you can use the Site Statistics functionality. If you have the required permissions, then *Site Statistics* can be accessed when you are creating a new or modifying an existing Canvas report via the “Admin” link and select Site Statistics as shown in Figure 6.

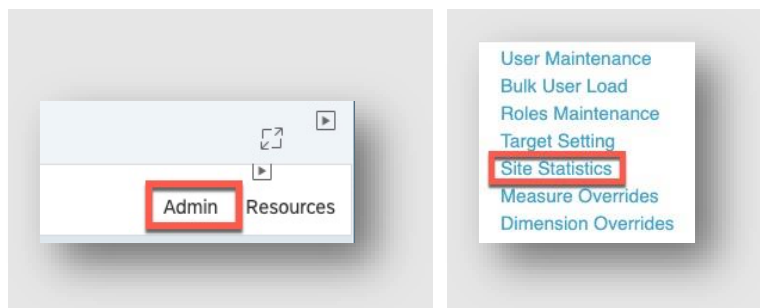


Figure 6: Site Statistics under Admin tab

When you select the timeframe, run the process and export the results to Excel then you can see who has been running which reports at what date / time as shown in Figure 7.

Exported to Excel on Monday, April 27, 2020

User ID	User Name	User Active	Login Time	Page	Measure	Time Visited	Extra Information
admin_ja_ESSHGAT0009	Jeroen Allonsius	Active	4/27/2020 2:46 PM	Test report 21Apr2020 - Page #1		4/27/2020 2:51 PM	Folder: ReportCenter, Book: Test report 21Apr2020
admin_ja_ESSHGAT0009	Jeroen Allonsius	Active	4/27/2020 2:46 PM	Planned Positions - Page #1		4/27/2020 2:52 PM	Folder: ReportCenter, Book: Planned Positions
admin_ja_ESSHGAT0009	Jeroen Allonsius	Active	4/27/2020 2:46 PM	Foundation Objects - Page #1		4/27/2020 2:52 PM	Folder: ReportCenter, Book: Foundation Objects
admin_ja_ESSHGAT0009	Jeroen Allonsius	Active	4/27/2020 2:46 PM	Learning Activities 2 - Page #1		4/27/2020 2:53 PM	Folder: ReportCenter, Book: Learning Activities 2

Figure 7: Result table of Site Statistics

Using the actions above will help you in identifying the actual usage of the reports in Report Center including Tiles that are used on the Homepage and the Insights Panel. This information will allow you to prioritize the need for each report to be rebuilt as a Story report.

Depending on your requirements, not all reports can be rebuilt yet as a Story. This will depend on the data source and specific usage. For example, Canvas reports using Workforce Analytics or Workforce Planning as a data source cannot yet be created, as well as Tiles that are used on the Homepage, Insights Panel or a mobile device. In the paragraphs below we will explain what to consider when rebuilding each specific report type (of the legacy tools) into a Story in People Analytics.

5.1.1 Table Reports

Table reports should be the easiest to rebuild as a Story. When you create or modify a Table report you are guided through a wizard using the following steps (with some domains there are additional steps which will be addressed further down this paragraph).

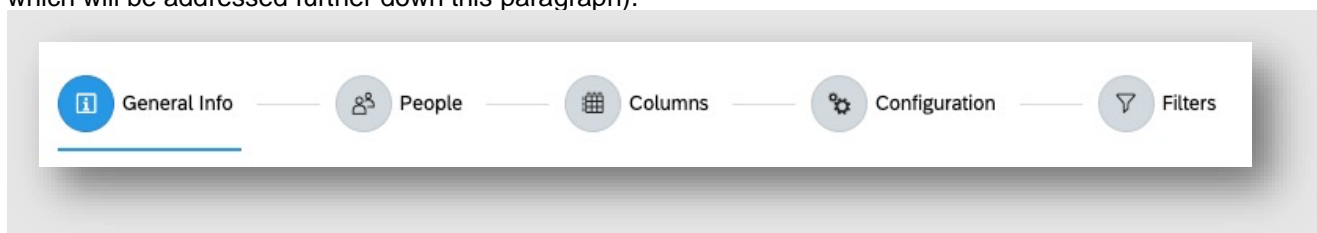


Figure 8: Wizard used in table reports

General Info: contains the report name
 People: sets the people scope based on the report consumers profile and RBP settings
 Columns: sets the columns / fields you would like to see in the output of the report
 Configuration: sets the date format and value or ID for Picklist values
 Filters: limits the report output based on certain criteria (for example COUNTRY = GBR and EMPLOYEE_TYPE = 'Employee')

Story reports have far richer functionality compared to the Table reports tool. In this section, we will focus on rebuilding a Table report in its current state only. When building a Story you will not be guided through a wizard. When you create a new Story, you will start by building a query (Data source) where you can select the columns, set the People Scope, set the configuration for Date fields and apply filters using the Query Designer.

General Info:

In Table reports, you are asked to enter the name of the report in the first step, General Info. With Story reports you can assign a name and save the report only when you have finished building your query or at later time when building the report.

Columns:

In Table reports you are first asked to complete the General Info and People steps before assigning columns and filters to the report. With Story reports you will start with building your query by selecting tables and columns and setting filters (optional). You start by selecting a domain and a table from the “Available Data” section and drag the selected table on to the query area. By selecting the table-like icon in the screenshot below (Figure 10.), you can select the columns that need to be included in the query.

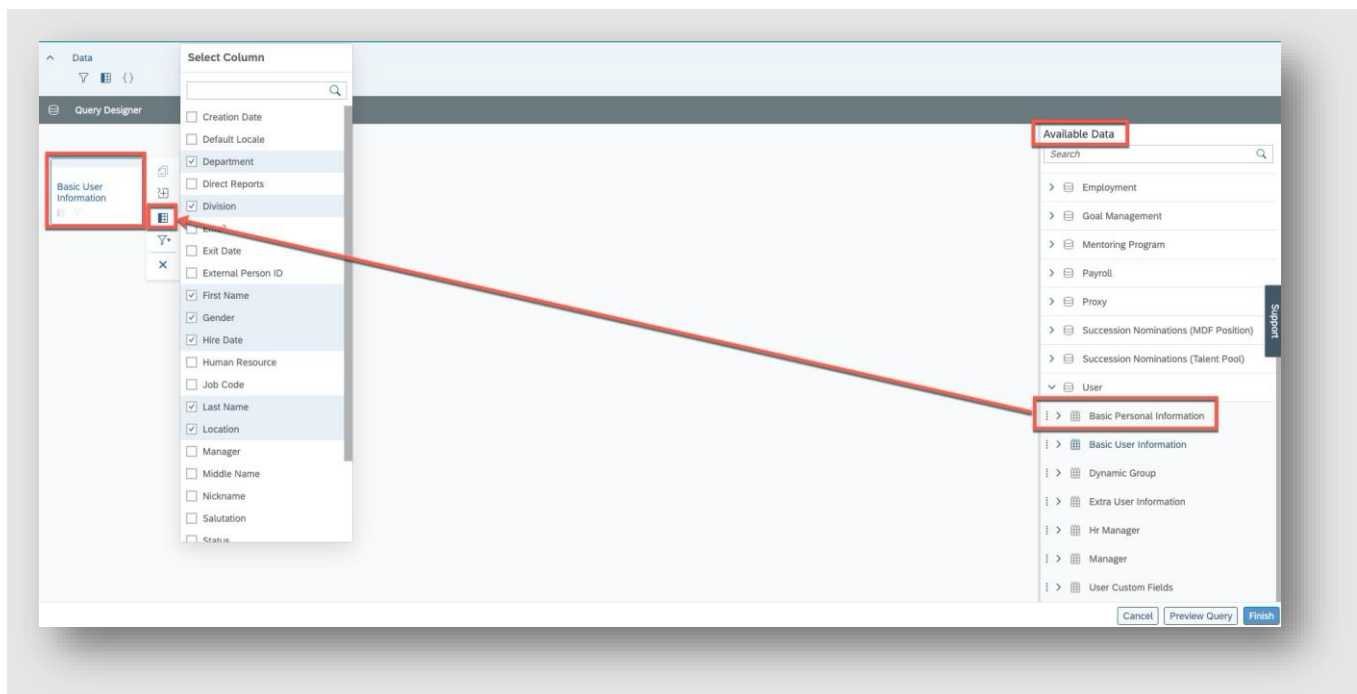


Figure 10: Query Designer in Story reports

More details on how to use other features like adding related tables, copying tables, setting table filters and using calculated columns are not described in this document and can be found in the user manual [here](#).

People Scope:

Compared to Table reports, the default People Scope is different in Story reports. While in Table reports the default setting is “Team View – Direct reports”, in Story reports the default setting is “All Data Included” (= Other Filters in Table reports). With this setting the People Scope returns all employees which the user is allowed to see based on their Target Population in RBP. Selecting the “Manager” option will give the “Team View – Direct reports” option from Table reports, including the option to select additional levels and to include inactive users. All other options are like the Table reports tool as shown in Figure 11 and Figure 12.

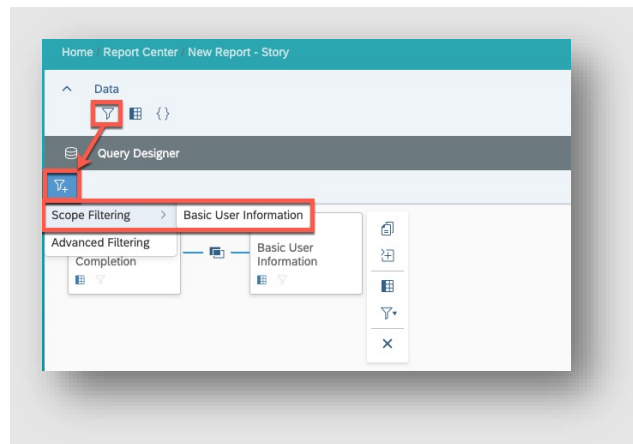


Figure 11: Filters in Query Designer of Story report

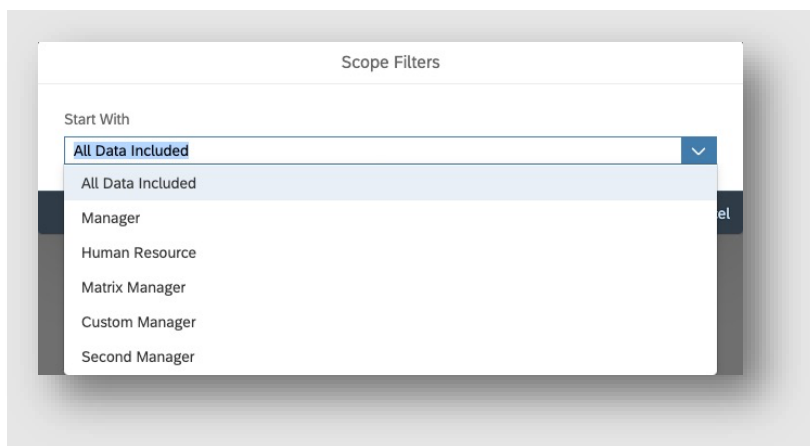


Figure 12: Accessing employee data based on RBP (Target Population)

Filters:

In Story reports, you can assign Query Filters and Table Filters. In comparison with Table reports, we will only focus on Query Filters. Table Filters will be described in section 5.1.3.2

To create a Query Filter, click on the Filter icon on the Data section of the Query Designer and select “Advanced Filtering” as shown in Figure 13 below (Scope Filtering has been described in the previous section “People Scope”).

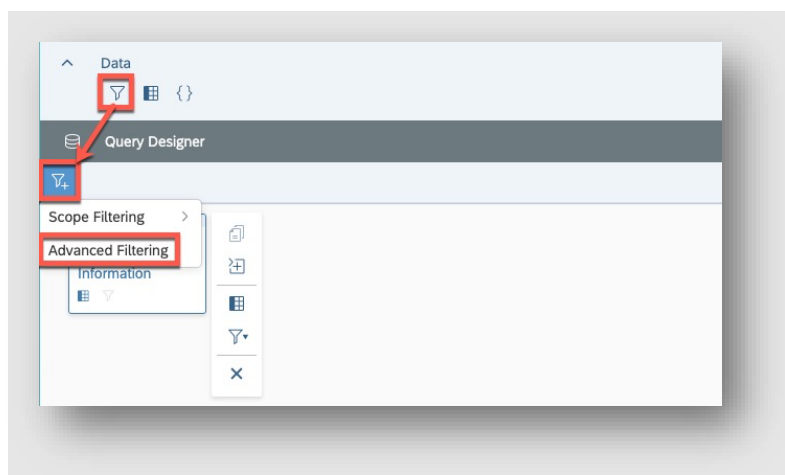


Figure 13: Advanced Filtering in Story report Query Designer

You need to assign a name to the filter and choose the columns you would like to set a filter on. Please note that you can only create a filter for columns of tables that are included in the query. You can also create AND / OR statements as shown in Figure 14.

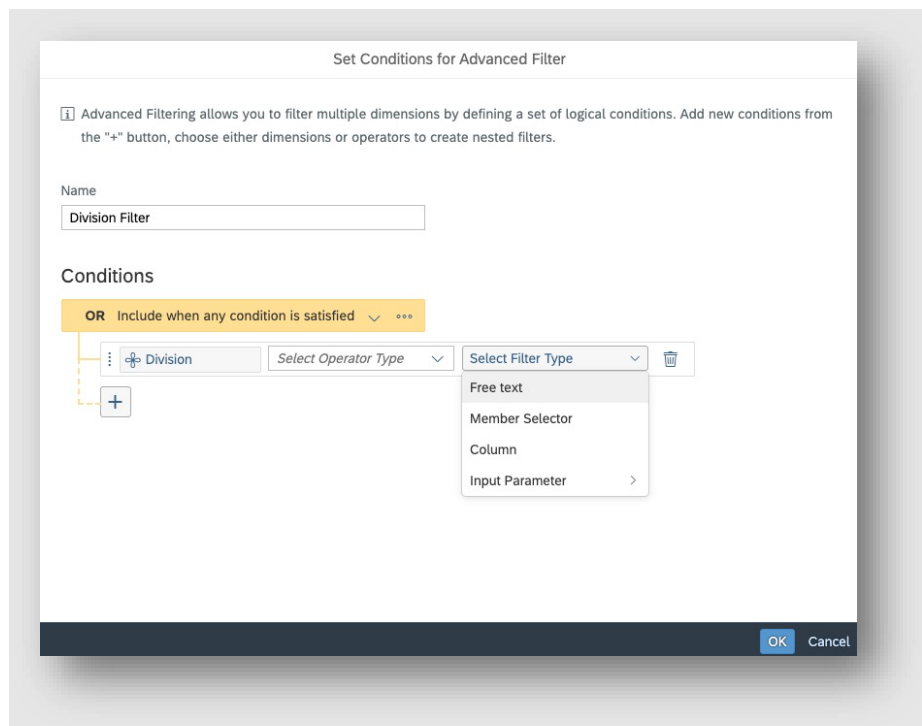


Figure 14: Set conditions for Advanced Filter in Story report

Once you have finished building your query you can look at the results by clicking on “Preview Query”. Assigning a correct order of columns is not possible when building the query, it is possible only at a later stage of the report building. Once you are satisfied with the query results, you can click on “Finished” and give the query (Data Source) a name. The next step is to add a widget to the Story report. The Story tool provides many types of widgets. but as we are describing the comparison with Tables you should add a Table widget as shown in Figure 15. Upon adding the widget, you will see a table with only one column (Count). On the Designer / Builder on the right side of the screen you can select “+Add Measures / Dimensions” to add all or some of the selected columns from the query. The Filters section allows you to set additional filters for this widget only and optionally make those filters editable by the user who is running the report.

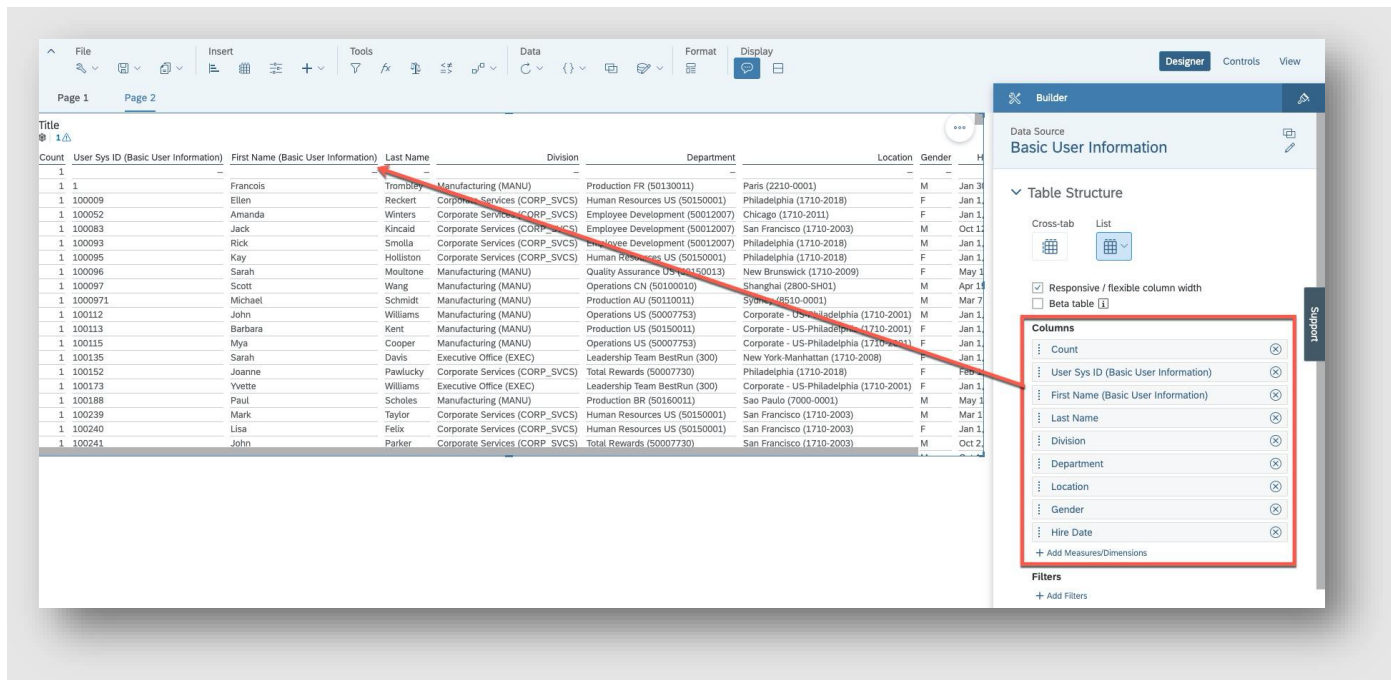


Figure 15: Table visualization in Story report

Finally, save your report by clicking on the Save icon (see Figure 16) and assign a name to the Story report. The Story will then be available in Report Center:

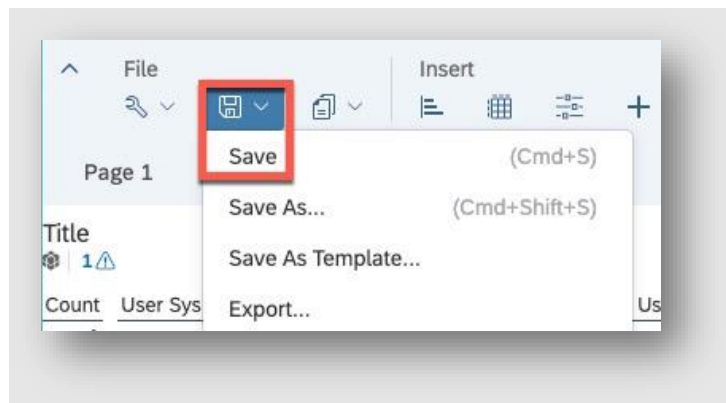


Figure 16: Save a Story report

5.1.2 Tiles and Dashboards

Tiles & Dashboards are graphical representations to support several HR business processes. A Tile shows aggregated data in a chart format (pie, bar or line) with the ability to drill to the details of the chart. For example, the number of open job requisitions per recruiter. A Dashboard is a combination of Tiles. Tiles can be shown on the Insights Panel, on a mobile device and in a Dashboard. A Tile cannot be run directly from Report Center.

