Workiva

Utilities Workshop

Guide to Wdata & Chains Build

Presenter Name
APAC SA's

Design Philosophy

Operational Excellence

Reliability

Security

Performance

Empathy

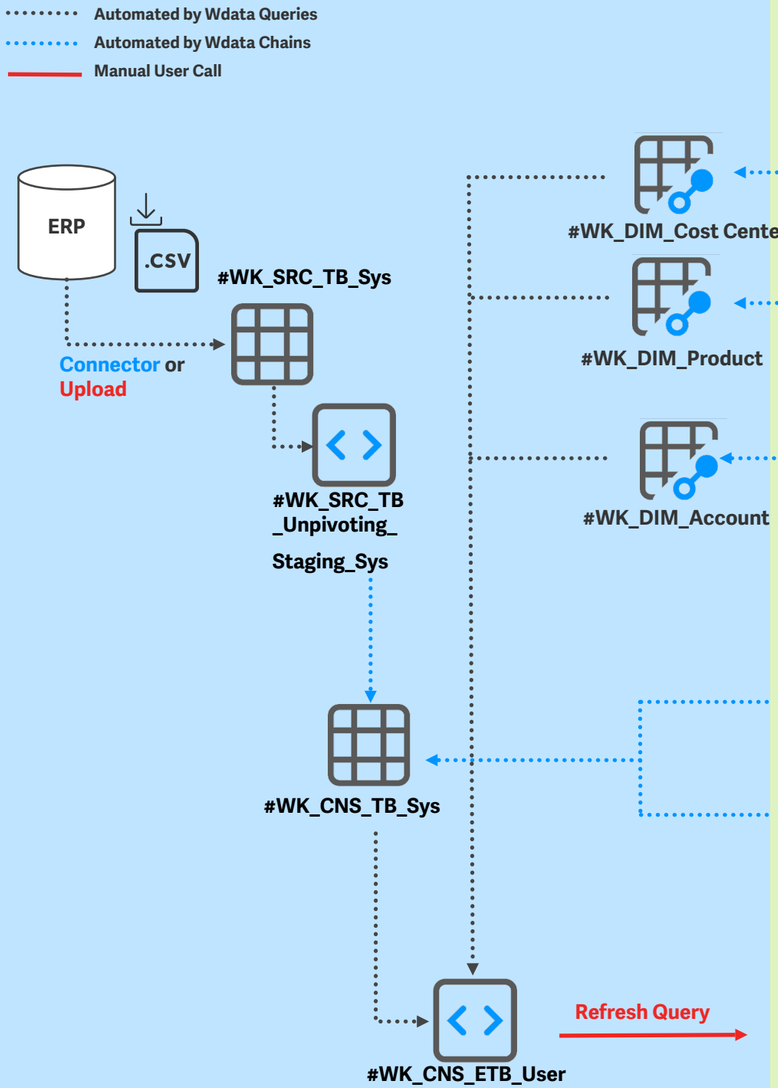
Client's Case Story

Our client, Dreamer Inc. has chosen Workiva to be their consultants in helping them improve their data workflow and processes. The end outcome that we agreed is for Workiva to generate a Master Spreadsheet with all the relevant data sets they need for their Financial Reports.

They have sent across their source files that flows to their Financial Reports that comprises of:

1. Trial Balance Actuals Dec'19, Nov'20, Dec'20 (.csv)
2. Adjustments & Budget Input Template Dec'20 (.csv)
3. Chart of Accounts (.csv)
4. Cost Center Mapping (.csv)
5. Product Mapping (.csv)

Workiva's objective is to setup a Workspace for Dreamer Inc. to design a model that automates their Monthly Financial Reporting process. Let us now take a look at the expected process of the data flow with Workiva.



Control Tower Sheet

- Controls the Year and Month to facilitate the Roll-forward process
- Feeds chain with spreadsheet IDs, query IDs, query parameters and data tags

Cost Centre Mapping

Account Code	Cost Centre Code	Cost Centre Name	ccru2	ccru3	ccru4
16-13-00-00-111	CC4337	Investment Solutions	Funds Management	Investment Solutions	
17-12-00-00-124	CC6173	Facilities Management	Support Units	Operations	Investment Operations

Product Mapping

productcode	productname	fmproductassetclass	fmproducttype	fmproductgeography	fmproductclass	productru
4062	Product - 284	Fixed Income	Fund	Australia	Institutional	Funds Management
4015	Product - 262	Fixed Income	Mandate	Australia	Institutional	Funds Management

Chart of Accounts / Account Hierarchy/ FS Mapping

GL Account	GL Description	FS	FS1	FS2
11-11-00-00-112	Intangible Assets - Software Licence	BS	Non current assets	Intangible assets
11-12-00-00-112	Intangible Assets Amortisation - Software Licence	BS	Non current assets	Intangible assets

Extended CoA

GL Account	GL Description	FS	FS1	FS2
BG001	Intangible assets	BS	Non current assets	Intangible assets
BG002	Bank loan - non current	BS	Non Current liability	Bank loan - non current

Adjustments

Current_FY	FS	Account_Code	Description	Adj_Jan	Adj_Feb	Adj_Mar	.....	YTD_Adj_Jan	YTD_Adj_Feb	YTD_Adj_Mar
2020	BS	13-11-11-00-111	Share Capital	25,960.00	37,809.00	22,557.00		27,980.00	27,980.00	27,980.00
2020	BS	13-11-11-00-112	Share Premium	30,100.00	68,985.00	14,328.00		32,120.00	32,120.00	32,120.00

Budget

Current_FY	FS	Account _Code	Description	Bud_Jan	Bud_Feb	Bud_Mar	.....	YTD_Bud_Jan	YTD_Bud_Feb	YTD_Bud_Mar
2020	BS	BG001	Intangible assets	8,671,304.80	1,143,195.83	10,999,007.36		8,673,324.80	8,673,324.80	8,673,324.80
2020	BS	BG002	Bank loan - non current	(0.12)	11,677.70	4,137,598.02		2,019.88	2,019.88	2,019.88

Extended Trial Balance

Current_FY	GL Account	GL Description	FS	FS1	FS2	Cost Centre	Product Code	CY_YTD	CY_FTM	CY_MoM	PY_YTD	PY_FTM
2020	11-11-00-00-112	Software Licence	BS	Non current assets	Intangible assets	General	7405	2,583,341	2,583,341	(726,565)	1,988,920	1,808,109
2020	11-12-00-00-112	Intangible Assets Amortisation	BS	Non current assets	Intangible assets	Mgmt Team	4102	(1,077,919)	(1,077,919)	303,165	(984,024)	(894,568)

Wdesk Workspace Set-up

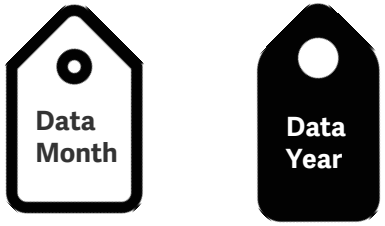
Objectives:

1. Import Reporting Package.tar.gz base spreadsheet template
2. Create Folders for organizing tables & queries

Points to note:

- Base template is purely for training purpose to provide the necessary resources & templates to build for this exercise. It does not serve to be used for client's projects
- Creating folders allows the separation and identification of Workiva's Resources & Client's Resources
  - We do not work in different environments (e.g. UAT, SIT, Production)

Creating Tags



Objectives:

1. Setting up Tag Names & Tag Values

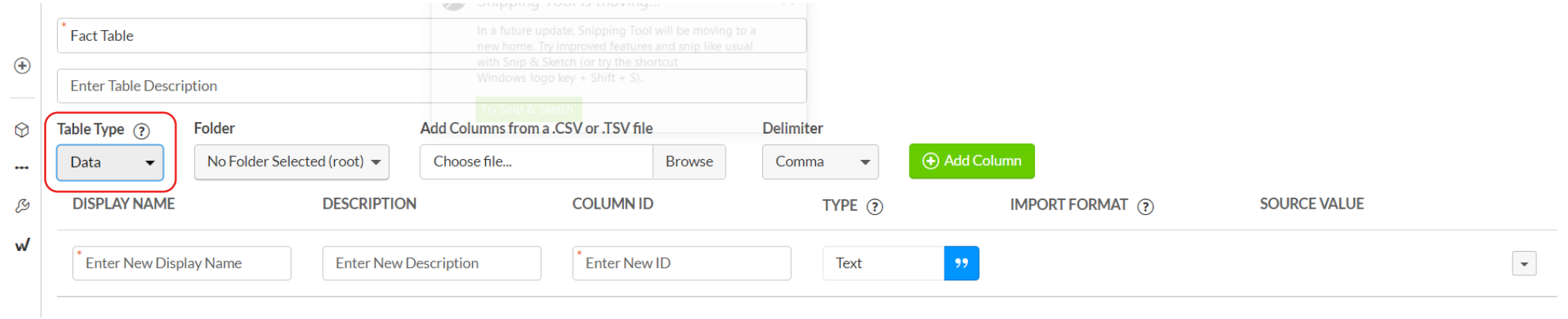
TB_Dec_2019

TB_Nov_2020

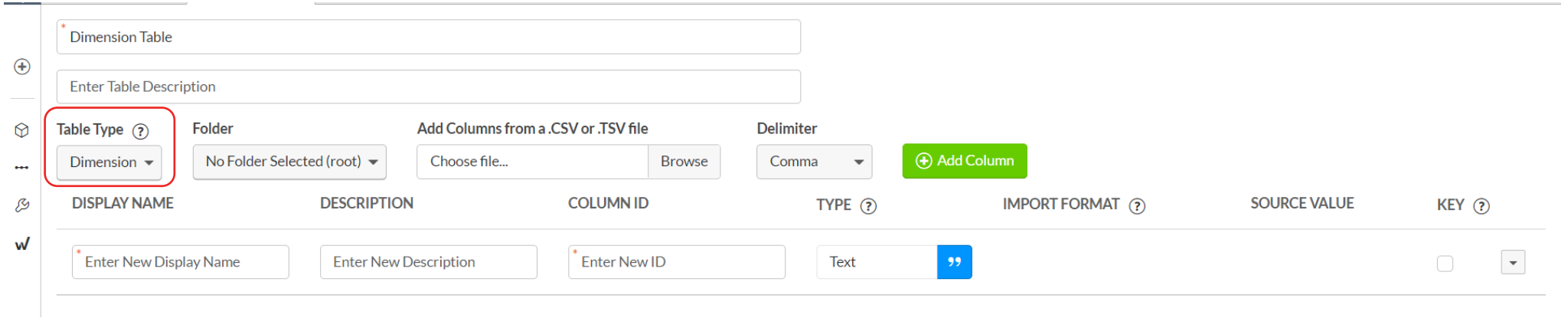
TB_Dec_2020

Wdata Tables

Data (Fact) Table



Dimension Table



workiva

Data (Fact) Table Vs Dimension Table

Data (Fact) Table	Dimension Table
• Typically to store transactional data i.e. GL transactions • Allows for duplicate transactions i.e. an account tied to a dimension in multiple rows • Typically imports and stacks data files every reporting cycle	• To store Dimension Data i.e. Data that defines and categorizes Fact Data • Does not allow any duplicate data as a dimension data cannot define & categorize the same fact data differently i.e. an account cant be defined as 2 different product codes • There is always a key found between the fact & dimension data to define and categorize the given transaction • Typically replaces Dimension Data File when it is updated instead of stacking files

Relationship between Fact & Dimension

Data (Fact) Table

year	month	entity	account_code	description	forward_balance	debit	credit	ending_balance	product_code
2020	8	c5854	11-11-00-00-112	Intangible Assets - Software Licence	1789743.384	0	0	1789743.384	7405
2020	7	c5854	11-12-00-00-112	Intangible Assets Amortisation - Software Licence	-240686.556	0	497214.476	-737901.032	4102
2020	12	c2452	13-11-11-00-111	INV| Subsidiary Share Capital	2.80E+07	0	0	2.80E+07	2010
2020	9	c2445	13-11-11-00-112	INV| Subsidiary Share Premium	6135797.868	0	0	6135797.868	2095
2020	9	c5940	13-11-11-00-115	INV| Other IC Share Capital	0	1.682208	0	1.682208	7898
2020	1	c2452	13-11-11-01-111	INV| Subsidiary Share Capital FMV	3.18E+07	2.65E+07	1.32E+07	4.51E+07	2112
2020	10	c5854	15-11-00-00-113	Deposits Receivable - Other Deposit & Guarantee {1Y+}	50929.25749	164393.9293	100349.6494	114973.5374	7152
2020	8	c5854	15-11-00-00-113	Deposits Receivable - Other Deposit & Guarantee {1Y+}	45783.2232	147783.108	90210.0408	103356.2904	7152
2020	6	c5854	15-11-00-00-113	Deposits Receivable - Other Deposit & Guarantee {1Y+}	46253.02566	149299.5776	91135.72703	104416.8763	7152
2020	9	c5917	16-13-00-00-111	ICoRec| (>50%) Current Account {1Y-}	1.4256	0	0	1.4256	2038
2020	5	c5854	16-13-00-00-111	ICoRec| (>50%) Current Account {1Y-}	82557.25546	3604.176	40741.6055	45419.82595	2038
2020	4	c2445	16-13-00-00-111	ICoRec| (>50%) Current Account {1Y-}	1122996.487	215182.656	0	1338179.143	2038
2020	3	c2452	16-13-00-00-111	ICoRec| (>50%) Current Account {1Y-}	0.16848	0	0	0.16848	2038
2020	11	c5940	16-13-00-00-111	ICoRec| (>50%) Current Account {1Y-}	14536.8432	0	0	14536.8432	2038

Dimension Table

productcode	productname	fmproductassetclass	fmproducttype	fmproductgeography	fmproductclass	productru
2010	Product - 101	Real Estate	Fund	Australia	Institutional	Funds Management
2038	Product - 113	Equities	Mandate	Australia	Institutional	Funds Management
2095	Product - 117	Fixed Income	Mandate	Australia	Institutional	Funds Management
2112	Product - 126	Equities	Fund	Australia	Retail	Funds Management
4102	Product - 305	Fixed Income	Fund	Australia	Institutional	Funds Management
7152	Product - 486	Alternatives	Mandate	Australia	Institutional	Funds Management
7405	Product - 499	Fixed Income	Fund	Australia	Retail	Funds Management
7898	Product - 598	Fixed Income	Mandate	Australia	Institutional	Funds Management

Examples of Fact/Dimension Relationships

Data (Fact) Table	Dimension Table

• Trial Balance
  † Account Number
• General Ledger
  † Product Code
• General Ledger
  † Cost Center Abbreviation
	• Chart of Accounts
	  † Key - Account Number
	• Product Listing/Table
	  † Key - Product Code
	• Cost Center Listing/Table
	  † Cost Center Abbreviation

Creating Dimension Tables

Objectives:

1. Create Dimension Tables for Product Mapping, Cost Center Mapping & Accounts Mapping
2. Configuring Column Types
3. Indicating which Field is the "Key" in each Dimension Table by checking the box
4. Import the client's mapping source files into the respective tables
5. Import Account Dim Extension from Spreadsheet into Accounts Mapping Dimension Table

Points to note:

- Most fields are represented as "Text" for column type information other than amount values
- "Key" configuration is only specific to Dimension Tables to prevent duplicate information from being imported
- Account Dim Extension exists as an approach Workiva recommends to allow clients to post Budget & Adjustments values at an aggregated level instead of breaking down to an individual Account level

How do Clients Post their Data?

Account Level

Current_FY	Entity	FS	Account_Code	Description	Product_Code	Adj_Jan	Adj_Feb	Adj_Mar	Adj_Apr	Adj_May	Adj_Jun	Adj_Jul	Adj_Aug	Adj_Sep	Adj_Oct	Adj_Nov	Adj_Dec
2020	c2445	BS	13-11-11-00-111	INV| Subsidiary Share Capital	2010	25,960	37,809	22,557	39,108	46,644	64,774	69,742	71,423	39,613	46,415	27,586	60,697
2020	c2445	BS	13-11-11-00-112	INV| Subsidiary Share Premium	2095	30,100	68,985	14,328	45,312	23,312	34,287	37,612	32,860	16,002	66,763	28,497	12,145
2020	c2445	BS	13-11-11-01-111	INV| Subsidiary Share Capital FMV	2112	13,145	37,001	75,120	22,842	55,305	70,156	73,254	69,435	16,729	11,260	52,675	20,679
2020	c2445	BS	13-14-11-00-117	Loan to| (>50%) IFL {1Y+}	2071	64,458	44,440	79,962	24,543	35,337	54,071	62,014	64,926	32,788	79,876	56,120	68,998
2020	c2445	BS	16-13-00-00-111	ICoRec| (>50%) Current Account {1Y-}	2038	60,705	38,141	73,732	37,875	58,770	70,585	41,757	22,825	79,841	69,032	77,370	68,830
2020	c2445	BS	17-12-00-00-111	BAN| General Accounts - Account 1	2000	65,529	22,553	10,002	32,600	63,133	23,984	40,719	64,643	32,525	70,598	47,909	51,762
2020	c2445	PL	61-11-00-00-111	Corporate Costs - Secretarial Services	2122	75,270	74,979	34,598	50,480	66,425	77,828	62,146	69,333	64,875	79,004	29,599	75,160
2020	c2445	PL	61-11-00-00-114	Corporate Costs - Domiciliation	2035	58,736	76,670	25,826	19,622	28,317	77,604	39,282	75,554	72,476	55,098	27,787	56,140
2020	c2445	PL	61-11-00-00-131	Corporate Costs - P-Codes	2110	26,123	49,738	63,155	40,596	52,465	54,669	16,833	56,935	42,606	64,408	53,113	33,125
2020	c2445	PL	61-17-00-00-111	BRE Management Fees Exp. - Accounting	2173	68,455	72,029	27,821	52,630	43,653	56,522	73,884	62,812	25,409	73,855	79,768	23,167
2020	c2445	PL	61-18-00-00-112	Professional Fees - Audit	2177	38,715	58,796	56,214	12,834	78,743	16,503	74,256	75,375	52,973	10,747	26,415	78,501
2020	c2445	PL	61-18-00-00-114	Professional Fees - Tax Compliance	2099	43,826	18,709	26,155	76,642	30,678	71,570	31,807	35,469	67,154	42,186	31,688	20,496
2020	c2445	PL	73-21-15-00-111	Revaluation Loss On Financial Assets At FVTPL	2312	43,826	18,709	26,155	76,642	30,678	71,570	31,807	35,469	67,154	42,186	31,688	20,496

Financial Statement Statutory Level

Current_FY	Entity	FS	Account_Code	Description	Bud_Jan	Bud_Feb	Bud_Mar	Bud_Apr	Bud_May	Bud_Jun	Bud_Jul	Bud_Aug	Bud_Sep	Bud_Oct	Bud_Nov	Bud_Dec
2020	c2445	BS	BG001	Intangible assets	8671304.8	1143195.83	10999007.36	100717.3	162315.27	928777.73	1222644.89	2899667.51	0	338942.38	0	2458195.2
2020	c2445	BS	BG002	Bank loan - non current	-0.12	11677.7	4137598.02	-11677.7	-3742858.86	11903	-78993022.55	4599499.89	-2087265.08	44343747.01	0	-1.59
2020	c2445	BS	BG003	Loan receivable - non current	1222644.89	2899667.51	0	338942.38	0	2458195.2	5592360.75	3705769.73	-2249745	-600566	-3342615.75	-315265
2020	c2445	BS	BG004	Other receivables - current	0.17	-0.12	11677.7	4137598.02	-11677.7	-3742858.86	11903	-78993022.55	4599499.89	-2087265.08	44343747.01	0
2020	c2445	BS	BG005	Loan receivable - current	8671304.8	1143195.83	10999007.36	100717.3	162315.27	928777.73	1222644.89	2899667.51	0	338942.38	0	2458195.2
2020	c2445	BS	BG006	Cash and cash equivalents	-0.12	11677.7	4137598.02	-11677.7	-3742858.86	11903	-78993022.55	4599499.89	-2087265.08	44343747.01	0	-1.59
2020	c2445	PL	BG007	Other income	1222644.89	2899667.51	0	338942.38	0	2458195.2	5592360.75	3705769.73	-2249745	-600566	-3342615.75	-315265
2020	c2445	PL	BG008	Loan Interest income	0.17	-0.12	11677.7	4137598.02	-11677.7	-3742858.86	11903	-78993022.55	4599499.89	-2087265.08	44343747.01	0
2020	c2445	PL	BG009	Dividend Income	8671304.8	1143195.83	10999007.36	100717.3	162315.27	928777.73	1222644.89	2899667.51	0	338942.38	0	2458195.2
2020	c2445	PL	BG010	Interest income	-0.12	11677.7	4137598.02	-11677.7	-3742858.86	11903	-78993022.55	4599499.89	-2087265.08	44343747.01	0	-1.59
2020	c2445	PL	BG011	Other expenses	1222644.89	2899667.51	0	338942.38	0	2458195.2	5592360.75	3705769.73	-2249745	-600566	-3342615.75	-315265
2020	c2445	PL	BG006	Income tax expense	8671304.8	1143195.83	10999007.36	100717.3	162315.27	928777.73	1222644.89	2899667.51	0	338942.38	0	2458195.2

workiva

Relationships between Accounts & FS Statutory

Account Level

Current_FY	Entity	FS	Account_Code	Description	Product_Code	Adj_Jan	Adj_Feb	Adj_Mar	Adj_Apr	Adj_May	Adj_Jun	Adj_Jul	Adj_Aug	Adj_Sep	Adj_Oct	Adj_Nov	Adj_Dec
2020	c2445	BS	13-11-11-00-111	INV| Subsidiary Share Capital	2010	25,960	37,809	22,557	39,108	46,644	64,774	69,742	71,423	39,613	46,415	27,586	60,697
2020	c2445	BS	13-11-11-00-112	INV| Subsidiary Share Premium	2095	30,100	68,985	14,328	45,312	23,312	34,287	37,612	32,860	16,002	66,763	28,497	12,145
2020	c2445	BS	13-11-11-01-111	INV| Subsidiary Share Capital FMV	2112	13,145	37,001	75,120	22,842	55,305	70,156	73,254	69,435	16,729	11,260	52,675	20,679
2020	c2445	BS	13-14-11-00-117	Loan to | (>50%) IFL {1Y+}	2071	64,458	44,440	79,962	24,543	35,337	54,071	62,014	64,926	32,788	79,876	56,120	68,998
2020	c2445	BS	16-13-00-00-111	ICoRec| (>50%) Current Account {1Y-}	2038	60,705	38,141	73,732	37,875	58,770	70,585	41,757	22,825	79,841	69,032	77,370	68,830
2020	c2445	BS	17-12-00-00-111	BAN| General Accounts - Account 1	2000	65,529	22,553	10,002	32,600	63,133	23,984	40,719	64,643	32,525	70,598	47,909	51,762
2020	c2445	PL	61-11-00-00-111	Corporate Costs - Secretarial Services	2122	75,270	74,979	34,598	50,480	66,425	77,828	62,146	69,333	64,875	79,004	29,599	75,160
2020	c2445	PL	61-11-00-00-114	Corporate Costs - Domiciliation	2035	58,736	76,670	25,826	19,622	28,317	77,604	39,282	75,554	72,476	55,098	27,787	56,140
2020	c2445	PL	61-11-00-00-131	Corporate Costs - P-Codes	2110	26,123	49,738	63,155	40,596	52,465	54,669	16,833	56,935	42,606	64,408	53,113	33,125
2020	c2445	PL	61-17-00-00-111	BRE Management Fees Exp. - Accounting	2173	68,455	72,029	27,821	52,630	43,653	56,522	73,884	62,812	25,409	73,855	79,768	23,167
2020	c2445	PL	61-18-00-00-112	Professional Fees - Audit	2177	38,715	58,796	56,214	12,834	78,743	16,503	74,256	75,375	52,973	10,747	26,415	78,501
2020	c2445	PL	61-18-00-00-114	Professional Fees - Tax Compliance	2099	43,826	18,709	26,155	76,642	30,678	71,570	31,807	35,469	67,154	42,186	31,688	20,496
2020	c2445	PL	73-21-15-00-111	Revaluation Loss On Financial Assets At FVTPL	2312	43,826	18,709	26,155	76,642	30,678	71,570	31,807	35,469	67,154	42,186	31,688	20,496

Financial Statement Statutory Level

GL Account	GL Description	FS	FS1	FS2	FS3	FS4	FS5
11-11-00-00-112	Intangible Assets - Software Licence	BS	Non current assets	Intangible assets	Intangible assets	Intangible assets	Intangible assets
11-12-00-00-112	Intangible Assets Amortisation - Software Licence	BS	Non current assets	Intangible assets	Intangible assets	Intangible assets	Intangible assets
12-11-17-00-111	Deferred Financing Fees - Loan Facility Fees	BS	Non Current liability	Bank loan - non current	Costs incurred	Bank loan - non current	Costs incurred
12-11-21-00-111	Formation Expenses	BS	Non current assets	Intangible assets	Intangible assets	Intangible assets	Intangible assets
12-12-17-00-111	Deferred Financing Fees Amortisation - Loan Facility Fees	BS	Non Current liability	Bank loan - non current	Accumulated amortisation	Bank loan - non current	Accumulated amortisation
13-11-11-00-111	INV| Subsidiary Share Capital	BS	Non current assets	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss
13-11-11-00-112	INV| Subsidiary Share Premium	BS	Non current assets	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss
13-11-11-00-113	INV| Associate Share Capital	BS	Non current assets	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss

Workiva

Approach Examples

Using Existing Accounts

GL Account	GL Description	FS	FS1	FS2	FS3	FS4	FS5
11-11-00-00-112	Intangible Assets - Software Licence	BS	Non current assets	Intangible assets	Intangible assets	Intangible assets	Intangible assets
11-12-00-00-112	Intangible Assets Amortisation - Software Licence	BS	Non current assets	Intangible assets	Intangible assets	Intangible assets	Intangible assets
12-11-17-00-111	Deferred Financing Fees - Loan Facility Fees	BS	Non Current liability	Bank loan - non current	Costs incurred	Bank loan - non current	Costs incurred
12-11-21-00-111	Formation Expenses	BS	Non current assets	Intangible assets	Intangible assets	Intangible assets	Intangible assets
12-12-17-00-111	Deferred Financing Fees Amortisation - Loan Facility Fees	BS	Non Current liability	Bank loan - non current	Accumulated amortisation	Bank loan - non current	Accumulated amortisation
13-11-11-00-111	INV| Subsidiary Share Capital	BS	Non current assets	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss
13-11-11-00-112	INV| Subsidiary Share Premium	BS	Non current assets	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss
13-11-11-00-113	INV| Associate Share Capital	BS	Non current assets	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss

Creating Dummy Accounts

gl_account	gl_description	fs	fs1	fs2	fs3	fs4	fs5
BG002	Bank loan - non current	BS	Non Current liability	Bank loan - non current			
BG003	Loan receivable - non current	BS	Non current assets	Loan receivable - non current			
BG009	Dividend Income	PL	Revenues	Dividend Income			
BG007	Other income	PL	Revenues	Other income			
BG011	Other expenses	PL	Expense	Other expenses			
BG006	Cash and cash equivalents	BS	Current assets	Cash and cash equivalents			
BG012	Income tax expense	PL	Expense	Income tax expense			
BG001	Intangible assets	BS	Non current assets	Intangible assets			
BG004	Other receivables - current	BS	Current assets	Other receivables - current			
BG010	Interest income	PL	Revenues	Interest income			
BG008	Loan Interest income	PL	Revenues	Loan Interest income			
BG005	Loan receivable - current	BS	Current assets	Loan receivable - current			

Creating Account Dimension Extended

Objectives:

1. Import the Account Dimension Extended from Spreadsheet into the Account Dimension Table created previously
2. Import is done through Wdata Connection through the spreadsheet

Points to note:

- Account Dimension Extended is created for the ease of training for defining the Budget & Adjustment Data
- Typically our Account Dimension (COA) is defined at the GL Account Level. However with Budget & Adjustments, clients typically do not post at such granular level, and instead post these values at a higher hierarchy level
- Hence, we created newly defined accounts specifically for the Budget & Adjustments Process which represents their mapping to be aligned with the main Account Dimensions Data
- Alternatively, another approach is for the client to use existing accounts with the same given mapping they want to post it to for the Budget & Adjustments Process

What Can Queries Do ?

Objectives:

1. Create Data Table for Trial Balance Import for Dec'19, Nov'20 & Dec'20
2. Configuring Column Types
3. Selecting the Tag Name & Tag Value for the file being imported
4. Adding New Datasets after the Data Table is created

Points to note:

- Most fields are represented as "Text" for column type information other than amount values
- Tags are an option to utilize when uploading more than a single file into a table. As shown earlier, tags help to "bookmark" each file uploaded with a specific tag name & value
- Tags can essentially be a way to parameterize the query results using the tag name and values in subsequent steps

Why Use Queries ?

Queries help to generate a more specific result output based on the user's business rules and methodology of calculations from a huge bank of data from one or many data sources.

Data (Fact) Table

year	month	entity	account_code	description	forward_balance	debit	credit	ending_balance	product_code
2020	8	c5854	11-11-00-00-112	Intangible Assets - Software Licence	1789743.384	0	0	1789743.384	7405
2020	7	c5854	11-12-00-00-112	Intangible Assets Amortisation - Software Licence	-240686.556	0	497214.476	-737901.032	4102
2020	12	c2452	13-11-11-00-111	INV| Subsidiary Share Capital	2.80E+07	0	0	2.80E+07	2010
2020	9	c2445	13-11-11-00-112	INV| Subsidiary Share Premium	6135797.868	0	0	6135797.868	2095
2020	9	c5940	13-11-11-00-115	INV| Other IC Share Capital	0	1.682208	0	1.682208	7898
2020	1	c2452	13-11-11-01-111	INV| Subsidiary Share Capital FMV	3.18E+07	2.65E+07	1.32E+07	4.51E+07	2112
2020	10	c5854	15-11-00-00-113	Deposits Receivable - Other Deposit & Guarantee {1Y+}	50929.25749	164393.9293	100349.6494	114973.5374	7152
2020	8	c5854	15-11-00-00-113	Deposits Receivable - Other Deposit & Guarantee {1Y+}	45783.2232	147783.108	90210.0408	103356.2904	7152
2020	6	c5854	15-11-00-00-113	Deposits Receivable - Other Deposit & Guarantee {1Y+}	46253.02566	149299.5776	91135.72703	104416.8763	7152
2020	9	c5917	16-13-00-00-111	ICoRec| (>50%) Current Account {1Y-}	1.4256	0	0	1.4256	2038
2020	5	c5854	16-13-00-00-111	ICoRec| (>50%) Current Account {1Y-}	82557.25546	3604.176	40741.6055	45419.82595	2038
2020	4	c2445	16-13-00-00-111	ICoRec| (>50%) Current Account {1Y-}	1122996.487	215182.656	0	1338179.143	2038
2020	3	c2452	16-13-00-00-111	ICoRec| (>50%) Current Account {1Y-}	0.16848	0	0	0.16848	2038
2020	11	c5940	16-13-00-00-111	ICoRec| (>50%) Current Account {1Y-}	14536.8432	0	0	14536.8432	2038

Financial Statement Statutory Level

GL Account	GL Description	FS	FS1	FS2	FS3	FS4	FS5
11-11-00-00-112	Intangible Assets - Software Licence	BS	Non current assets	Intangible assets	Intangible assets	Intangible assets	Intangible assets
11-12-00-00-112	Intangible Assets Amortisation - Software Licence	BS	Non current assets	Intangible assets	Intangible assets	Intangible assets	Intangible assets
12-11-17-00-111	Deferred Financing Fees - Loan Facility Fees	BS	Non Current liability	Bank loan - non current	Costs incurred	Bank loan - non current	Costs incurred
12-11-21-00-111	Formation Expenses	BS	Non current assets	Intangible assets	Intangible assets	Intangible assets	Intangible assets
12-12-17-00-111	Deferred Financing Fees Amortisation - Loan Facility Fees	BS	Non Current liability	Bank loan - non current	Accumulated amortisation	Bank loan - non current	Accumulated amortisation
13-11-11-00-111	INV| Subsidiary Share Capital	BS	Non current assets	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss
13-11-11-00-112	INV| Subsidiary Share Premium	BS	Non current assets	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss
13-11-11-00-113	INV| Associate Share Capital	BS	Non current assets	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss	Investments at fair value through profit or loss

Query Result

year	account_code	ending_balance	FS1	FS2
2020	11-11-00-00-112	1,789,743	Non current assets	Intangible assets
2020	13-11-11-00-111	1,808,109	Non current assets	Investments at fair value through profit or loss
2020	16-13-00-00-111	1,990,911	Current assets	Other receivables - current
2020	16-16-11-00-112	1,809,919	Current assets	Other receivables - current
2020	17-12-00-00-113	1,628,927	Current assets	Cash and cash equivalents

workiva

What Can Queries Do ?