We cannot yet present a graphical representation or table on the Homepage, Insights Panel or a mobile device using a Story. If that is a requirement, then you should keep using Tiles for the time being.

Using Story reports, you can create a report showing one or more graphical representations and include a table with the details to create a drill to detail functionality. You can achieve this by creating a chart widget for the graphical representation and a table widget for the detailed list. When you use filters and/or input controls and link the widgets to each other, you can simulate the drill to detail effect.

In the example in Figure 17 below, you can see a Story report that has a table widget, chart widget and an input control.

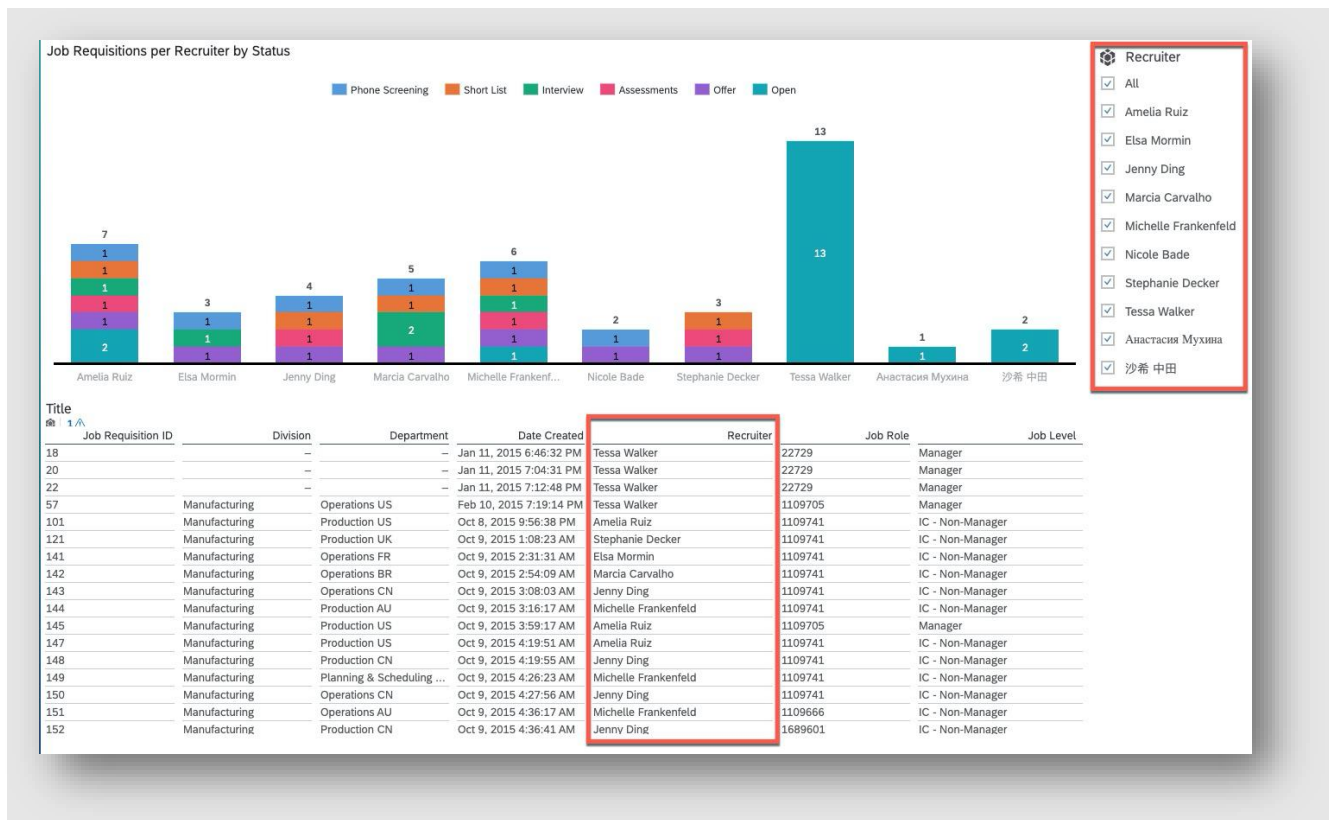


Figure 17: Story report with a table widget, chart widget and an input control

The chart widget and input control have a linked analysis setting for all widgets on the page as shown in Figure 18.

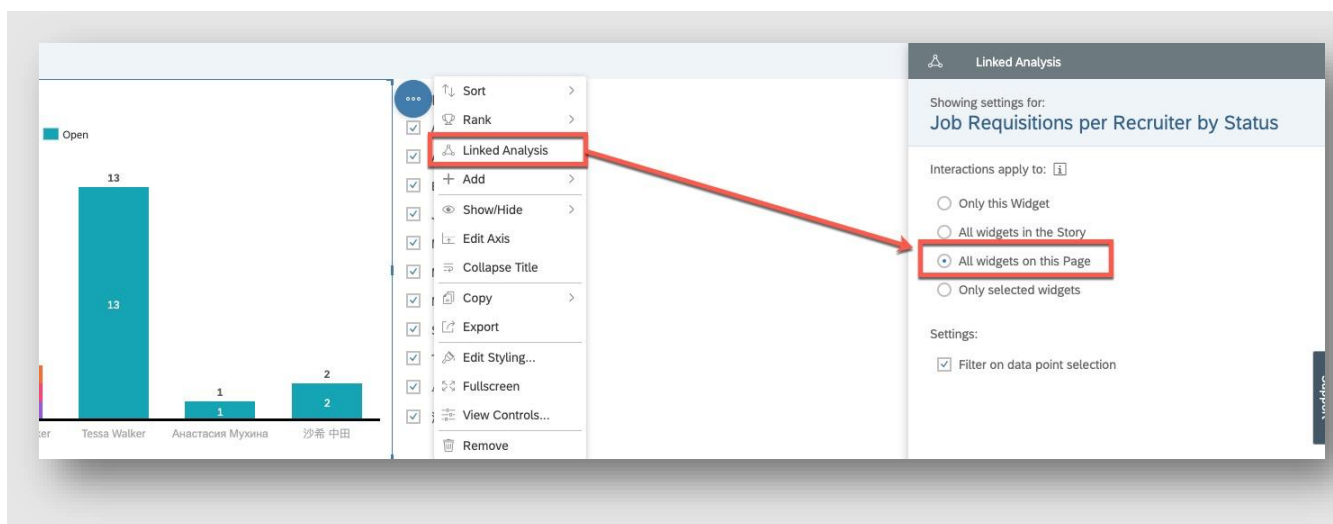


Figure 18: Linked analysis in Story report

With a linked analysis, you can either select a bar in the chart and apply a filter (Figure 19.) or use the input control (Figure 20.) to (de)select recruiters. Both the chart and the table will update with the selection.

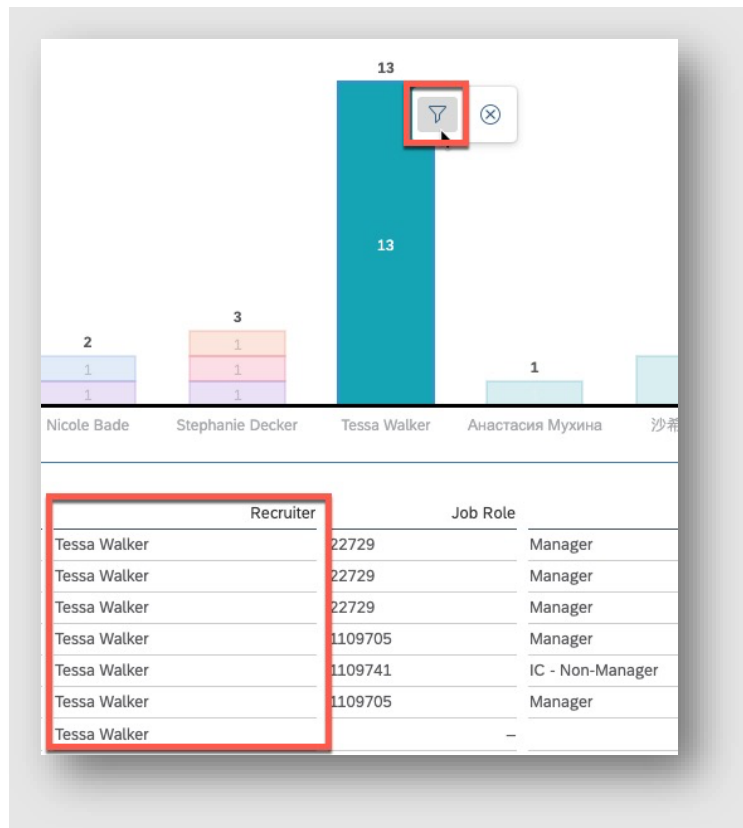


Figure 19: Filter option in the chart and the impact on the corresponding table

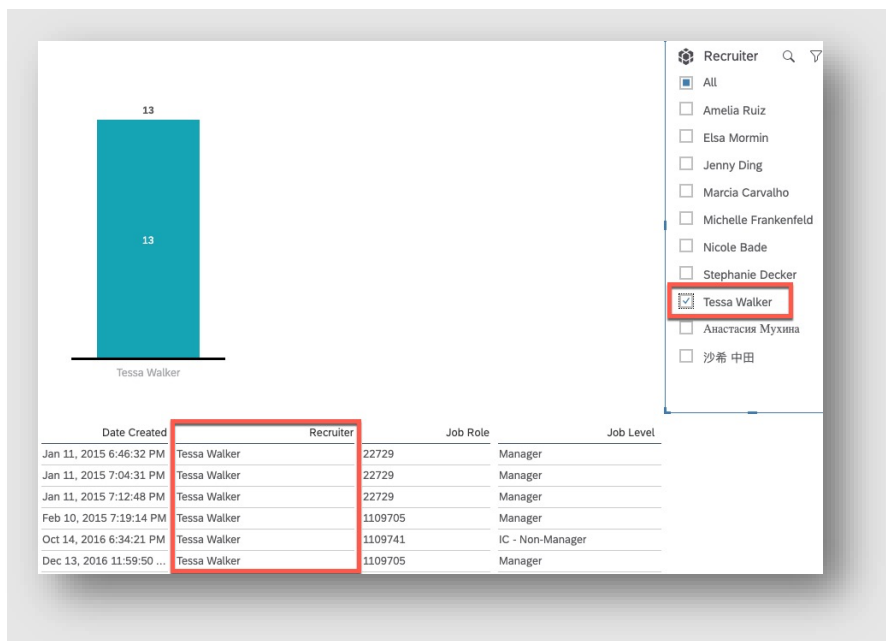


Figure 20: Using input control on the chart and the impact on the table

5.1.3 Canvas Reports

Canvas reports are reports that can be formatted in many different ways to make them fit to company branding and colors. You can create multipage and presentation ready reports using list reports, aggregated lists and charts. You have the ability to include text and image components, use calculated columns, transpose rows to columns, conditional formatting and do date effective reporting for Employee Central data. This section will explain and provide tips for migrating a Canvas report into a Story report. This section will explain the 3 different query tools, Detailed Reporting, Advanced Reporting and Workforce Analytics and Planning. In addition, this section will show how to use the other components and formatting in Story reports.

5.1.3.1 Detailed Reporting

Detailed Reporting is the query tool that uses the same reporting schema as the Table report tool, but has an additional feature to create calculated columns. In Detailed Reporting you can create three (3) data type calculated columns (Text, Number, Date) and you can include conditional logic using “If Then Else” statements as shown in Figure 21.

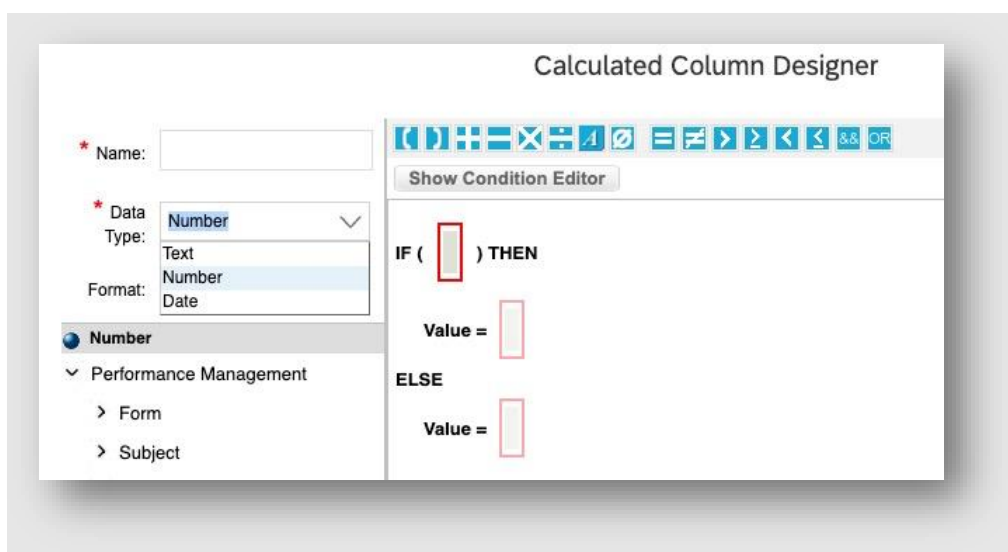
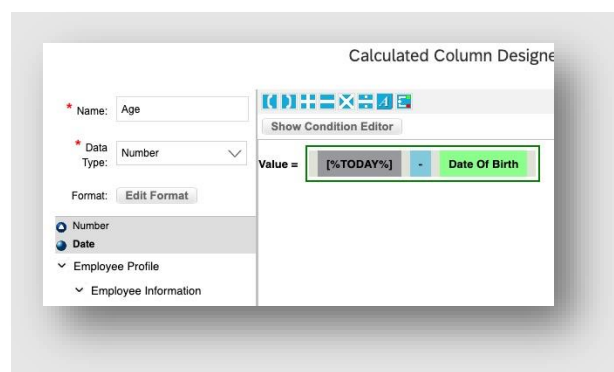


Figure 21: Detailed reporting calculated column data types

We will address some commonly used calculated columns and how to create those in Story reports.

Age Calculation:

With a calculated column in Canvas reports, you can calculate the age of an employee or the number of years of organization tenure based on the Date of Birth or Original Start Date respectively. You can calculate the number of days between “Today” and the Date of birth or you can use the “Age” function which returns the age in years (see Figure 22).



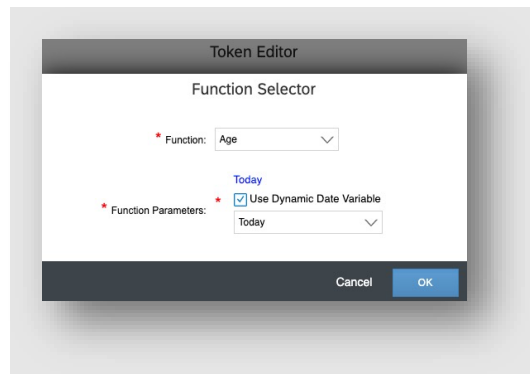


Figure 22: Using calculated column and age function in Canvas reports

With Story reports you can also create calculated columns while in Query Designer. In Figure 23 below you can see how to access the calculated columns and create new, modify existing or delete calculated columns.

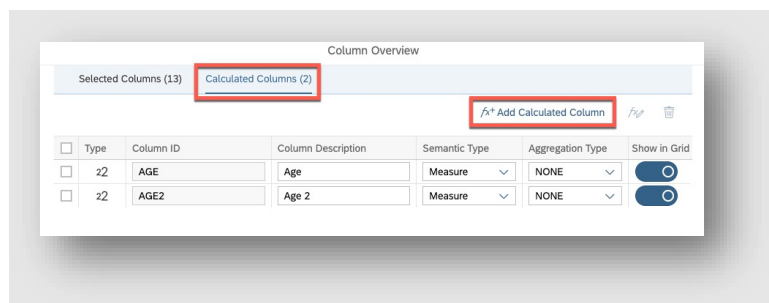
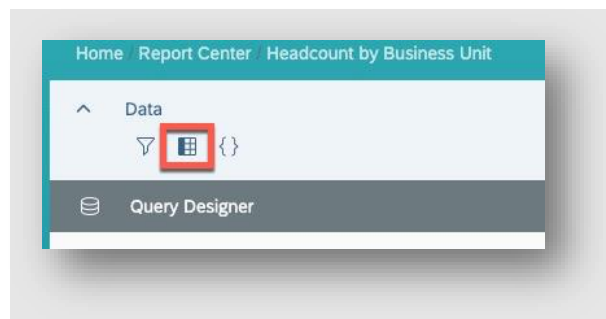


Figure 23: Using calculated columns in Story reports

To calculate the age in days you can use the “DAYS_BETWEEN” function to calculate the number of days between “Today” and the Date of Birth or another date field (see Figure 24).

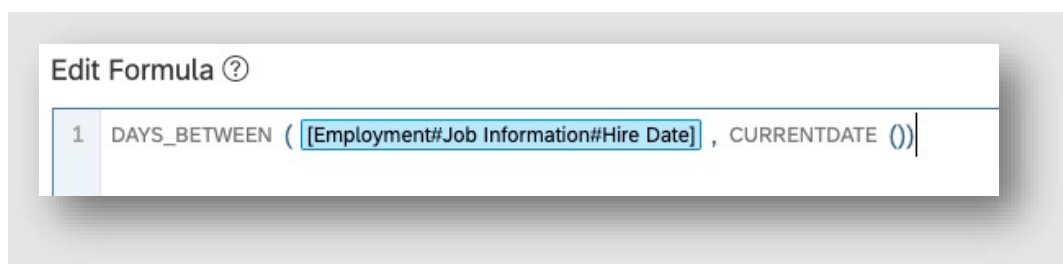


Figure 24: Formula DAYS_BETWEEN in Story reports

To calculate the age in years you need to create a different calculation as follows (see Figure 25).

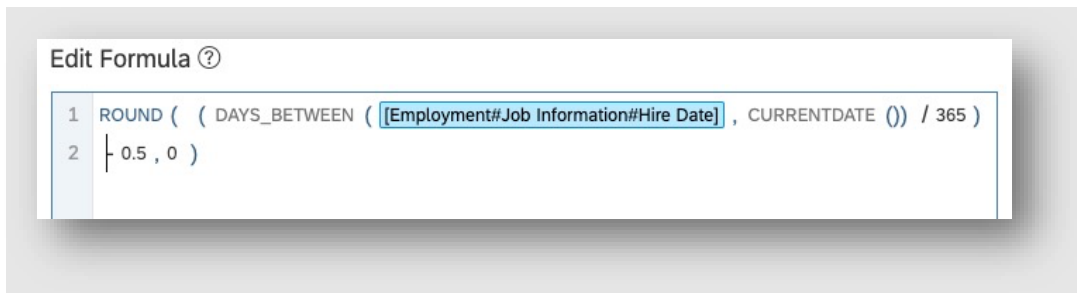


Figure 25: Formula DAYS_BETWEEN to calculate years in Story reports

This formula returns a rounded number (age in years) as of “Today” but still has 2 decimals (like 23.00). To remove the decimals, you need to add the column to a Table widget and use the Styling functionality to change from 2 to 0 decimals as shown below (see Figure 26).

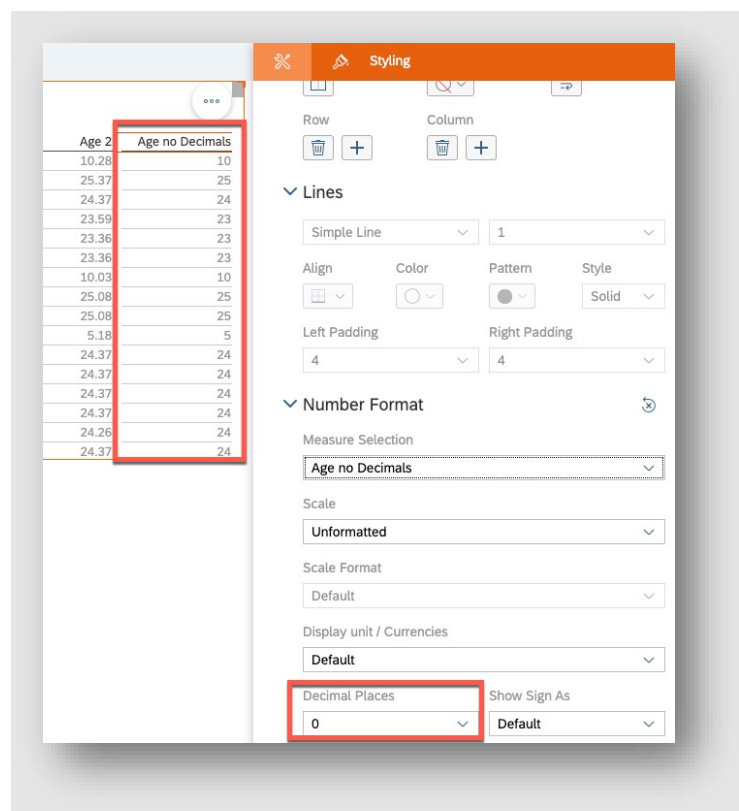


Figure 26: Setting decimal places for a column in a table of a Story report

Text Concatenation:

With Canvas reports a common calculated column is to concatenate text values into one column. For example, to create a “Full Name” column using the “First Name” and “Last Name” columns. Figure 27 is an example of how to concatenate fields in Story reports where also a blank space is added between the 2 columns.