Client Challenges:

- † Finance teams handle immense amounts of data on a regular basis
- † They export raw data from source systems to allow them to "massage" the data to the required state
- † This includes filtering, pivoting, formularizing, transposing and many more
- † This is repeated over & over again every reporting cycle

What clients do not realize is that such a process is based on business rules & logic that are used repeatedly every cycle

Solution:

1. Queries pull data from data sources
2. Queries can help to:
    a. Filter Data
    b. Transform Data
    c. Formularize Data
    d. Repeat Output consistently without manual intervention

Creating Current Year Query

Objectives:

1. Add Data Sources
2. Adding Fields from Data Sources into the Query
3. Creating a Relationship between Different Data Sources
4. Creating a Formulated Result for Monthly & Year-To-Date Fields
5. Creating a Filter with Tags
6. Sorting Data Result

Points to note:

- Query Results will follow the order of fields arranged in the query as you drag them from the data sources into the center panel
- Relationships must be defined if more than one source exist. A primary key or a similar field must exist between different sources to define the relationship.

Replicating Current Year Query for Prior Year

Objectives:

1. Copy an existing current year query for the prior year
2. Renaming the fields from current year to prior year
3. Setting the filters for prior year

Points to note:

- Filters help to pull specific data results, in this case, using the value of the year in our data table.
- Earlier on when the data table was created, "Year" field was defined as a text column
- Because of that, in the filter, we have to cast it as an integer first to minus 1 from the current year, before casting it back to text
- This would allow us to pull prior year's data 1 year earlier from the current year selected as the basis

Staging/Transforming Data

Original File Format:

year	month	entity	account_code	description	forward_balance	debit	credit	ending_balance	product_code
2020	1	c5854	11-11-00-00-112	Intangible Assets - Soft	1355866.2	0	0	1355866.2	7405
2020	2	c5854	11-11-00-00-112	Intangible Assets - Soft	1627039.44	0	0	1627039.44	7405
2020	3	c5854	11-11-00-00-112	Intangible Assets - Soft	1757202.595	0	0	1757202.595	7405
2020	4	c5854	11-11-00-00-112	Intangible Assets - Soft	1809918.673	0	0	1809918.673	7405
2020	5	c5854	11-11-00-00-112	Intangible Assets - Soft	1628926.806	0	0	1628926.806	7405
2020	6	c5854	11-11-00-00-112	Intangible Assets - Soft	1808108.754	0	0	1808108.754	7405
2020	7	c5854	11-11-00-00-112	Intangible Assets - Soft	1491452.82	0	0	1491452.82	7405
2020	8	c5854	11-11-00-00-112	Intangible Assets - Soft	1789743.384	0	0	1789743.384	7405
2020	9	c5854	11-11-00-00-112	Intangible Assets - Soft	1932922.855	0	0	1932922.855	7405
2020	10	c5854	11-11-00-00-112	Intangible Assets - Soft	1990910.54	0	0	1990910.54	7405
2020	11	c5854	11-11-00-00-112	Intangible Assets - Soft	1791819.486	0	0	1791819.486	7405
2020	12	c5854	11-11-00-00-112	Intangible Assets - Soft	1988919.63	0	0	1988919.63	7405
2020	1	c5854	11-12-00-00-112	Intangible Assets Amor	-218805.96	0	452013.16	-670819.12	4102
2020	2	c5854	11-12-00-00-112	Intangible Assets Amor	-262567.152	0	542415.792	-804982.944	4102
2020	3	c5854	11-12-00-00-112	Intangible Assets Amor	-283572.5242	0	585809.0554	-869381.5795	4102
2020	4	c5854	11-12-00-00-112	Intangible Assets Amor	-292079.6999	0	603383.327	-895463.0269	4102
2020	5	c5854	11-12-00-00-112	Intangible Assets Amor	-262871.7299	0	543044.9943	-805916.7242	4102
2020	6	c5854	11-12-00-00-112	Intangible Assets Amor	-291787.6202	0	602779.9437	-894567.5639	4102
2020	7	c5854	11-12-00-00-112	Intangible Assets Amor	-240686.556	0	497214.476	-737901.032	4102
2020	8	c5854	11-12-00-00-112	Intangible Assets Amor	-288823.8672	0	596657.3712	-885481.2384	4102
2020	9	c5854	11-12-00-00-112	Intangible Assets Amor	-311929.7766	0	644389.9609	-956319.7375	4102
2020	10	c5854	11-12-00-00-112	Intangible Assets Amor	-321287.6699	0	663721.6597	-985009.3296	4102
2020	11	c5854	11-12-00-00-112	Intangible Assets Amor	-289158.9029	0	597349.4938	-886508.3966	4102

ETB Query Result to achieve:

current_fy	entity	account_code	product_code	description	fs	jan	feb	mar	jan_ytd	feb_ytd	mar_ytd	py_jan	py_feb	py_mar	py_jan_ytd	py_feb_ytd	py_mar_ytd	carry_forward	adj_jan	adj_feb	adj_mar	ytd_adj_jan	ytd_adj_feb	ytd_adj_mar	bud_ja
2020	c2445	61-18-00-00-1	2099	Professional F	PL	#####	#####	#####	44094	97006	154152														
2020	c2452	47-21-11-00-1	2160	Other Op. Inc	PL	-7.61	-9.1	-9.86	-7.61	-16.742	-26.605														

workiva

Agenda

1. Wdata Guide
2. Wdata Chains Starters Guide
3. Chain : Manage Redundant Files
4. Chain : Run Query & Upload Files
5. Chain : File Importer
6. Chain : Creating Template Chain
7. Chain : External to Source
8. Chain : Source to Staging
9. Chain : Budget & Adjustments
10. Chain : Master Chain

Workiva

Design Philosophy

Operational Excellence

Reliability

Security

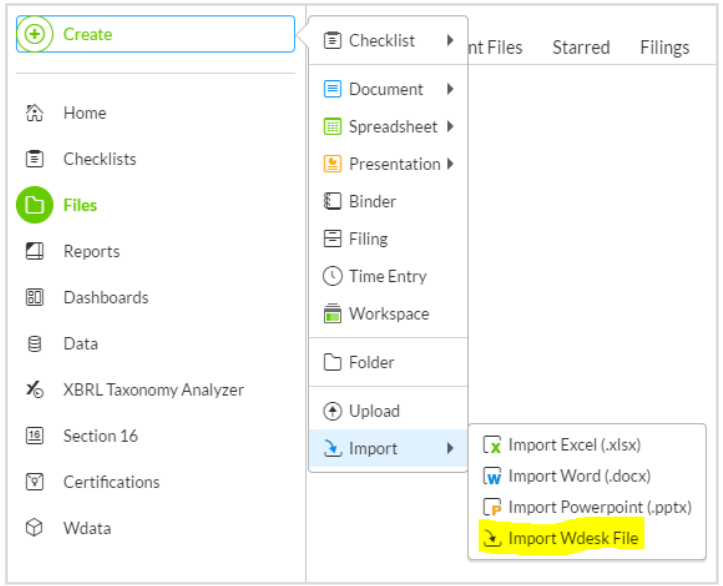
Performance

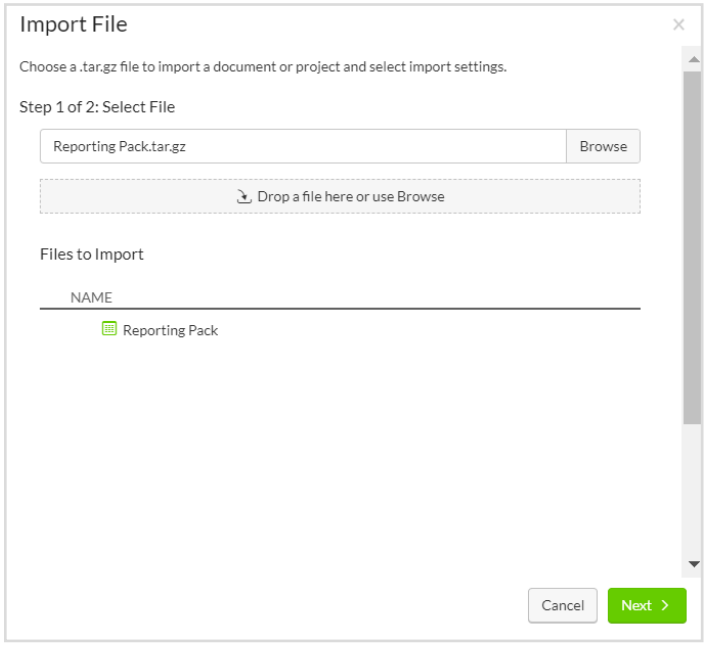
Empathy

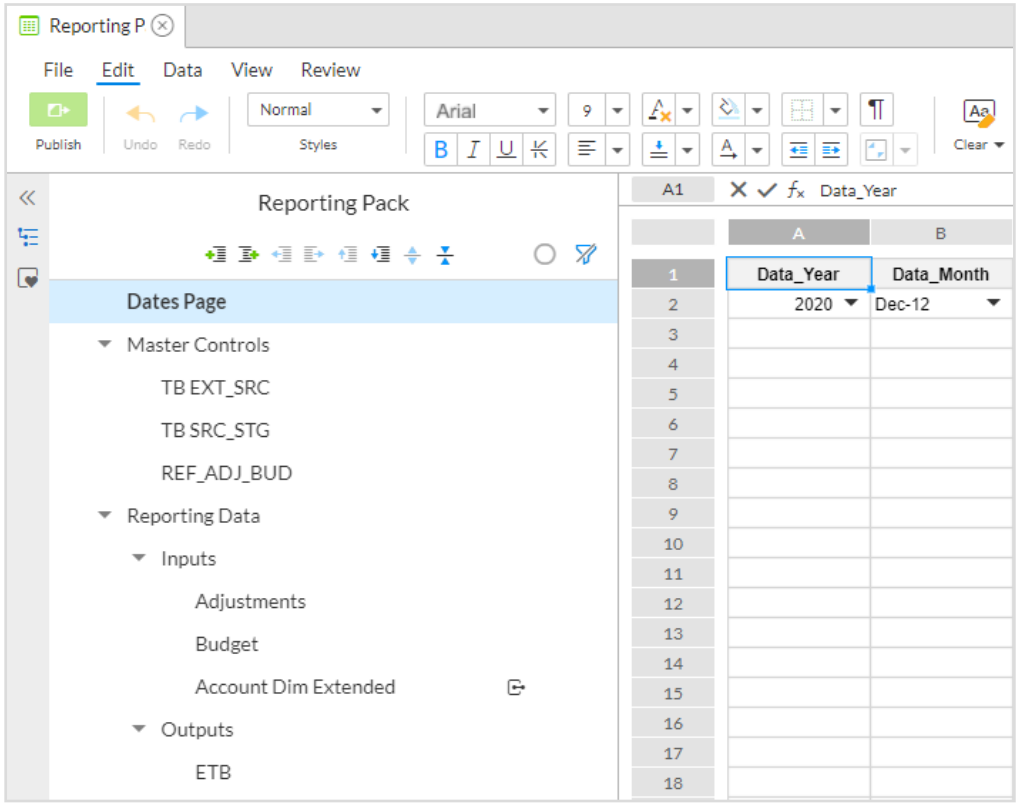
Wdata Tables, Queries

Workiva

Wdesk Set-Up



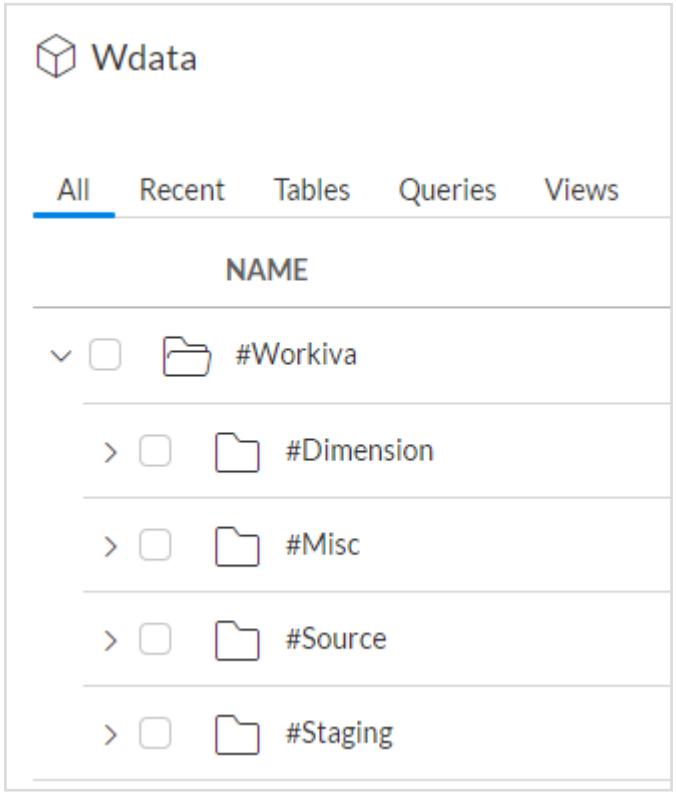


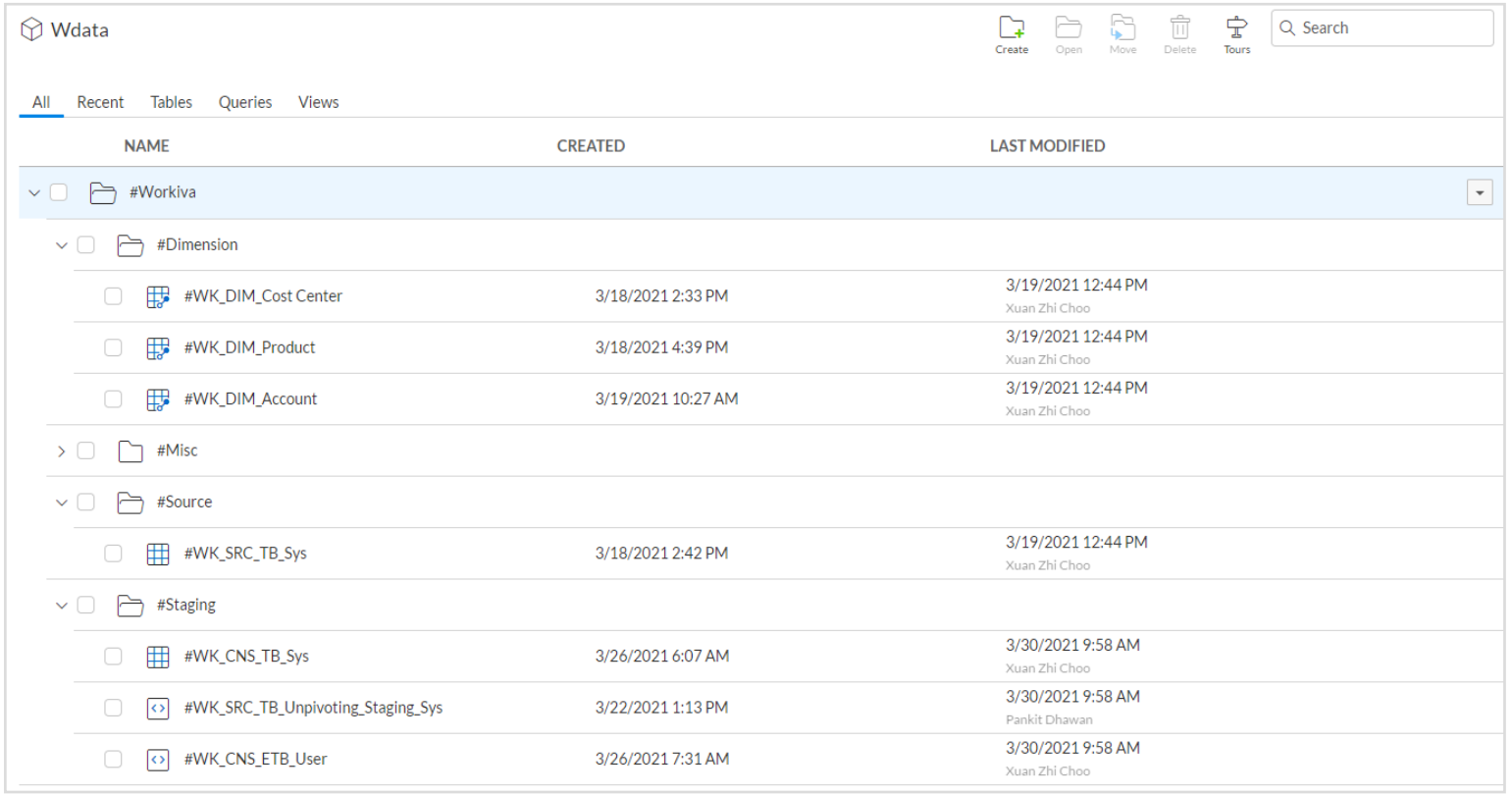


To set-up Wdesk, import the "Reporting Pack.tar.gz" into Wdesk. This file would establish the required spreadsheet structure for this training.

Workiva

Wdata Folder Structure





Wdata Folders are structured as follows:

i. #Dimension - stores all the Dimension tables (i.e. COA, Product, Cost Center)

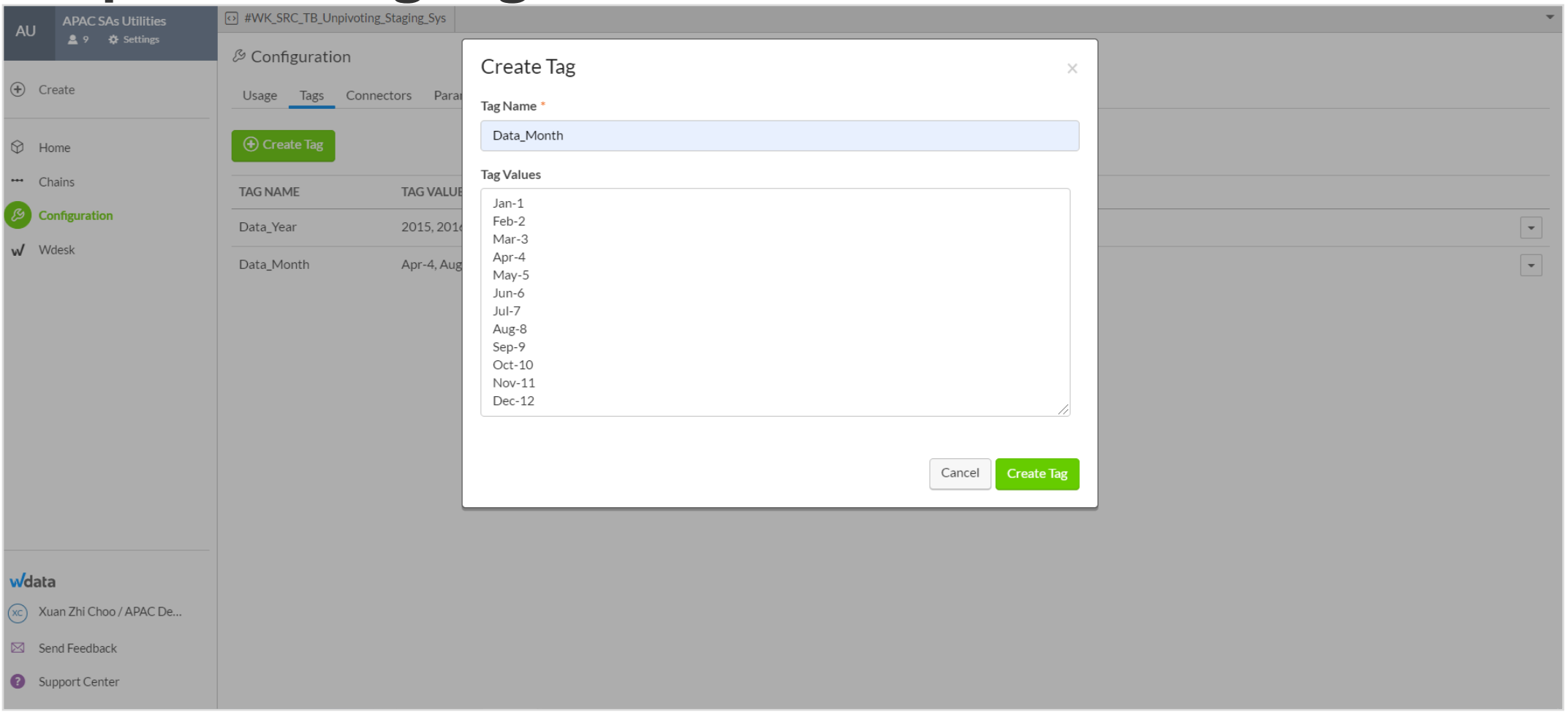
ii. #Misc - stores miscellaneous (testing, troubleshooting) that is not required for the built

iii. #Source - stores the Source table (TB GL Data)

iv. #Staging - stores the staging query, consumption table, ETB query that drives the final figures

Workiva

Step 4: Creating Tags



Creating 'tags' allows tagging of data during import & creation of data tables. Create 2 tags, name 'Data_Year' & 'Data_Month'.

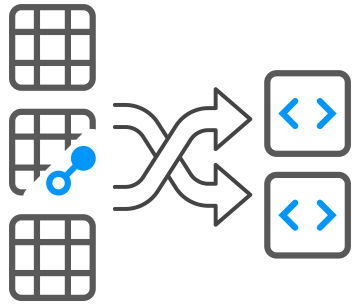
Workiva

Step 5: Overview - Steps to Build

Steps	Description
1	First create Dimension tables using the 'COA, Product, Cost Center' data sets.
2	Next, create the TB source (SRC) table using the PY & CY Trial Balance Data.
3	Create a 'Staging' query ('#WK_SRC_TB_Unpivoting_Staging_Sys'), that Converts TB Fact Data with months Pivoted to 48 Columns (Jan to Dec movement + YTD Jan to YTD Dec) for Current & Prior Year.
4	The output from the 'Staging' query is used as an input into the consumption table ('#WK_CNS_TB_Sys').
5	Connect 'Budget' & 'Adjustments' spreadsheet to the consumption table ('#WK_CNS_TB_Sys').
6	Create the Consumption query (ETB) that drives the final results. This query would be connected to the 'Actuals ETB' sheet that would populate the final figures.
7	Set up 3 chains to automate the refresh of the Consumption table (#WK_CBS_TB_Sys) with the Staging query; i.e. step 3 & 4.
8	**Glossary:** i) DIM - Dimension tables (e.g. Mapping tables such as COA) ii) FACT - Actual data tables that stores financials data (e.g. Trial Balance)

Wdata Tables & Queries

Data tables , also known as fact tables, contain information that builds up over time. For example, you can use data tables to curate trial balance data.



Dimension tables contain relational information, such as rollups by department or for mapping, like dates for fiscal year accounting.

Queries, slice-and-dice from multiple data tables, joins and curate information (financials) for analysis and reporting.

Workiva

Wdata Tables - Data Transformation

Source Data from client's source systems.

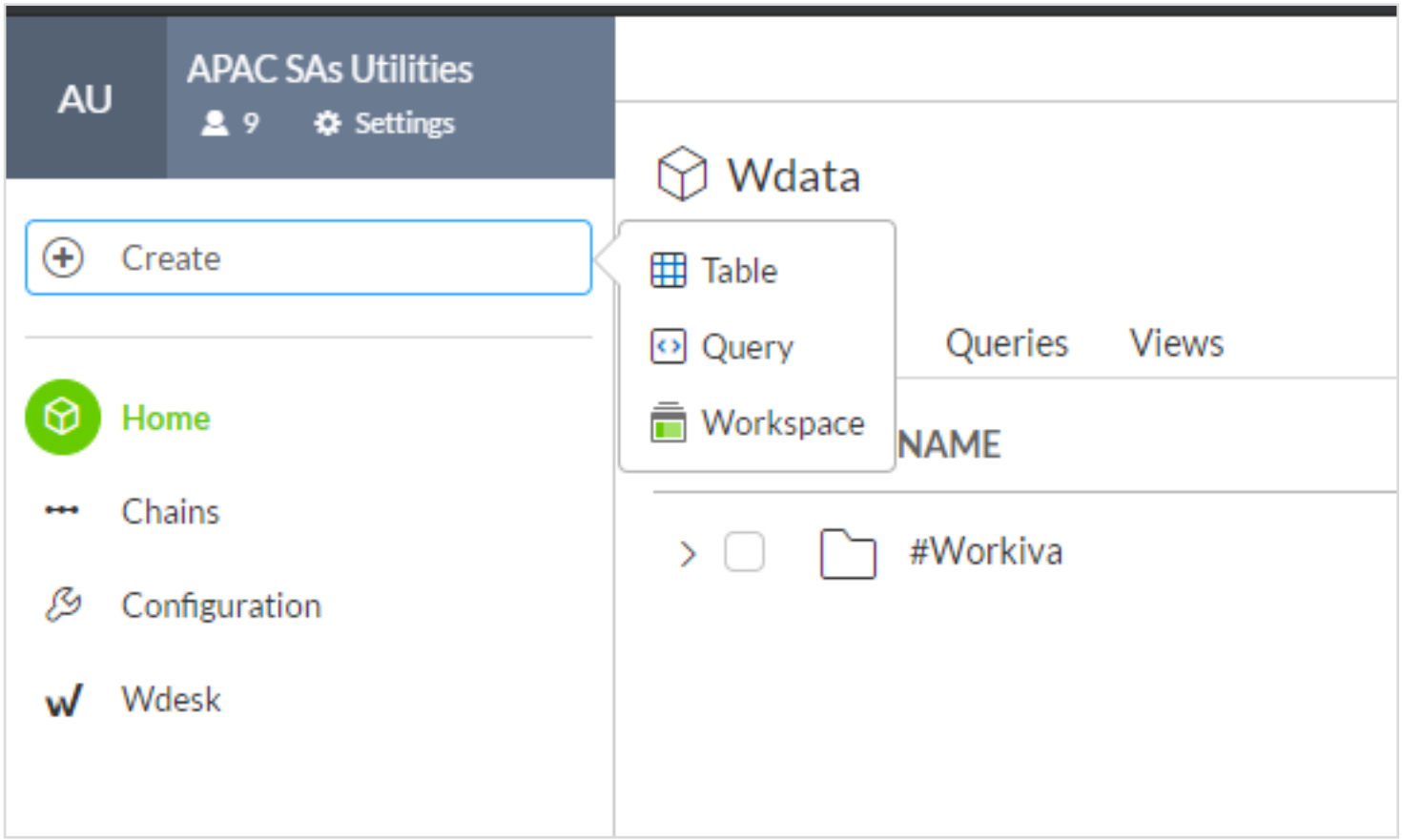
- Trial Balance
- Adjustments information

Dimension table - stores the main mapping data.

- Chart of Accounts (CoA)

Queries are created to join both source and dimension data to transform

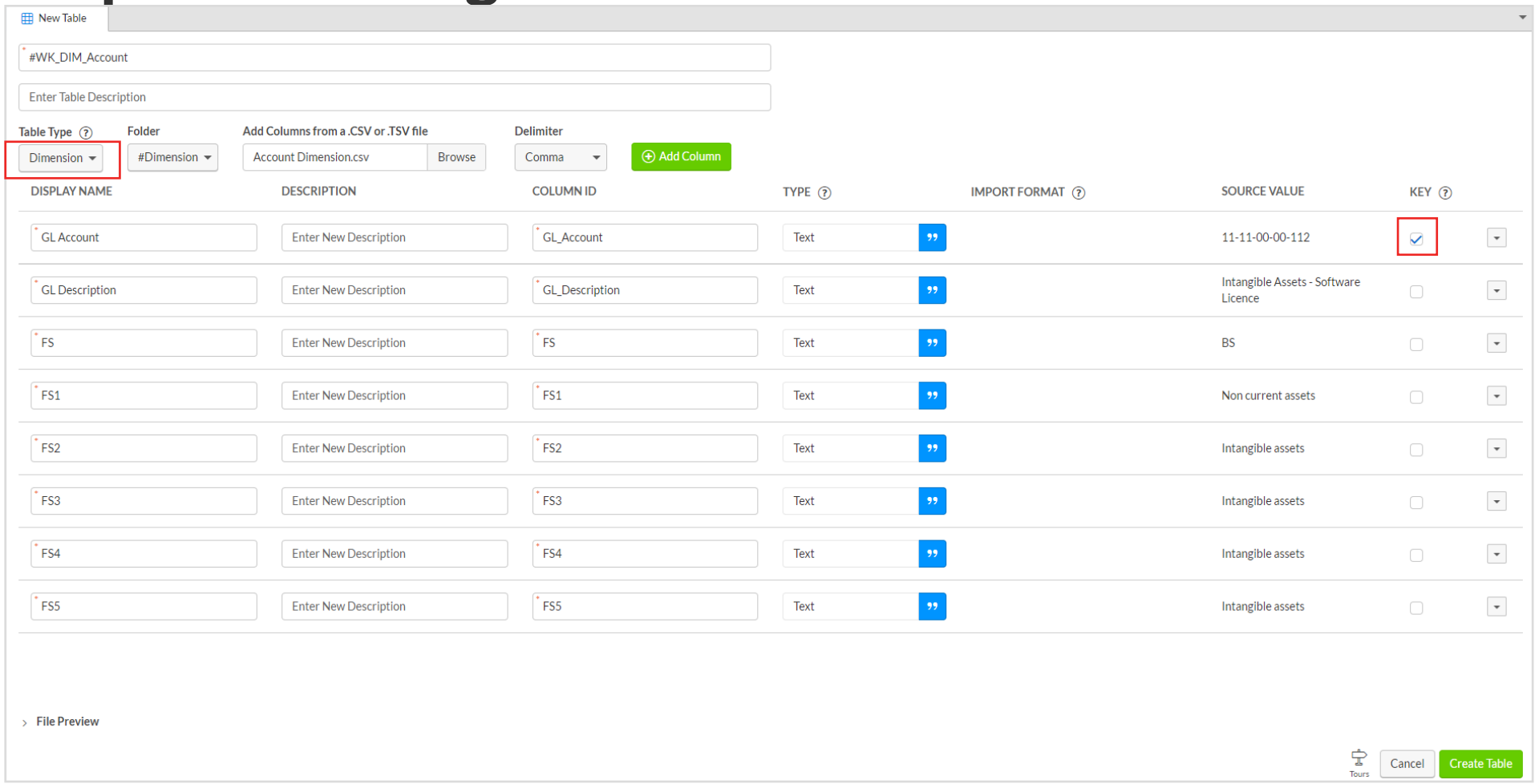
Step 5.1: Creating COA Table



To create a table, click on 'Create', select 'Table' on the left panel

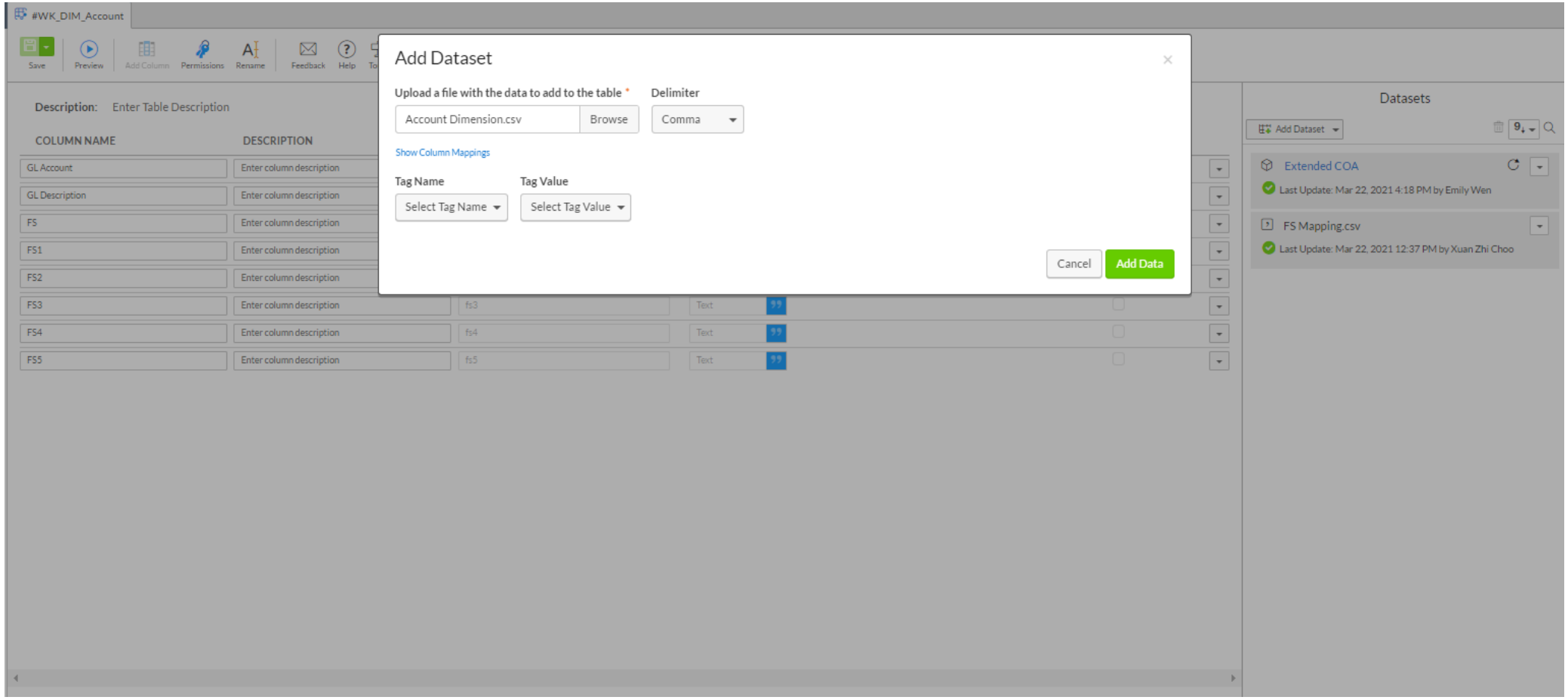
workiva

Step 5.2: Creating COA Table



To create a table, we would need to import the Account Dimension.csv file that is provided via the 'Browse'. The first table that would be created is a COA/Account Dimension table that stores the Accounts data, setting the 'GL Account' as the primary key.

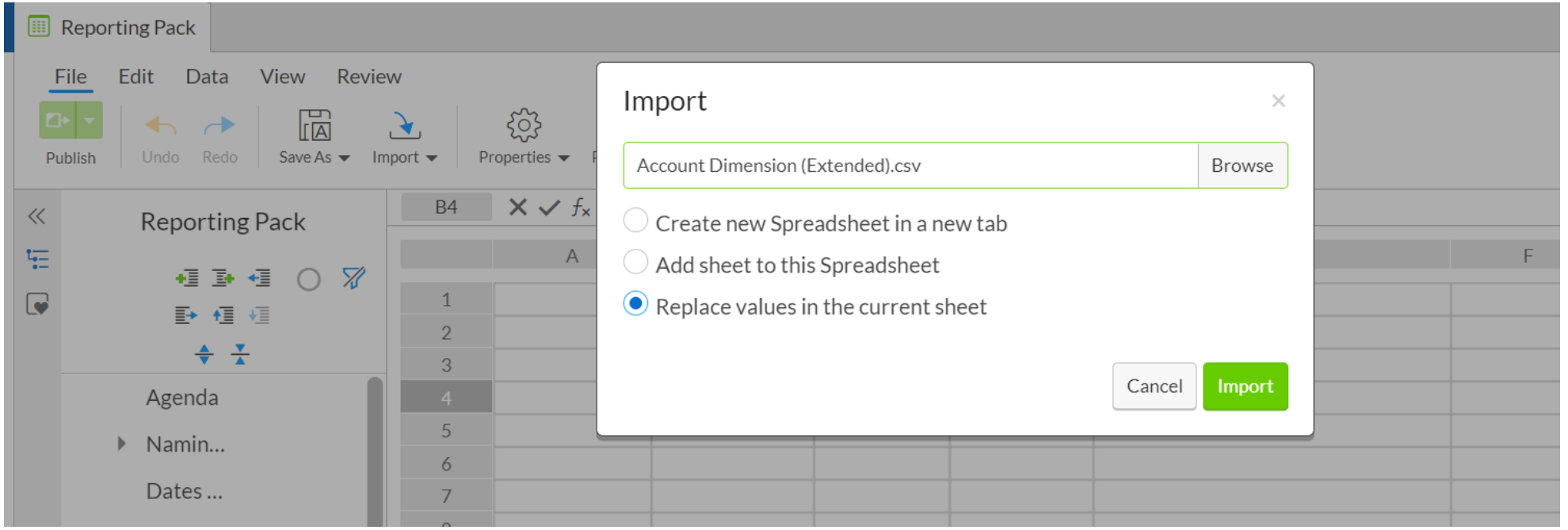
Step 5.3: Creating COA DIM Table P2



Once the table is created, you would be directed to the page where you would be required to 'Add Dataset'. Simply add the .csv file that was uploaded previously.

Workiva

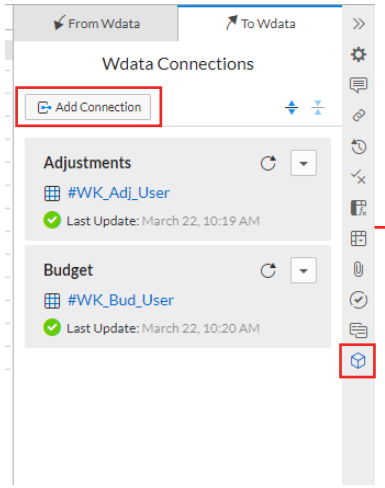
Step 5.4: Creating the Extended COA P1

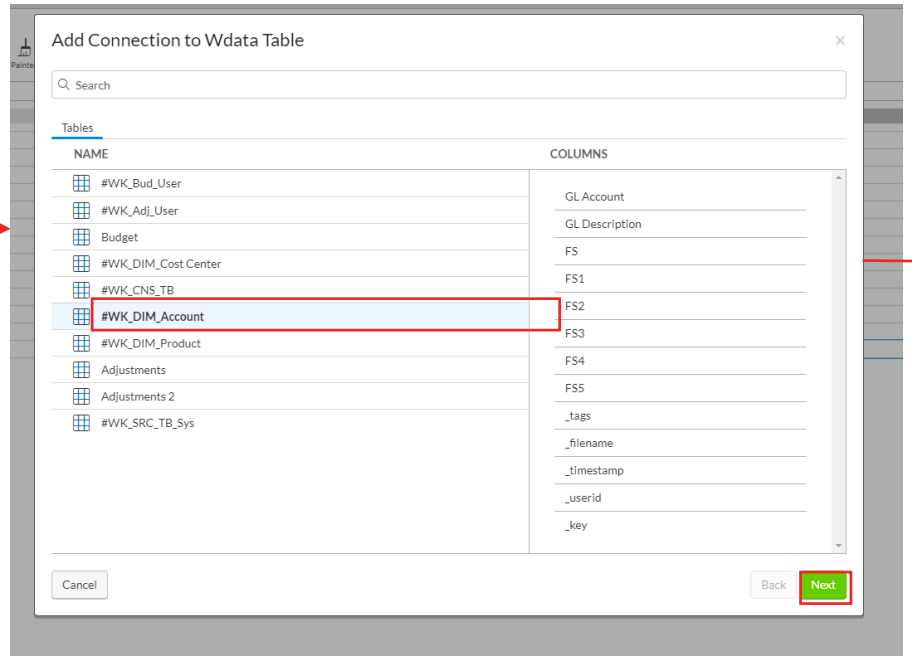


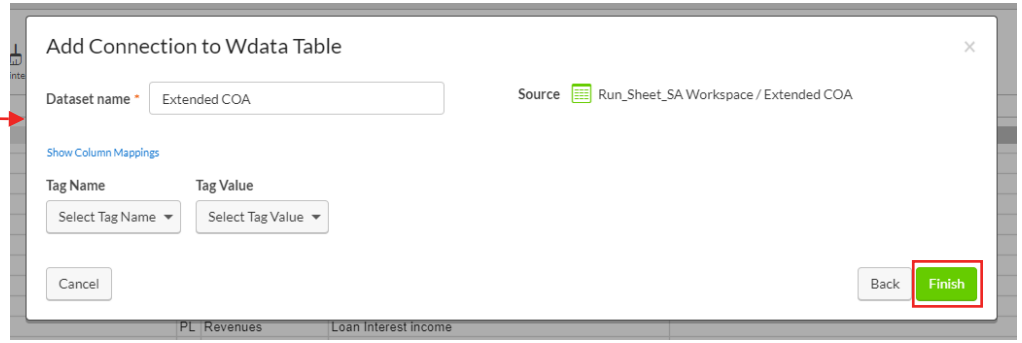
In order to create the staging query for the budget data, one would need to append the budget account codes in the COA DIM Table. To achieve this, import the file 'Account Dimension (Extended).csv" file into a blank Wdesk spreadsheet in the Reporting Pack. Name the sheet as "Extended COA". (Note: in most cases, the budget accounts are manually input by the users)

workiva

Step 5.5: Creating the Extended COA P2





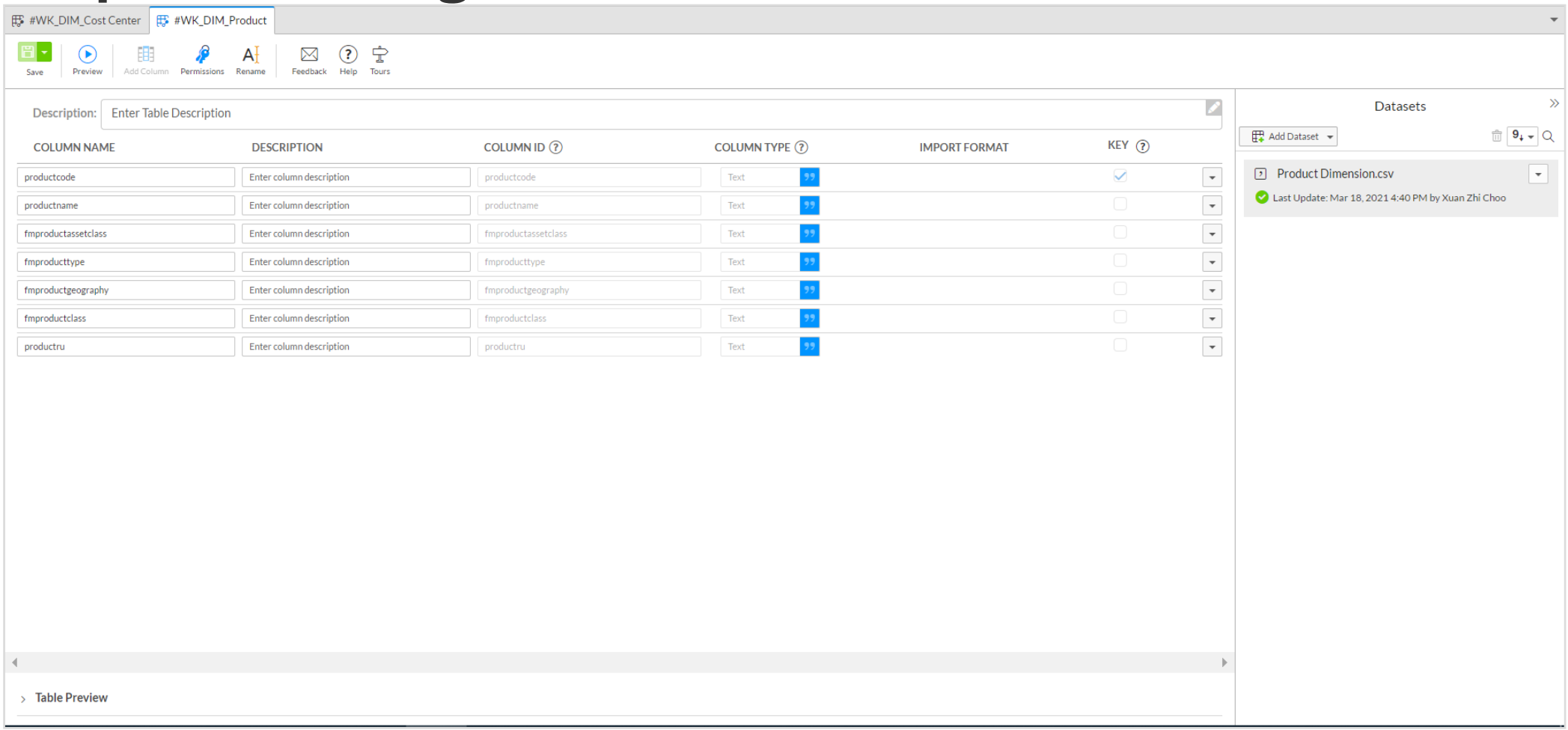


In the Spreadsheet, click the connection icon on the right panel and click add connection. Then, select table "#WK_DIM_Account" and click Next.

In the next page, click Finish and the Extended COA sheet is now appended to the COA DIM Table.

Workiva

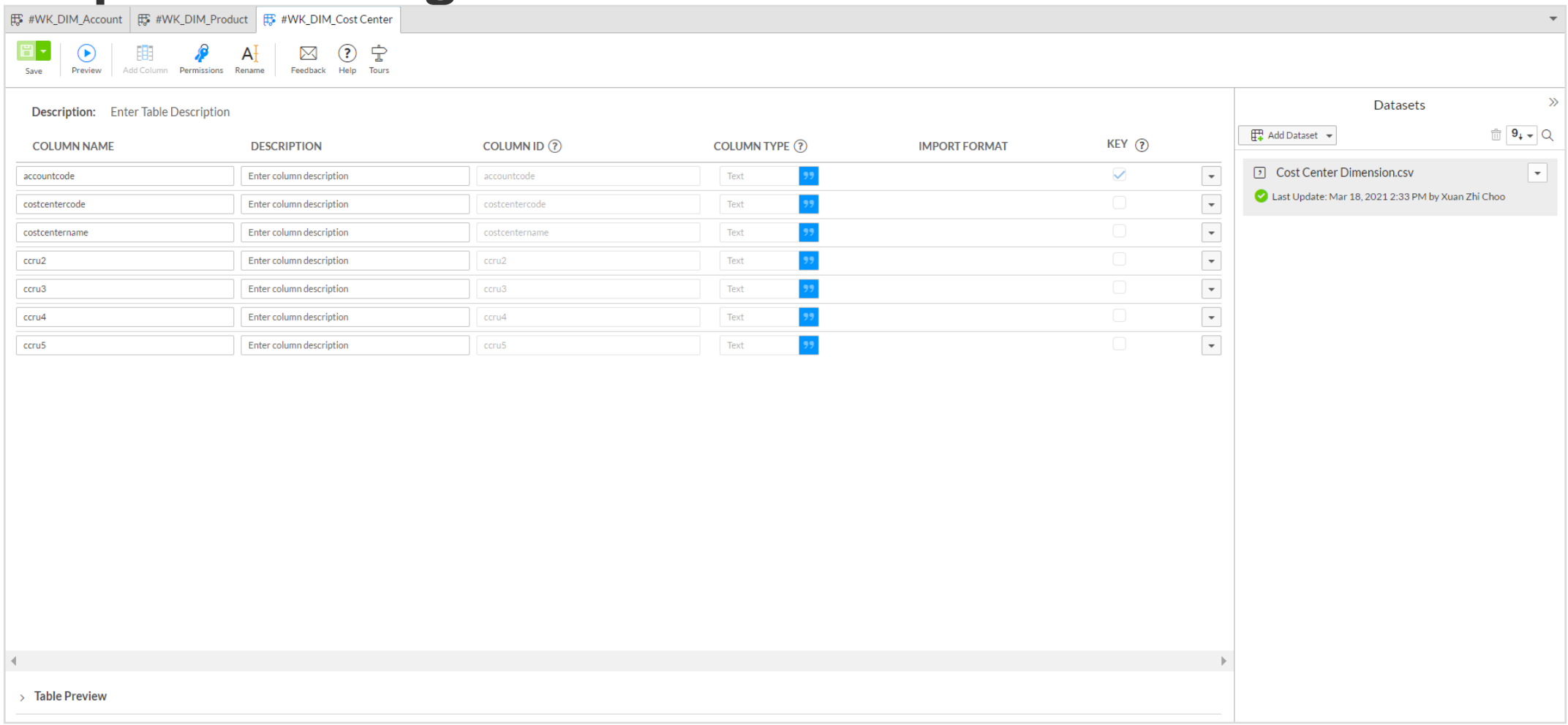
Step 5.6: Creating Product DIM Table



The next table that would be created is a Product Dimension table that stores the Product Mapping data, setting the 'Product Code' as the primary key.

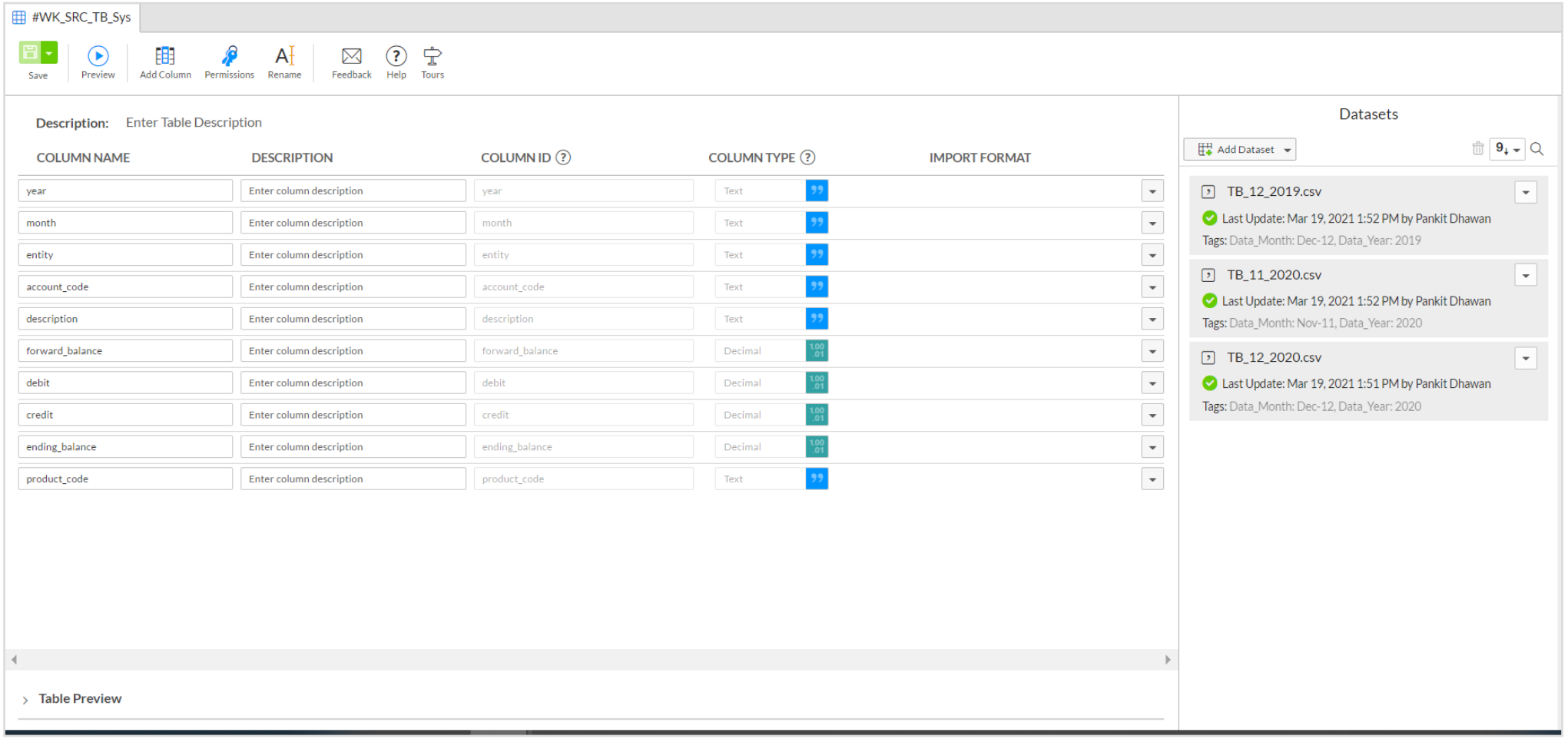
workiva

Step 5.7: Creating Cost Center DIM Table



The next table that would be created is a Cost Center Dimension table that stores the Cost Center data, setting the 'GL Account' as the primary key.

Step 6: Creating TB Source (SRC) Table



Creating a 'Data' type TB Table which would store data from the client's TB data, importing PY & CY Trial Balance data.

Workiva

Step 7: Objective for Creating Staging Query

↪ Consumption query helps us derive the month balances for both the Profit & Loss & Balance Sheet

P&L

- Always represented in number incurred for the month.
- For our example data, ending balance represents the closing balance for the months.
- Calc Representation Example:

FS Mapping	Jan_YTD	Feb_YTD	Mar_YTD	Apr_YTD	May_YTD
PL	Jan	Jan + Feb	Jan + Feb + Mar	Jan + Feb + Mar + Apr	...

BS

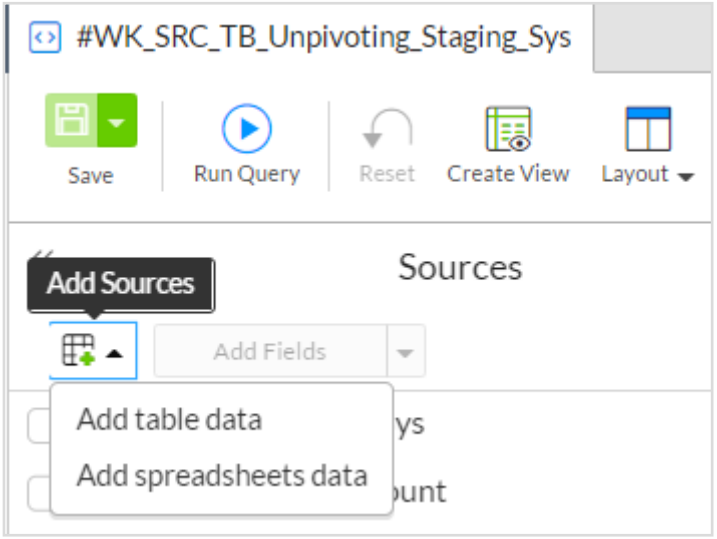
- Always represented in closing balance numbers.
- For our example data, ending balance represents the closing balance for the months.
- Calc Representation Example:

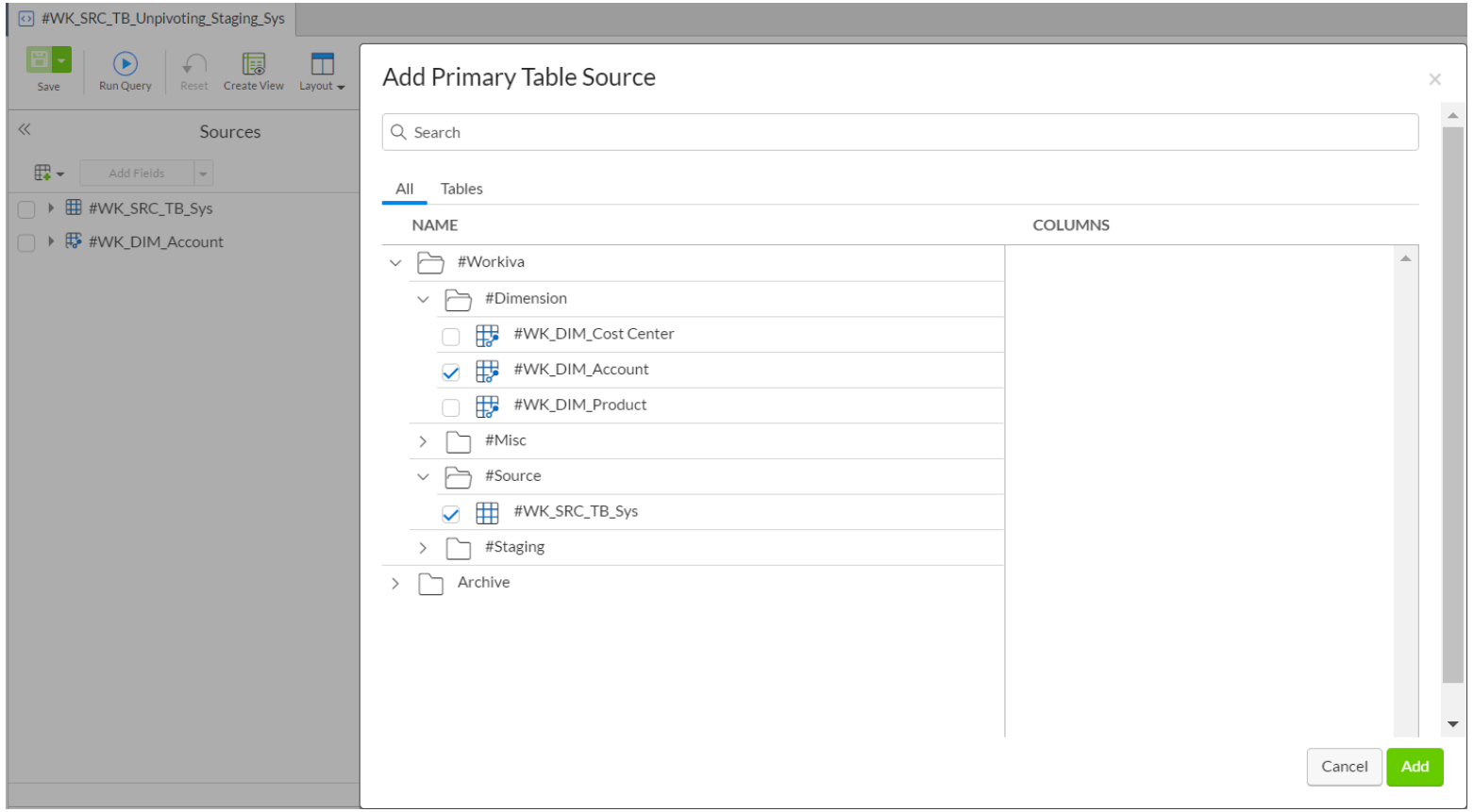
FS Mapping	Jan	Feb	Mar	Apr	May
BS	Jan	Feb	Mar	Apr	...

Step 7.1: Steps for creating the Staging Query

Steps	Description
1	First, build the query for Current Year.
2	Based on the Current Year query, replicate the query to retrieve Prior Year's data.
3	Do a 'Full Outer Join' for both Current & Prior Year's query.
4	As sometimes current year would not have data, we would use "Advance SQL" mode to get rid of the metadata to make it more presentable.
5	The following steps would be illustrated in the slides below.
6	**Naming Convention:** i) Queries & Tables with '_Sys' at the back of the name are system based, and should not be editable by users ii) Queries & Tables with '_User' at the back of the name are user type query, and should be editable by users

Step 7.2: Creating TB Staging Query - Adding table data source

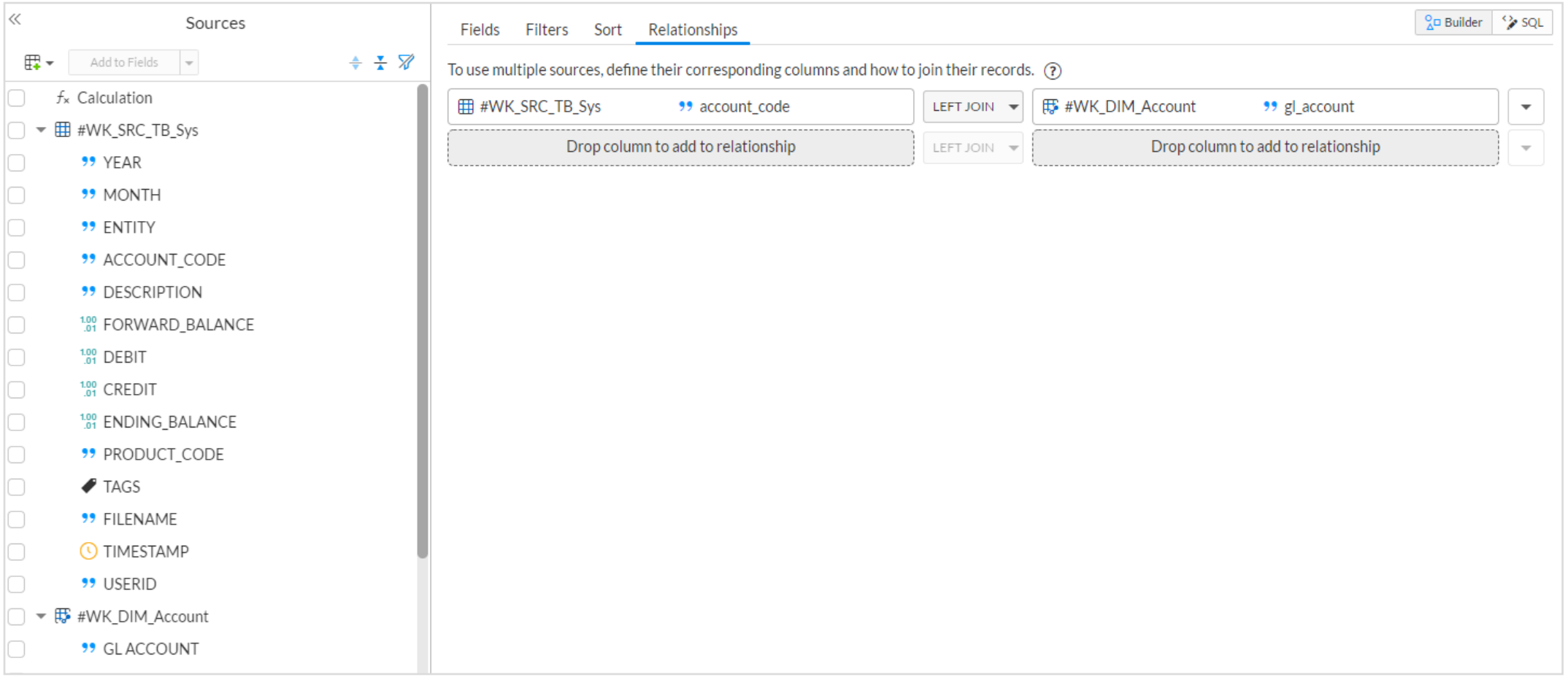




To create a new query, click on the 'Create' -> 'Query'. Upon creation of a new query from the left panel, add all the relevant data tables that would be used in the query. Name the query as "#WK_SRC_TB_Unpivoting_Staging_Sys". Note: Always hit 'Save'!