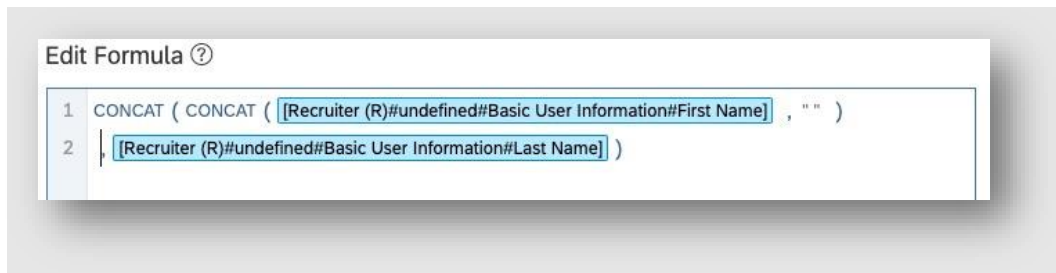


Figure 27: Using CONCAT function for text concatenation in Story reports

Using IF THEN ELSE Statements:

In Detailed (and Advanced) Reporting you can use logical conditions in a calculated column using an "If Then Else" statement. An example can be to change a code to a more understandable label which makes the report easier to read. For example:

IF [Gender] = 'F' then 'Female' else 'Male'

Another example is to create categories which you can use in aggregated tables or charts as a dimension. Below is an example of a calculated column (navigation to calculated column is already shown in Figure 23.) for the creation of an Age Category in Story reports.

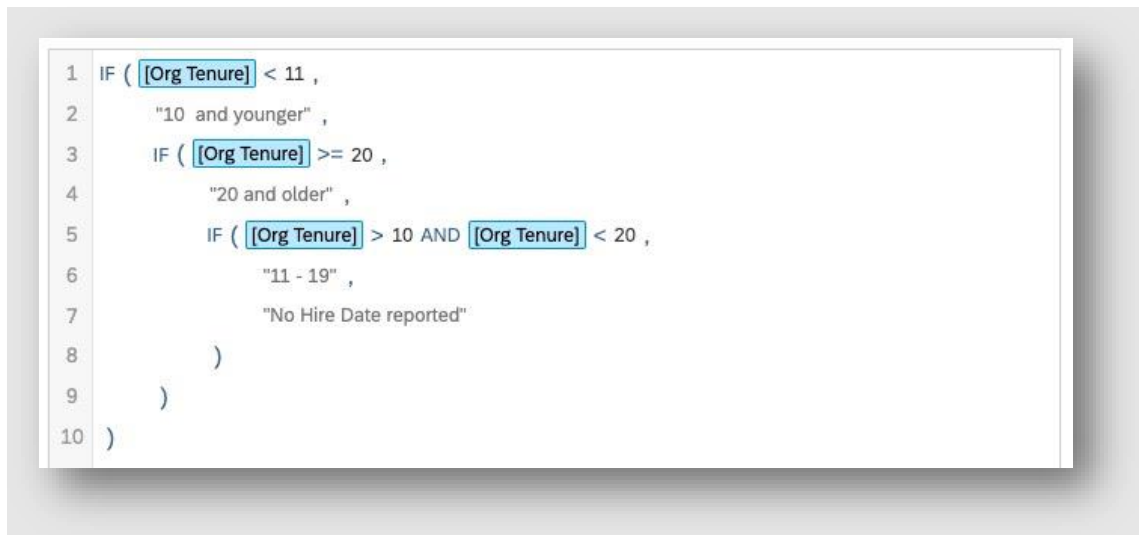


Figure 28: Using IF Statements in Story reports

5.1.3.2 Advanced Reporting

The Advanced Reporting query tool within Canvas reports can be used to report on Employee Central, Performance Management, Goals Management and Succession Management. Most of Employee Central data is effective dated. This allows you not to only see employee data as of today but also what the data looked like in the past and if needed in the future. Every time there is a change to employee data, a new row of data is added with an effective start and an effective end date and the change that was made (for example from Part Time to Full Time or a Salary change). The Advanced Reporting query tool allows you to report on any value (history, current, future) using Date options (see Figure 29).

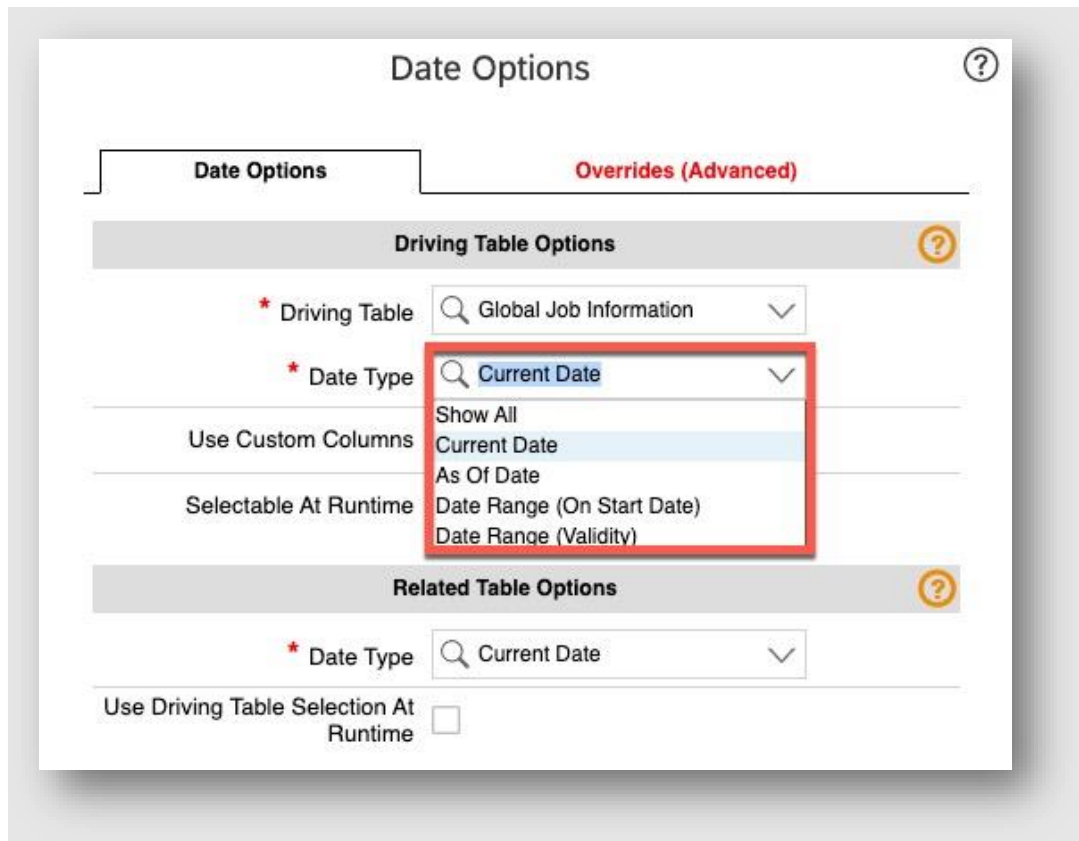


Figure 29: Using Date Options in Advanced Reporting

In Advanced Reporting there are 5 Date Type options:

- Show all: returns all the rows from the selected table
- Current Date: returns only the row that is active 'Today' from the selected table
- As Of Date: returns only the row that is active on the selected "As Of Date" from the selected table
- Date Range (On Start date): returns all rows that have an Effective Start Date which falls in the selected date range
- Date Range (Validity): returns all rows that are active in the selected date range

Date options can be set separately for each table in the query. The default setting is Current Date for most tables.

With Story reports, you can also define which rows should be returned for each table using the Time Filter option. The default setting is "Today". To change this setting, you need to be in the Query Designer and open the Table Actions (see Figure 30).

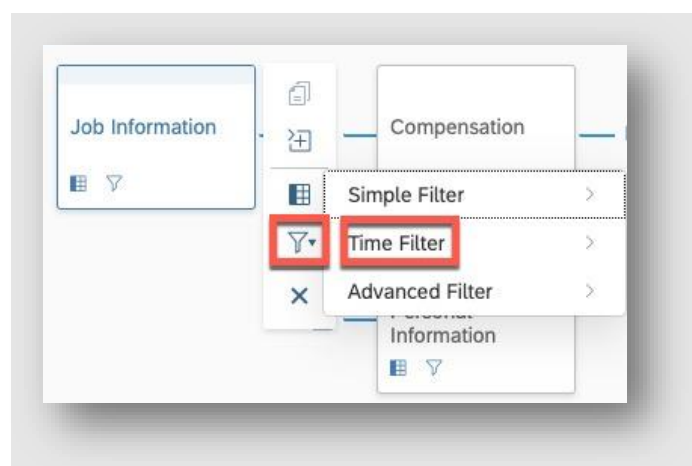


Figure 30: Time filter in Story reports

You can then select the Time Filter and make changes. When you Edit the filter you can choose a fixed date or another Dynamic Date Variable (see Figure 31).

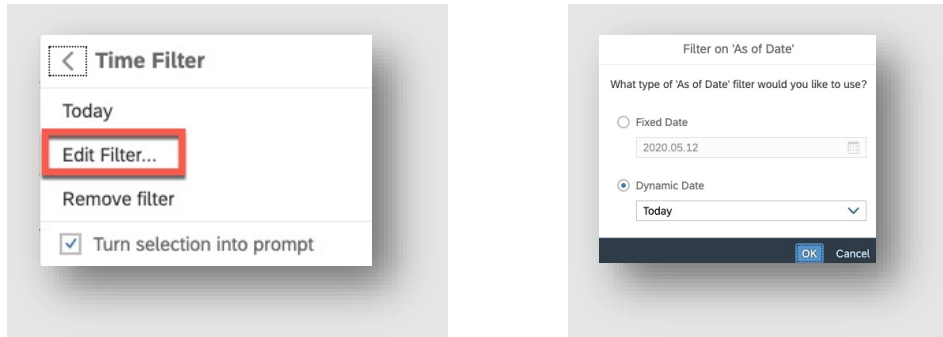


Figure 31: Using Edit Filter within Time Filter of Story reports

When you remove the filter then you will get all rows returned for that table which is the same as “Show All” in the Date Options within Advanced Reporting (see Figure 32).

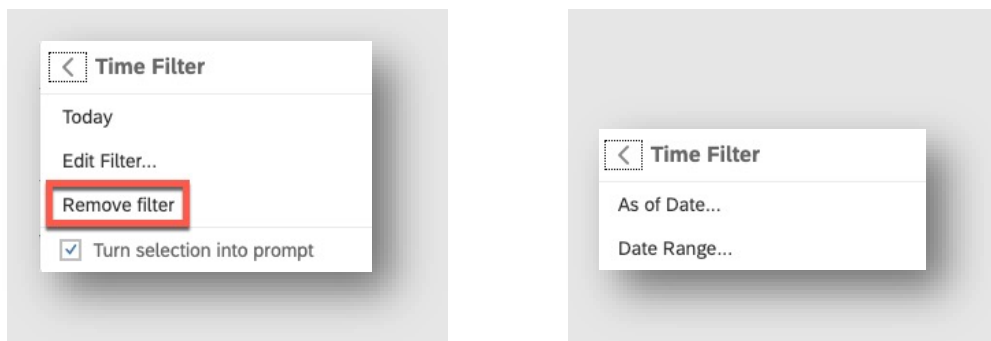


Figure 32: Remove filter and Date Range option in Time Filter of Story reports

You can now also select a Date Range where you can set the Range Type (Dynamic or Fixed), the Granularity (Year, Quarter, Month, Week, Day), the Range Period (this year, previous year or next year), the Range Extensions and the Range Parameters as shown in Figure 33.

Figure 33: Set Date Range in Story reports

Additionally, you can define which records need to be returned within the Date Range. The first option as shown in Figure 34 is the same as Date Range (Validity) in Advanced Reporting and the second option is the same as Date Range (On Start Date):

Figure 34: Records Returned in Date Range of Story report

Table Filters:

In Advanced Reporting you can set table filters. A table filter only impacts the selected table whereas a query filter has effect on all tables in the query. A table filter can be useful if you would like to transpose rows to columns. As an example, when you add the Compensation Table to the query employees with more than one pay component will have a row for each pay component on their record. By copying the compensation table and applying a table filter on each instance of the table, you can show one column for each pay component. This will then result in only one row of data per employee.

In the example below you see that you will get multiple rows of data for one employee:

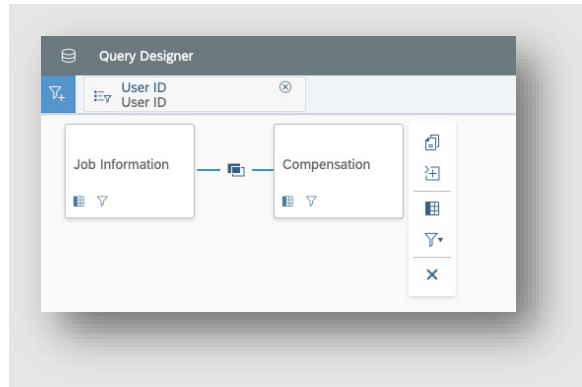


Figure 35: Query Designer with Job Information and Compensation datasets in Story report

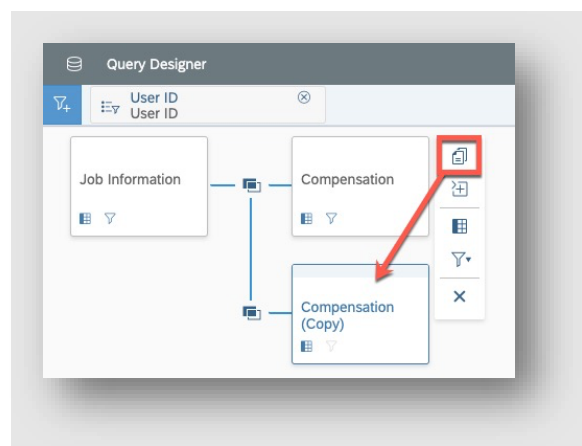
^ Data

Query Designer

AA User ID	z2 Amount	AA Currency Code	AA Name	AA Pay Component ID
101006	500.00	GBP	Home Fuel Allowance UK	HF_ALL_UK
101006	12,800.00	GBP	EE Basic UK	EEB_UK

Figure 36: Data retrieved in the table of Story report

By adding a copy of the Compensation Table and applying a table filter on the Pay Component we can transpose the rows to columns:



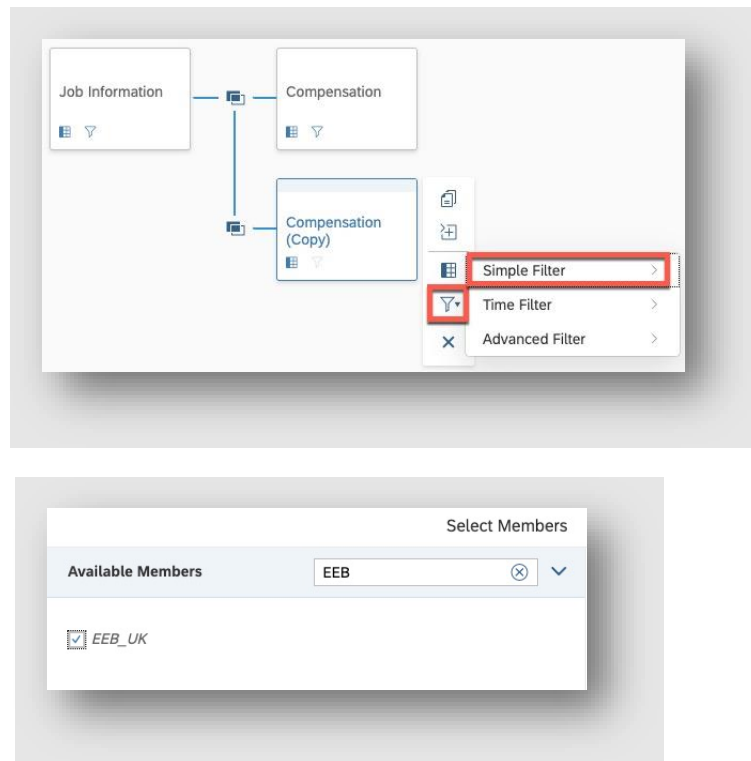


Figure 37: Additional copy of Compensation Dataset in Story report

In the example above we have set a Simple Filter for Pay Component “HF_ALL_UK” on the first Compensation Table and on the copy of the table we have set a Simple Filter for Pay Component “EEB_UK” resulting in only one row for the employee as shown in Figure 38.

AA User ID	I2 Amount (Compensation)	I2 Amount (Compensation (Copy))	AA Name	AA Pay Component ID	AA Name (undefined)	AA Pay Component ID (undefined)
101006	500.00	12,800.00	EE Basic UK	EEB_UK	Home Fuel Allowance UK	HF_ALL_UK

Figure 38: Simple Table Filter in Story report

5.1.3.3 Aggregated Tables

With Canvas reports you can create pivot tables or aggregated tables using a query from either Detailed or Advanced Reporting. With Story reports it is also possible to aggregate data from a query / data source and present it in a table. There are two types of table widgets you can use for pivoting / aggregating data: aggregated list table widget or a cross tab table widget.