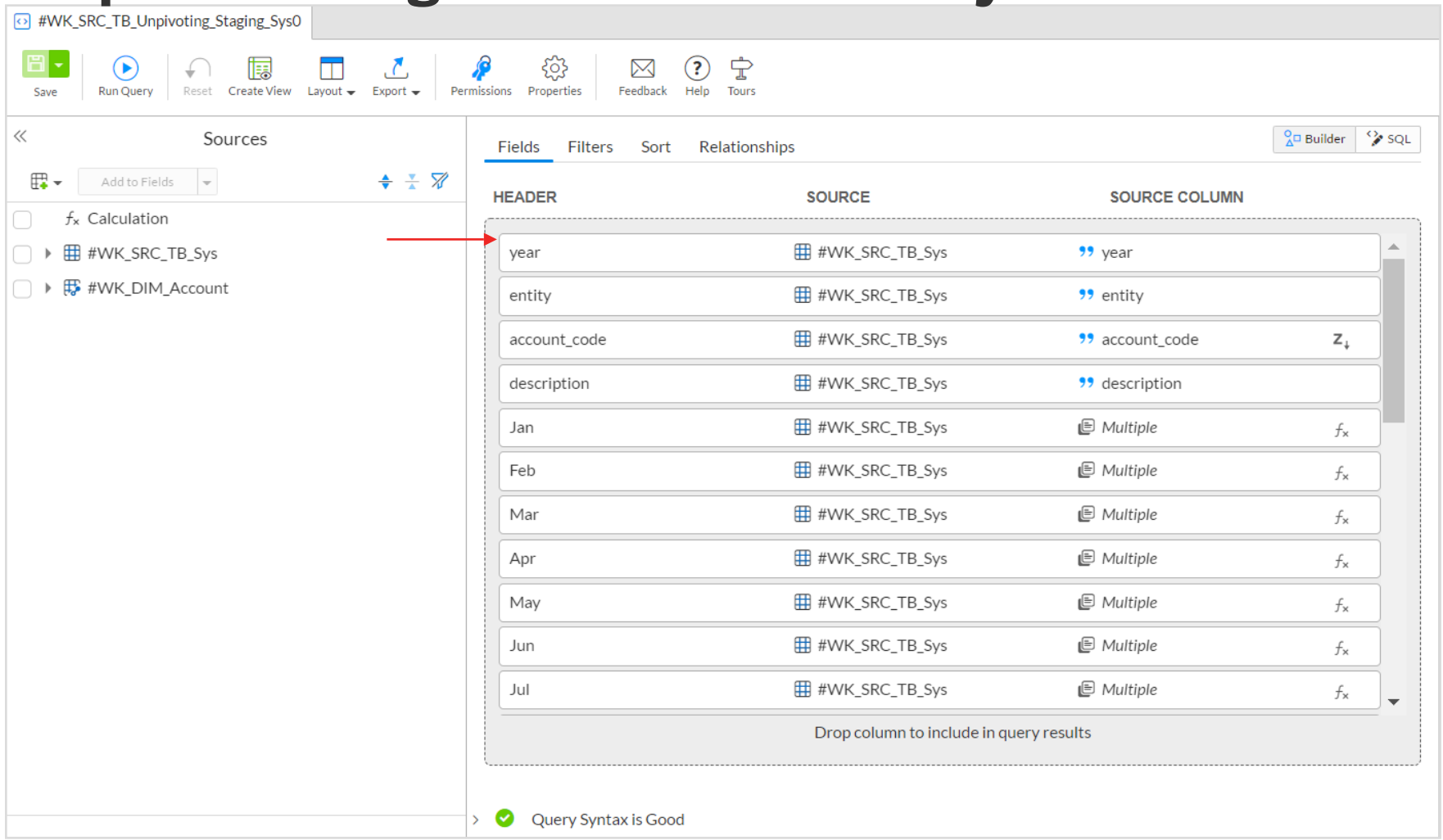
Workiva

Step 8: Establishing data table relationship



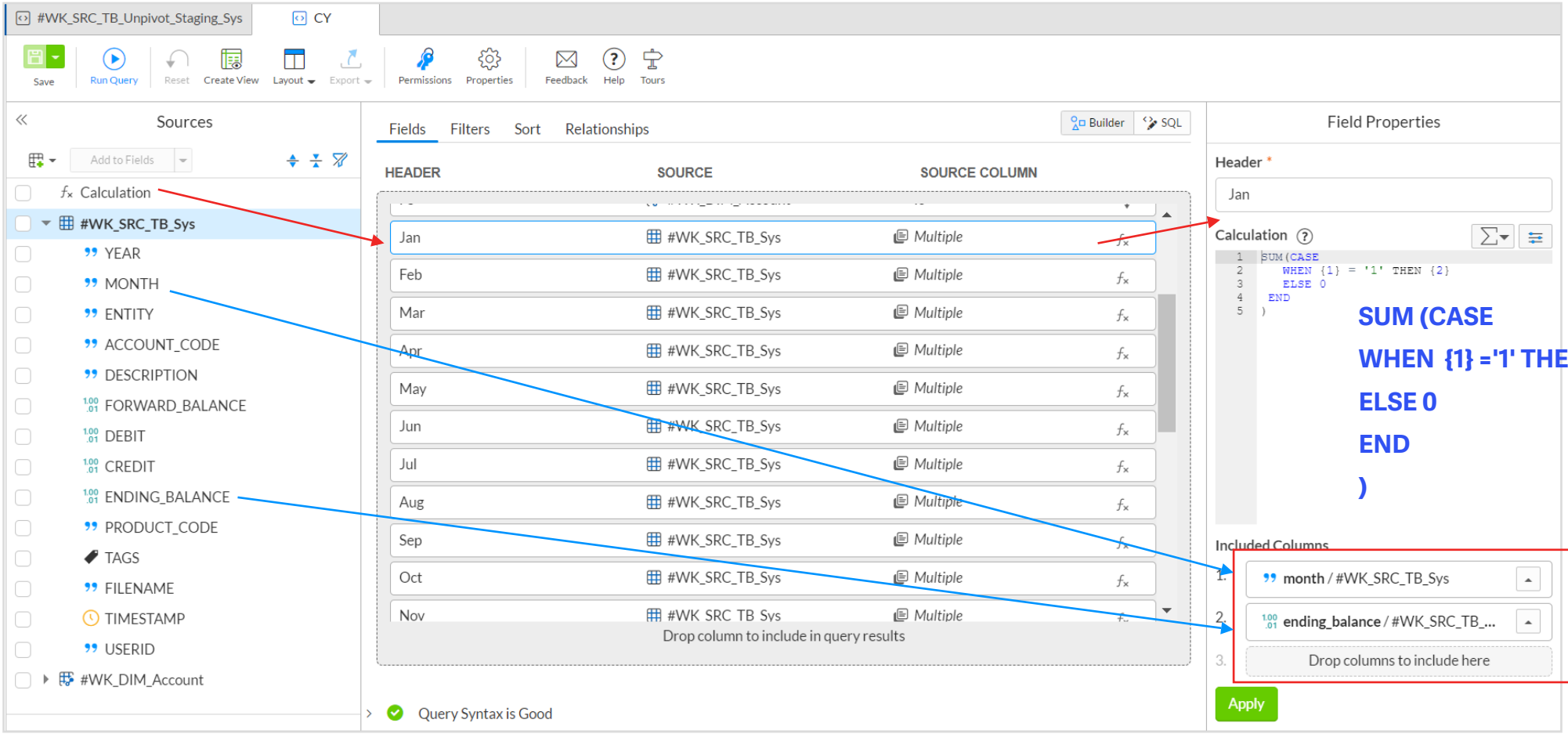
As there are 2 data tables, a relationship would need to be established between the "#WK_SRC_TB_Sys" & "#WK_Dim_Account" tables via a common identifier. The common identifier is the 'GL Account' code

Step 9: Building Fields for the Query



To insert fields in the query, tick the intended columns and drag it to the space under 'Fields'. In this step, the query would only be extracting Current Year's values. The following steps will illustrate how the Prior Year fields will be setup.

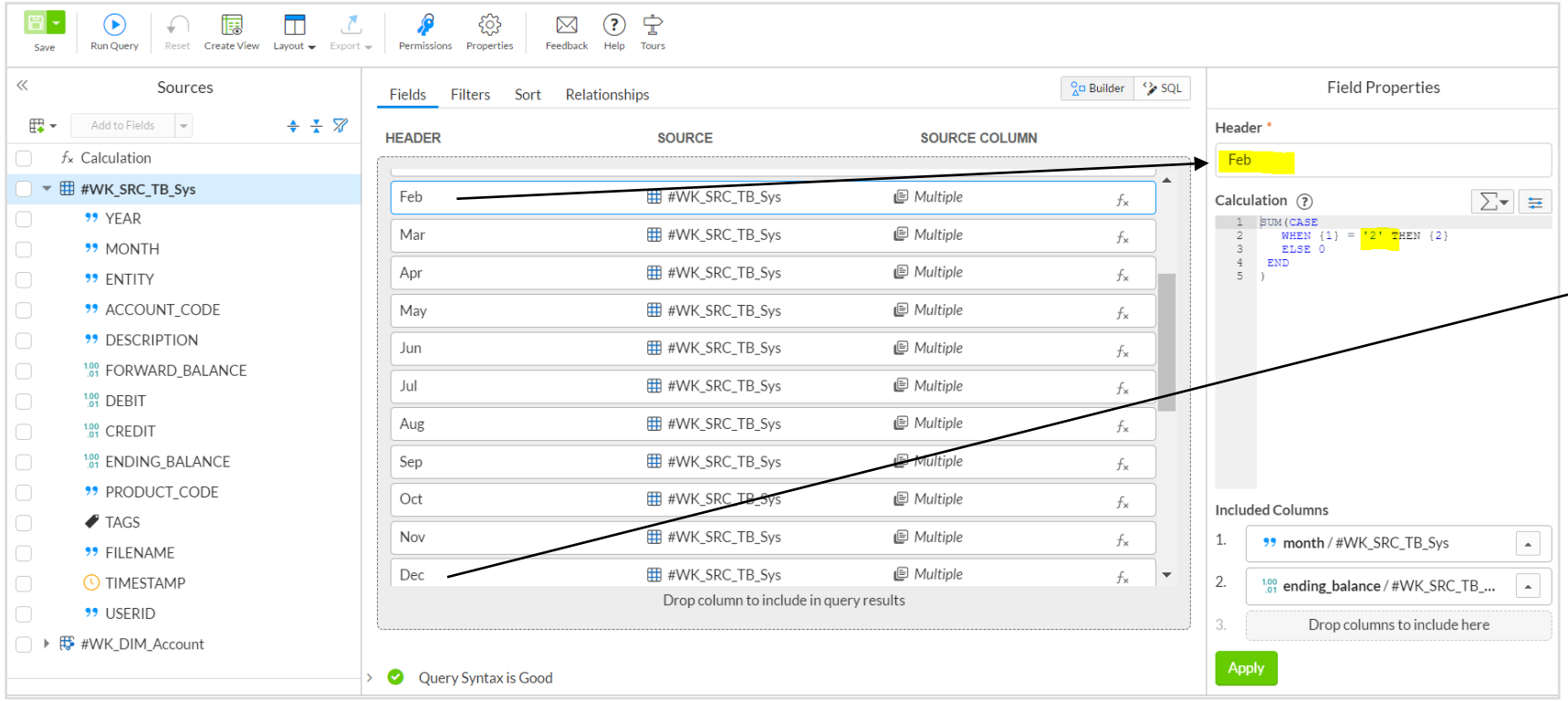
Step 9.1: Inserting Calculation Fields & Month Formula P1

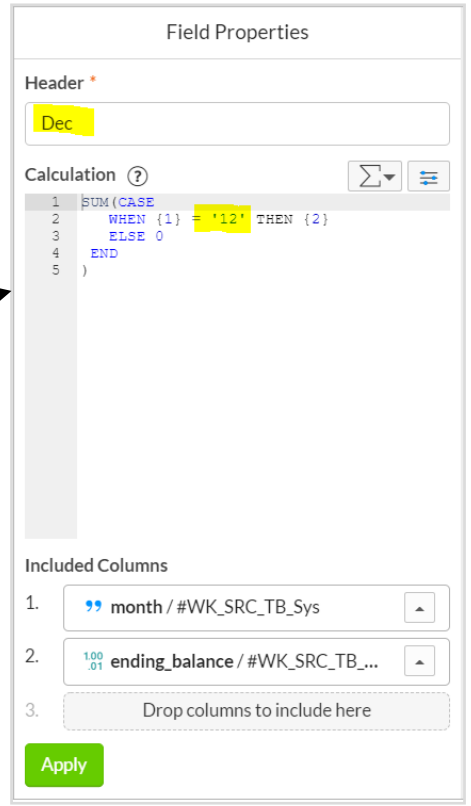


To calculate for 'Jan' monthly amount field, drag 'Calculation' into the Fields and name it 'Jan'.

Drag the " month, ending_balance" into the 'Included Columns'. With an SUM CASE formula, we instruct the code to identify the particular month (i.e. January) and display the balance for that month.

Step 9.2: Inserting Calculation Fields & Month Formula P2

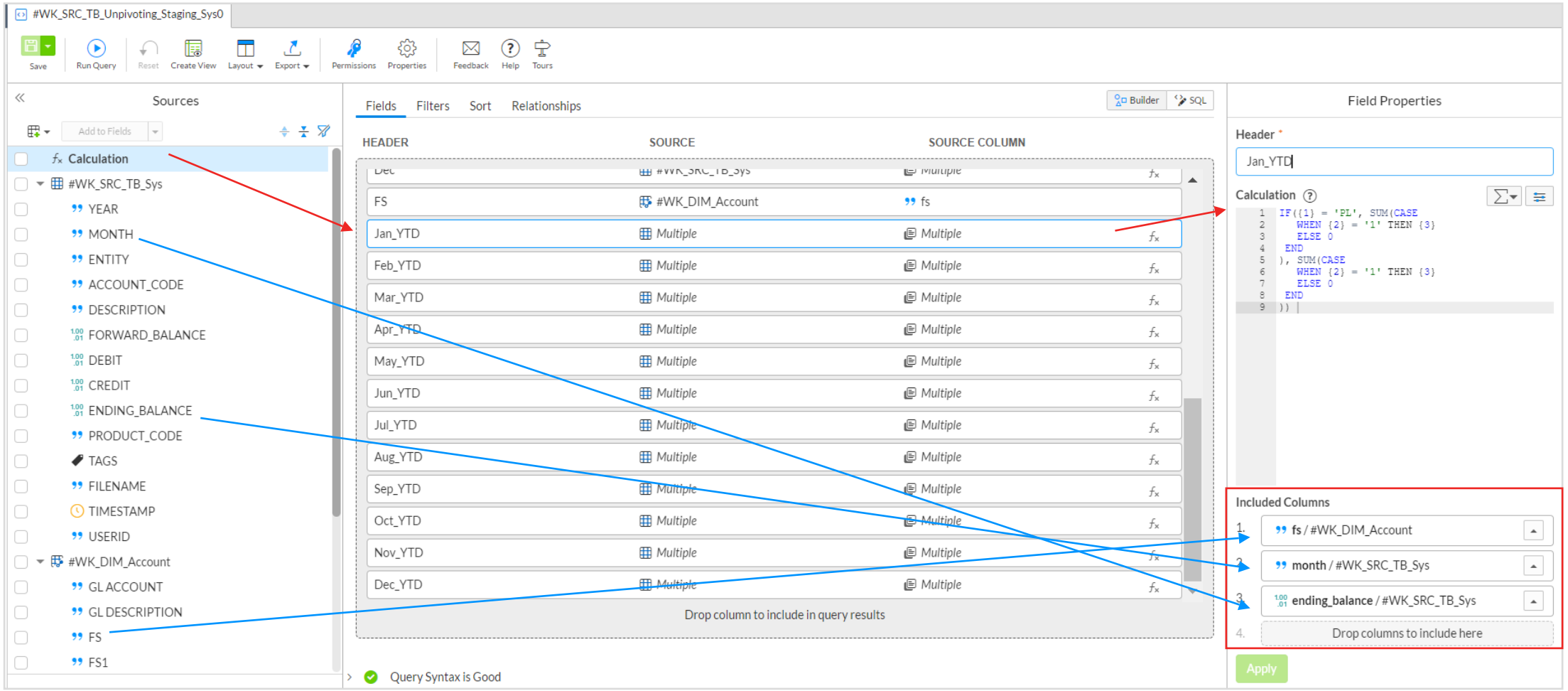




To calculate monthly amount fields for the remaining months, duplicate the "Jan" column and change the part of the calculation as highlighted above. (i.e. if we are creating the field for Feb, change from 1 to 2) Create the monthly amount fields for 12 months.

workiva

Step 10: Inserting Calculation Fields & YTD Formula P1

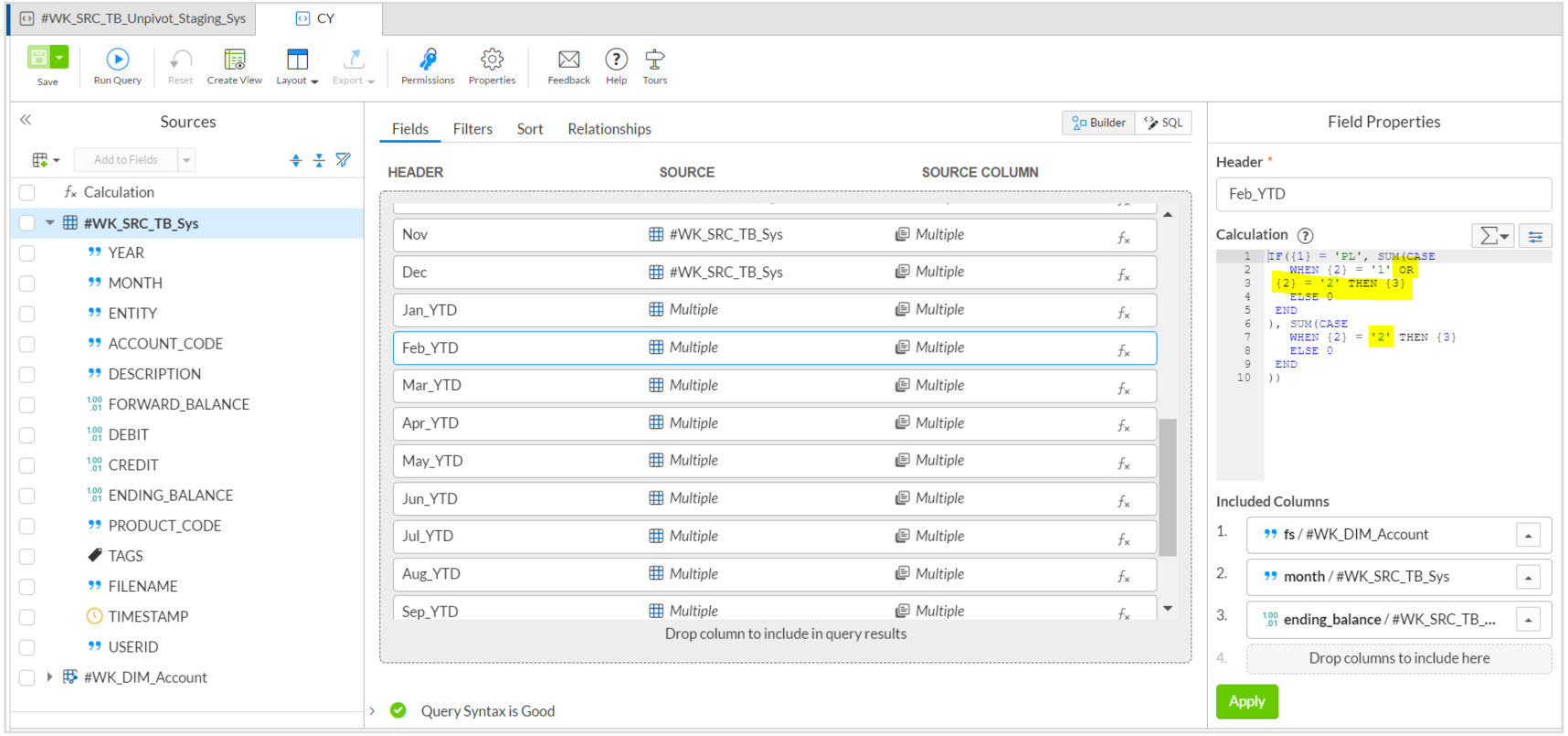


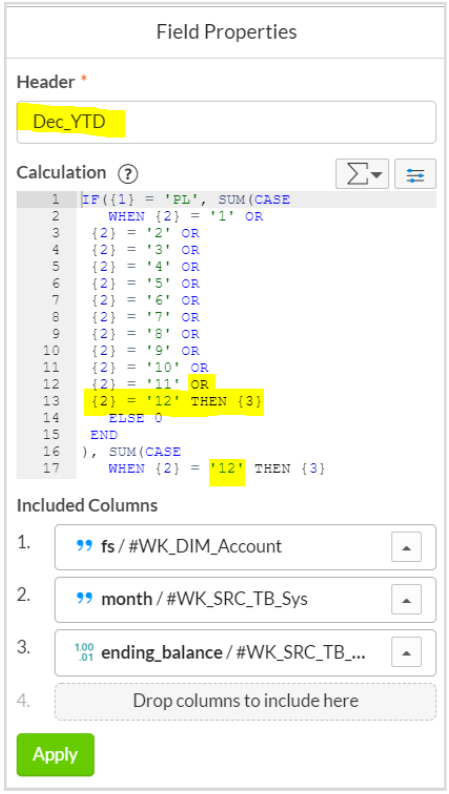

```
IF({1} = 'PL', SUM(CASE
  WHEN {2} = '1' THEN {3}
  ELSE 0
 END
), SUM(CASE
  WHEN {2} = '1' THEN {3}
  ELSE 0
 END
))
```

To calculate for 'Jan_YTD', drag 'Calculation' into the Fields and name it 'Jan_YTD'. Drag the "fs, month, ending_balance" into the 'Included Columns'. With an IF formula, we instruct the code to identify whether it is a PL/BS item, and if the month is January, display the balance for January.

Workiva

Step 10.1: Inserting Calculation Fields & YTD Formula P2

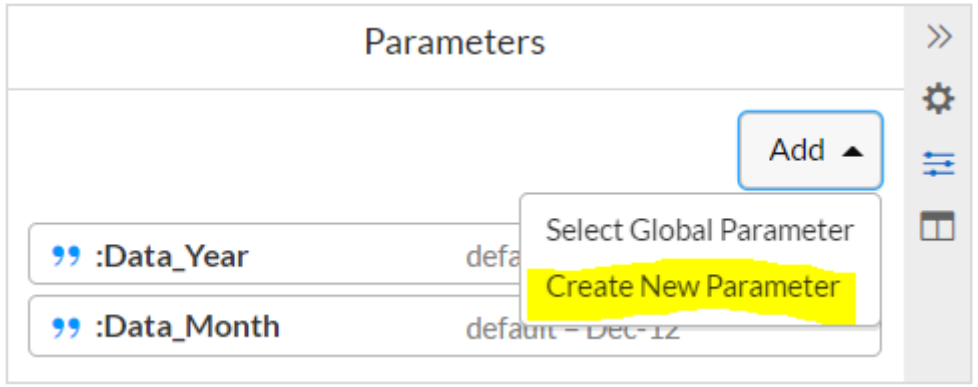


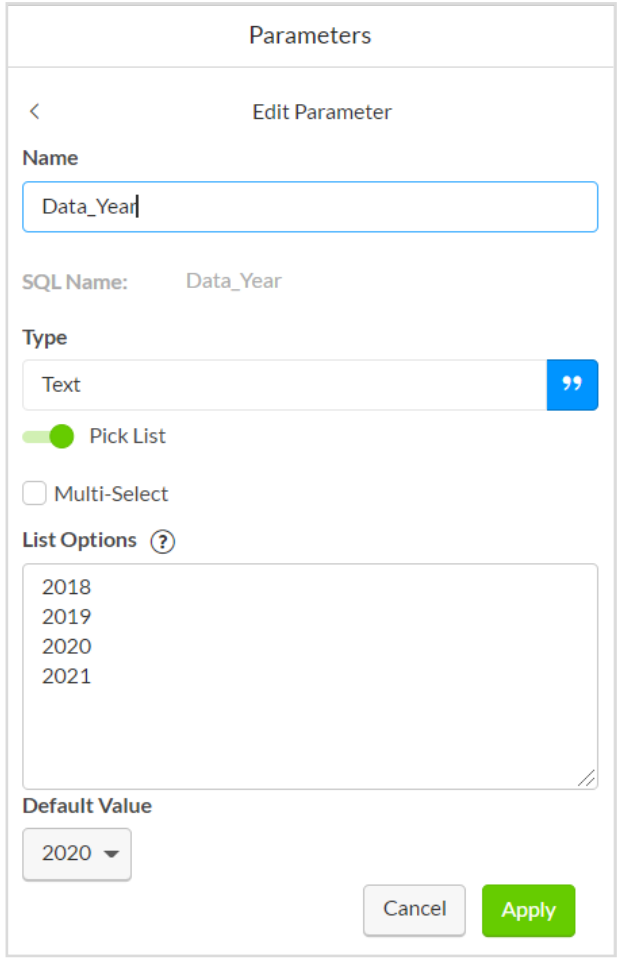


To calculate monthly YTD amount fields for the remaining months, duplicate the "Jan_YTD" column and change the part of the calculation as highlighted above. (i.e. if we are creating the field for Feb, add in another OR statement) Create the monthly YTD amount fields for 12 months.

Workiva

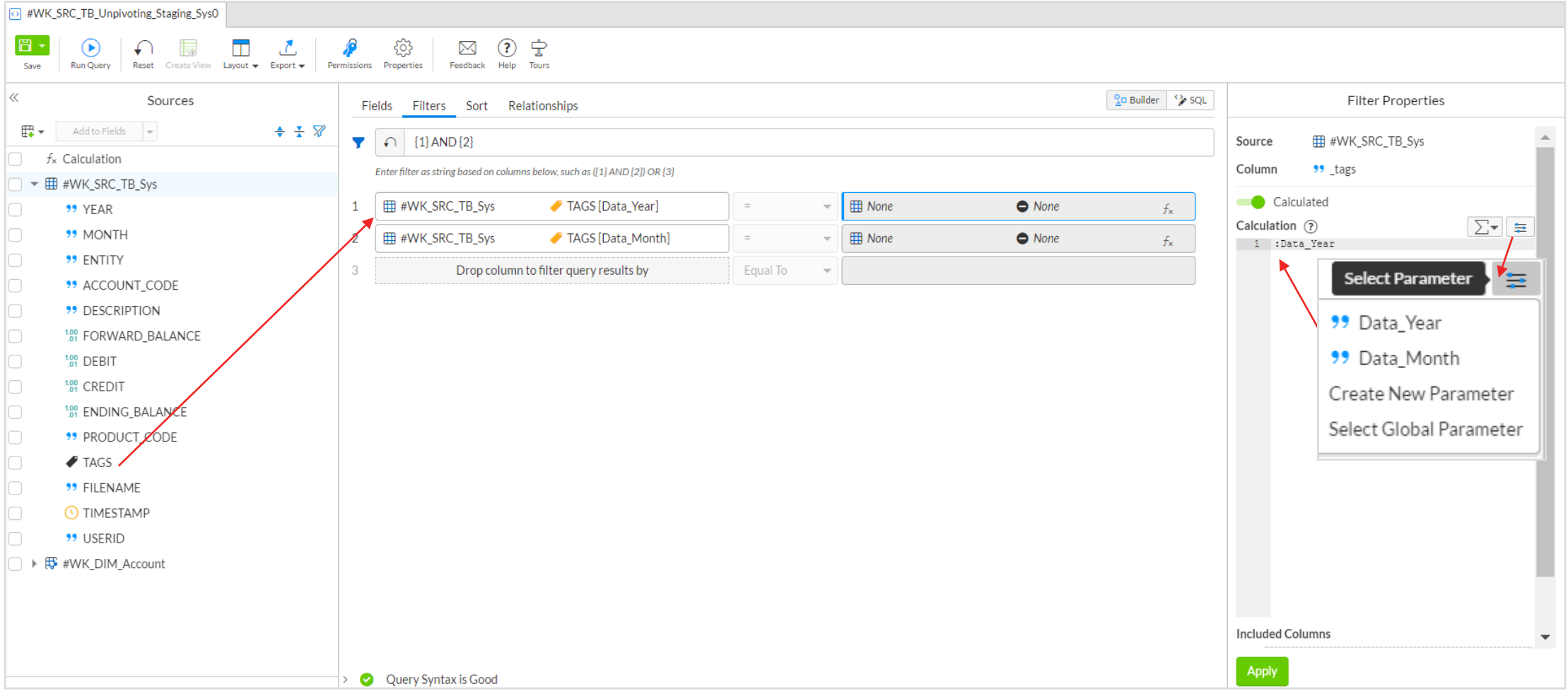
Step 11: Creating Parameters





Click on the Parameters icon at the top right of the screen, create 2 Parameters ('Data_Year' & 'Data_Month'). Ensure that the parameter names matches the Tags name which was created in Step 0.

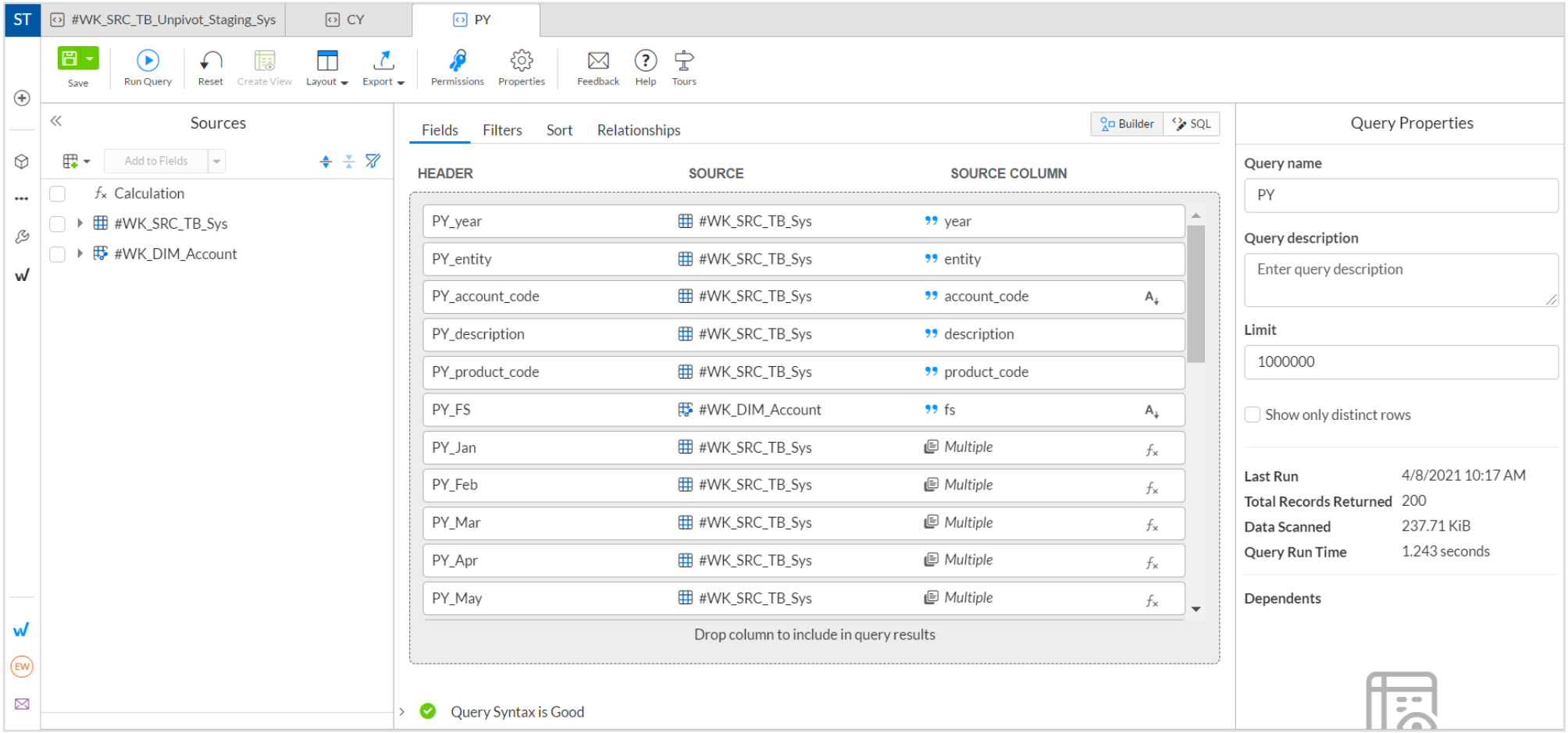
Step 11.1: Creating Filters



Create 2 filters ('Data_Year' & 'Data_Month'). Drag "Tags" into the 'Filters' and assign Filter values where 'Data_Year' = ":Data_Year" ; 'Data_Month = ":Data_Month"

Workiva

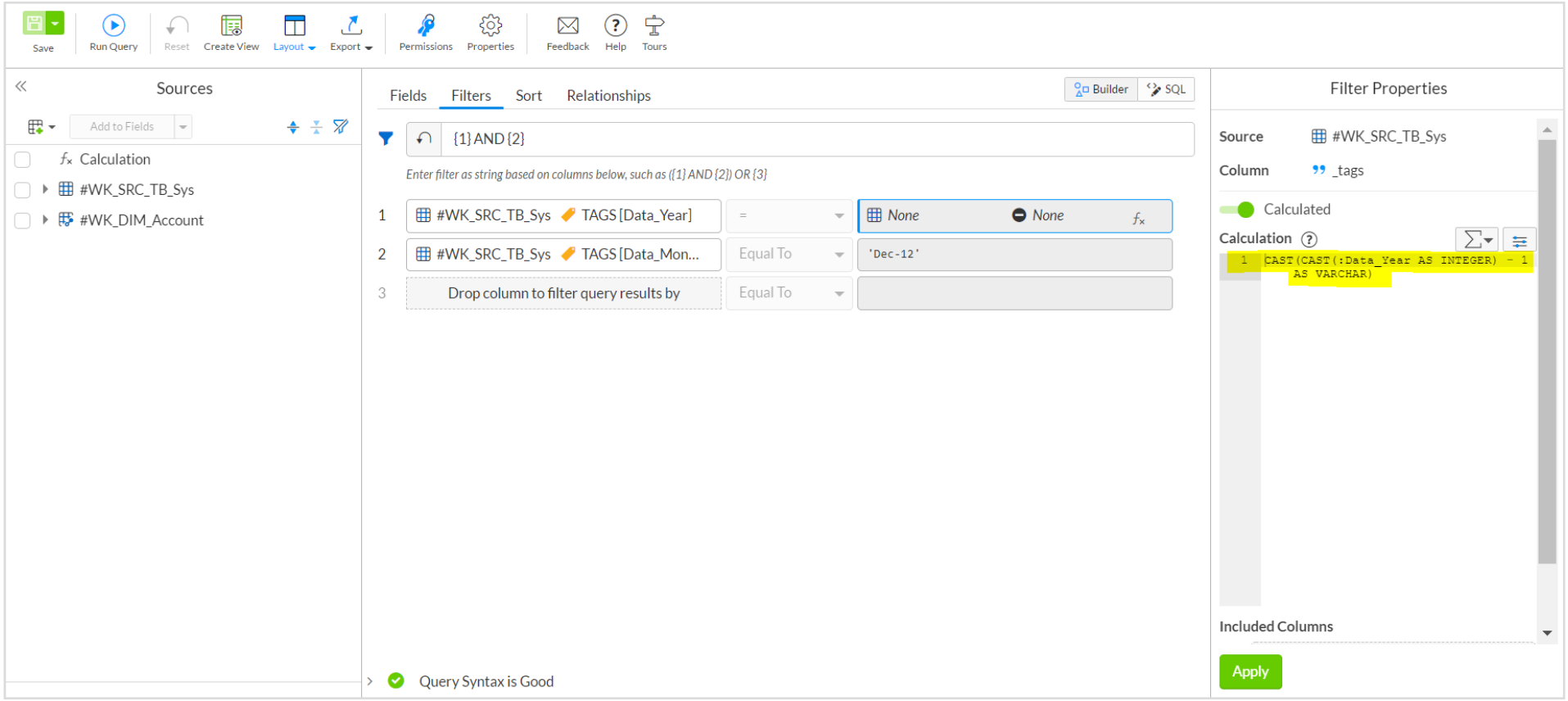
Step 12: Replicating for Prior Year's Data P1



After creating all the necessary columns, save this query as "CY". Copy the CY query and name it as "PY". In the PY query, change the front naming convention to begin with 'PY'; e.g. "PY_year".