Aggregated List Table:

When adding a table widget to your Story the default setting is a non-aggregated list. You can change this setting to an aggregated list or a cross tab table (Figure 39):

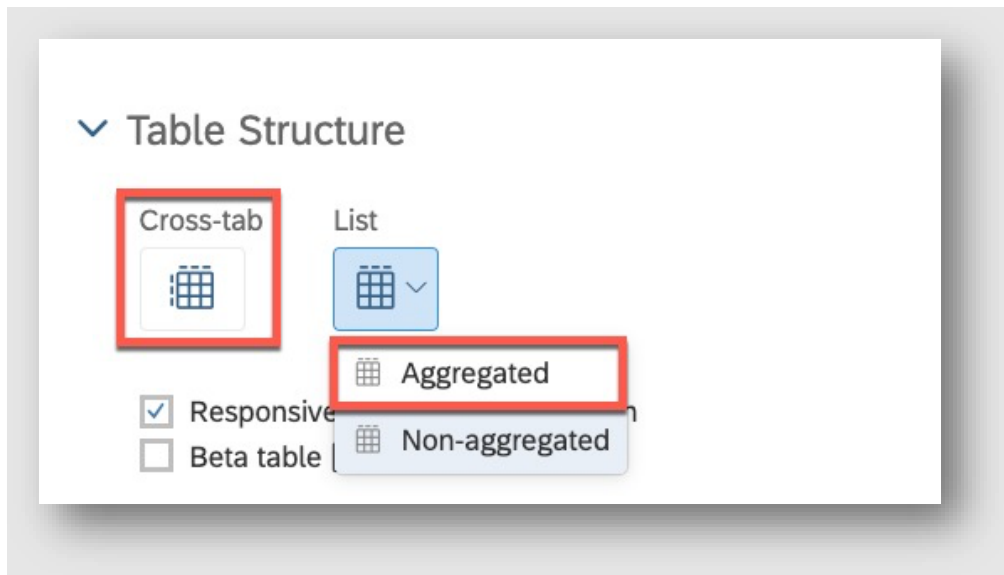


Figure 39: Aggregated table structure

When you select a non-aggregated list you can do simple counts only (like a headcount or sum of FTE) by selecting one or multiple measures. You can also select one or multiple dimensions (like Business Unit and Gender), but the dimensions will only be shown on the rows of the table. You can also set a filter on one or more dimensions:

Business Unit Name	Count	FTE
Corporate	457	473.55
Products	683	689.20
Sales	59	58.10
Retail	20	19.50

Figure 40: Selecting measures & filters in non-aggregated table

Crosstab Table:

When you need to show dimensions on the columns or both on the rows and columns then you need to use a crosstab table. With a crosstab table, you can also create and add more complex calculations. In the example below we use a crosstab table where we are showing the dimension gender on the columns, the dimension business unit on the rows (Figure 42) and created a calculated measure to calculate the average organization tenure (Figure 41):

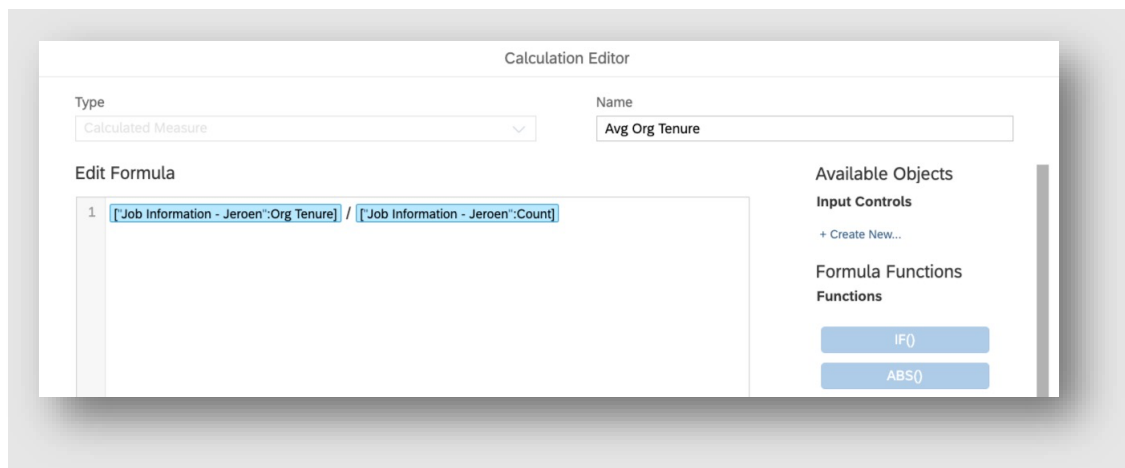


Figure 41: Calculated measure *Average Organizational Tenure* of an employee

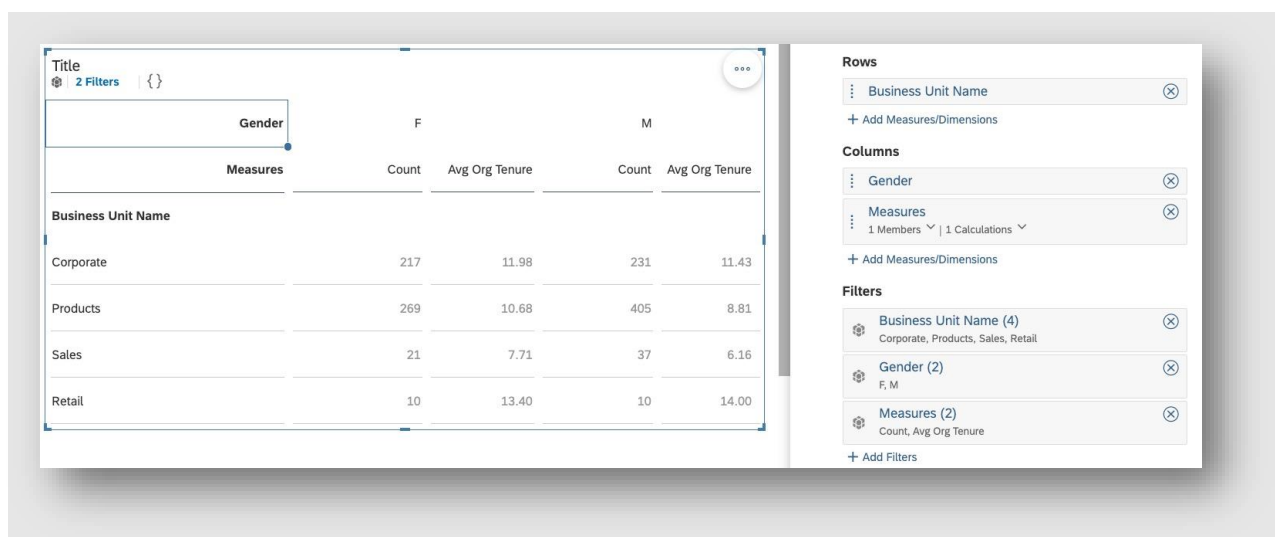


Figure 42: Crosstab table showing Business Unit Rows and Avg Org Tenure as columns

Using aggregated tables in Story reports also gives you more flexibility by showing or hiding dimension names and changing the order of the dimension values. These are known limitations with Canvas reports.

Story designer enables an improved table experience by checking the “Beta Table” checkbox in the Builder Panel for :

- Smooth scrolling – also when your cursor hovers over the table (no click needed)
- Improved performance of in-cell charts and thresholds
- Ability to specify exact row height or use the predefined Fiori heights (Default, Condensed, Cozy)
- Ability to specify exact column width for non-responsive tables
- Deselect linked analysis members with ESC

In the near future, this “Beta Table” format will be used by default for all tables.

5.1.3.4 Charts

With Story reports, there is a large variety of chart types available. Working with charts is similar to working with crosstab tables where you need to select your measures and dimension(s) and can create and use more complex calculations. Depending on the chart type, you are presented with different options. In the example below we are showing a vertical stacked bar chart showing the average organization tenure by business unit split by gender:

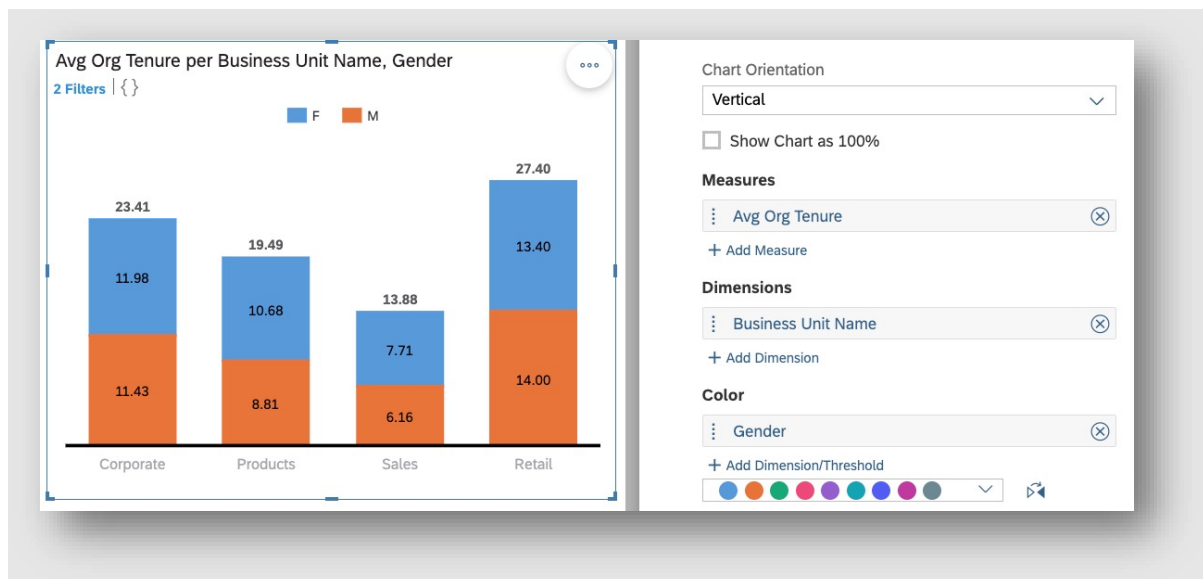


Figure 43: Vertical stacked bar (chart type) showing Organization Tenure by Business Unit

Explore this site for more information on chart types: <https://www.sapanalytics.cloud/resources-chart-type-guide/>

5.1.3.5 Workforce Analytics and Planning

Workforce Analytics (WFA) and Workforce Planning (WFP) data sources are currently not yet available for use in Story reports. Therefore, any reports that are currently built using these data sources cannot be migrated to a Story report. For the time being, keep on using Canvas reports in these scenarios.

5.1.3.6 Report Formatting

It is recommended that you design your Story report with responsive pages instead of canvas or gride page types. Responsive pages allow your Story content to re-flow depending on the size of the screen it is being viewed on. This format is also compatible with mobile apps ensuring that once mobile is ready your reports will already be mobile compatible. Figure 44 shows the different page types.

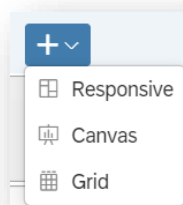


Figure 44: Types of pages in Story reports (using Responsive page type)

If you want to control the page size (see Figure 45), you might prefer to use the canvas page type. This page type is useful in cases where you expect your report to be printed or saved to PDF in a specific

format. Sometimes it is better to use the grid page type for large tables in order to facilitate the export feature.

Figure 45: Using Page Size in Story report

Create a report template in order to easily initiate a new report that follows a standard format you wish to employ. A report template allows you to easily create reports with the same format, text, background and picture. You can create a report template by defining your brand / style and by selecting 'save as template'. You will be able to apply this template in the format section of the story editor for any other Story reports.

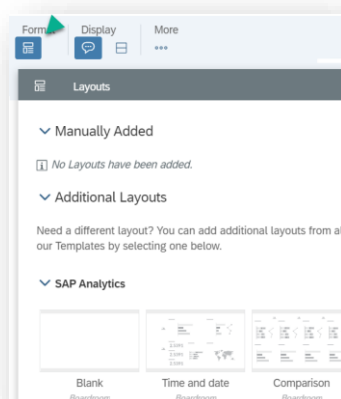


Figure 46: Format section

in Story Editor

As it is not possible to share a query for use in another Story, we recommend the creation of different story templates with not only the design, but also with single and simple queries which can be enhanced later. As an example, a report template with key Job Info and Personal Information details could be leveraged later to produce reports on performance data or compensation reports. You can initiate a new story report by copying a story template. Keep in mind that the query definition is stored in each report and if you perform a change in one, it will not change the others.

Story reports will support multiple languages soon allowing you translate the different fixed texts in a story. Data and field names will be updated from the SuccessFactors configuration (fields and picklists).

5.2 Query Designer & Advanced Filters

5.2.1 Selecting Data from Left Outer Join and Inner Join

In query designer, data elements can be joined in two different ways. Based on the requirement, the query creator can select the type of join.

- **All Primary Data (Left Outer Join):** This is equivalent to a left outer join. All the rows from the left table are returned irrespective of whether there is a match in the right table. If a matching id is found in the right, it is returned, otherwise a null is appended. This is the default join type when two tables are joined as shown in Figure 51.

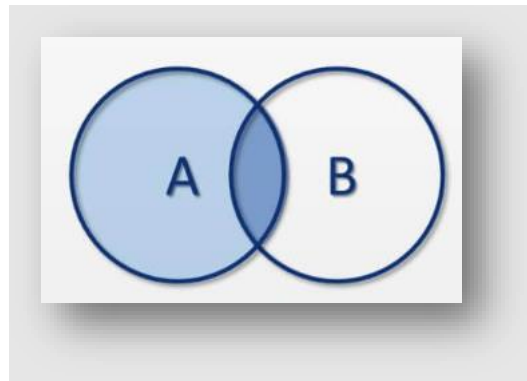


Figure 51: Left outer join dataset

- **Intersecting Data (Inner Join):** Returns only the data that has a match on the join condition from both tables as shown in Figure 52.

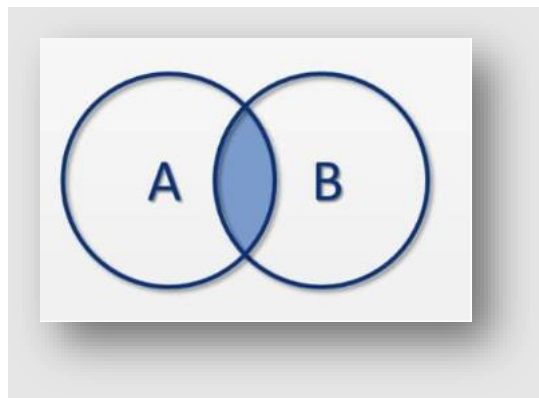


Figure 52: Inner join dataset

With your mouse on top of the connection, you can understand the type of join between the two data elements and you can change the type of join as shown in Figure 53.

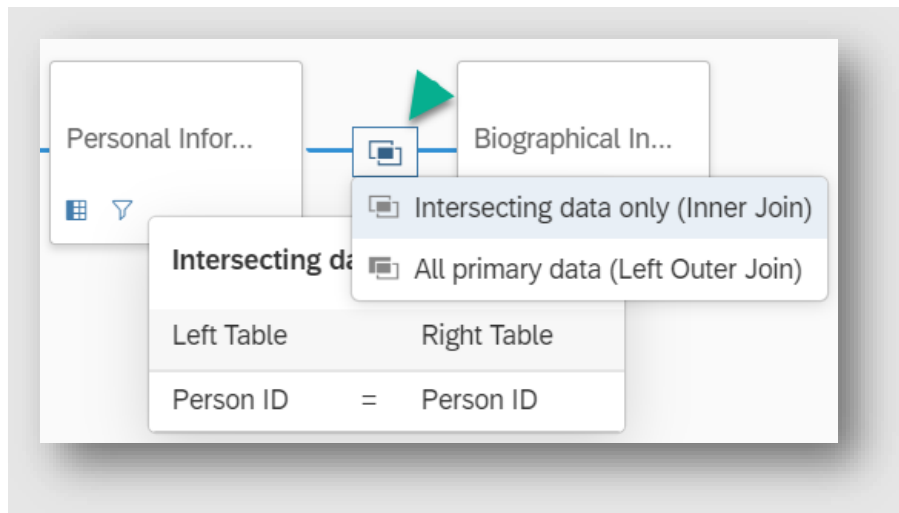


Figure 53: Available join types in Story report

5.2.2 Advanced Filters at Table Level and Query Level

Query designer supports advanced filtering that can be created at the table and query level. It impacts reports in the following ways:

- Table level: Filtering will be limited to the specific tables on which filters are applied. Columns on which filters can be applied will be limited to columns present in the table.
- Query level: Provide the ability to apply filters across the query. Filters can be applied on any column from any of the tables included in the query.

5.2.3 Using Input Parameter in Advanced Filters

Input parameter allows the report designer to provide parameters to the user to input select values for the filter and save its value. For example, instead of predefining filters in the filter definition, input parameters allow the user to input filter values while previewing or finishing a query. This increases the flexibility of the report without making modifications to the filter definition.

As an example, if a user wants to limit the report output to Analyst job title in one instance and in a separate instance wants to limit the report output to job title Manager, advanced filters with input control can be used. This prevents the user from making any changes to the filter definition in each case.

An input parameter must exist prior to using it in advanced filters. To create an input parameter, follow below steps.

1. Click on Create Input Parameter icon as shown below (Figure 54)

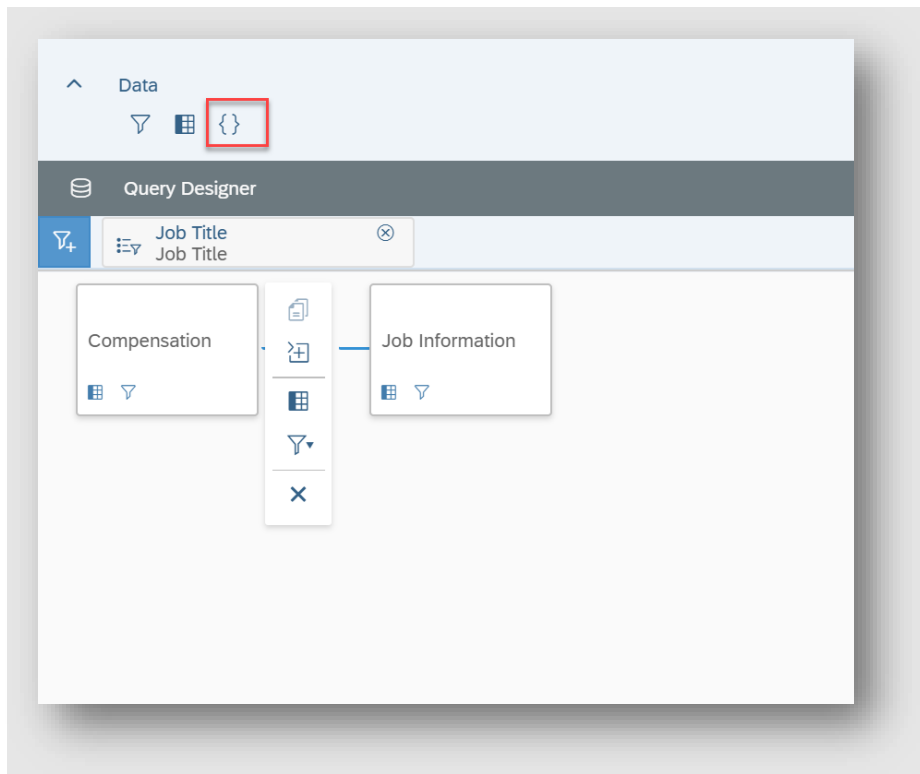


Figure 54: Input Parameter in Story report

2. In the dialog box define parameter name, value type (String, Integer etc.) and default value (Figure 55).

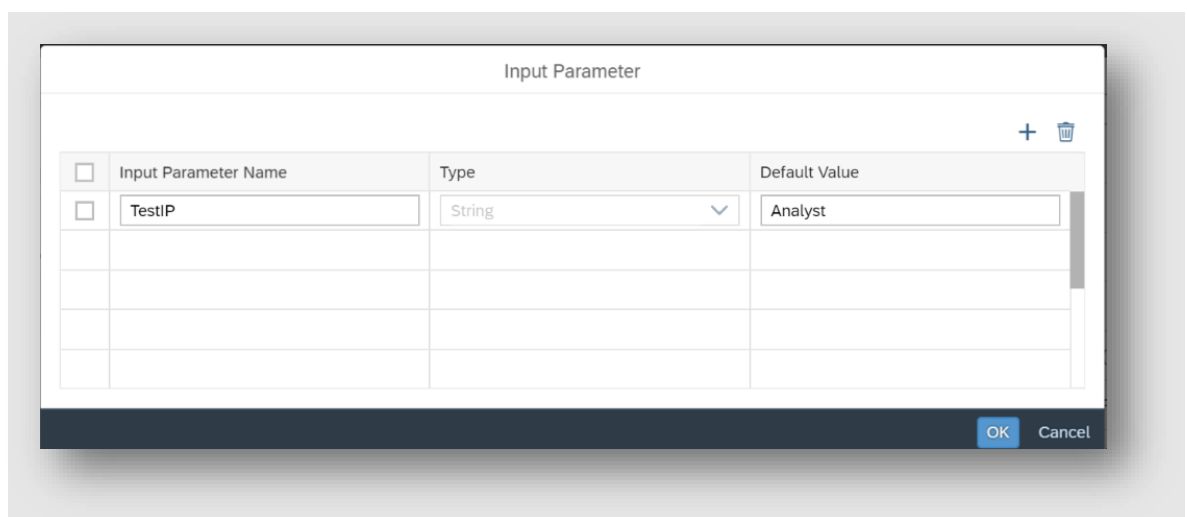


Figure 55: Input Parameter Type in Story report

3. Create advanced filter either on table or query. In the advanced filter dialog, select filter type as Input Parameter. System will list all the input parameters that exist in the query (Figure 56). Select required parameter from the available list.

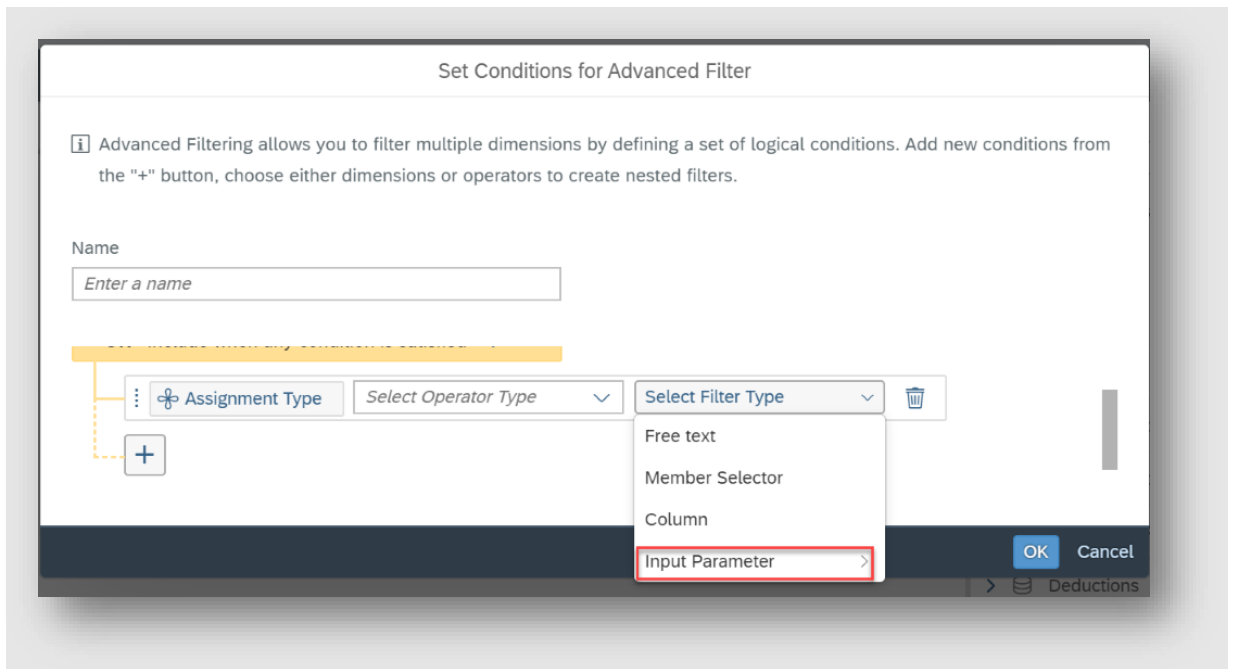


Figure 56: Input Parameter in Advanced Filter

4. User input will be asked when previewing/finishing the query and upon inputting the value the filter will be applied.

Figure 57: User input in Story report

5.2.4 Aggregation Types in Query Designer

Query designer allows the report designer to define the aggregation type that should be applied on the measure when data is aggregated in story designer. To define aggregation type on measures, follow the below steps.

1. Navigate to Column Overview dialog as shown in Figure 58.

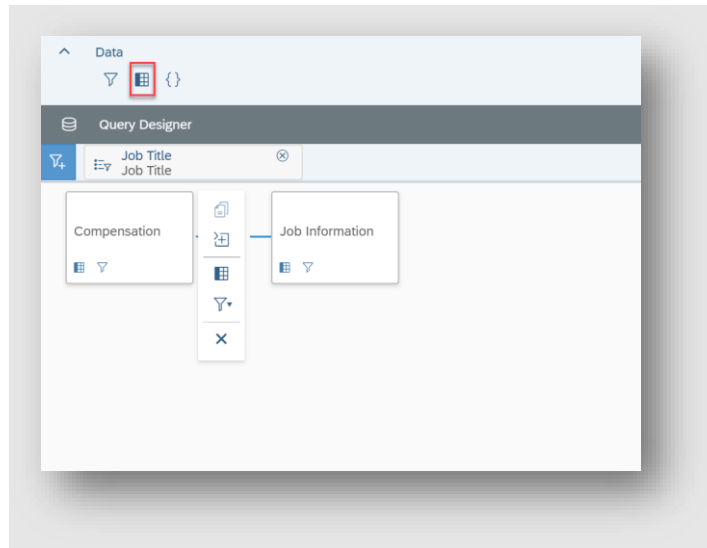


Figure 58: Column Overview in Story report

2. In the Column Overview window, the report designer can define the aggregation type for each of the measures included in the query. It is possible to select aggregation type on user selected measures (columns) and calculated columns (measure) as shown in Figure 59.

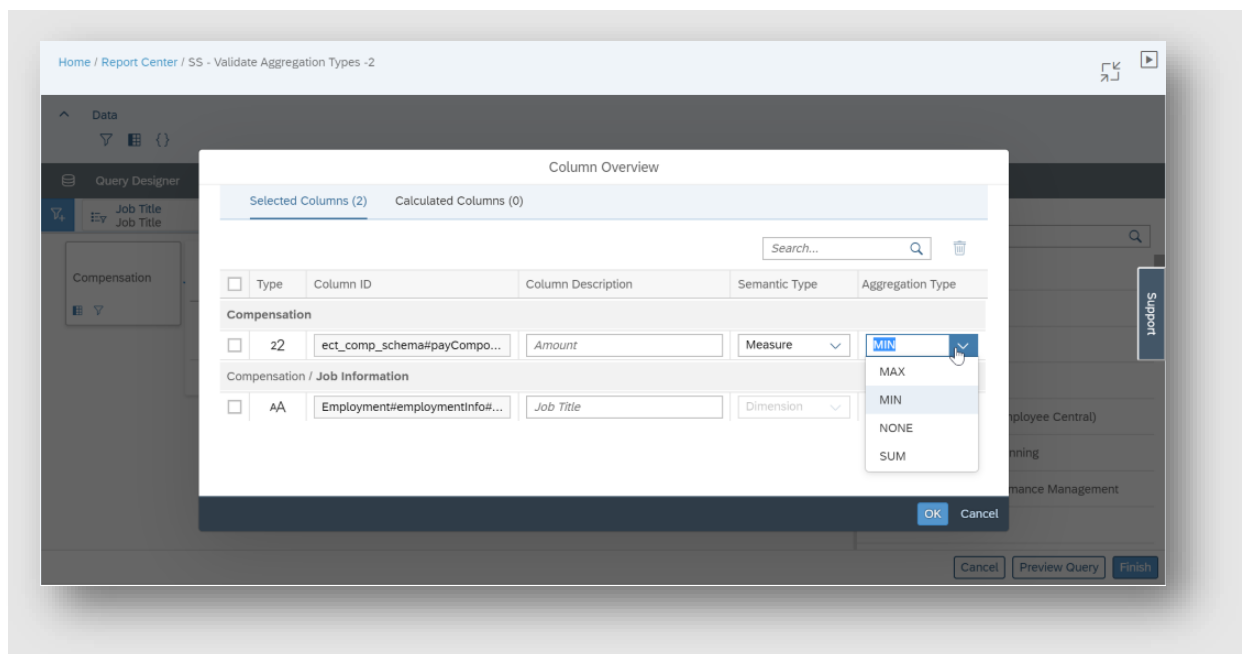


Figure 59: Column Overview and Aggregation Type in Story report

Following are the available aggregation types in Story reports and their meaning.

- **MIN:** When data is aggregated in story designer (users aggregated lists or charts) MIN of measure is considered. E.g. The data source includes Job Title (dimension) and Salary (measure), when an aggregated list is visualized MIN of (Salary) will be shown.
- **MAX:** When data is aggregated in story designer (users aggregated lists or charts) MAX of measure is considered. E.g. The data source includes Job Title (dimension) and Salary (measure), when an aggregated list is visualized MAX of (Salary) will be shown.
- **SUM:** When data is aggregated in story designer (users aggregated lists or charts) SUM of measure is considered. E.g. The data source includes Job Title (dimension) and Salary (measure), when an aggregated list is visualized SUM of (Salary) will be selected.
- **NONE:** Default aggregation (SUM) is considered

5.3 Handling Currency Conversion

The currency exchange rate is not included in the story reports. To convert values into **Euros** for Story reports, two fields were established within the Compensation section. Calculated columns were then employed within the story to perform the actual Euro amount calculations.

First step: To acquire the numerical currency exchange rate, it is essential to incorporate two custom fields within the compensation section and include an onSave rule in the compensation section under Manage Business Configuration.

Technical Configuration:

1st field: Currency Exchange Rate to do the Lookup (custom-string with Currency Exchange Rate Object)

custom-string1 Currency Exchange Rate ⓘ Yes No [Details](#)

1st Field: Currency Exchange Rate

Identifier	custom-string1
Label	Currency Exchange Rate ⓘ
Enabled	Yes
Mandatory	No
Default Label	?
Data Type	STRING ?
Type of Reference Object	Foundation Object
Reference Object	CurrencyExchangeRate

2nd field: Currency Exchange Rate in Number (custom-double)

custom-double1 Exchange Rate in Number ⓘ Yes No [Details](#)

Business rule to set the currency exchange rate numerical value.

Assumption: Each employee has assigned at least either the base salary or the contractual salary component and the rule will trigger on import (see Details). Otherwise, the rule won't work.

Please be aware to adjust the rule according to your specific configuration.

The screenshot shows a business rule configuration window with two main sections: 'If' and 'Else If'.

If Section:

- Condition:** Compensation Information Model.Compensation Information.Compensation.Currency is not equal to EUR. The rule selects one entry from the collection "Compensation" where... Pay Component is equal to Basic Salary (BasicSalary) OR Pay Component is equal to Contractual Salary (ContractualSalary).
- Then:** Set Compensation Information Model.Exchange Rate in Number.Value to be equal to Lookup(Currency Exchange Rate) Select Exchange Rate where... Effective Date is equal to System Context.Effective Date. Currency Exchange Rate.Exchange Rate = Compensation Information Model.Currency Exchange Rate.Value.Exchange Rate.

Else If Section:

- Condition:** Compensation Information Model.Compensation Information.Compensation.Currency is equal to EUR. The rule selects one entry from the collection "Compensation" where... Pay Component is equal to Basic Salary (BasicSalary) OR Pay Component is equal to Contractual Salary (ContractualSalary).
- Then:**
 - Set Compensation Information Model.Currency Exchange Rate.Value to be equal to Null
 - Set Compensation Information Model.Exchange Rate in Number.Value to be equal to 1

Details of Business rule:

The screenshot shows the 'Details' tab for a business rule.

- Base Object:** Compensation Information Model
- Event Type:** onSave
- Rules:** COMP_DEFAULT_EXCHANGERATE (COMP_DEFAULT_EXCHANGERATE) [E]
- Enabled:** Yes
- Rule Contexts:**
 - Edit UI (MSS / ESS): Yes
 - History UI: Yes
 - Imports: Yes
 - Mass Changes: Yes
 - New Hire/Rehire UI: Yes
 - OffCycle Batch Events: Yes
 - Termination UI: Yes
 - Publish from Compensation Planning: Yes
 - Onboarding: Yes
 - Report No-Show UI: Yes
 - Position to jobInfo Sync: Yes
 - API: Yes

Second step: Import Compensation Data with currency exchange rate object

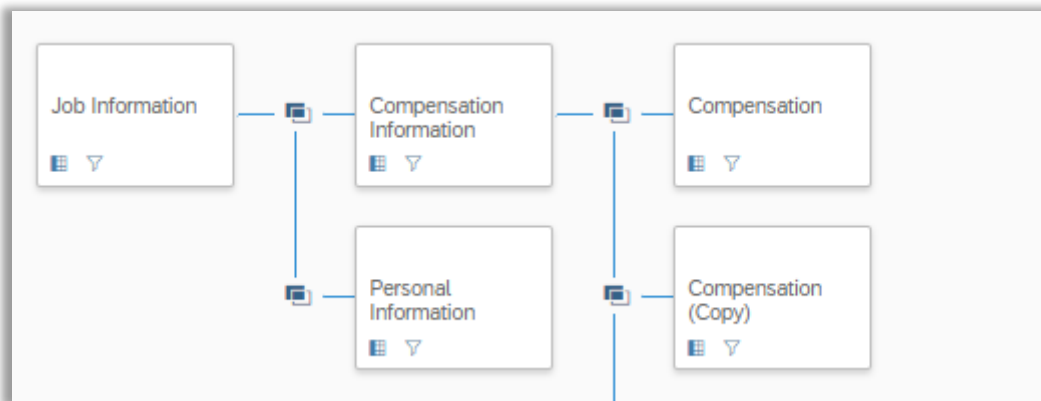
Update the employee record with an import of compensation data to a specific date or update the complete history – depending on your needs. When importing the data the rule will trigger and set the currency exchange rate in number value.

View of Compensation Information

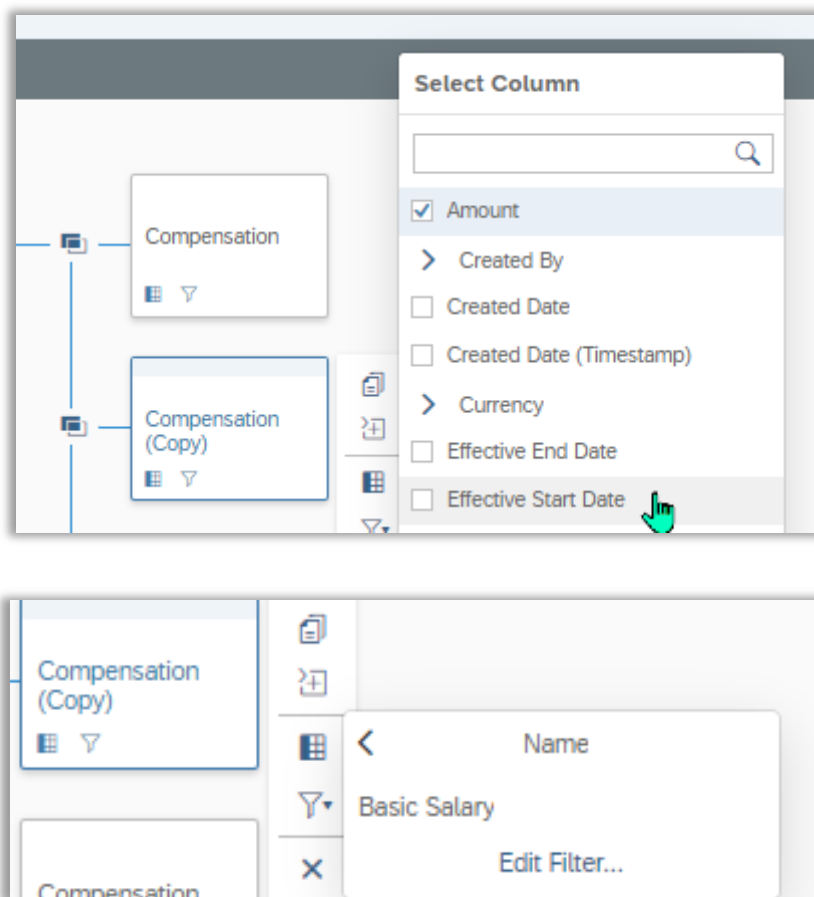
The screenshot shows a compensation information view with the following details:

- Currency Exchange Rate:** DEFAULT-USD-EUR (DEFAULT-USD-EUR)
- Exchange Rate in Number:** 1.4
- Basic Salary (BasicSalary):** 5,000 USD Monthly (MON)

In the story itself you must work with calculated columns. The Query is structured in the following way:



The Compensation (Copy) table refers to a specific pay component by selecting the Amount column and setting a filter. In the figure below, the filter is set for the basic salary pay component. This can be done for each pay component to get the values column by column.



To get the value in Euro we must add calculated columns for each pay component: By dividing the Amount for the related pay component through the Exchange Rate in Number we get the amount in Euro – see screenshot.

```
IF ( [Compensation (Employee Central)#Compensation Information#Exchange Rate in Number] != 0 ,
[Compensation (Employee Central)#Compensation (Copy)#Amount] /
[Compensation (Employee Central)#Compensation Information#Exchange Rate in Number] ,
[Compensation (Employee Central)#Compensation (Copy)#Amount] )
```

In the story itself we must add the respective calculated column for each pay component to display the amount in Euros.

Exchange Rate in Number	Basic Salary in EUR
1.40	3,571.43

For sure this can be done for each currency by following the steps above and adjusting the calculated columns.

5.4 Localization Considerations

In Employee Central, some dimensions are only relevant for specific countries or based on the configuration, the same field could have a different format between two countries.

Employee Central manages those fields in localized (or country specific) tables, for example see Figure 60:

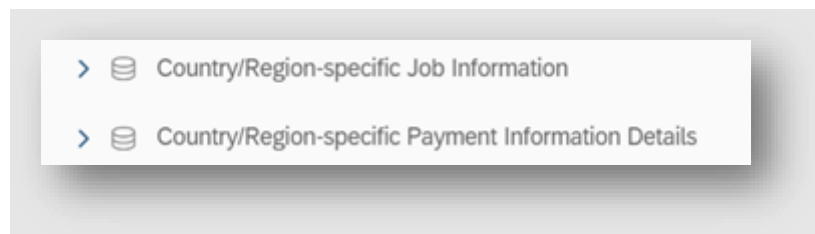


Figure 60: Country specific datasets (tables) in Story report

In order to avoid a specific report per country or having columns for each country for similar fields, it is possible to combine in one report information coming from different localized tables by concatenating similar fields together in the query panel. For example, Contract Type is generally a country specific field, but you would not want your report to have a column for each country just to display the Contract Type.

1. Identify the countries implemented in your Employee Central instance and connect to the localized tables via the root table (keeping an outer join between those tables as shown in Figure 61).

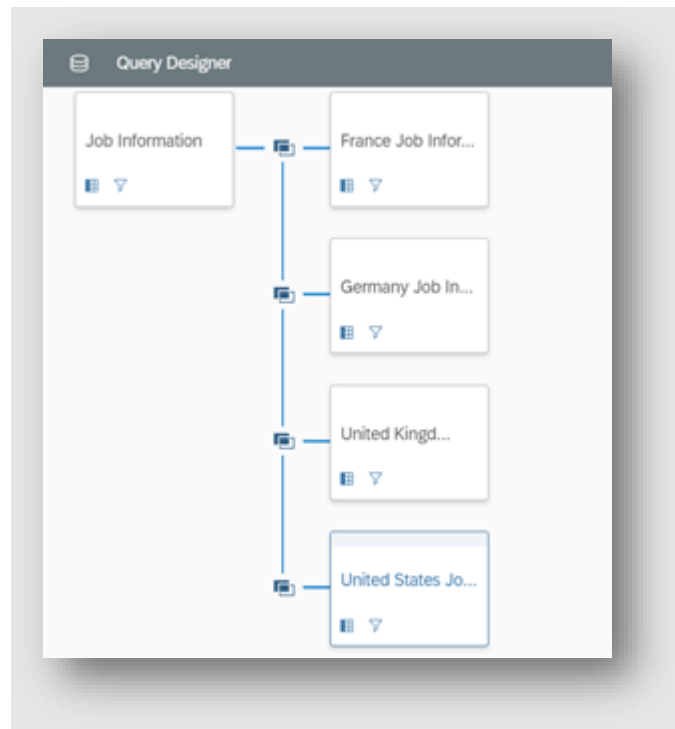


Figure 61: Example of query to support 4 countries with outer join in Story report

2. Rename the objects in the field editor as shown in Figure 62.

The 'Field Editor' window displays a table with the following columns: Type, Column ID, Column Description, Semantic Type, and Aggregation Type. The table contains several rows of data, including a row for 'Country' and rows for 'Contract Type' for each of the four countries (FRA, GER, GBR, USA). The 'Contract Type' rows are grouped under their respective country headers.

Type	Column ID	Column Description	Semantic Type	Aggregation Type
<input type="checkbox"/>	AA	FoundationObjects#location#n...	Country	
Job Information / France Job Information				
<input type="checkbox"/>	AA	Employment#jobInfo_FRA#CO...	Contract Type (FRA)	Dimension
Job Information / Germany Job Information				
<input type="checkbox"/>	AA	Employment#jobInfo_DEU#CO...	Contract Type (GER)	Dimension
Job Information / United Kingdom Job Information				
<input type="checkbox"/>	AA	Employment#jobInfo_GBR#CO...	Contract Type (GBR)	Dimension
Job Information / United States Job Information				
<input type="checkbox"/>	AA	Employment#jobInfo_USA#CO...	Contract Type (USA)	Dimension

At the bottom right of the window are 'OK' and 'Cancel' buttons.