Workiva

Step 12.1: Replicating for Prior Year's Data P2

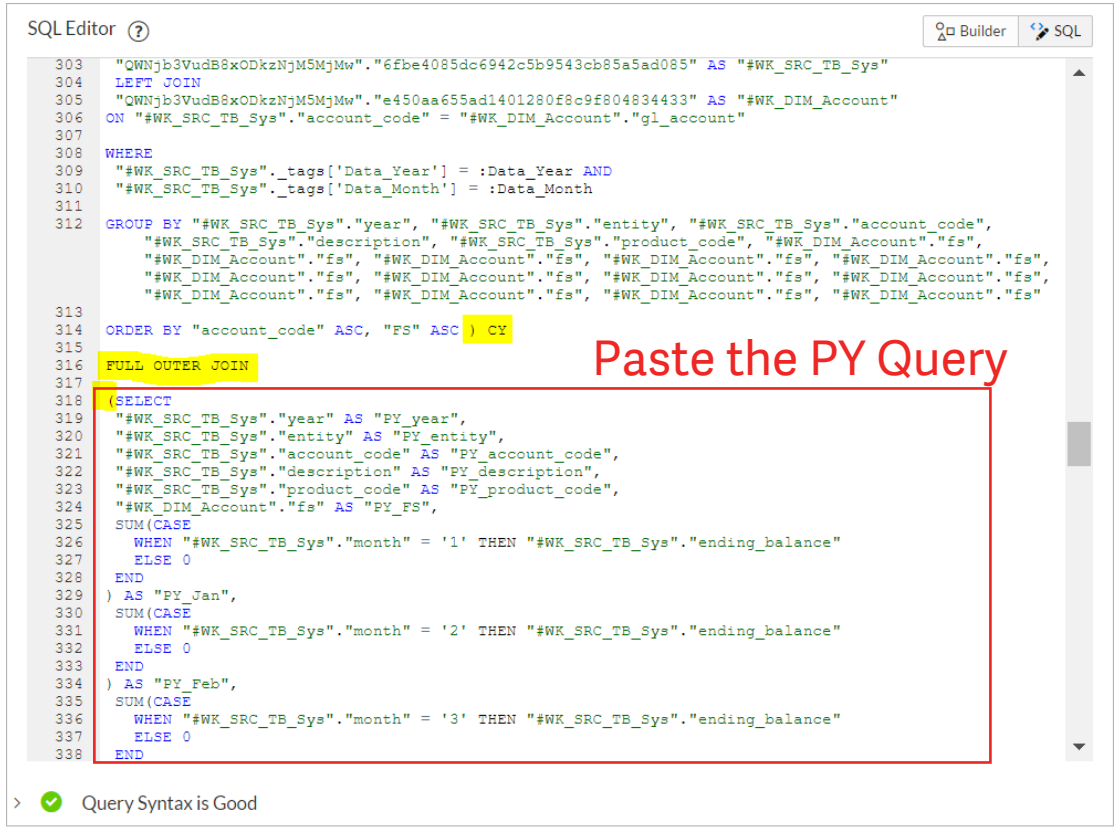


In the PY query, go to Filters and change the Filter Value for 'Data_Year' with the calculation highlighted above - 'CAST (CAST(:Data_Year AS INTEGER) -1 AS VARCHAR)'. Set the filter value for 'Data_Month' to always be as "Dec-12".

In here, we are essentially adding on the "WHERE" clause for the Prior Year's query to cater for prior year.

Workiva

Step 13: Joining Current & Prior Year Query P1




```
303   "QWNjb3VudB8xODkzNjM5MjMw"."6fbe4085dc6942c5b9543cb85a5ad085" AS "#WK_SRC_TB_Sys"
304   LEFT JOIN
305   "QWNjb3VudB8xODkzNjM5MjMw"."e450aa655ad1401280f8c9f804834433" AS "#WK_DIM_Account"
306  ON "#WK_SRC_TB_Sys"."account_code" = "#WK_DIM_Account"."gl_account"
307
308  WHERE
309   "#WK_SRC_TB_Sys"._tags['Data_Year'] = :Data_Year AND
310   "#WK_SRC_TB_Sys"._tags['Data_Month'] = :Data_Month
311
312  GROUP BY "#WK_SRC_TB_Sys"."year", "#WK_SRC_TB_Sys"."entity", "#WK_SRC_TB_Sys"."account_code",
       "#WK_SRC_TB_Sys"."description", "#WK_SRC_TB_Sys"."product_code", "#WK_DIM_Account"."fs",
       "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs",
       "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs",
       "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs"
313
314  ORDER BY "account_code" ASC, "FS" ASC ) CY
315
316  FULL OUTER JOIN
317
318  (SELECT
319   "#WK_SRC_TB_Sys"."year" AS "PY_year",
320   "#WK_SRC_TB_Sys"."entity" AS "PY_entity",
321   "#WK_SRC_TB_Sys"."account_code" AS "PY_account_code",
322   "#WK_SRC_TB_Sys"."description" AS "PY_description",
323   "#WK_SRC_TB_Sys"."product_code" AS "PY_product_code",
324   "#WK_DIM_Account"."fs" AS "PY_FS",
325   SUM(CASE
326     WHEN "#WK_SRC_TB_Sys"."month" = '1' THEN "#WK_SRC_TB_Sys"."ending_balance"
327     ELSE 0
328   END
329  ) AS "PY_Jan",
330   SUM(CASE
331     WHEN "#WK_SRC_TB_Sys"."month" = '2' THEN "#WK_SRC_TB_Sys"."ending_balance"
332     ELSE 0
333   END
334  ) AS "PY_Feb",
335   SUM(CASE
336     WHEN "#WK_SRC_TB_Sys"."month" = '3' THEN "#WK_SRC_TB_Sys"."ending_balance"
337     ELSE 0
338   END
```

In this step, we switch to 'SQL' Mode. A "FULL OUTER JOIN" is used to stitch both the Current Year query with Prior Year query. The join criteria would be a concatenation of 'entity' & 'GL Account Code' (this is illustrated in the next slide).

Go back to query "#WK_SRC_TB_Unpivoting_Staging_Sys" and scroll down to the bottom.

Delete the last row "LIMIT 1000000" and add in the code as highlighted in the bottom. Then, paste the code from query "PY" to do a full outer join.

Workiva

Step 13.1: Joining Current & Prior Year Query P2

SQL Editor

Builder | SQL

```
546   "#WK_SRC_TB_Sys"."month" = '2' or
547   "#WK_SRC_TB_Sys"."month" = '3' or
548   "#WK_SRC_TB_Sys"."month" = '4' or
549   "#WK_SRC_TB_Sys"."month" = '5' or
550   "#WK_SRC_TB_Sys"."month" = '6' or
551   "#WK_SRC_TB_Sys"."month" = '7' or
552   "#WK_SRC_TB_Sys"."month" = '8' or
553   "#WK_SRC_TB_Sys"."month" = '9' or
554   "#WK_SRC_TB_Sys"."month" = '10' or
555   "#WK_SRC_TB_Sys"."month" = '11' or
556   "#WK_SRC_TB_Sys"."month" = '12' THEN "#WK_SRC_TB_Sys"."ending_balance"
557     ELSE 0
558  END
559 ), SUM(CASE
560     WHEN "#WK_SRC_TB_Sys"."month" = '12' THEN "#WK_SRC_TB_Sys"."ending_balance"
561     ELSE 0
562  END
563 )) AS "PY_Dec_YTD"
564
565 FROM
566  "QWNjb3VudB8xOTA1MjA5MjE1"."e36418c3a92846debcebbb3f3708a555" AS "#WK_SRC_TB_Sys"
567  LEFT JOIN
568  "QWNjb3VudB8xOTA1MjA5MjE1"."817b61205a154bd98c835c294cd33240" AS "#WK_DIM_Account"
569 ON "#WK_SRC_TB_Sys"."account_code" = "#WK_DIM_Account"."gl_account"
570
571 WHERE
572  "#WK_SRC_TB_Sys"._tags['Data_Year'] = CAST(CAST(:Data_Year AS INTEGER) - 1 AS VARCHAR) AND
573  "#WK_SRC_TB_Sys"._tags['Data_Month'] = 'Dec-12'
574
575 GROUP BY "#WK_SRC_TB_Sys"."year", "#WK_SRC_TB_Sys"."entity", "#WK_SRC_TB_Sys"."account_code",
        "#WK_SRC_TB_Sys"."description","#WK_SRC_TB_Sys"."product_code", "#WK_DIM_Account"."fs",
        "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs",
        "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs",
        "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs"
576
577 ORDER BY "PY_account_code" DESC ) PY
578
579 ON CONCAT (CY."entity" , CY."account_code" , CY."product_code") = CONCAT( PY."PY_entity" , PY
      ."PY_account_code" , PY."PY_product_code")
580
```

Query Syntax is Good

Go to the bottom of the combined query and delete the last row "LIMIT 1000000".

Add in the code highlighted on the left.

In here, we are joining the two sets of codes based on the concatenation of 'entity' & 'GL Account Code'.

workiva

Step 14: Unpivoting Months Columns

SQL Editor

```
SELECT

SUM(CY."Jan")as"Jan",
SUM(CY."Feb")as"Feb",
SUM(CY."Mar")as"Mar",
SUM(CY."Apr")as"Apr",
SUM(CY."May")as"May",
SUM(CY."Jun")as"Jun",
SUM(CY."Jul")as"Jul",
SUM(CY."Aug")as"Aug",
SUM(CY."Sep")as"Sep",
SUM(CY."Oct")as"Oct",
SUM(CY."Nov")as"Nov",
SUM(CY."Dec")as"Dec",
SUM(CY."Jan_YTD")as"Jan_YTD",
SUM(CY."Feb_YTD")as"Feb_YTD",
SUM(CY."Mar_YTD")as"Mar_YTD",
SUM(CY."Apr_YTD")as"Apr_YTD",
SUM(CY."May_YTD")as"May_YTD",
SUM(CY."Jun_YTD")as"Jun_YTD",
SUM(CY."Jul_YTD")as"Jul_YTD",
SUM(CY."Aug_YTD")as"Aug_YTD",
SUM(CY."Sep_YTD")as"Sep_YTD",
SUM(CY."Oct_YTD")as"Oct_YTD",
SUM(CY."Nov_YTD")as"Nov_YTD",
SUM(CY."Dec_YTD")as"Dec_YTD",
SUM(PY."PY_Jan")as"PY_Jan",
SUM(PY."PY_Feb")as"PY_Feb",
SUM(PY."PY_Mar")as"PY_Mar",
SUM(PY."PY_Apr")as"PY_Apr",
SUM(PY."PY_May")as"PY_May",
SUM(PY."PY_Jun")as"PY_Jun",
SUM(PY."PY_Jul")as"PY_Jul",
SUM(PY."PY_Aug")as"PY_Aug",
SUM(PY."PY_Sep")as"PY_Sep",
SUM(PY."PY_Oct")as"PY_Oct",
SUM(PY."PY_Nov")as"PY_Nov",
SUM(PY."PY_Dec")as"PY_Dec",
SUM(PY."PY_Jan_YTD")as"PY_Jan_YTD",
SUM(PY."PY_Feb_YTD")as"PY_Feb_YTD",
SUM(PY."PY_Mar_YTD")as"PY_Mar_YTD",
SUM(PY."PY_Apr_YTD")as"PY_Apr_YTD",
```

Query Syntax is Good

Builder SQL

SQL Editor

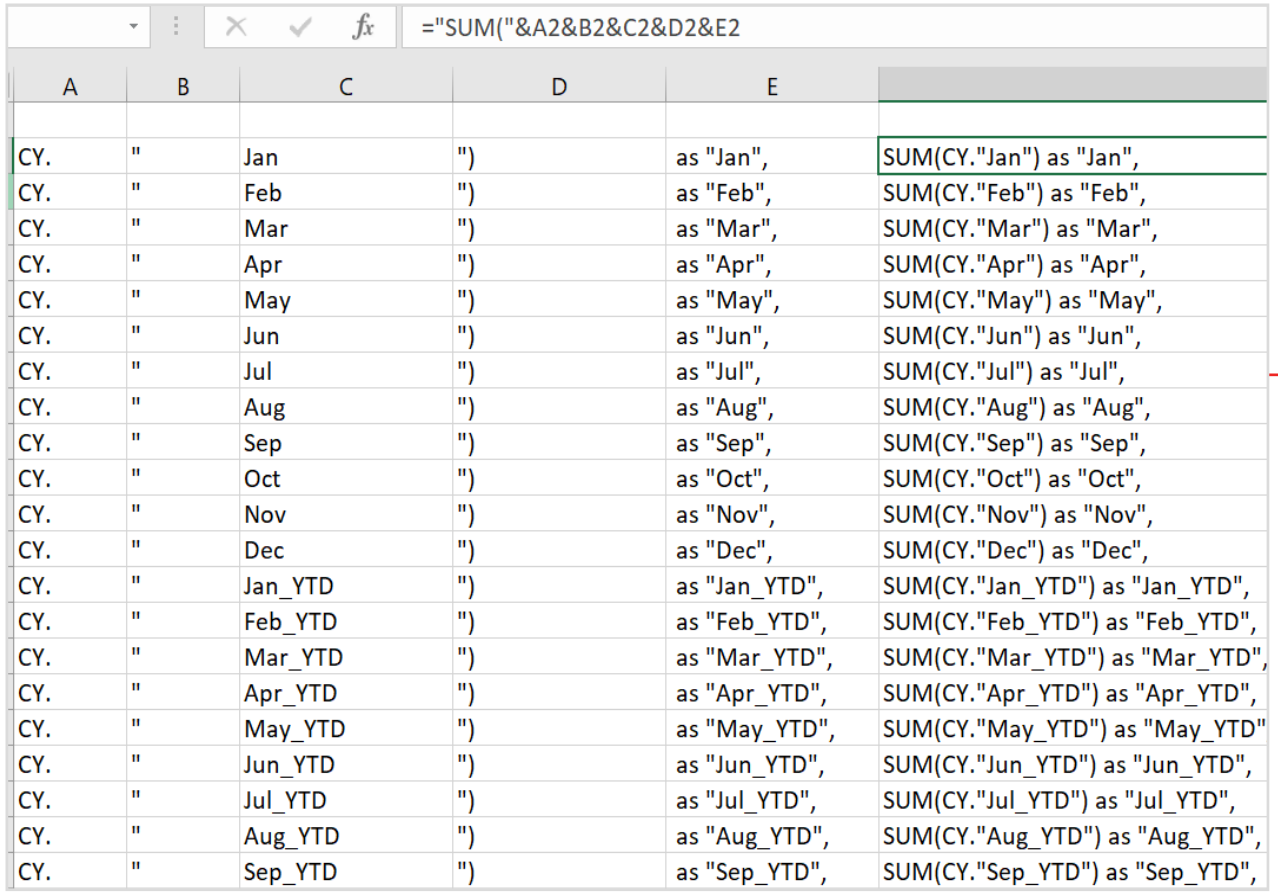
```
SUM(PY."PY_Aug")as"PY_Aug",
SUM(PY."PY_Sep")as"PY_Sep",
SUM(PY."PY_Oct")as"PY_Oct",
SUM(PY."PY_Nov")as"PY_Nov",
SUM(PY."PY_Dec")as"PY_Dec",
SUM(PY."PY_Jan_YTD")as"PY_Jan_YTD",
SUM(PY."PY_Feb_YTD")as"PY_Feb_YTD",
SUM(PY."PY_Mar_YTD")as"PY_Mar_YTD",
SUM(PY."PY_Apr_YTD")as"PY_Apr_YTD",
SUM(PY."PY_May_YTD")as"PY_May_YTD",
SUM(PY."PY_Jun_YTD")as"PY_Jun_YTD",
SUM(PY."PY_Jul_YTD")as"PY_Jul_YTD",
SUM(PY."PY_Aug_YTD")as"PY_Aug_YTD",
SUM(PY."PY_Sep_YTD")as"PY_Sep_YTD",
SUM(PY."PY_Oct_YTD")as"PY_Oct_YTD",
SUM(PY."PY_Nov_YTD")as"PY_Nov_YTD",
SUM(PY."PY_Dec_YTD")as"PY_Dec_YTD"

FROM
(SELECT
 "#WK_SRC_TB_Sys"."year" AS "year",
 "#WK_SRC_TB_Sys"."entity" AS "entity",
 "#WK_SRC_TB_Sys"."account_code" AS "account_code",
 "#WK_SRC_TB_Sys"."description" AS "description",
 "#WK_SRC_TB_Sys"."product_code" AS "product_code",
 "#WK_DIM_Account"."fs" AS "FS",
 SUM(CASE
   WHEN "#WK_SRC_TB_Sys"."month" = '1' THEN "#WK_SRC_TB_Sys"."ending_balance"
   ELSE 0
 END
) AS "Jan",
 SUM(CASE
   WHEN "#WK_SRC_TB_Sys"."month" = '2' THEN "#WK_SRC_TB_Sys"."ending_balance"
   ELSE 0
 END
) AS "Feb",
 SUM(CASE
   WHEN "#WK_SRC_TB_Sys"."month" = '3' THEN "#WK_SRC_TB_Sys"."ending_balance"
   ELSE 0
 END
) AS "Mar",
 SUM(CASE
   WHEN "#WK_SRC_TB_Sys"."month" = '4' THEN "#WK_SRC_TB_Sys"."ending_balance"
```

Query Syntax is Good

Then, at the top, we add in the SUM statements as shown above. Also add in the code as highlighted. What we are trying to achieve here is to unpivot and sum the 48 monthly columns (Jan to Dec movement + YTD Jan to YTD Dec) for Current & Prior Year.

Workiva

Step 15: Tips for Efficient Query Building



=="SUM("&A2&B2&C2&D2&E2

A	B	C	D	E	
CY.	"	Jan	")	as "Jan",	SUM(CY."Jan") as "Jan",
CY.	"	Feb	")	as "Feb",	SUM(CY."Feb") as "Feb",
CY.	"	Mar	")	as "Mar",	SUM(CY."Mar") as "Mar",
CY.	"	Apr	")	as "Apr",	SUM(CY."Apr") as "Apr",
CY.	"	May	")	as "May",	SUM(CY."May") as "May",
CY.	"	Jun	")	as "Jun",	SUM(CY."Jun") as "Jun",
CY.	"	Jul	")	as "Jul",	SUM(CY."Jul") as "Jul",
CY.	"	Aug	")	as "Aug",	SUM(CY."Aug") as "Aug",
CY.	"	Sep	")	as "Sep",	SUM(CY."Sep") as "Sep",
CY.	"	Oct	")	as "Oct",	SUM(CY."Oct") as "Oct",
CY.	"	Nov	")	as "Nov",	SUM(CY."Nov") as "Nov",
CY.	"	Dec	")	as "Dec",	SUM(CY."Dec") as "Dec",
CY.	"	Jan_YTD	")	as "Jan_YTD",	SUM(CY."Jan_YTD") as "Jan_YTD",
CY.	"	Feb_YTD	")	as "Feb_YTD",	SUM(CY."Feb_YTD") as "Feb_YTD",
CY.	"	Mar_YTD	")	as "Mar_YTD",	SUM(CY."Mar_YTD") as "Mar_YTD",
CY.	"	Apr_YTD	")	as "Apr_YTD",	SUM(CY."Apr_YTD") as "Apr_YTD",
CY.	"	May_YTD	")	as "May_YTD",	SUM(CY."May_YTD") as "May_YTD"
CY.	"	Jun_YTD	")	as "Jun_YTD",	SUM(CY."Jun_YTD") as "Jun_YTD",
CY.	"	Jul_YTD	")	as "Jul_YTD",	SUM(CY."Jul_YTD") as "Jul_YTD",
CY.	"	Aug_YTD	")	as "Aug_YTD",	SUM(CY."Aug_YTD") as "Aug_YTD",
CY.	"	Sep_YTD	")	as "Sep_YTD",	SUM(CY."Sep_YTD") as "Sep_YTD",

```
Select

Case when CY."year" is NULL then cast (Cast(PY."PY_year" as INTEGER) + 1 as VARCHAR) else CY."year" end as "Current_Year",
Case when CY."entity" is NULL then PY."PY_entity" else CY."entity" end as "Entity",
Case when CY."account_code" is NULL then PY."PY_account_code" else CY."account_code" end as "Account_Code",
Case when CY."description" is NULL then PY."PY_description" else CY."description" end as "Description",
Case when CY."FS" is NULL then PY."PY_FS" else CY."FS" end as "FS",


SUM(CY."Jan") as "Jan",
SUM(CY."Feb") as "Feb",
SUM(CY."Mar") as "Mar",
SUM(CY."Apr") as "Apr",
SUM(CY."May") as "May",
SUM(CY."Jun") as "Jun",
SUM(CY."Jul") as "Jul",
SUM(CY."Aug") as "Aug",
SUM(CY."Sep") as "Sep",
SUM(CY."Oct") as "Oct",
SUM(CY."Nov") as "Nov",
SUM(CY."Dec") as "Dec",
SUM(CY."Jan_YTD") as "Jan_YTD",
SUM(CY."Feb_YTD") as "Feb_YTD",
SUM(CY."Mar_YTD") as "Mar_YTD",
SUM(CY."Apr_YTD") as "Apr_YTD",
SUM(CY."May_YTD") as "May_YTD",
SUM(CY."Jun_YTD") as "Jun_YTD",
SUM(CY."Jul_YTD") as "Jul_YTD",
SUM(CY."Aug_YTD") as "Aug_YTD",
SUM(CY."Sep_YTD") as "Sep_YTD",
SUM(CY."Oct_YTD") as "Oct_YTD",
SUM(CY."Nov_YTD") as "Nov_YTD",
SUM(CY."Dec_YTD") as "Dec_YTD",
SUM(PY."PY_Jan") as "PY_Jan",
SUM(PY."PY_Feb") as "PY_Feb",
SUM(PY."PY_Mar") as "PY_Mar",
SUM(PY."PY_Apr") as "PY_Apr",
SUM(PY."PY_May") as "PY_May",
SUM(PY."PY_Jun") as "PY_Jun",
SUM(PY."PY_Jul") as "PY_Jul",
SUM(PY."PY_Aug") as "PY_Aug",
SUM(PY."PY_Sep") as "PY_Sep",
SUM(PY."PY_Oct") as "PY_Oct",
SUM(PY."PY_Nov") as "PY_Nov",
SUM(PY."PY_Dec") as "PY_Dec",
```

Excel formulas can be leveraged when building & structuring queries. As illustrated above, we utilised excel formulas to build the SUM for CY & PY's representation.

Workiva

Step 16: Grouping - Group By

CY query

```
  "#WK_SRC_TB_Sys"."month" = '9' OR
  "#WK_SRC_TB_Sys"."month" = '10' OR
  "#WK_SRC_TB_Sys"."month" = '11' OR
  "#WK_SRC_TB_Sys"."month" = '12' THEN "#WK_SRC_TB_Sys"."ending_balance"
    ELSE 0
  END
), SUM(CASE
    WHEN "#WK_SRC_TB_Sys"."month" = '12' THEN "#WK_SRC_TB_Sys"."ending_balance"
    ELSE 0
  END
)) AS "Dec_YTD"

FROM
  "QWNjb3VudB8xODkzNjM5MjMw"."6fbe4085dc6942c5b9543cb85a5ad085" AS "#WK_SRC_TB_Sys"
  LEFT JOIN
  "QWNjb3VudB8xODkzNjM5MjMw"."e450aa655ad1401280f8c9f804834433" AS "#WK_DIM_Account"
ON "#WK_SRC_TB_Sys"."account_code" = "#WK_DIM_Account"."gl_account"

WHERE
  "#WK_SRC_TB_Sys"._tags['Data_Year'] = :Data_Year AND
  "#WK_SRC_TB_Sys"._tags['Data_Month'] = :Data_Month

GROUP BY "#WK_SRC_TB_Sys"."year", "#WK_SRC_TB_Sys"."entity", "#WK_SRC_TB_Sys"."account_code", "#WK_SRC_TB_Sys"
    ."description", "#WK_SRC_TB_Sys"."product_code", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs",
    "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs",
    "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs",
    "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs"

ORDER BY "account_code" ASC, "FS" ASC ) CY

FULL OUTER JOIN

(SELECT
  "#WK_SRC_TB_Sys"."year" AS "PY_year",
  "#WK_SRC_TB_Sys"."entity" AS "PY_entity",
  "#WK_SRC_TB_Sys"."account_code" AS "PY_account_code",
```

PY query

```
  IF("#WK_DIM_Account"."fs" = 'PL', SUM(CASE
    WHEN "#WK_SRC_TB_Sys"."month" = '1' OR
  "#WK_SRC_TB_Sys"."month" = '2' OR
  "#WK_SRC_TB_Sys"."month" = '3' OR
  "#WK_SRC_TB_Sys"."month" = '4' OR
  "#WK_SRC_TB_Sys"."month" = '5' OR
  "#WK_SRC_TB_Sys"."month" = '6' OR
  "#WK_SRC_TB_Sys"."month" = '7' OR
  "#WK_SRC_TB_Sys"."month" = '8' OR
  "#WK_SRC_TB_Sys"."month" = '9' OR
  "#WK_SRC_TB_Sys"."month" = '10' OR
  "#WK_SRC_TB_Sys"."month" = '11' OR
  "#WK_SRC_TB_Sys"."month" = '12' THEN "#WK_SRC_TB_Sys"."ending_balance"
    ELSE 0
  END
), SUM(CASE
    WHEN "#WK_SRC_TB_Sys"."month" = '12' THEN "#WK_SRC_TB_Sys"."ending_balance"
    ELSE 0
  END
)) AS "PY_Dec_YTD"

FROM
  "QWNjb3VudB8xODkzNjM5MjMw"."6fbe4085dc6942c5b9543cb85a5ad085" AS "#WK_SRC_TB_Sys"
  LEFT JOIN
  "QWNjb3VudB8xODkzNjM5MjMw"."e450aa655ad1401280f8c9f804834433" AS "#WK_DIM_Account"
ON "#WK_SRC_TB_Sys"."account_code" = "#WK_DIM_Account"."gl_account"

WHERE
  "#WK_SRC_TB_Sys"._tags['Data_Year'] = CAST(CAST(:Data_Year AS INTEGER) - 1 AS VARCHAR) AND
  "#WK_SRC_TB_Sys"._tags['Data_Month'] = :Data_Month

GROUP BY "#WK_SRC_TB_Sys"."year", "#WK_SRC_TB_Sys"."entity", "#WK_SRC_TB_Sys"."account_code", "#WK_SRC_TB_Sys"
    ."description", "#WK_SRC_TB_Sys"."product_code", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs",
    "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs",
    "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs",
    "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs", "#WK_DIM_Account"."fs"

ORDER BY "PY_account_code" ASC, "PY_FS" ASC) PY

ON CONCAT(CY."entity",CY."account_code",CY."product_code")= CONCAT(PY."PY_entity",PY."PY_account_code",PY
    ."PY_product_code")
```

For both Current & Prior Year's query, at the end of the query we would group them by the following:

i. Year

ii. Entity

iii. Account_Code

iv. Description

v. Product_Code

The GROUP BY statements are indicated above.