Figure 62: Renaming the objects in Field Editor of Story report

3. Create a calculated column using the CONCAT() function as shown in Figure 63. This works because one employment is only defined in one localized table, meaning each employment can only have the country specific field/table for one country table.

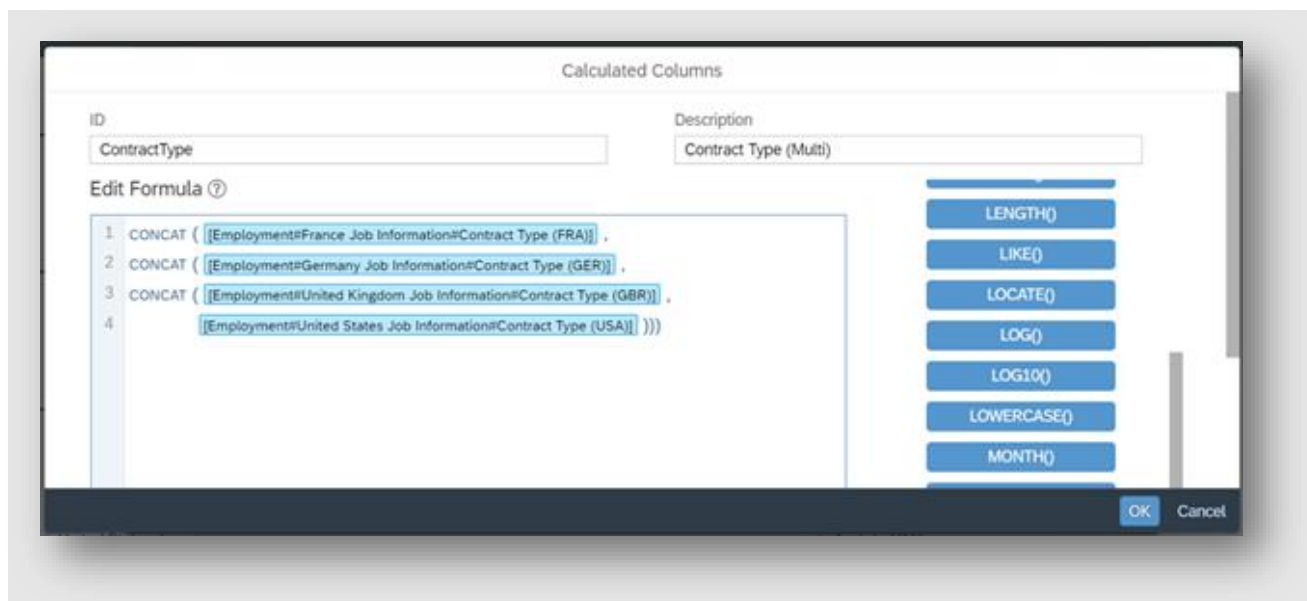


Figure 63: Calculated Columns in Story report

4. Consume the new concatenated field in your Story report as shown below in Figure 64.

User ID	Location	Contract Type (Multi)	Contract Type (FRA)	Contract Type (USA)	Contract Type (GBR)
10007	DEU01 Berlin	orientation	–	orientation	–
1004	Lyon	orientation	orientation	–	–
1017	Marseille	solidarity	solidarity	–	–
1219	Lyon	solidarity	solidarity	–	–
14	–	Fixed Term Contract	Fixed Term Contract	–	–
1511Migration	Loc_FRA	Job-creating Program	Job-creating Program	–	–
1511_ec2mdf	Loc_FRA	Job-creating Program	Job-creating Program	–	–
1605UserA	Paris	Permanent	Permanent	–	–
2201	DEU02 München	apprenticeship	–	apprenticeship	–
45	–	trainee	trainee	–	–
50	FRA010002	orientation	orientation	–	–

Figure 64: Story report output using calculated columns for different countries

5.5 Guardrails

Below are expected actions from the user on hitting threshold while using Story report artifacts:

Scenario	Expected Actions
----------	------------------

Maximum parallel query executions per customer instance (Company ID)	Currently there is a restriction that no more than 100 queries can be executed in parallel in a customer instance. Upon reaching the limit, report execution will be blocked. Users will be required to wait a couple of minutes prior to attempting to rerun the report so that parallel running queries are less than 100.
Maximum cell (rows * columns) limit per query	There is an upper limit of 1 million cells that a query can retrieve. If the number of cells in a report exceeds the limit, users can use the following options to limit the data retrieved in the query: <ol style="list-style-type: none"> 1. Use a different permission model to limit the data 2. Use static and advanced filters 3. Modify target group of the report using scope filter
Maximum number of columns allowed in a story	There is an upper limit of 120 columns than can be selected in a query. On crossing the limit users will be warned in query designer. Users are advised to limit the number of columns in a query to less than 120 for optimal performance.
Maximum number of tables allowed in a story	There is an upper limit of 30 tables than can be joined in a query. Users are advised to limit the number of tables in a query to less than 30 tables.

5.6 Data Analyzer

The Explorer mode isn't supported in optimized view mode. This can be replaced by Data Analyzer in Optimized Design Experience.

5.7 Do's & Troubleshooting Tips of Report Center

5.7.1 Do's

5.7.1.1 Use Labels in Report Center

Adding labels and assigning them to reports in Report Center allows you to group the reports in a folder-like structure that you can access under the Labels tab of the Report Center page.

- Go to Report Center, select the report you would like to create/add a label to and click on *Label As* under the action dropdown as shown in Figure 67 below.

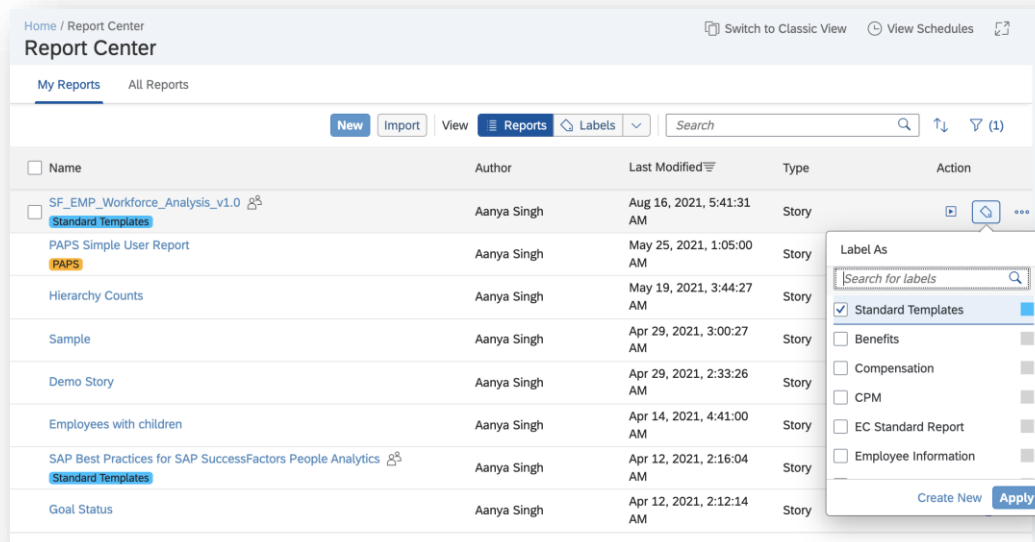


Figure 67: Label reports in Report Center

- The *Manage Label* dialog should then open as shown in Figure 68. From this menu you can name a label, nest a label inside another, set the label to public/private (only reporting admins can create a public label, or convert a private label into a public label) or select the color of a label.

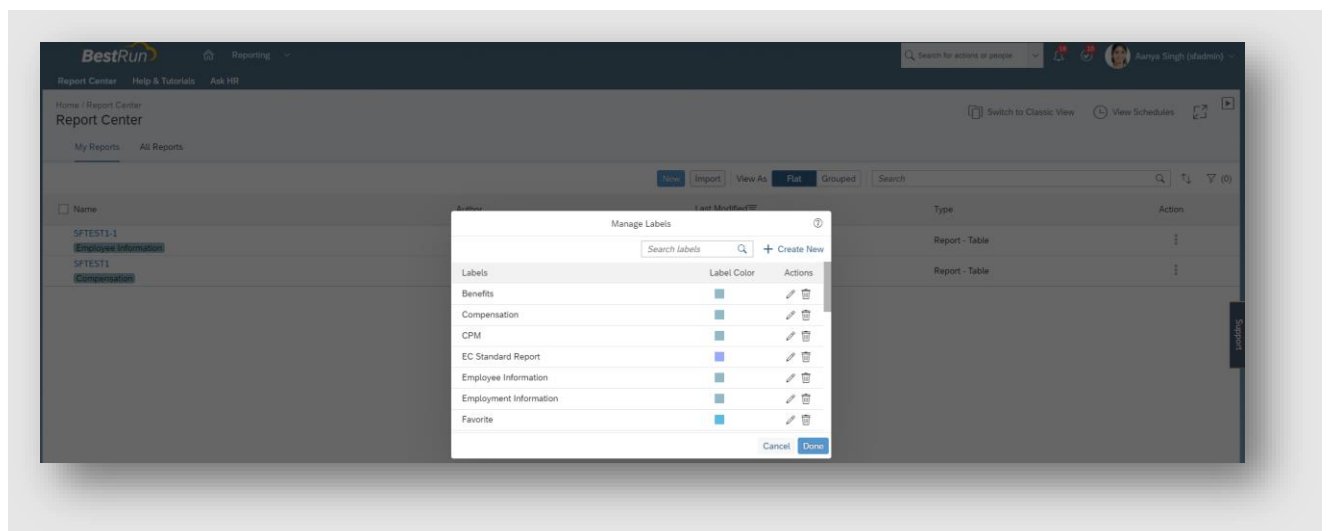


Figure 68: Manage Labels in Report Center

Note: Labels are not case-sensitive. For example, the labels *Retail* and *retail* are considered as one label. You cannot create more than one public label with the same name. Private labels with the same name can exist in a tenant, provided that the labels are not created by the same user.

5.7.1.2 Import and Export of Story Reports

The current best practice is to first build story reports in your test instance and then move them to the production instance once completed using the Import and Export option in Report Center.

- Using the *Export* option on the Action menu, or the Export icon on the Report Center menu bar, you can export one or more story reports and save them as an ACN file on your local drive.

- Using the *Import* button on the Report Center menu bar, you can import story reports into your instance, using an ACN file.

The ACN file format is used for storing story reports, and is native to the application. You can import stories if you have the *Create Report Story* permission. You can export stories if any of the following conditions are met:

- You are the author of the story
- A story has been shared with you, and you have the permissions to edit the story
- You have been assigned the *Report - Story Admin* permission

5.7.1.3 Schedule Reports

For non EC customers to ensure you can schedule reports in Report Center, you must select *Enable Generic Objects* — requires “Enable the Attachment Manager” in company settings of provisioning as shown in Figure 69 below.

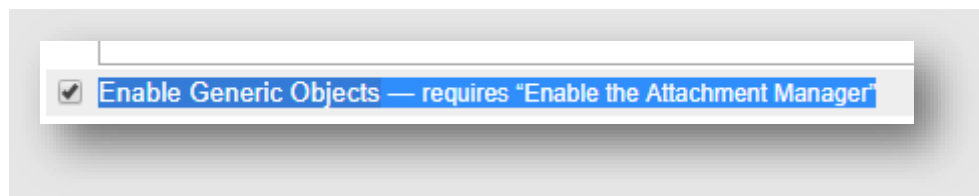


Figure 69: Provisioning settings of Company Settings to schedule reports

Once the feature is enabled from provisioning the *New Schedule* option is available and can be used for report scheduling. If you do not have access to partner provisioning reach out to SAP Cloud Support.

5.7.1.4 Exporting Reports Permissions

Below are the different permissions needed to export various kinds of report definitions from Report Center:

- To export a table report, you must be the author of the report and the *Create Report* permission must be enabled for you.
- To export a Canvas report, the *Report Canvas Designer Admin* permission must be enabled for you or you must be the author of the report and the Report Canvas Designer permission must be enabled for you.
- To export Tiles and Dashboards, the *Analytics Tiles and Dashboards* permission must be enabled for you.
- To export a story report, you must be able to edit the report. This is shown in Figure 70.

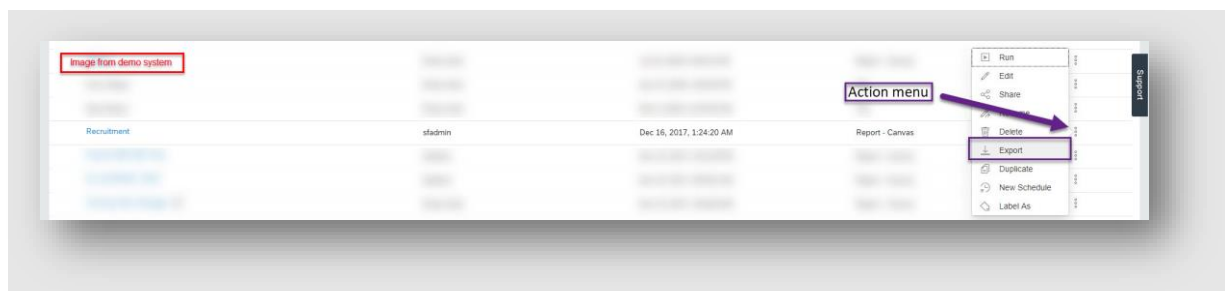


Figure 70: Export in action menu of Report Center

5.7.1.5 Include Quick link for Report Center

Quick links can be added to the Home Page in the Welcome Portlet, the Company Resources Page within the SAP SuccessFactors Application. For example Deeplinks: (from my Internal Sales Demo environment).

- **SP-Initiated Login**
URL: [https://salesdemo.successfactors.eu/sf/inform?landingPage=ADVANCEDREPORTING\(\)&company=XXXXXX&loginMethod=SSO](https://salesdemo.successfactors.eu/sf/inform?landingPage=ADVANCEDREPORTING()&company=XXXXXX&loginMethod=SSO)
- **Replace** 'salesdemo.successfactors.eu' with your own Datacenter domain and 'XXXXXX' with your specific Company ID

5.7.1.6 Sharing Reports in Report Center

You can assign the *Share Reports to Groups & Roles* report permission as shown in Figure 72 with the following steps:

- From Admin Center - Manage Permission Roles then select the desired role from the list
- Select Permission then Manage Dashboards/Reports
- Make sure the Share Reports to Groups & Roles option is selected for those roles with users who you wish to allow the ability to share reports to other users/groups/roles

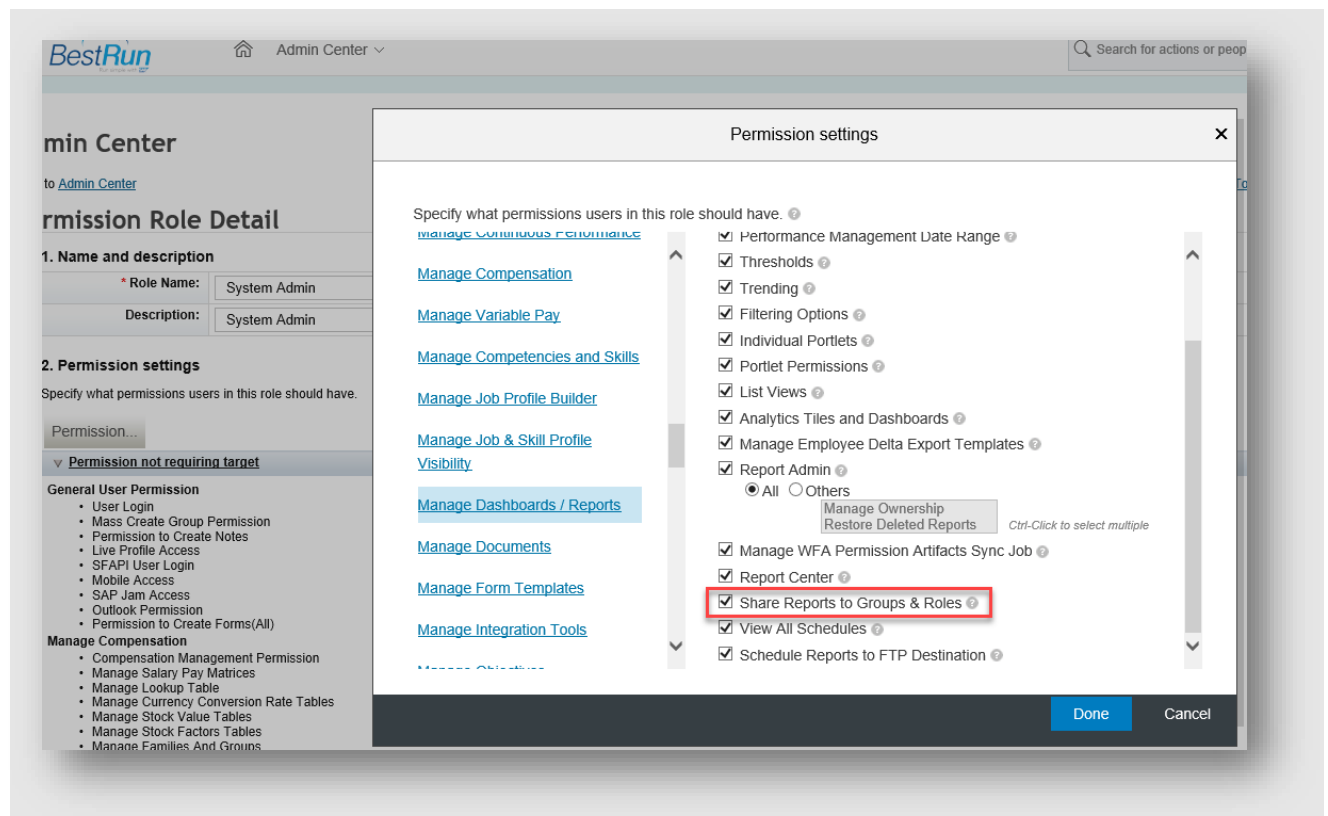


Figure 72: Share Reports to Groups & Roles permission settings

- The Groups and Role option will then be available when you select the share report function via Report Center as shown in Figure 73.

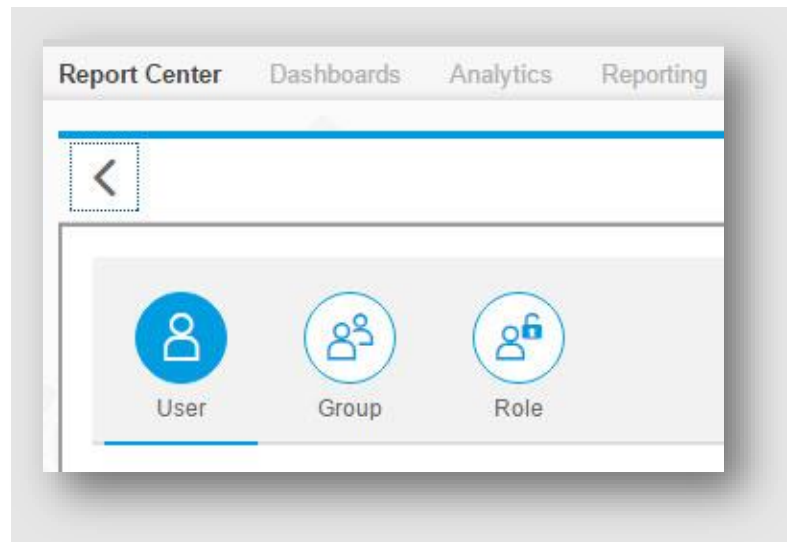


Figure 73: Share reports to User, Group or a Role

5.7.2 Troubleshooting Tips

5.7.2.1 Reports are Missing

Reports are missing in report center for certain users and is caused due to applied filters as shown in Figure 74.