Workiva

Step 16.1: Optimizing the Fields

SQL Editor

Builder | SQL

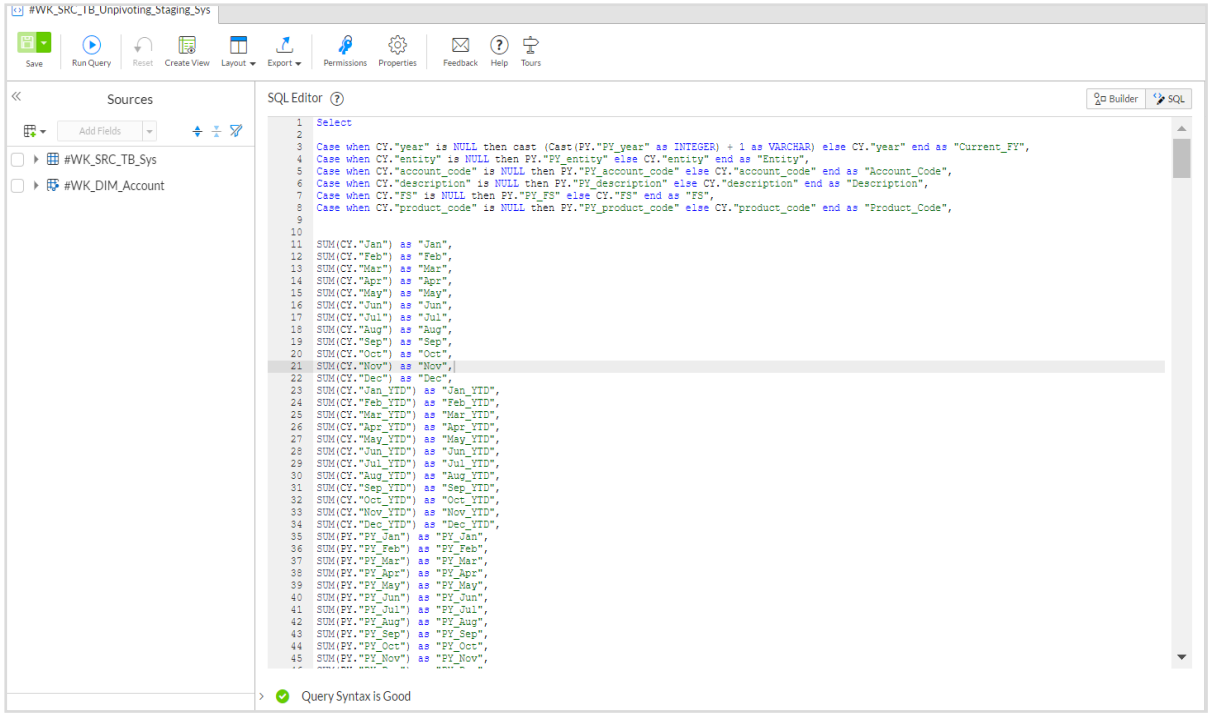
```
1  Select
2
3  Case when CY."year" is NULL then cast (Cast(PY."PY_year" as INTEGER) + 1 as VARCHAR) else CY."year" end as
       "Current_FY",
4  Case when CY."entity" is NULL then PY."PY_entity" else CY."entity" end as "Entity",
5  Case when CY."account_code" is NULL then PY."PY_account_code" else CY."account_code" end as "Account_Code",
6  Case when CY."description" is NULL then PY."PY_description" else CY."description" end as "Description",
7  Case when CY."FS" is NULL then PY."PY_FS" else CY."FS" end as "FS",
8  Case when CY."product_code" is NULL then PY."PY_product_code" else CY."product_code" end as "Product_Code",
9
```

As for rolling period, sometimes the current year's data would not be made available and this would result in 'blank / NULL output result fields. To cater for 'blank / NULL fields' of the results, we would include the following code snippet at the beginning.

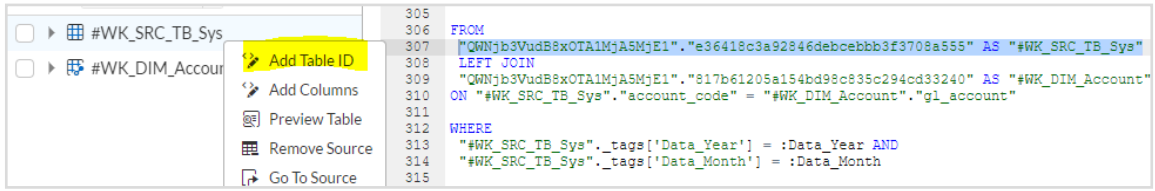
This code snippet simply means, if any of the Current Year's columns is 'blank / NULL', we would append & display the Prior Year's data instead

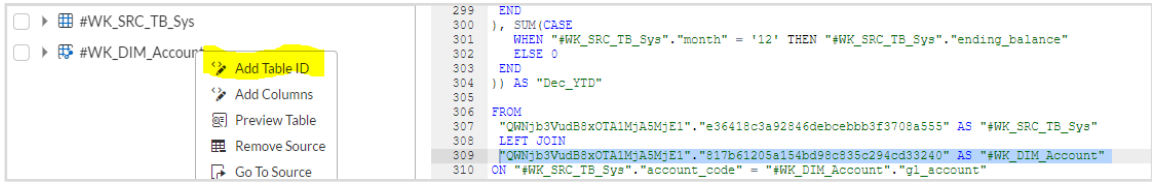
Step 16.2: Quick-Hack- #WK_SRC_TB_Unpivoting_Staging_Sys

#WK_SRC_TB_Unpivoted_Staging_Sys.txt



Replace Table IDs:



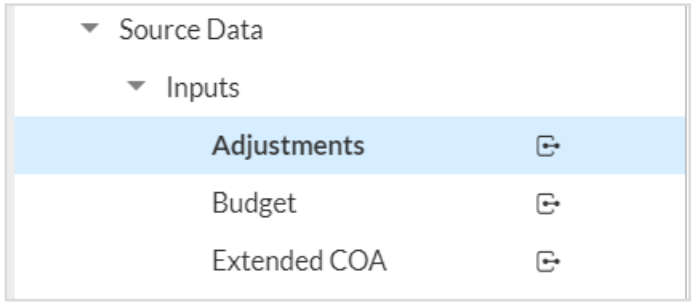


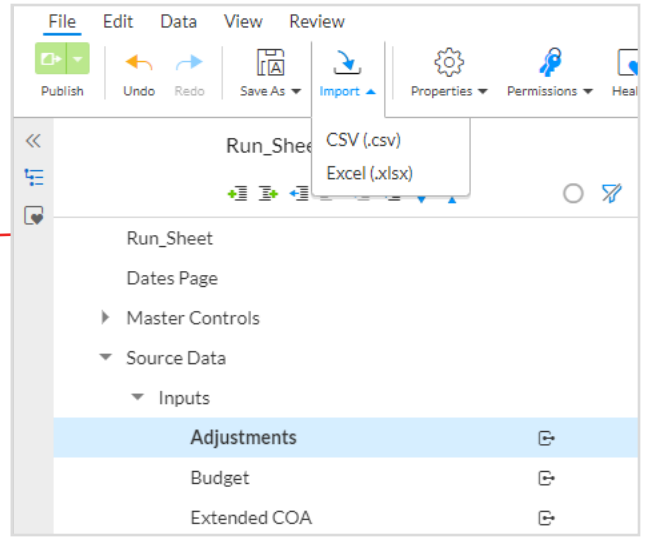
If your query doesn't work, copy paste the provided script into the ' SQL Editor', but remember to replace the 'Wdata Table ID at Line 307, 309, 566, 568. To replace the table ID, first highlight the entire line, right click on the '#WK_SRC_TB_Sys' table, and select "Add Table ID". Repeat for '#WK_DIM_Account'

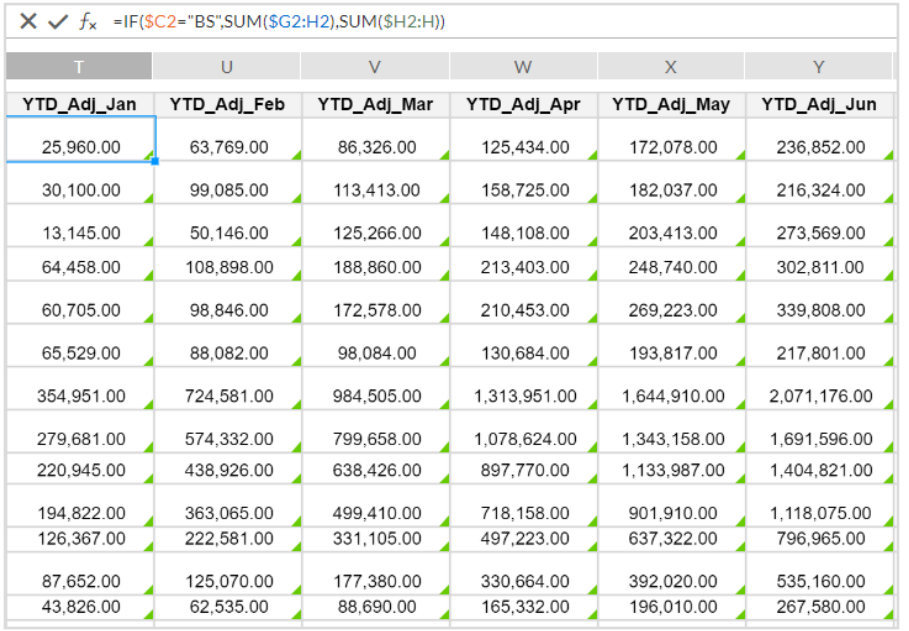
Workiva

Step 17: Objective of the 'Adjustment' Sheet

↪For users to populate late adjustments, these adjustment figures would be factored together with the TB amounts







=IF($C2="BS",SUM($G2:H2),SUM($H2:H))

T	U	V	W	X	Y
YTD_Adj_Jan	YTD_Adj_Feb	YTD_Adj_Mar	YTD_Adj_Apr	YTD_Adj_May	YTD_Adj_Jun
25,960.00	63,769.00	86,326.00	125,434.00	172,078.00	236,852.00
30,100.00	99,085.00	113,413.00	158,725.00	182,037.00	216,324.00
13,145.00	50,146.00	125,266.00	148,108.00	203,413.00	273,569.00
64,458.00	108,898.00	188,860.00	213,403.00	248,740.00	302,811.00
60,705.00	98,846.00	172,578.00	210,453.00	269,223.00	339,808.00
65,529.00	88,082.00	98,084.00	130,684.00	193,817.00	217,801.00
354,951.00	724,581.00	984,505.00	1,313,951.00	1,644,910.00	2,071,176.00
279,681.00	574,332.00	799,658.00	1,078,624.00	1,343,158.00	1,691,596.00
220,945.00	438,926.00	638,426.00	897,770.00	1,133,987.00	1,404,821.00
194,822.00	363,065.00	499,410.00	718,158.00	901,910.00	1,118,075.00
126,367.00	222,581.00	331,105.00	497,223.00	637,322.00	796,965.00
87,652.00	125,070.00	177,380.00	330,664.00	392,020.00	535,160.00
43,826.00	62,535.00	88,690.00	165,332.00	196,010.00	267,580.00

1: Create the following spreadsheet tabs

2: Import the Adjustment CSV data that is provided

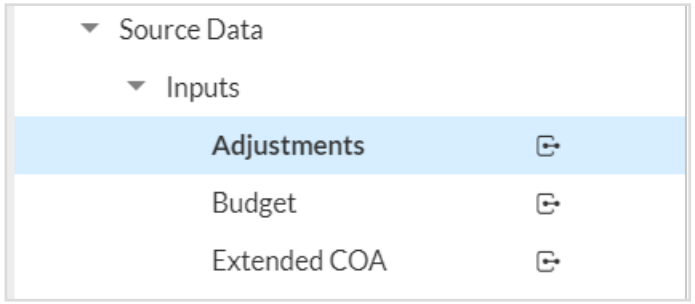
3: As the imported Adjustment spreadsheet only has up to 'Adj_Dec', we would need to create the 'YTD_Adj' calculations.

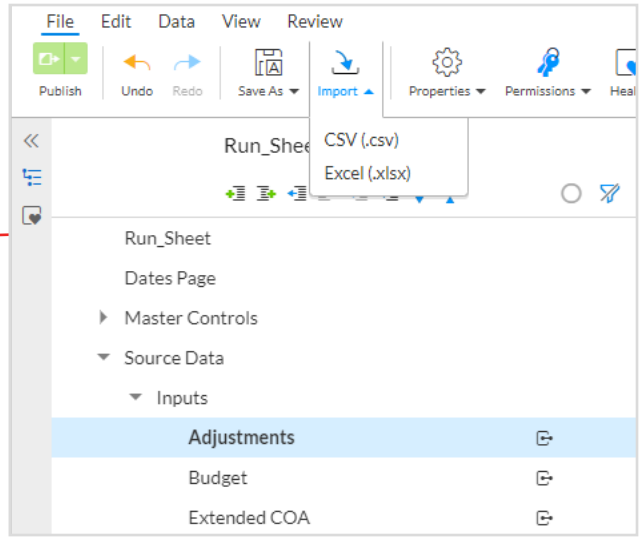
Sample Formula for 'YTD_Adj_Jan':
=IF($C2="BS",SUM($G2:H2),SUM($H2:H2))

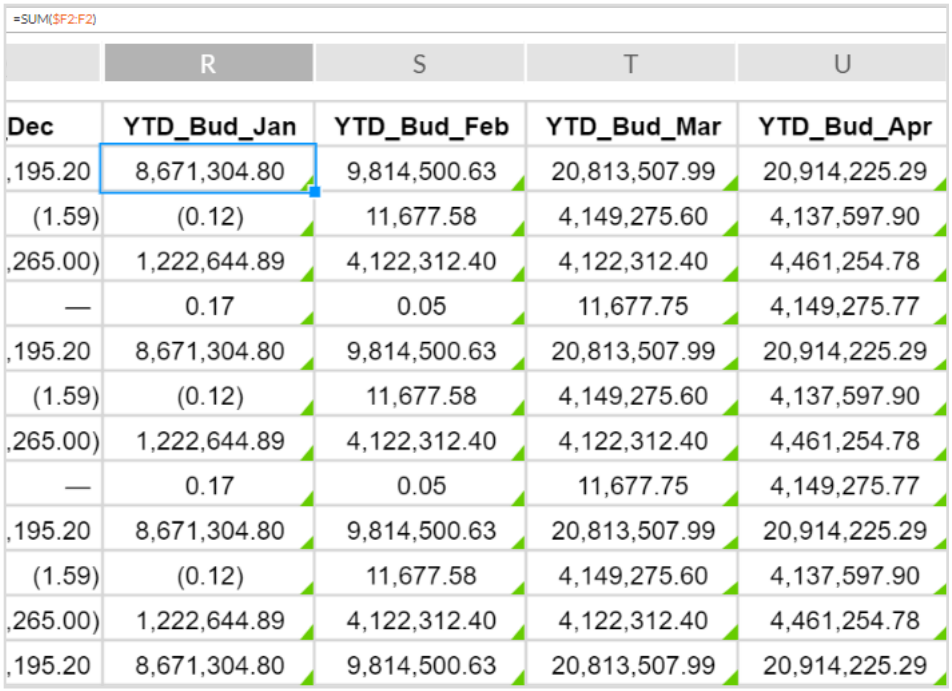
Workiva

Step 17.1: Objective of the 'Budget' Sheet

↪For users to populate budget amounts, these budget figures would be combined in the TB table







=SUM($F2:F2)

	R	S	T	U
Dec	YTD_Bud_Jan	YTD_Bud_Feb	YTD_Bud_Mar	YTD_Bud_Apr
,195.20	8,671,304.80	9,814,500.63	20,813,507.99	20,914,225.29
(1.59)	(0.12)	11,677.58	4,149,275.60	4,137,597.90
,265.00)	1,222,644.89	4,122,312.40	4,122,312.40	4,461,254.78
—	0.17	0.05	11,677.75	4,149,275.77
,195.20	8,671,304.80	9,814,500.63	20,813,507.99	20,914,225.29
(1.59)	(0.12)	11,677.58	4,149,275.60	4,137,597.90
,265.00)	1,222,644.89	4,122,312.40	4,122,312.40	4,461,254.78
—	0.17	0.05	11,677.75	4,149,275.77
,195.20	8,671,304.80	9,814,500.63	20,813,507.99	20,914,225.29
(1.59)	(0.12)	11,677.58	4,149,275.60	4,137,597.90
,265.00)	1,222,644.89	4,122,312.40	4,122,312.40	4,461,254.78
,195.20	8,671,304.80	9,814,500.63	20,813,507.99	20,914,225.29

1: Create the following spreadsheet tabs

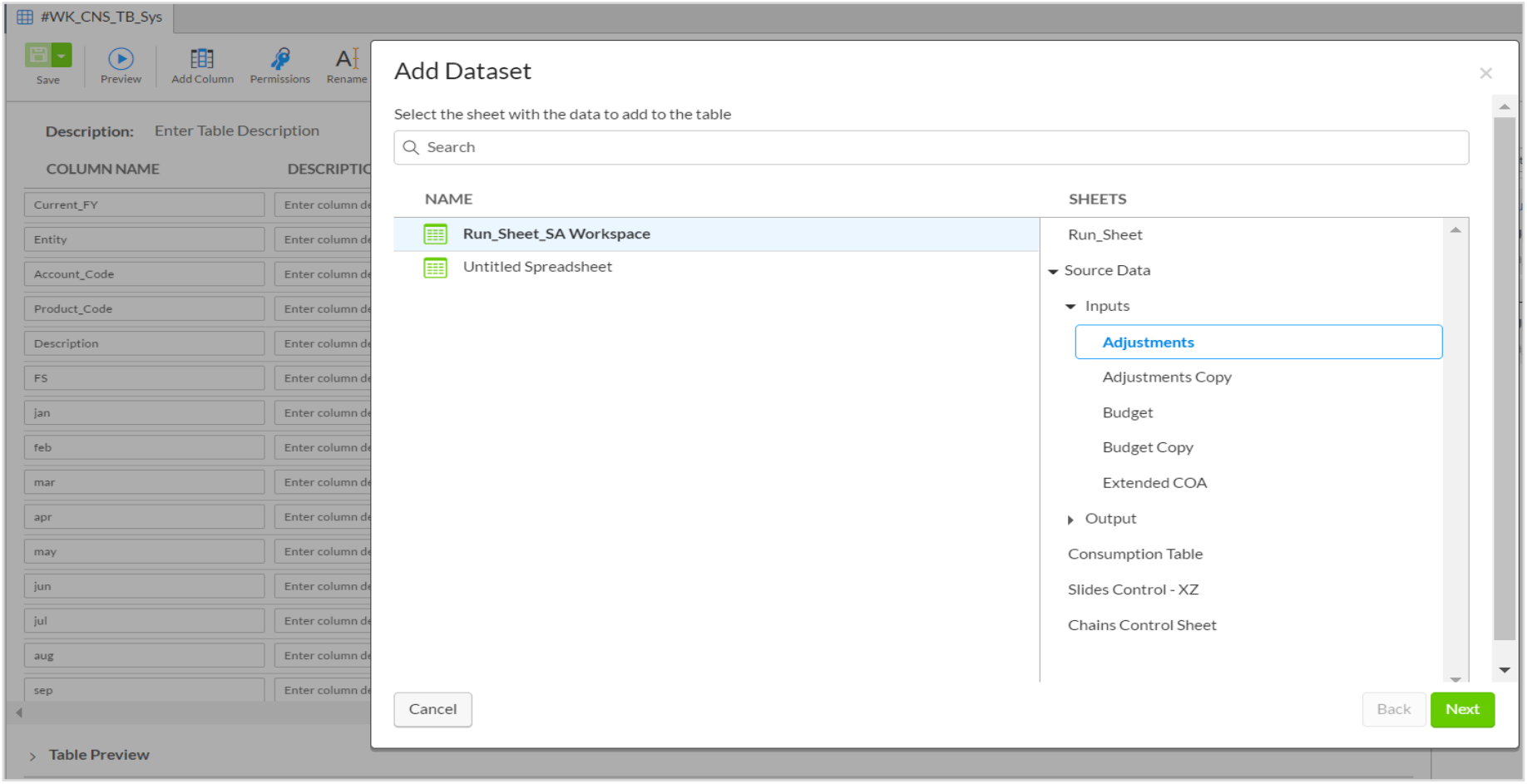
2: Import the Budget CSV data that is provided

3: As the imported Budget spreadsheet only has up to 'Bud_Dec', we would need to create the 'YTD_Bud_Jan' calculations.

Sample Formula for 'YTD_Bud_Jan':
=SUM($F2:F2)

Workiva

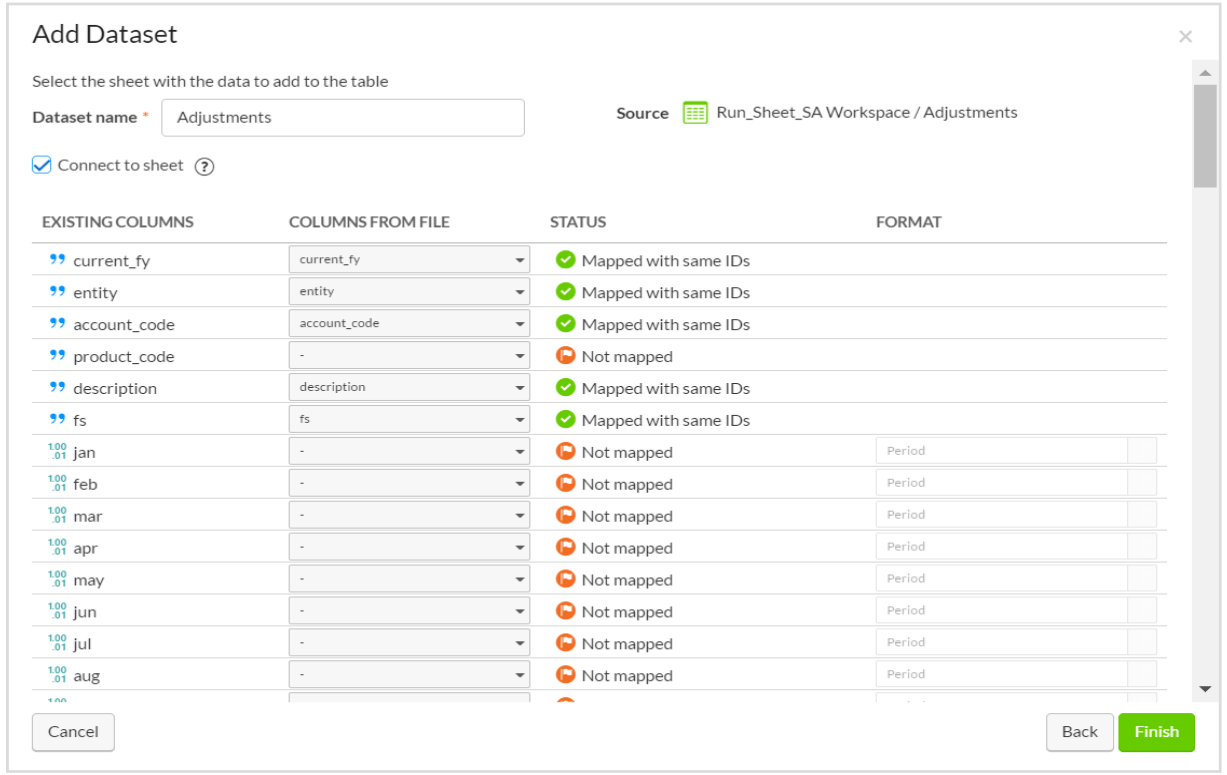
Step 17.2: Adjustment' Connected Sheet P1

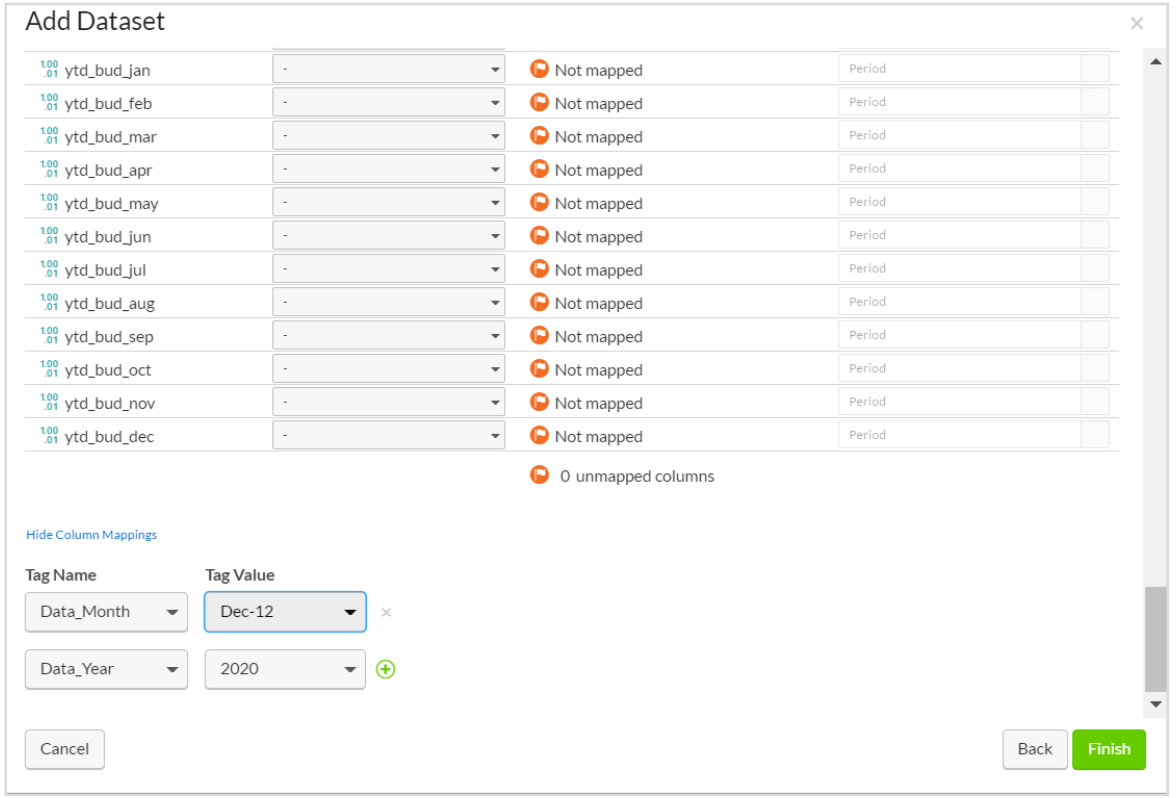


In this step, we illustrate how to connect the 'Adjustment' spreadsheet on Wdesk to the Consumption table ("#WK_CNS_TB_Sys"). We click 'Add Dataset', and choose 'Spreadsheet', add the 'Adjustment' spreadsheet from Wdesk.

Workiva

Step 17.3: Adjustment' Connected Sheet P2

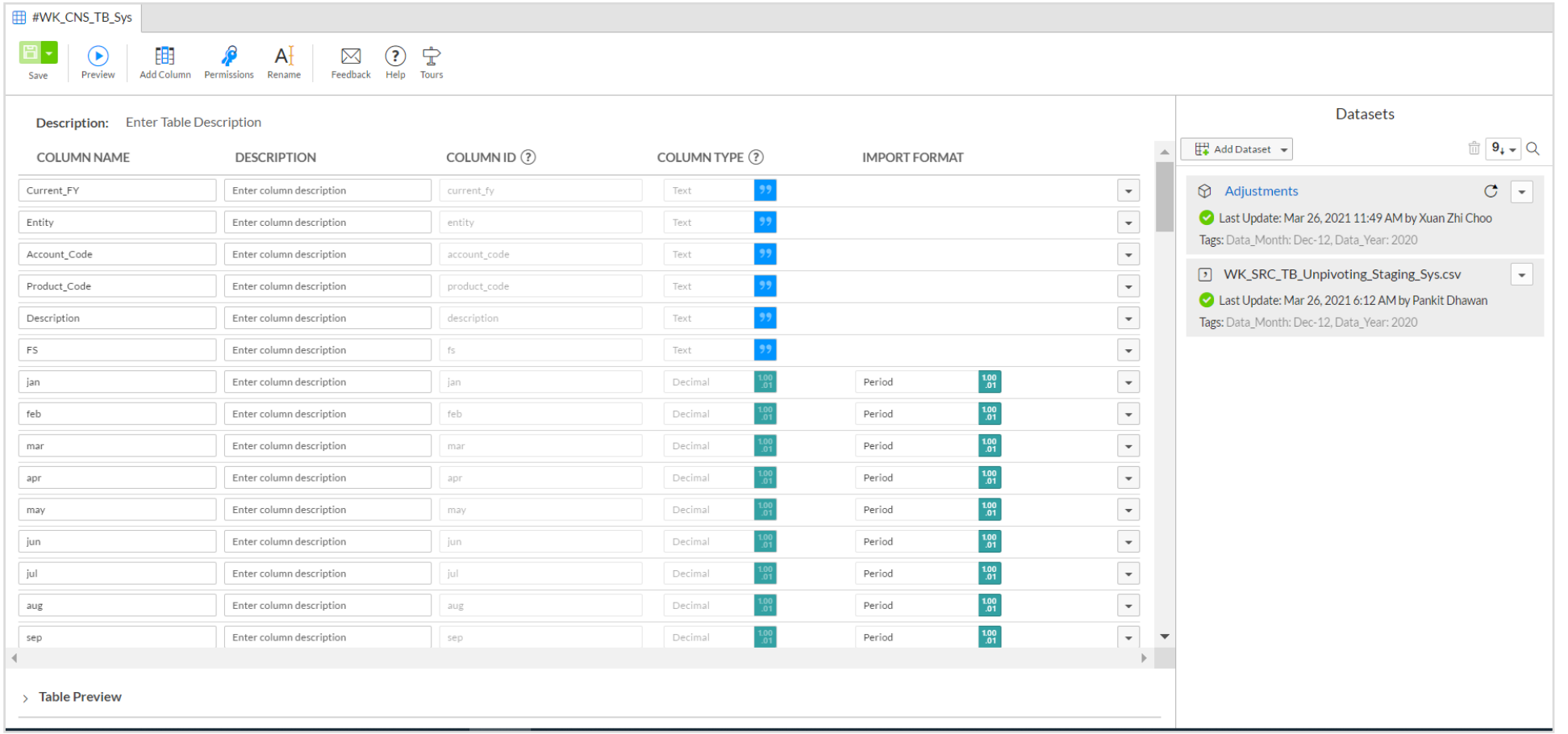




When 'Next' is clicked, tick the 'Connect to sheet' checkbox. Scroll down and add 'Data_Year' & 'Data_Month' as tags. Click 'Finish' to create the connection.

workiva

Step 17.4: Adjustment' Connected Sheet P3



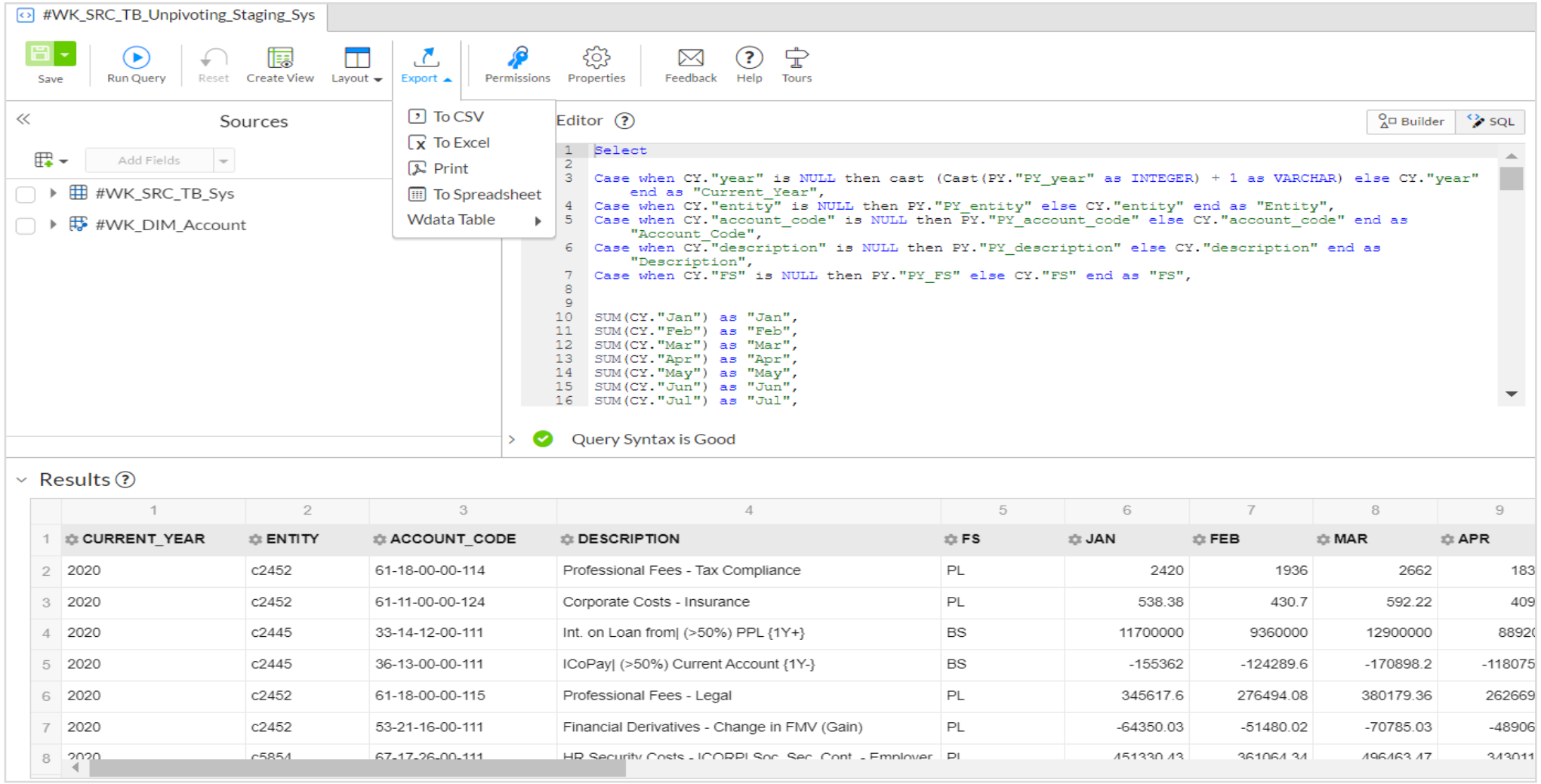
The 'Adjustment' spreadsheet is now connected to the Consumption table. Whenever the spreadsheet is refreshed, this table gets updated as well. Repeat the same steps to connect the '**Budget**' spreadsheet to this table as well. In the next step, we would be mixing all the tables that was created to form the Extended Trial Balance (ETB) Query.

Step 18: Creating the Budget Sheet

1: Repeat the steps for Adjustments but using Budget data.

2: Formulas for Budget will be different, the formula will be =SUM($F2:F2)

Step 19: Creating the Consumption Table P1 - WK_CNS_TB



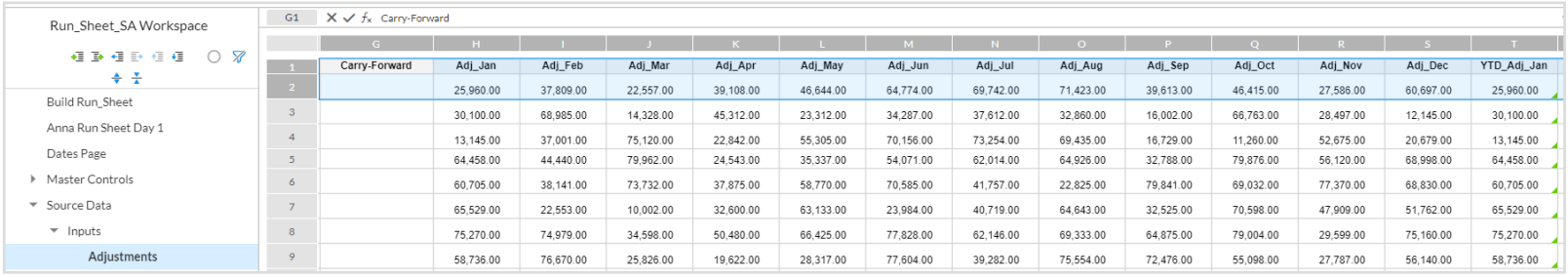
	1	2	3	4	5	6	7	8	9
1	CURRENT_YEAR	ENTITY	ACCOUNT_CODE	DESCRIPTION	FS	JAN	FEB	MAR	APR
2	2020	c2452	61-18-00-00-114	Professional Fees - Tax Compliance	PL	2420	1936	2662	183
3	2020	c2452	61-11-00-00-124	Corporate Costs - Insurance	PL	538.38	430.7	592.22	409
4	2020	c2445	33-14-12-00-111	Int. on Loan from| (>50%) PPL {1Y+}	BS	11700000	9360000	12900000	88920
5	2020	c2445	36-13-00-00-111	ICoPay| (>50%) Current Account {1Y-}	BS	-155362	-124289.6	-170898.2	-118075
6	2020	c2452	61-18-00-00-115	Professional Fees - Legal	PL	345617.6	276494.08	380179.36	262669
7	2020	c2452	53-21-16-00-111	Financial Derivatives - Change in FMV (Gain)	PL	-64350.03	-51480.02	-70785.03	-48906
8	2020	c5854	67-17-26-00-111	HR Security Costs - ICORPL Soc. Sec. Cont. - Employer	PL	451330.43	361064.34	496463.47	343011

After the Staging query has been built & successfully ran, we export the results to 'CSV'.

workiva

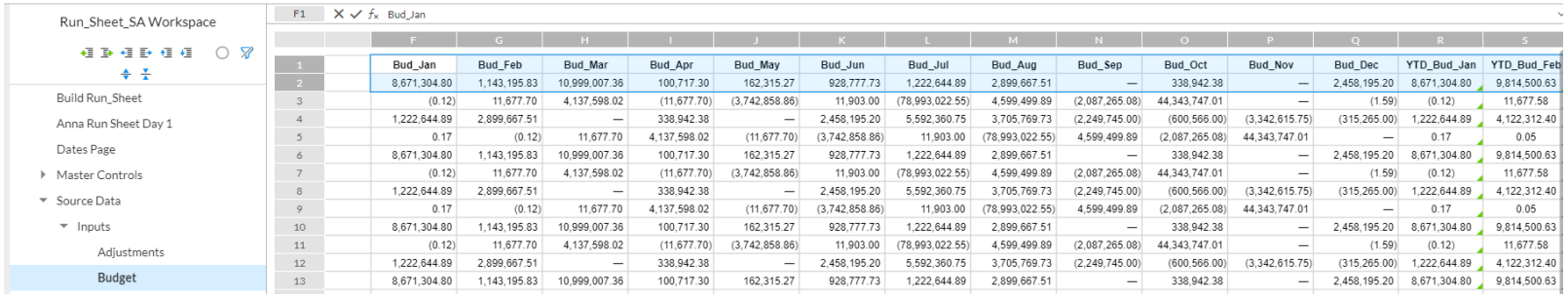
Step 19.1: Building the 'WK_CNS_TB' Table Schema

From the Adjustments spreadsheet, copy the columns from 'Carry Forward', all the way to the end, and append to the query export csv file.



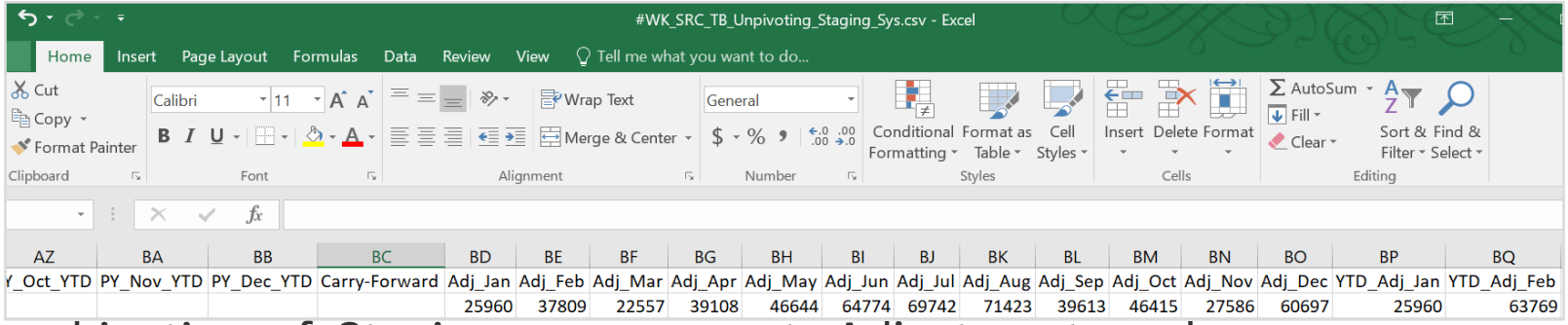
Carry-Forward	Adj_Jan	Adj_Feb	Adj_Mar	Adj_Apr	Adj_May	Adj_Jun	Adj_Jul	Adj_Aug	Adj_Sep	Adj_Oct	Adj_Nov	Adj_Dec	YTD_Adj_Jan
	25,960.00	37,809.00	22,557.00	39,108.00	46,644.00	64,774.00	69,742.00	71,423.00	39,613.00	46,415.00	27,586.00	60,697.00	25,960.00
	30,100.00	68,985.00	14,328.00	45,312.00	23,312.00	34,287.00	37,612.00	32,860.00	16,002.00	66,763.00	28,497.00	12,145.00	30,100.00
	13,145.00	37,001.00	75,120.00	22,842.00	55,305.00	70,156.00	73,254.00	69,435.00	16,729.00	11,260.00	52,675.00	20,679.00	13,145.00
	64,458.00	44,440.00	79,962.00	24,543.00	35,337.00	54,071.00	62,014.00	64,926.00	32,788.00	79,876.00	56,120.00	68,998.00	64,458.00
	60,705.00	38,141.00	73,732.00	37,875.00	58,770.00	70,585.00	41,757.00	22,825.00	79,841.00	69,032.00	77,370.00	68,830.00	60,705.00
	65,529.00	22,553.00	10,002.00	32,600.00	63,133.00	23,984.00	40,719.00	64,643.00	32,525.00	70,598.00	47,909.00	51,762.00	65,529.00
	75,270.00	74,979.00	34,598.00	50,480.00	66,425.00	77,828.00	62,146.00	69,333.00	64,875.00	79,004.00	29,599.00	75,160.00	75,270.00
	58,736.00	76,670.00	25,826.00	19,622.00	28,317.00	77,604.00	39,282.00	75,554.00	72,476.00	55,098.00	27,787.00	56,140.00	58,736.00

From the Budget spreadsheet, copy the columns from 'Bud_Jan', all the way to the end, and append to the query export csv file.



Bud_Jan	Bud_Feb	Bud_Mar	Bud_Apr	Bud_May	Bud_Jun	Bud_Jul	Bud_Aug	Bud_Sep	Bud_Oct	Bud_Nov	Bud_Dec	YTD_Bud_Jan	YTD_Bud_Feb
8,671,304.80	1,143,195.83	10,999,007.36	100,717.30	162,315.27	928,777.73	1,222,644.89	2,899,667.51	—	338,942.38	—	2,458,195.20	8,671,304.80	9,814,500.63
(0.12)	11,677.70	4,137,598.02	(11,677.70)	(3,742,858.86)	11,903.00	(78,993,022.55)	4,599,499.89	(2,087,265.08)	44,343,747.01	—	(1.59)	(0.12)	11,677.58
1,222,644.89	2,899,667.51	—	338,942.38	—	2,458,195.20	5,592,360.75	3,705,769.73	(2,249,745.00)	(600,566.00)	(3,342,615.75)	(315,265.00)	1,222,644.89	4,122,312.40
0.17	(0.12)	11,677.70	4,137,598.02	(11,677.70)	(3,742,858.86)	11,903.00	(78,993,022.55)	4,599,499.89	(2,087,265.08)	44,343,747.01	—	0.17	0.05
8,671,304.80	1,143,195.83	10,999,007.36	100,717.30	162,315.27	928,777.73	1,222,644.89	2,899,667.51	—	338,942.38	—	2,458,195.20	8,671,304.80	9,814,500.63
(0.12)	11,677.70	4,137,598.02	(11,677.70)	(3,742,858.86)	11,903.00	(78,993,022.55)	4,599,499.89	(2,087,265.08)	44,343,747.01	—	(1.59)	(0.12)	11,677.58
1,222,644.89	2,899,667.51	—	338,942.38	—	2,458,195.20	5,592,360.75	3,705,769.73	(2,249,745.00)	(600,566.00)	(3,342,615.75)	(315,265.00)	1,222,644.89	4,122,312.40
0.17	(0.12)	11,677.70	4,137,598.02	(11,677.70)	(3,742,858.86)	11,903.00	(78,993,022.55)	4,599,499.89	(2,087,265.08)	44,343,747.01	—	0.17	0.05
8,671,304.80	1,143,195.83	10,999,007.36	100,717.30	162,315.27	928,777.73	1,222,644.89	2,899,667.51	—	338,942.38	—	2,458,195.20	8,671,304.80	9,814,500.63
(0.12)	11,677.70	4,137,598.02	(11,677.70)	(3,742,858.86)	11,903.00	(78,993,022.55)	4,599,499.89	(2,087,265.08)	44,343,747.01	—	(1.59)	(0.12)	11,677.58
1,222,644.89	2,899,667.51	—	338,942.38	—	2,458,195.20	5,592,360.75	3,705,769.73	(2,249,745.00)	(600,566.00)	(3,342,615.75)	(315,265.00)	1,222,644.89	4,122,312.40
8,671,304.80	1,143,195.83	10,999,007.36	100,717.30	162,315.27	928,777.73	1,222,644.89	2,899,667.51	—	338,942.38	—	2,458,195.20	8,671,304.80	9,814,500.63

The consolidated schema in the CSV file:

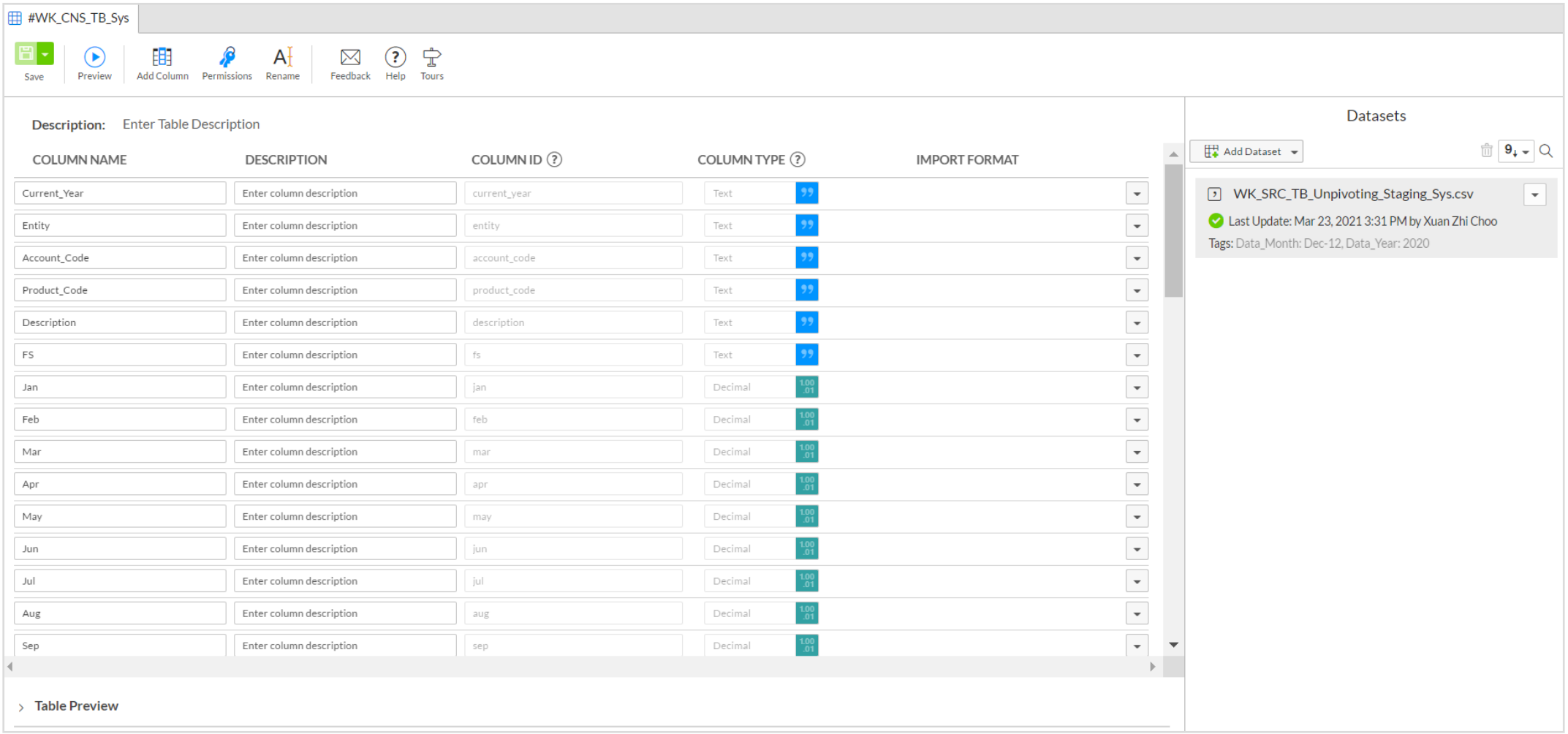


AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ
Y_Oct_YTD	PY_Nov_YTD	PY_Dec_YTD	Carry-Forward	Adj_Jan	Adj_Feb	Adj_Mar	Adj_Apr	Adj_May	Adj_Jun	Adj_Jul	Adj_Aug	Adj_Sep	Adj_Oct	Adj_Nov	Adj_Dec	YTD_Adj_Jan	YTD_Adj_Feb
				25960	37809	22557	39108	46644	64774	69742	71423	39613	46415	27586	60697	25960	63769

The 'WK_CNS_TB' table schema, is a combination of Staging query export, Adjustment, and Budget columns. From the previous step, after the query was exported, copy the columns from the Adjustments and Budget Spreadsheet and append to the columns behind to build the schema for the 'WK_CNS_TB' wdata table.

Workiva

Step 19.2: Creating the Consumption Table P2 - WK_CNS_TB



The updated file from previous step is used to create the consumption table "#WK_CNS_TB_Sys". The schema of consumption table would encapsulate the 'Pivoted Actual data', 'Adjustments' and 'Budget' data fields.

Step 20: Uploading Data into Consumption Table (Adv Query, Adj, Bud).

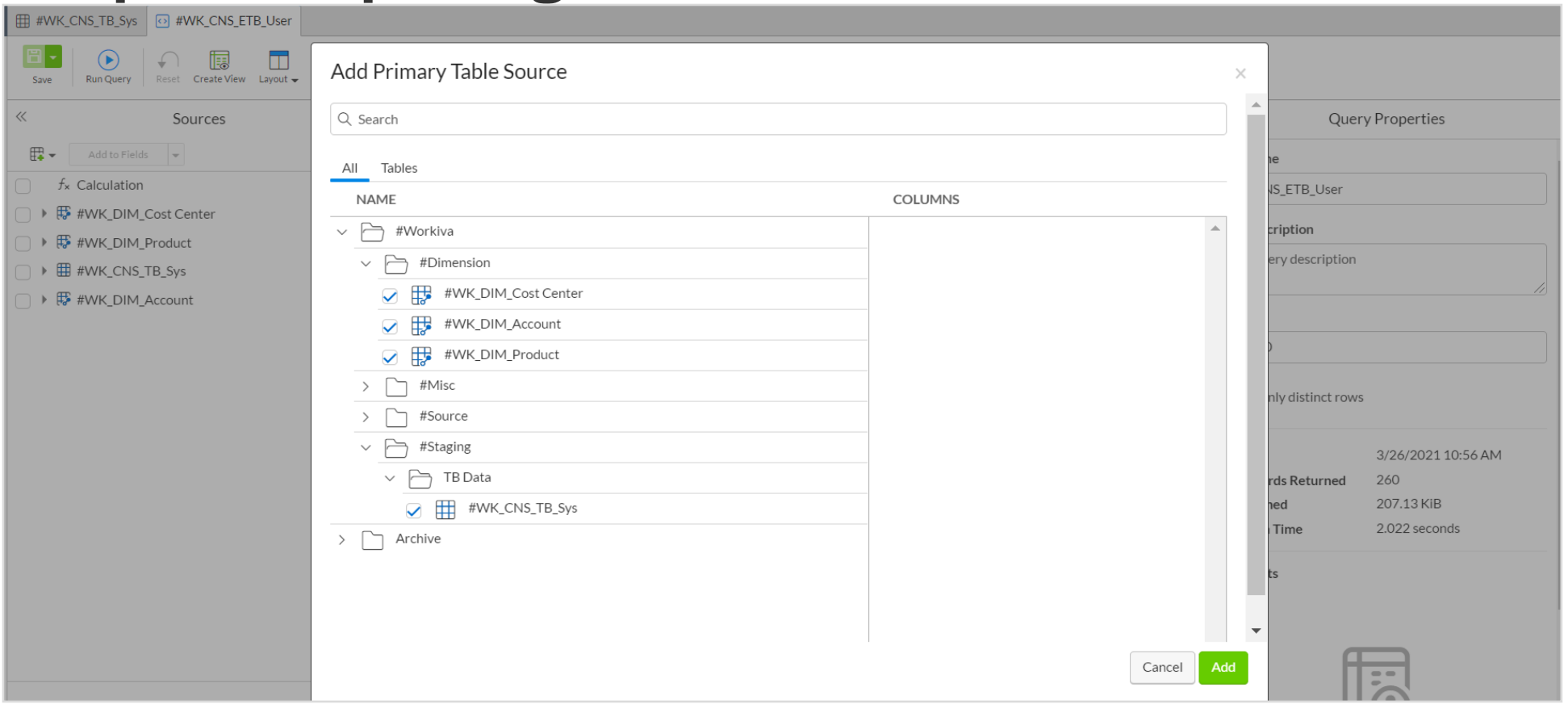
Uploaded the transformed TB data, filename:

1. Act_Dec-12_2020.csv
2. Adj_Dec-12_2020.csv
3. Bud_Dec-12_2020.csv

Step 21: Setting up Consumption Query

Steps	Description
0	**Based on the previous steps, in general:** i. Actual Accounts can be used for Adjustments ii. Actual Accounts can be used for Budget iii. Specially created Budget Accounts should never be used for Adjustments iv. Specially created Adjustment Accounts should never be used for Budget v. (THE ABOVE 2 have to be mutually exclusive) This relates to the extended COA.
1	**We first import all the tables that was created previously, namely:** i. #WK_CNS_TB_Sys ii. #WK_DIM_Cost Center iii. #WK_DIM_Product iv. #WK_DIM_Account
2	**Establish relationships between the tables:** i. #WK_CNS_TB_Sys LEFT JOIN #WK_DIM_Account on account_code ii. #WK_CNS_TB_Sys LEFT JOIN #WK_DIM_Cost Center on account_code iii. #WK_CNS_TB_Sys LEFT JOIN #WK_DIM_Product on product_code
3	Create calculation fields to display Quarter, YTD data which we would illustrate in the next few steps.

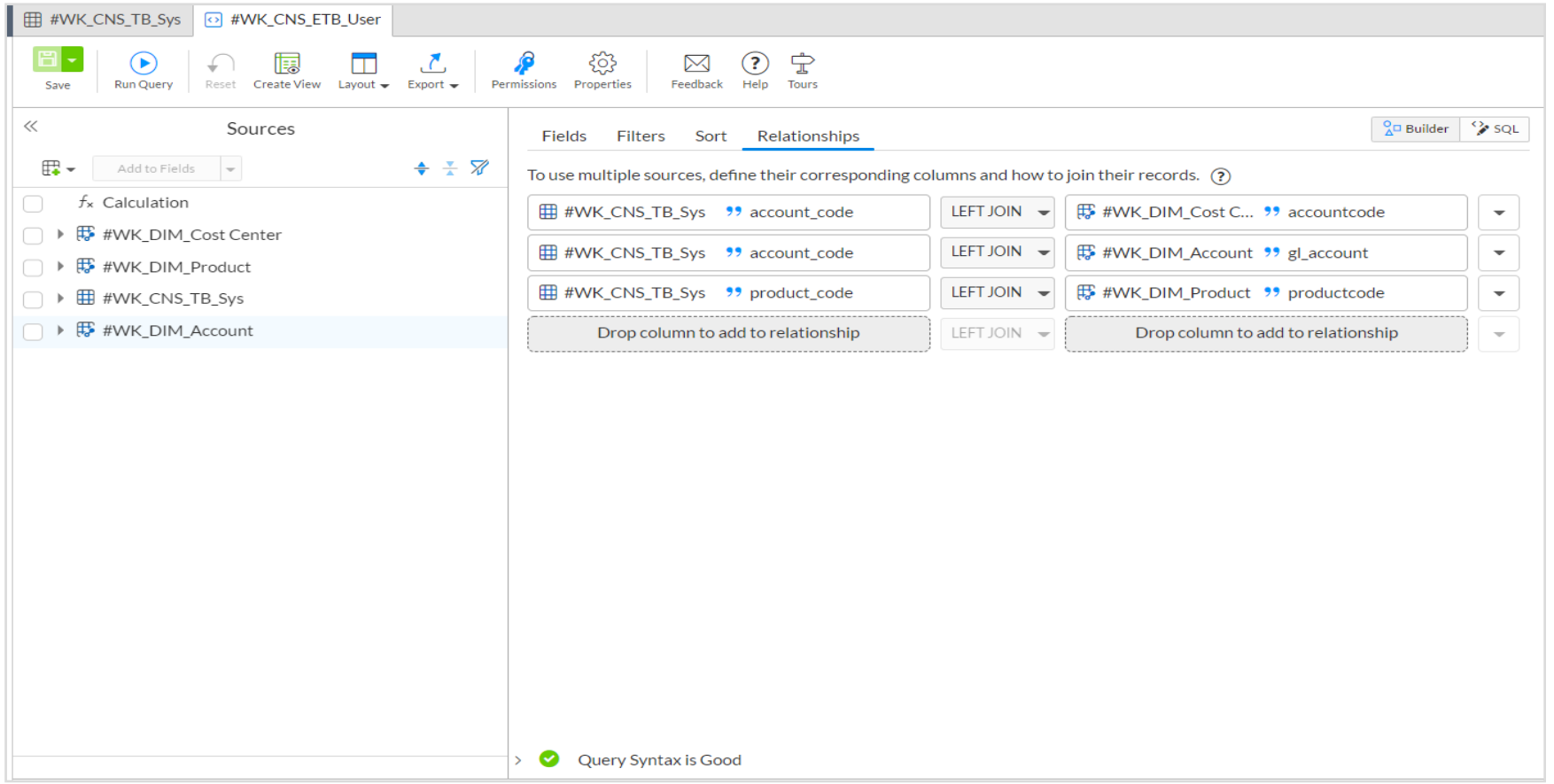
Step 21.1: Importing the Tables



Create a new query, import the following tables:

i. #WK_DIM_Cost Center
ii. #WK_DIM_Account
iii. #WK_DIM_Product
iv. #WK_CNS_TB_Sys

Step 21.2: Establish Table Relationships

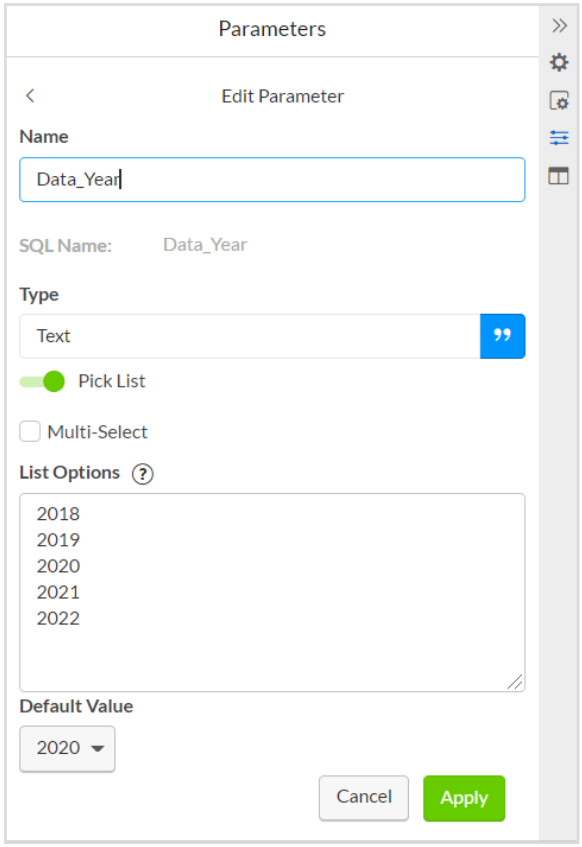


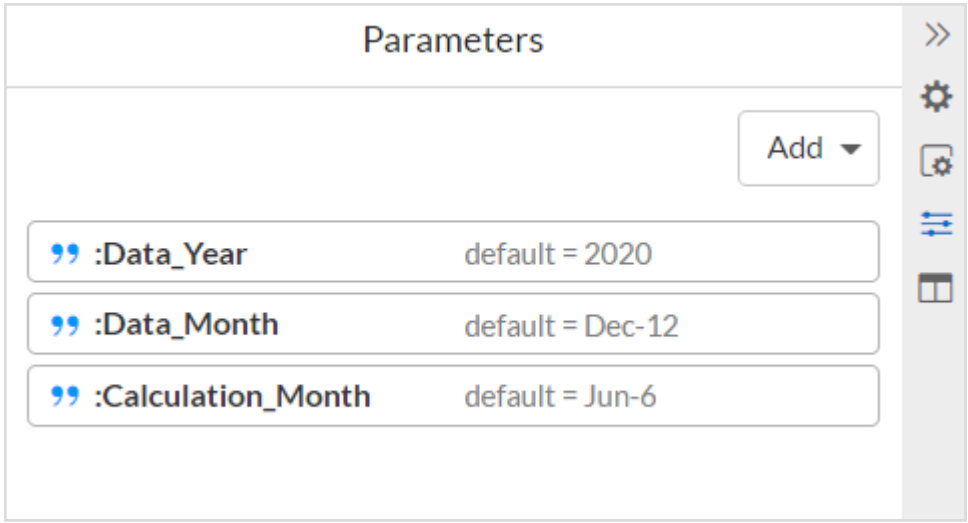
Establish relationships between:

i. #WK_CNS_TB_Sys LEFT JOIN #WK_DIM_Cost Center on Account Code
ii. #WK_CNS_TB_Sys LEFT JOIN #WK_DIM_Account on GL Account
iii. #WK_CNS_TB_Sys LEFT JOIN #WK_DIM_Product on Product Code

Workiva

Step 21.3: Create Parameters





Create the following Parameters:

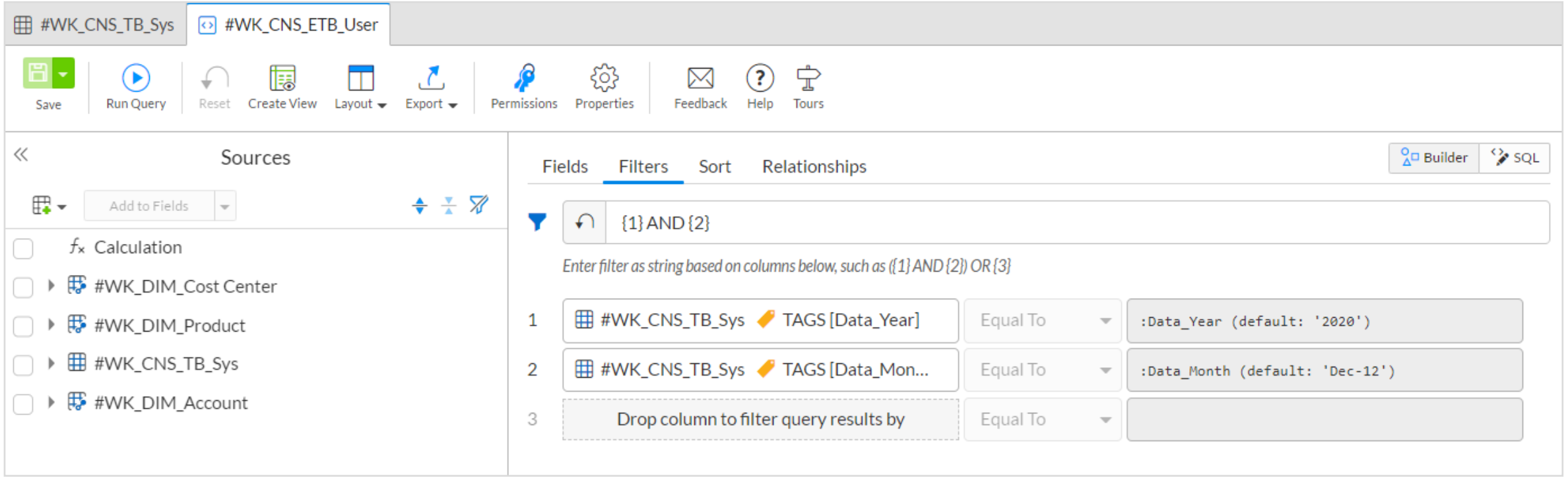
i. Data_Year - Data for the year that was imported

ii. Data_Month - Data for the month that was imported

iii. Calculation_Month - Derive the YTD data (selected by user). (Note that the calculation month needs to be equal or earlier than Data_Month)