- Edit the filters by clicking the filter icon
- Either add the desired report types (top left) or clear the filter all together (top right)
- Click 'Go'

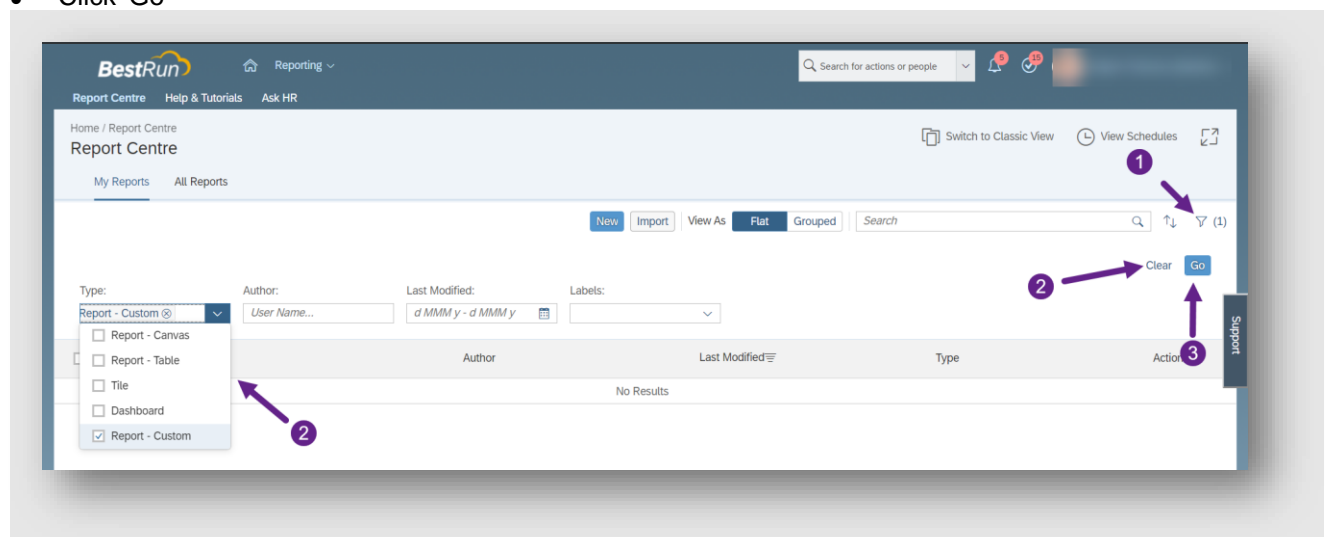


Figure 74: Filter option in Report Center

5.7.2.2 Search Bar is Missing in Report Center

The Search functionality for the Report Center is disabled in Admin Center

- You can check if in Admin Center > Reporting & Analytics > Manage Report Center the *Enable Report Center Search Functionality* is switched on.
- If not, enable it and then click Save. When you navigate back to Report Center, you should be able to see the search bar.

Notes:

- The switch for search functionality is usually enabled along with *Use SOLR to load Report Center*
- If you cannot find the *Manage Report Center* link in Admin Center, you will have to permission it to the RBP role from Manage Permission Roles > Administrator Permissions > Manage Dashboard/Reports > Report Center

5.7.2.3 Errors Fetching Reports in Report Center

While accessing Report Center sometimes an error like „Error occurred while fetching reports!“ is displayed as shown in Figure 75.

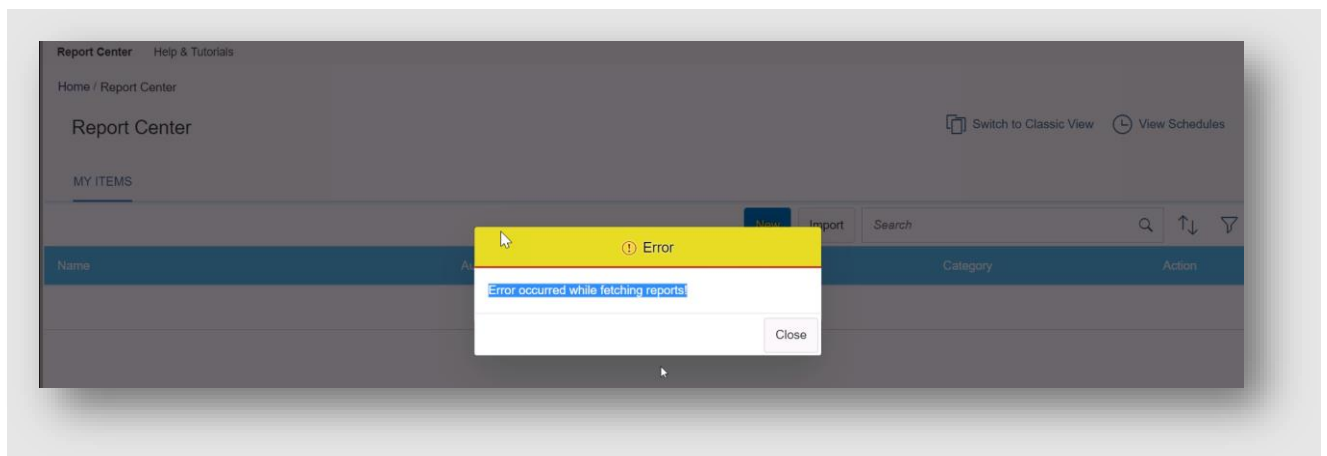


Figure 75: Error in Report Center

- Go to Upgrade Center
- Click on View Recently Completed Upgrades
- Search for Report Center
- Click on Undo
- Log out, wait for a few minutes, clear the cache and log in SuccessFactors
- Go to Upgrade Center > Enable the Report Center upgrade again (this will upgrade the instance with the proper configuration)
- Logout from the instance and clear your browser cache
- Wait for few minutes and open Report Center and check if the reports are appearing in the report listing page without any error

5.7.2.4 Missing Reports in Report Center

Follow the steps below to check and take action for missing reports in Report Center.

- Select Admin Center
- Click on Reporting & Analytics icon & Select "Sync Report Center Search" (Figure 76)

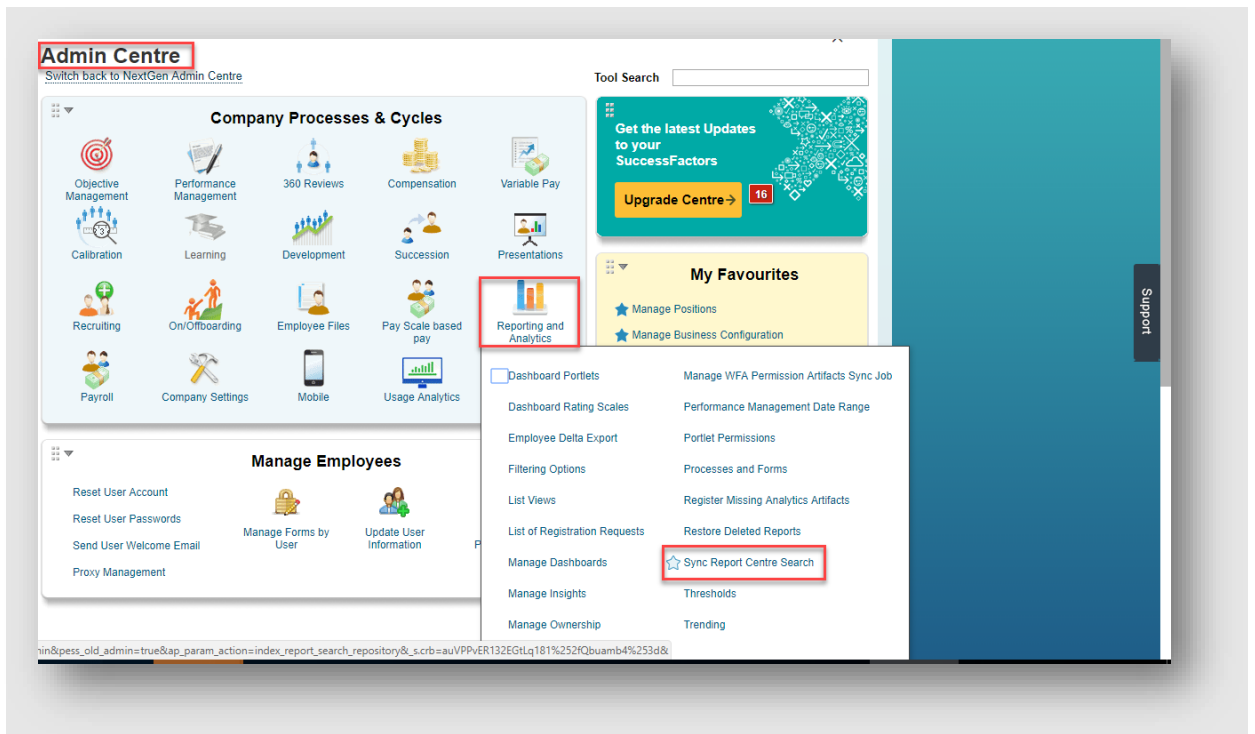


Figure 76: Sync Report Center Search

- Enter a Job Name
- Select a Job Priority - Job Priority is used by the Job scheduler to determine the position of the job in the queue
- Select the Job Configuration for the type of reports that are missing
- Select a date range
- Click on Start Sync (Figure 77)

The screenshot shows the 'Sync Report Centre Search' configuration page. The page has a header 'Admin Centre' and a sub-header 'Sync Report Centre Search'. Below the header, there is a note: 'Syncing between the Report Centre and their original tools should be automatic, however if you notice anything is out of sync, please use this tool to correct it. Please note: syncing could take some time, if you have any issues, please contact Support to check the sync status'. The 'Configure Job Parameters' section contains the following fields:

- Job Name:
- Job Priority:
- Job Configuration: ☒ Report - Table, ☒ Tiles and Dashboards, ☒ Report - Canvas
- Date Range:

At the bottom, there are two buttons: 'Reset' and 'Start Sync'.

Figure 77: Configuration in Sync Report Center Search

- Depending on the workload involved and priority selected it may take up to 24h for the job to be completed.
- You can check the progress of the Sync Job only in provisioning via *Monitor Jobs* and *Index Report Search Repository* job type.

5.7.2.5 Error While Deleting Dashboards from the Report Center

- When attempting to delete a Dashboard, the following error appears "An error occurred while deleting the report" as shown in Figure 78.

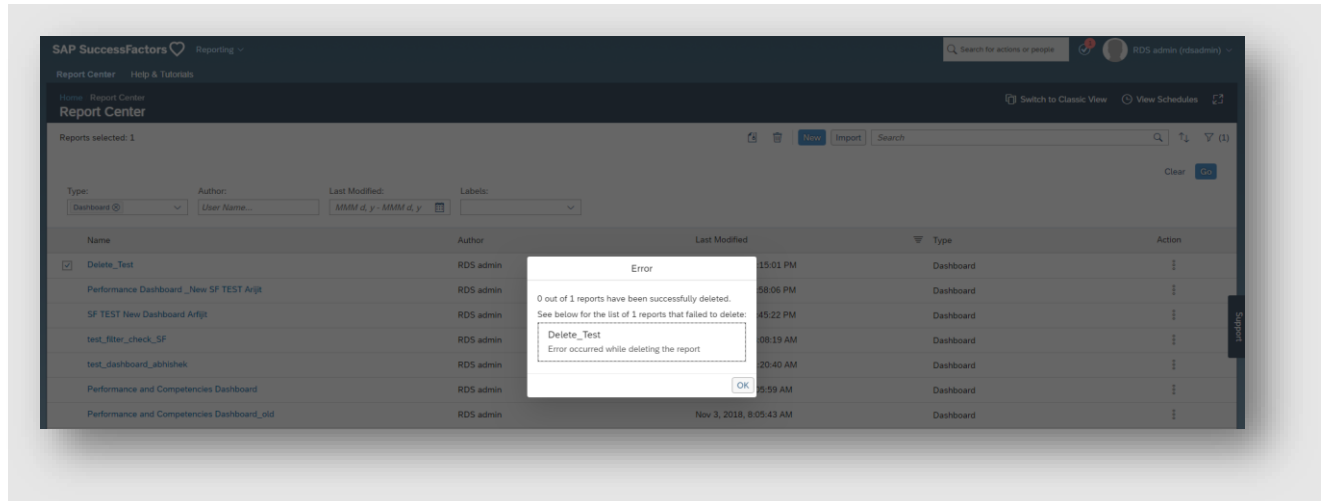


Figure 78: Error while deleting Dashboard from Report Center

- Navigate to Admin Center > Manage Dashboards > Manage Tile-Based Dashboards
- Identify the Dashboard you need to delete then select the Delete Icon and confirm
- The Dashboard will not be displayed in the Report Center anymore

5.7.2.6 FTP error login has no write permission

FTP Test Connection in Report Center is showing the error "login has no write permission" as seen in Figure 79 in Report Center, Schedule Job Provisioning and Report Distributor.

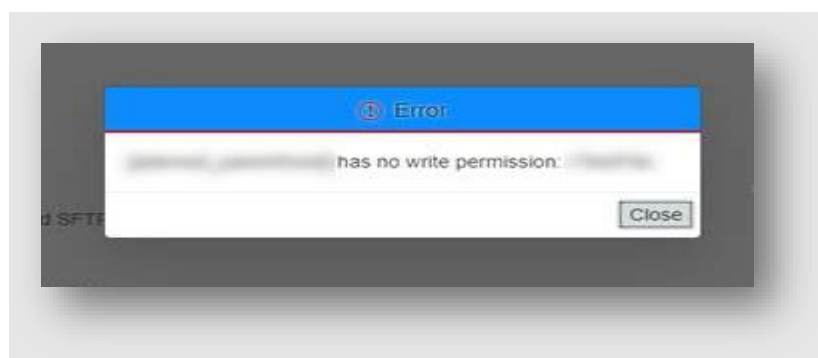


Figure 79: FTP Test Connection

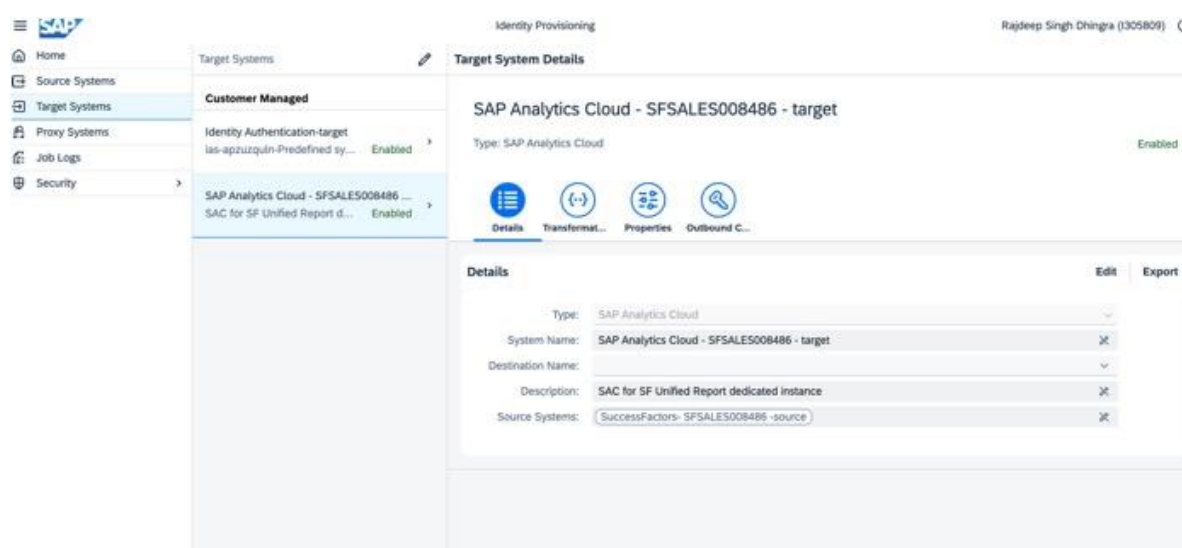
- Scenario 1: The SAP SuccessFactors outgoing Public IP address is blocked by your environment. You need to whitelist the SAP SuccessFactors outgoing Public IP address to allow the connection into your environment.
- Scenario 2: The login user does not have permission to the FTP root folder
 - You can configure the FTP path to "/incoming"
 - Click on Test File Put Permission
 - It will be successful

4. The Test Connection button may still not work. However, when you run the scheduled job it will work.

See KBA [2580876 - FTP error login has no write permission](#) for further information.

5.7.2.7 User Sync Issues

- Avoid configuring multiple sources or target in IPS system as it may lead to user sync issues. Recommendation is to use the dedicated IPS system to sync users between BizX and SAC systems. Please note, the supported architecture is 1:1 i.e. One BizX has a dedicated IAS, IPS and SAC. Using IPS system of Dev system to sync users from BizX Prod (or any such combination where one IPS is used to sync users between multiple source or target) may lead to user sync issues.
- Recommendations is to have only one source system selected in SAC as Target. Leaving it blank or selecting multiple can lead to issues with user sync process.



IPS configuration with SuccessFactors Source System

6 BEST PRACTICES FOR PERFORMANCE

SuccessFactors People Analytics Story reports leverage SAP Analytics Cloud that enables you to perform sophisticated analysis on live HR data. These complex scenarios can sometimes lead to less than ideal performance times for end-users. The following section aims to help users in best practices and guidelines that you can keep in mind when designing models and stories that will help Story reports run at optimal performance levels.

The section is divided into two parts: General-Purpose Guidelines & Complex Scenario Guidelines as explained below.

6.1 General Purpose Guidelines

6.1.1 General

- Always use Google Chrome, recommended browser, to take advantage of the latest performance improvements.
 - Take advantage of improved performance with browser caching of story and Report Center. As long as no structural changes are made in the story, cache is still valid. It becomes important for Stories with multiple pages, charts or models to make use of browser cache. This is applicable to Chrome users in non-incognito model only.
- Report Center first login may seem slower, but subsequent operations with caching enabled would lead to faster login and access to Report Center.
- Label View helps you find your data much faster especially if you work with a large number of reports on day-to-day basis

6.1.2 Modelling and Data

Query design has some of the most significant impact on performance and hence queries must be designed judiciously.

- Stories leverage auto suggested joins for the schemas when you drag and drop the schemas for common schema elements making it easier for users to not worry about remembering how to join the data yet also considering the performance while design.
- In case you need to join tables from different modules, try to keep all the intra-schema table joins and then go for inter-schema table joins unless design needs to consider otherwise. Try to keep the number of schemas in a query to ten or less unless really needed. Each join between schemas/tables adds complexity. Depending on the join chosen automatically by subdomain schema (Left Outer/ Inner join) we may look at the complexity of $O(m*n)$ in few cases where m and n are sizes of the tables in a single join and O here represents Big O notation (Big O notation describes the limiting behavior of a function when the argument leans towards a particular value or infinity). Hence, we should be cautious of introducing the number of joins.
- Linked dimensions in a story increase time to load much more than inter query joins used in the query design.
- Being a reporting tool, Stories, suggests complex hierarchies should be built as custom MDF objects exposed for reporting instead of calculated columns.
- Refer to Data Source Design section for complex modelling requirements covered later in part 2.

6.1.3 Story Design

- Instead of creating a Story with many pages, you can limit your pages and create different stories for each use-case or audience which serves the viewers better.
- Try to keep the number of individual widgets on each page of your story limited to six or less.
- Users can club/stack measures for some chart types. Eg:- Numeric Point chart type can be used to club and show more than one measure in a single widget instead of having multiple numeric point chart widgets. This also ensures faster loading by reducing number of client-server calls and hence improving performance without reducing total measures compared to earlier.
- Limit the number of content-rich widgets like tree maps or charts containing a high volume of data points on each page. It will increase load-time and causes over loading issues due to which it will be harder for the viewer to consume. If need be, apply top N filter.
 - Tip: Use different pages on a Story based to capture data based on the type of information or category, with first page being made available for your most viewed content that is easily accessible.
- Nesting of calculated columns should be avoided where possible. Instead, wherever feasible calculated columns should be built over source columns directly instead of other calculated columns. Nesting impacts performance due to the sequential nature of processing instead of parallel processing. For example, Building Calculated Dimension "Generation" (Group of people born at the same time) with values GenX, GenY, Gen Z and others would be built

better if directly calculated over “date of birth” instead of calculating over “Age” which in turn is calculated over “date of birth” (sequential).

- Ensure that the images are smaller than 1MB and are sized for web.
 - Tip: Image files of SVG vector are great without compromising the quality even though they are a small file size. For any reason, you can't make use of SVG image file type then PNG image files perform better than JPG.
- Avoid creating linked dimensions on Calculated Dimension when working with blended data and also keep the number of models linked to a minimum to improve the overall performance.
- Story performance can be impacted when your story has a lot of widgets on a page, particularly data-related (process-heavy) widgets. For example, you have several charts, tables, or filters on a page. Processing data from those widgets can slow down the story's performance.
 - To determine the best performance for a story, the widgets are assigned a unit weight, with the maximum ideal weight per page set to 5 units.
 - The following table shows the unit weight per widget:

Type of Widget	Unit Weight
Table	0.6
Chart	0.3
Filters, expanded	0.4
Filters, non-expanded	0.2
Section on page	0.5
Text box with dynamic text	0.2
Dynamic Image	0.3
Text Box	0.1
Image/ Shape	0.1

6.1.4 Charts and Tables

- In your Stories, reduce the number of individual data points to make the information readable and easier to analyze.
- When adding a table, try to limit your tables to a maximum of 500 rows and 60 columns to ensure readability and easy navigation. You can edit the drill limitation if you have to display more data. Increase in the number of cells means the Story must work harder to display.
 - **Tip:** Using Chart filters reduces the volume of information when there is more than 1000+ data points. Where applicable use table filters to keep a manageable amount of information visible on the tables.
- It is recommended to avoid Formatting Rules for tables with large dataset
- Some examples of widgets which deal with large amount of data and may see slower load times are:
 - Table widgets with data filter of million cell values
 - Non-aggregated widgets trying to load nested calculated columns as this leads to cascading of computation and waits for previous query to compute rather than processing queries in parallel
 - Non-aggregated widgets with non-time dimension that may have thousands of unique values are not a good fit and should be filtered with Top-N value filter

6.1.5 Filters

- Query filters help in reducing the amount of data stored and accessed in the model. If a filter is applicable on whole data and is non-configurable, try putting it in query rather than inside story.
- Limit the number of facts in your story to one million or less. It has been set purposefully to help you work with specific meaningful information rather than high volumes.
- In case you need only limited values that should be selected from a filter, try to limit that during filter value selection.

- Try adding [story filter](#) capabilities instead of individual filters for pages, charts, or tables for a Story that has different elements for displaying the same information. The exception to this recommendation is for filtering on information from models that contain linked dimensions. In these cases, add a filter to the individual tile when filtering on dimensions not used for linking.
 - **Tip:** Page filters (eg. Product below) take up less space in your story and are easier to work with than the expanded page filters (eg. Location below) .

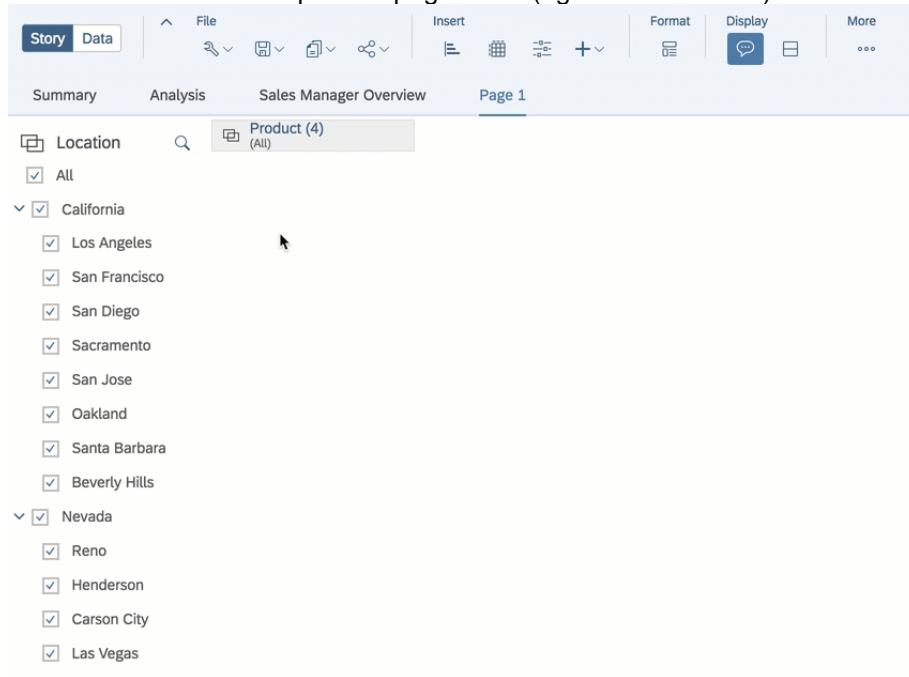


Figure 80: Page filters as tokens and as expanded

- Select specific members rather than selecting “All Members” when setting up a filter. Filters with fewer members are easier to use during analysis and when viewing a story.
- Limit the number of members in a dimension used in Member Selectors
- Limit the filters to the minimum possible number of Members.
- Leave the input controls collapsed for improved performance on a page.
- For the collapsed Story, page or topic filters will be delayed until the user selects the filter. Therefore, greater priority is given to displaying data on expanded input controls, table and charts.
- If you plan to select multiple filter values for a story filter on single dimension, you should select “Confirm Selection on Apply” instead of “Confirm Selection on Select” (Default behaviour) in the Story details settings in the File Menu for runtime performance benefits.

6.1.6 Story Maintenance

- Any unused pages in your Story has to be removed to help streamline the information and improve navigation and performance for the users.

Please refer to [this blog](#) for more details.

6.2 Complex Scenario Guidelines

Businesses come in different shapes and sizes. As businesses grow, their reporting needs grow as well. Businesses that are on the mature side of reporting processes have more complex scenarios and expectations from People Analytics. In the last few releases, SuccessFactors has taken significant steps to simplify and address the complex needs of customers.

This section refers to a few of the best design practices users should follow when creating complex reports which require larger number of columns sourced from a larger number of tables and split

across multiple data sources. Whenever customers start designing such a query, they can refer to some of the design principles mentioned below.

6.2.1 Data Source Design for Complex Scenarios

There might be scenarios where users would have to create and add multiple data sources to the query. This would mean categorizing and assigning fields to different data sources. If done haphazardly this can create performance issues.

When we need to split a query into two or more data-sources, designers need to be wary of the following design principles and should use the following framework in defining and selecting fields that go into each of the data sources.

6.2.1.1 Guardrails

The current Guardrails stand at 120 columns per data source which were relaxed post b2105 release. There are other guidelines and limitations to be considered. Do refer to [KBA-3025688](#) for further understanding.

The previous section on Guardrails covers other guardrails, but for story design we will focus on Column Level Query Design Guardrails

6.2.1.2 Principles

- Aim for limited tables and joins in each data source. Joins increase performance impact and hence should be used sparingly.
- Aim to having **one story widget map to one data source** and not do browser blending of data sources unless necessary.
- In case you need to use to join tables from different modules, try to keep all the intra-schema table joins and then go for inter-schema table joins unless design needs to consider otherwise.
- The field selection for data sources should aim to be **Mutually Exclusive and Collectively Exhaustive (MECE)** for maximum performance benefit (except common fields to be present across data sources, e.g. User ID)
 - **Mutually Exclusive** - Items can only fit into one category at a time (a field can fit in only one data source: Exception Primary Key and Common Field)
 - **Collectively Exhaustive** – All items can fit into one of the categories. (All fields fit into number of the data sources we define, and no field is left out)
- Criteria for grouping tables in a query should be based on the splitting into smaller functional groups by domain while keeping the 120-column limit in mind and for better performance, one should aim for maximizing the fields that come from the same table and then same schema unless unavoidable.
- In case customers try to recklessly create complex reports by adding more and more data sources, report loading may time out and in extreme cases may even lead to browser crashing. Hence, customers are requested to use the data sources sparingly and only where needed with the below framework in mind.

6.2.1.3 Framework

- Identify the widgets that you would be using for communicating the message you would want as part of reporting needs.

- Once widgets are identified, define what measures and dimensions are required for each of the widgets. For table widgets, grouping needs to be done based on use case. For example, Compensation Manager needs to review the salary and all its component for India based employees on CXO level band. This would let's say translate to 30 fields (25 standard fields, 5 calculated fields)
 - What if, instead of 30 in the above example we have 125, you find that it is going to be more than 120 fields (current Guardrail limit). We have two options:
 - We need to rework the table widget design to split into much smaller functional grouping. In the above example of Compensation, had we more than 120 columns of data, we would then use 2 table widgets: one for fixed component and one for variable components. The functional grouping of fields needs to be defined by Domain /Module expert. Rather than having everything in the same table widget, we can place it in two or more table widgets. Here MECE principle should be utilized. The suggested limit here is 120 columns per data sources in a query. Please refer to KBA [3025688](#) for more details on this. Beyond that it might lead to performance degradation in the model.
 - Another alternative is to create calculated fields at the story level rather than in query designer. This can be used if your use case has combined standard and calculated fields greater than 120
- Identify reporting schemas and tables where the measures and dimensions are sourced from
 - Let's say in the above compensation example the 30 fields for salary come from 4 tables in 3 schemas (2 Tables from Compensation and Payroll - Salary data, 1 from Employment - Job Information, 1 from User - Personal Information)
 - In case we have fields exceeding 30, we need to break down the widget's field grouping into further sub-grouping as covered in 2.a.1 point and here we aim to keep table selection among fields to be Mutually Exclusive and Collectively Exhaustive.

Let's take an example below to demonstrate.

Assume for the compensation example mentioned above we have now 50 fields identified that we want to include in a table widget and the guardrail limit is 30. The 50 fields would be as below.

- **Field 1-3** - User ID, First Name, Last Name to be sourced from User schema.
- **Field 4-5** - Job Title, Hire Date from Job Information, Employment Schema
- **Field 6-35** – Salary Fields like Basic, Daily Allowance, Bonus, Benefits etc. from Compensation and VarPay Schemas
- **Field 35-50** – Calculated columns over Fields 6-35. Mapping table shared below.

Due to the guardrail limitation of 30 in this example, data source cannot have more than 30 fields and we have more columns than the guardrail limit (50>30) [b2105 column guardrails are relaxed to 120 and not to be confused with this example]

- Assigning fields to a data source and splitting data sources.
 - Now, we can split the table widget into 2 data sources (assuming we can fit these 50 into 2 data sources with 30 limit each)
 - The next step is subgrouping into 2 smaller groups by domain expert. This can be done in multiple ways depending on the discretion of the story designer who assumingly is also a domain expert:
 - Splitting fields 6-50 between Fixed Salary Components to Variable Pay Components (we will be using this in below example)
 - Splitting fields 6-50 between Compensation Schema Fields and VarPay schema fields
 - Splitting between monthly components to annual components

- Splitting between cash and non-cash components
- While doing this we need to make sure source fields and calculated columns over those source fields are in the same data source and other fields are in separate data source (refer to the tables below)
- In case we have a scenario where calculated field needs to come from both sources, e.g. termination rate has terminations coming from one data source and headcount coming from another, the only option is to have a linked model in the story that joins both data source across the primary key (e.g. user id) and create a calculated column over it. This is to be done in very rare case as some of these can be too performance intensive and lead to story timeout. Such stories are to be created cautiously as joining data sources on browser (browser-based blending) is very performance expensive and is not recommend it unless really necessary. In some cases, it may even be disabled.
- Common fields – Field 1-5 would be present in both data sources leaving 25 fields available for each data source. This is to minimize linked dimension needs unless necessary and ensure we need not create linked data source. This also ensures we do one to one mapping of data source to table widget rather than one to many mapping of table widgets to data sources.
- Next, we choose standard and calculated fields grouping meeting the limit of 25 available fields in each data source and allocate accordingly. In the below example we split the 45 fields 6-50 among the 2 data sources with one having 24 and other having 21 fields. Fields that were not part of any mapping were chosen to be kept where they are needed in data source or one where we still have not reached the 30-column limit.
- Criteria for grouping should be based on the splitting into smaller functional group keeping the 30-column limit in mind and for better performance, one should aim for maximizing the fields that come from same table and then same schema unless not avoidable.

Calculated Fields to Standard Field and Data Source Mapping			
Calculated Field	Source Standard Field	Functional Group	Data Source Selection
Field 36	Field 6,7	Fixed	1
Field 37	Field 8,9,10	Fixed	1
Field 38	Field 11,12	Fixed	1
Field 39	Field 14	Fixed	1
Field 40	Field 15	Fixed	1
Field 41	Field 16,17,18,19	Fixed	1
Field 42	Field 20	Fixed	1
Field 43	Field 21	Fixed	1
Field 44	Field 22	Variable	2
Field 45	Field 23	Variable	2
Field 46	Field 24	Variable	2
Field 47	Field 25	Variable	2
Field 48	Field 26	Variable	2
Field 49	Field 27,28,29	Variable	2
Field 50	Field 30,31	Variable	2
-	Field 32,33,34,35	Variable	2
-	Field 1,2,3,4,5	Common	1,2

Original data source design vs new data source design

	Original Data Source with 50 fields		Data Source 1 with 29 columns		Data Source 2 with 26 columns
Common Fields	Field 1	Common Fields	Field 1	Common Fields	Field 1
	Field 2		Field 2		Field 2
	Field 3		Field 3		Field 3
	Field 4		Field 4		Field 4
	Field 5		Field 5		Field 5
Standard Fields	Field 6	Standard Fields	Field 6	Standard Fields	Field 22
	Field 7		Field 7		Field 23
	Field 8		Field 8		Field 24
	Field 9		Field 9		Field 25
	Field 10		Field 10		Field 26
	Field 11		Field 11		Field 27
	Field 12		Field 12		Field 28
	Field 13		Field 13		Field 29
	Field 14		Field 14		Field 30
	Field 15		Field 15		Field 31
	Field 16		Field 16		Field 32
	Field 17		Field 17		Field 33
	Field 18		Field 18		Field 34
	Field 19		Field 19		Field 35
	Field 20		Field 20	Calculated Fields	Field 44
	Field 21		Field 21		Field 45
	Field 22	Calculated Fields	Field 36		Field 46
	Field 23		Field 37		Field 47
	Field 24		Field 38		Field 48
	Field 25		Field 39		Field 49
	Field 26		Field 40		Field 50
	Field 27		Field 41		
	Field 28		Field 42		
	Field 29		Field 43		
	Field 30				
	Field 31				
	Field 32				
	Field 33				
	Field 34				
	Field 35				
Calculated Fields	Field 36				
	Field 37				
	Field 38				
	Field 39				

	Field 40
	Field 41
	Field 42
	Field 43
	Field 44
	Field 45
	Field 46
	Field 47
	Field 48
	Field 49
	Field 50

Note: This is a suggested approach. Experts may use this framework at their discretion.

6.2.1.4 *Relevance of Filters with Regard to the Framework*

- If we need to apply filters on non-common fields, it is not an issue and should work well. In case you apply filters on common fields. The normal approach is to apply page filter and do linked dimension on common fields across data sources. We suggest not to do the same only for such scenarios rather apply filters individually on the same common fields separately for each data source. This may seem a redundancy but ensures complexity of the scenario does not impact the performance of the story.

6.2.1.5 *Where not to use the Framework*

- The framework hinges on insisting of forced redundancy of common fields across multiple data sources in its aim to ensure one to one mapping of data source to widgets rather than browser based blending and linked dimension. In case the common fields need to be present as common a filter and need to be linked, the idea should be to go for single data source with the current guardrail limit of 120 columns for a single story and then splitting into multiple stories rather than into multiple data sources and joining them via Linked Dimension. This is to avoid any performance issues and browser memory issues that are introduced due to complex scenarios.

7 **APPENDIX**

• ***Setting Analytics Default Autocreate Role***

After ORD (Online Report Designer) is activated in provisioning, a default user role has to be enabled in the Analytics backend to allow all new users to be created automatically when they access Canvas reports (ORD).

The first user to access Analytics will automatically receive the admin role Implementation Partner. Please ensure this is a BizX admin user. This user can open a new Canvas report from Report Center, and from the page editor select Admin Roles Maintenance, select the role and from the Tools dropdown make it the default role.

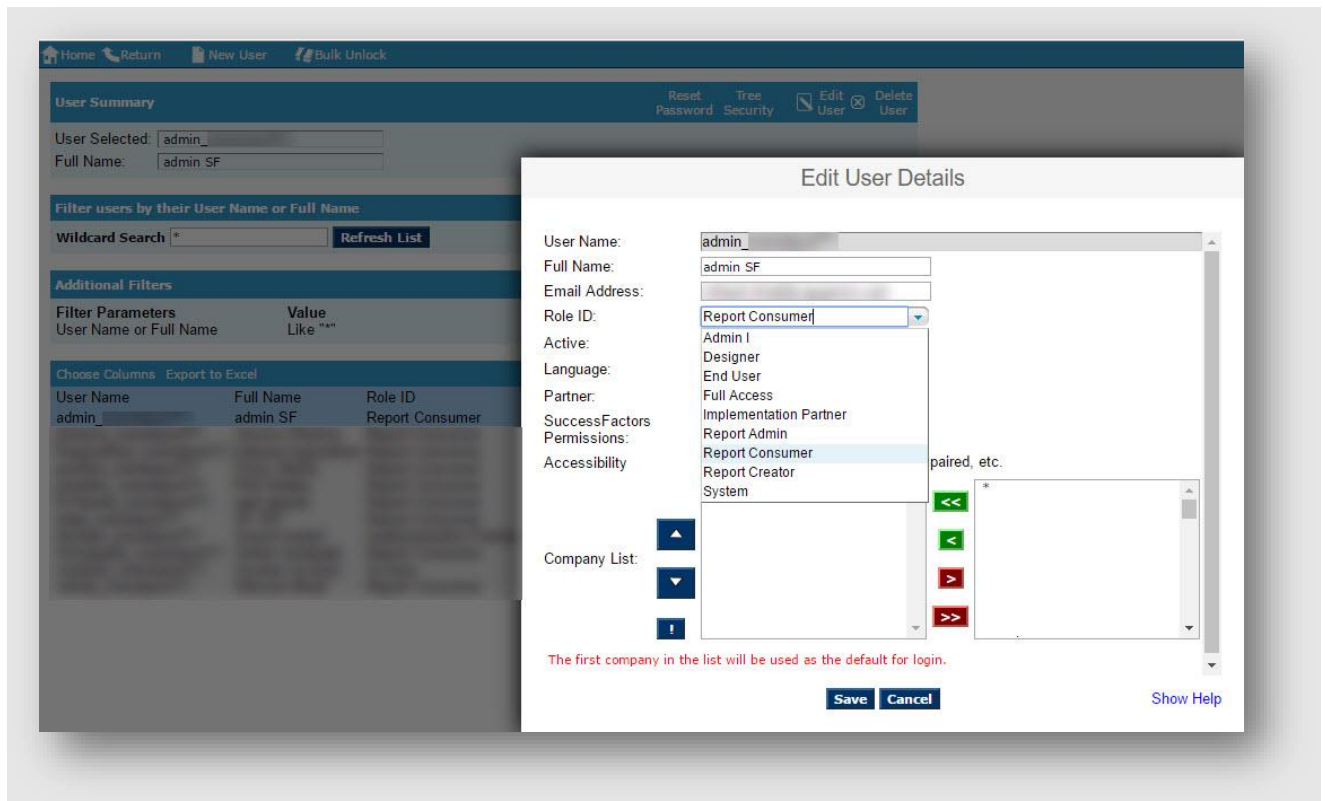


Figure 71: Edit User Details

Note: Only SAP Cloud Support can enable the default Role in Analytics backend. Customers have to raise an incident and request Default Role to be set as *Report Consumer*.

- ***User Unable to Create / Access Canvas reports in Report Center***

If two users have the same UserID in BizX even if with different letter casing, the second user will never not be able to create / access Canvas reports in Report Center. The second user's UserID has to be changed.

8 REFERENCES

SAP Help Portal

- Implementation Guide
[SAP SuccessFactors People Analytics: Stories in People Analytics](#)

KBAs

- People Analytics Story - Columns limits for Story Creation
<https://launchpad.support.sap.com/#/notes/3025688>
- Stories in People Analytics - Known Issues
<https://launchpad.support.sap.com/#/notes/3050294>

Blogs

- [Known Issues, Product Gaps and Limitations within Stories in People Analytics blog post on Customer Community](#)

SAP Analytics Cloud

- <https://www.sapanalytics.cloud/resources-performance-best-practices/>
- <https://saphanajourney.com/sap-analytics-cloud/learning-article/best-practices-for-performance/>

www.sap.com/contactsap

© 2022 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See www.sap.com/copyright for additional trademark information and notices.