Workiva

Step 21.4: Create Filters



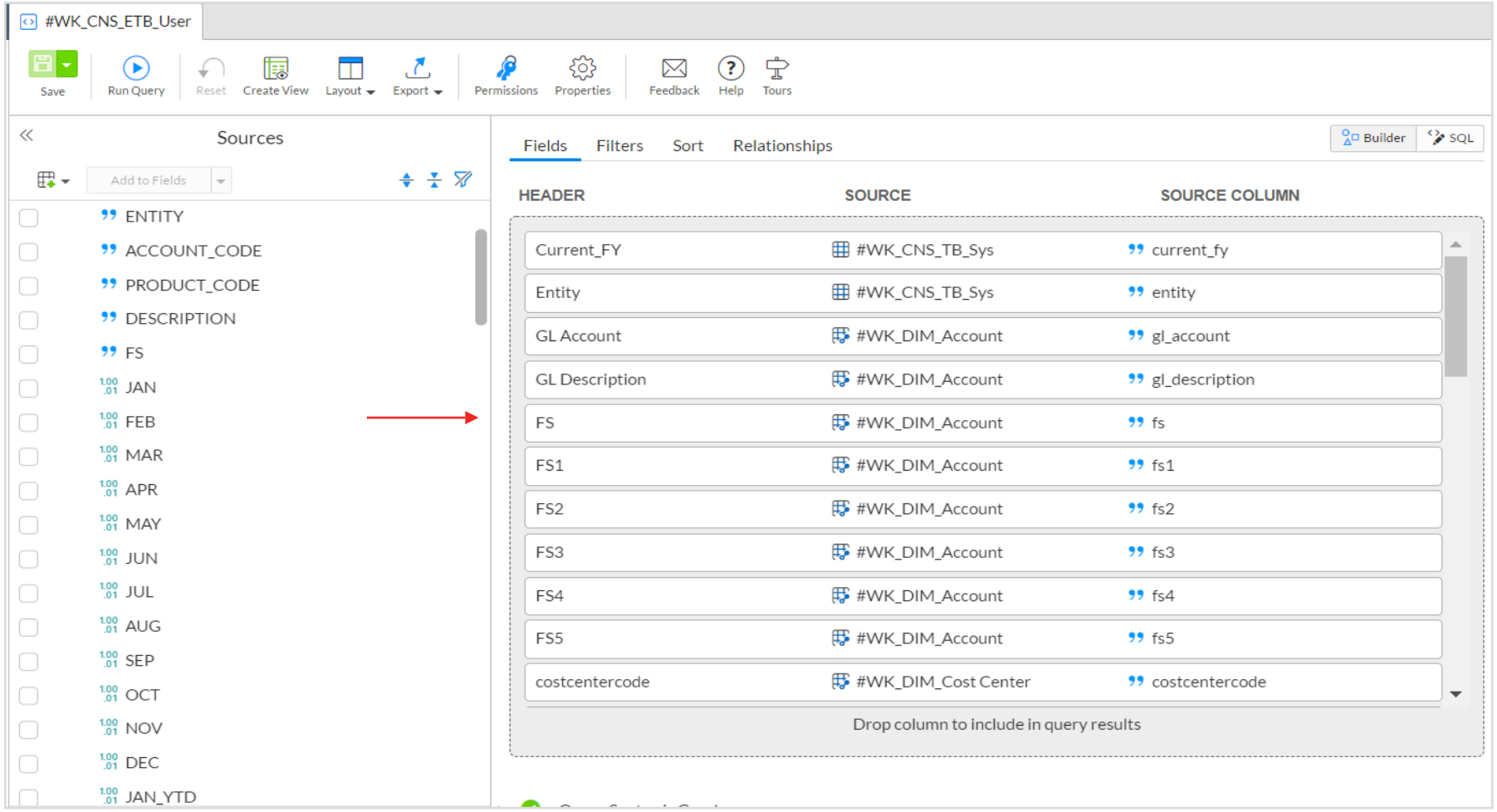
Create the following filters:

i. #WK_CNS_TB_Sys [Data_Year] = 'Data_Year' parameter

ii. #WK_CNS_TB_Sys [Data_Month] = 'Data_Month' parameter

workiva

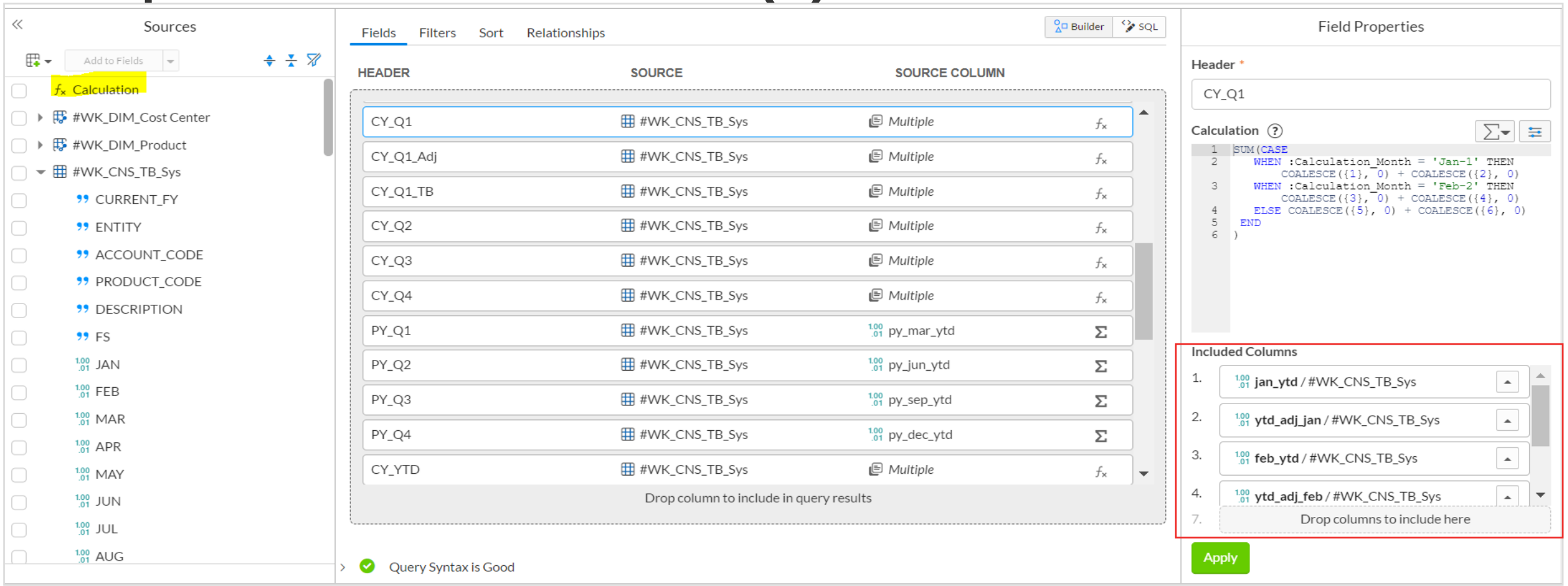
Step 22: Building Fields for the Query



Drag the required columns to the 'Fields' space. Note: The required columns should mirror the columns in the 'ETB' spreadsheet in Wdesk.

Workiva

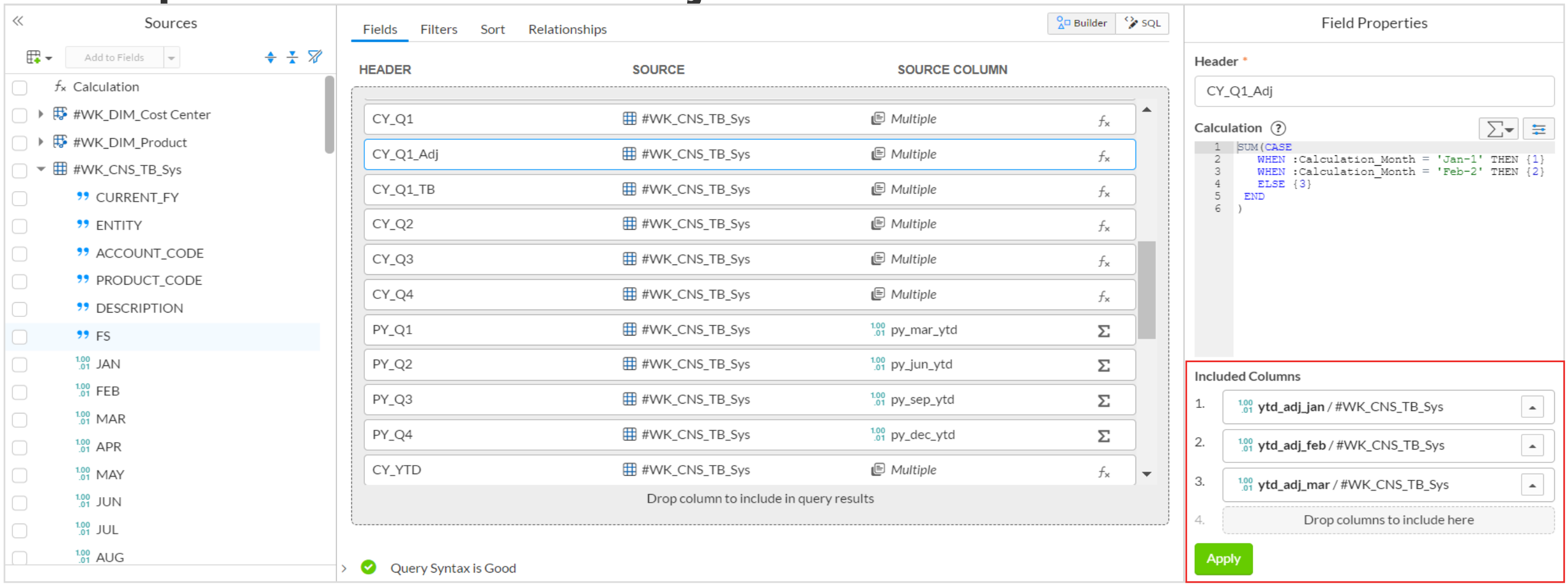
Step 22.1: Current Year Quarter (Q) Fields



To calculate the Current Year Quarter, we would be summing the YTD numbers & Adjustments. First pull a 'Calculation' into the 'Fields' space, then include "ytd" & "ytd_adj" columns into the space highlighted in red, along with the SUM CASE formula. As this is for Q1, we would only be including up to Mar' numbers.

Workiva

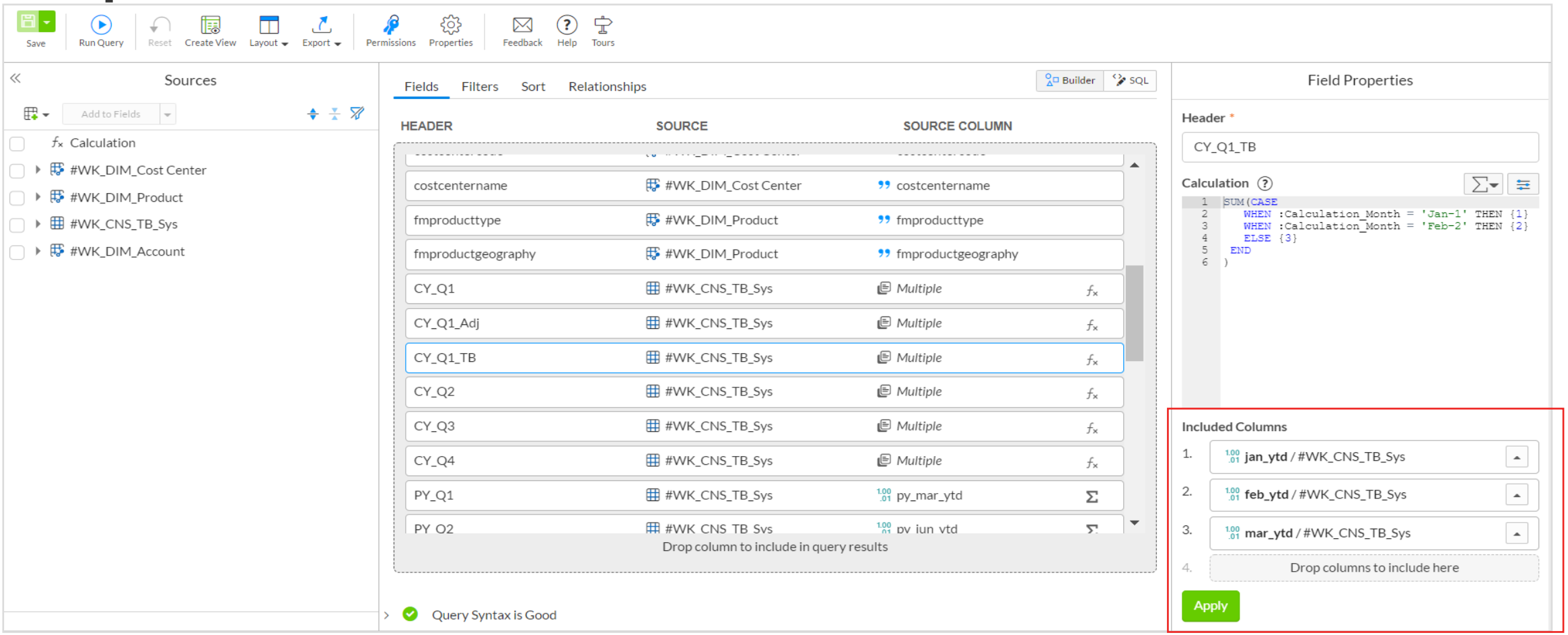
Step 22.2: Current Year Q Adjustment



To display adjustments figures by quarter, pull a 'Calculation' into the 'Fields' space, then include 'ytd_adj' columns along with the SUM CASE formula. As this is for Q1, we would only be including up to Mar' numbers.

Workiva

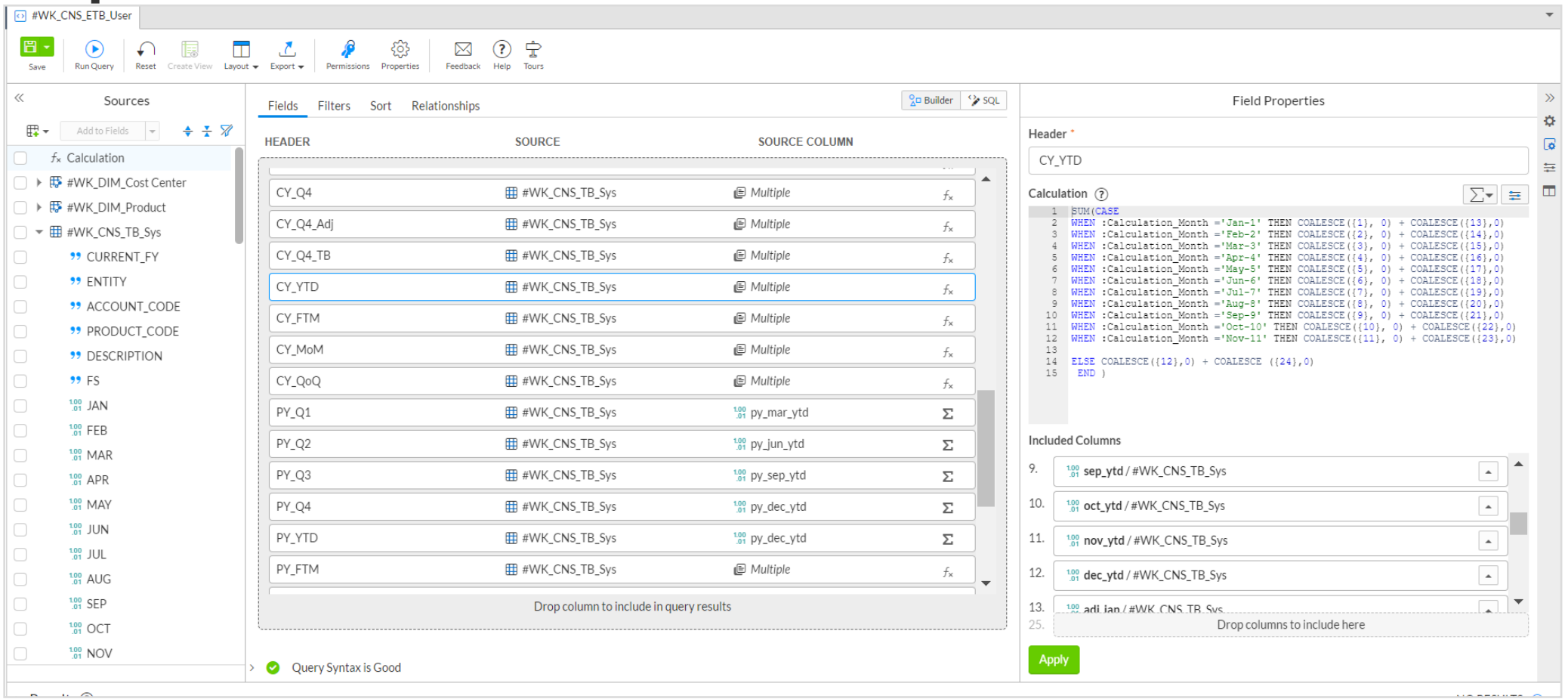
Step 22.3: Current Year Q TB



To display TB figures by quarter, pull a 'Calculation' into the 'Fields' space, then include '_ytd' columns along with the SUM CASE formula. As this is for Q1, we would only be including up to Mar' numbers.

Workiva

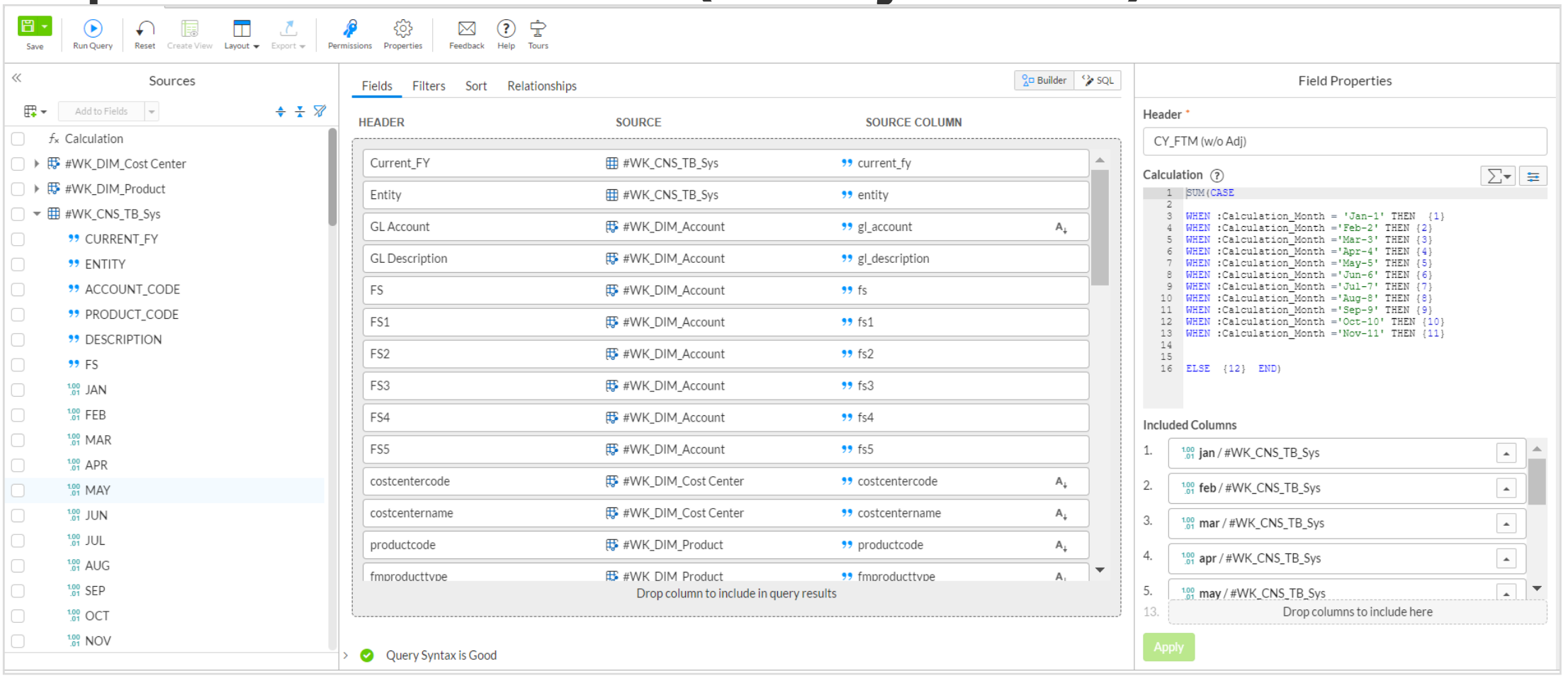
Step 23: Current Year YTD



To display Current Year's YTD, pull a 'Calculation' into the 'Fields' space, then include '_ytd' columns, along with the YTD adjustments columns along with the SUM CASE formula. This would serve as an aggregated figures for the whole year.

Workiva

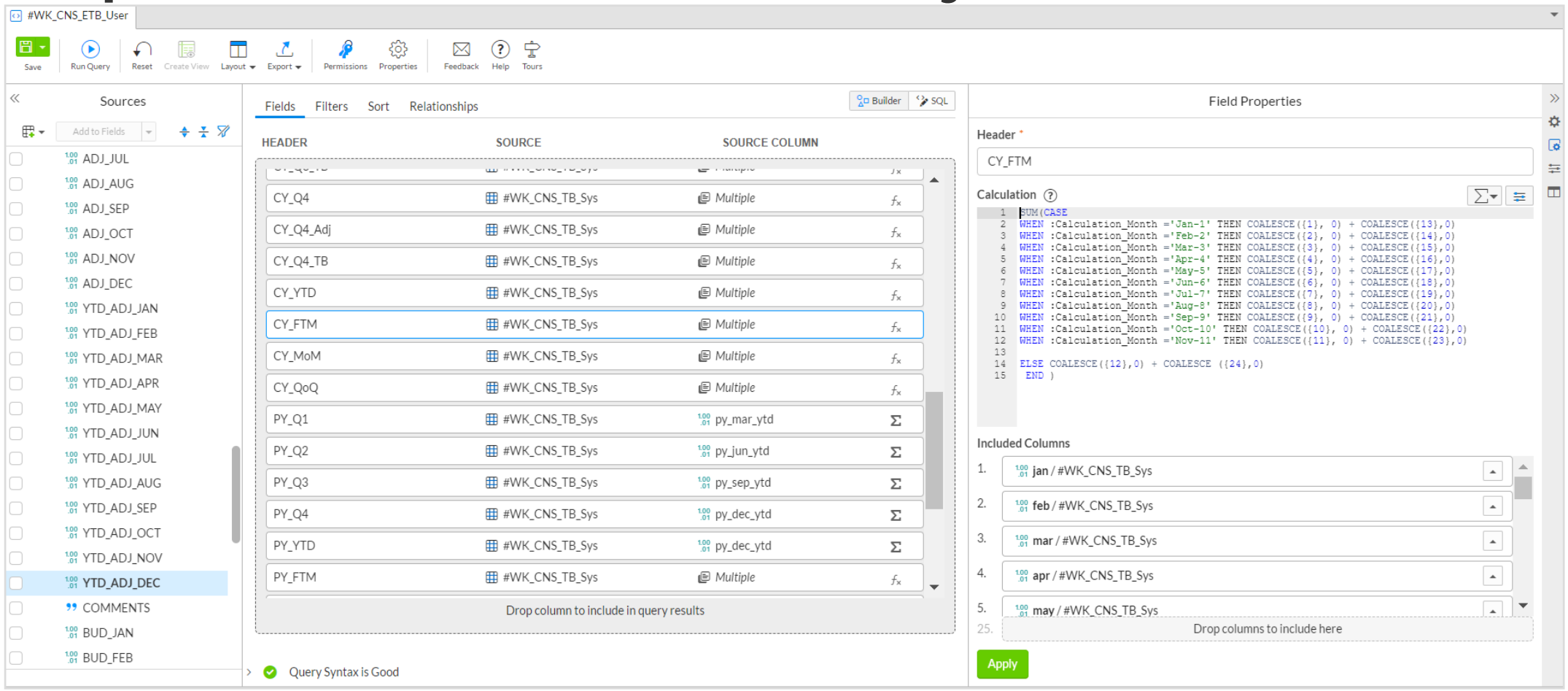
Step 23.1: Current Year FTM (w/o adjustments)



To display Current Year's For the Month (FTM), pull a 'Calculation' into the 'Fields' space, then include 'Jan-Dec', along with the SUM CASE formula. Note that this query did not account for adjustments. Next slide illustrates an example which accounts for adjustments.

Workiva

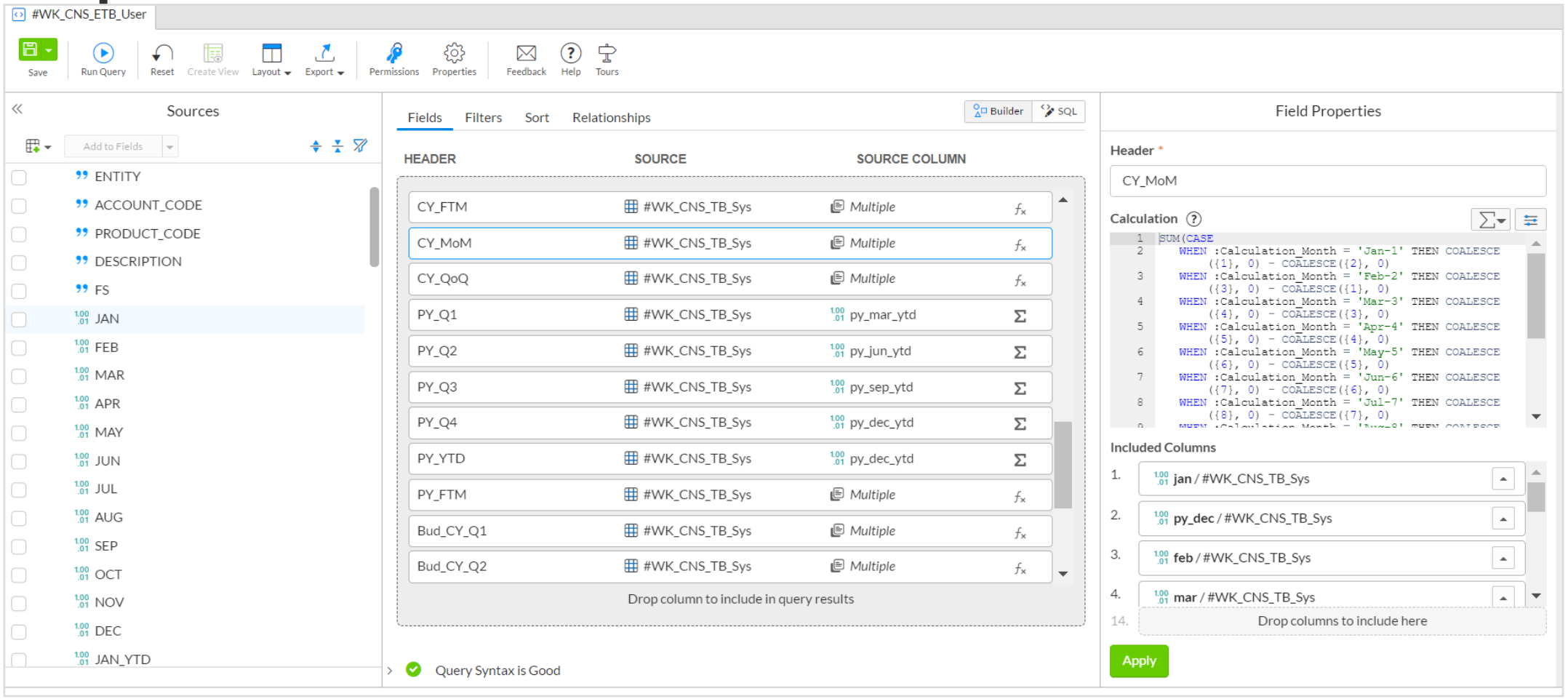
Step 23.2: Current Year FTM (with Adjustments)



To display Current Year's For the Month (FTM), pull a 'Calculation' into the 'Fields' space, then include 'Jan-Dec', along with the adjustment columns along with the SUM CASE formula.

Workiva

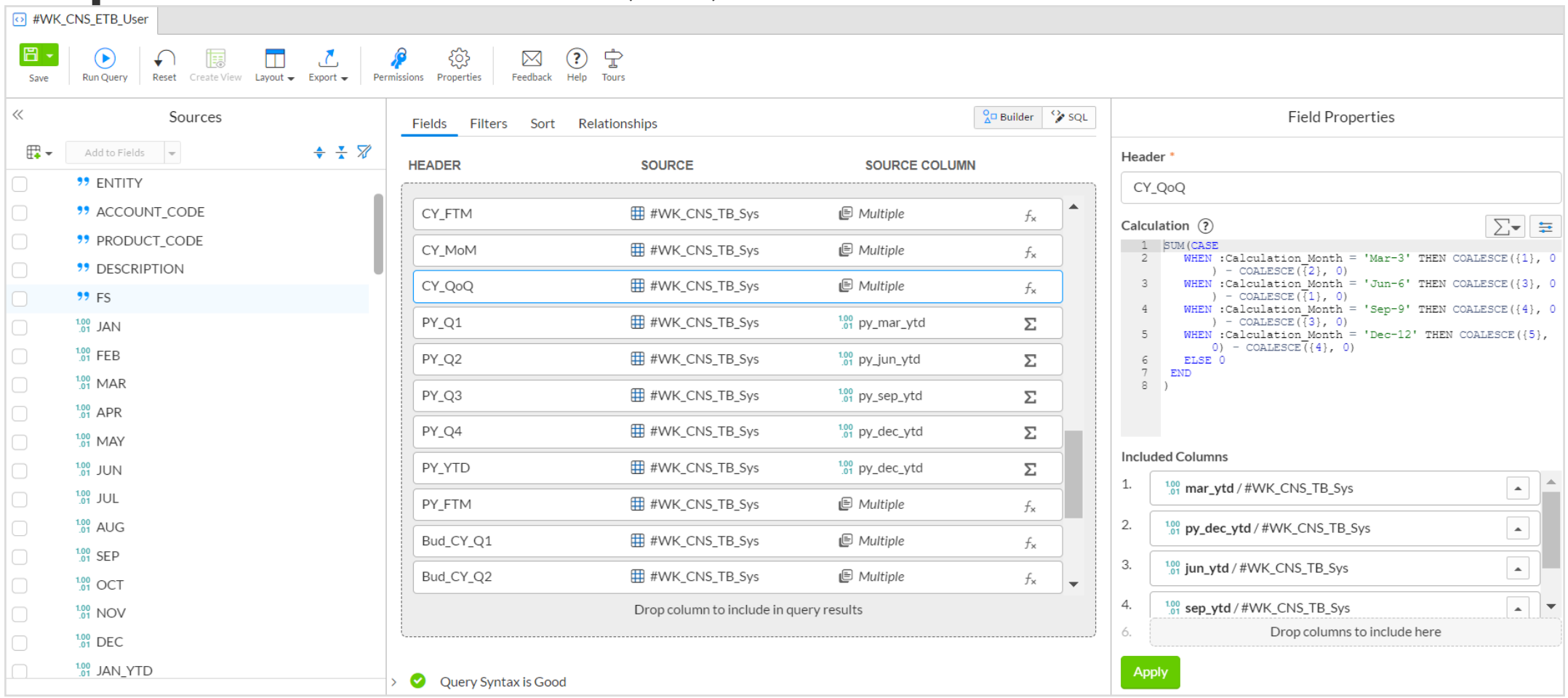
Step 23.3: Current Year MOM



To display Current Year's Month on Month (MOM), pull a 'Calculation' into the 'Fields' space, then with the SUM CASE formula, derive the difference in figures between months encapsulated with 'COALESCE'

Workiva

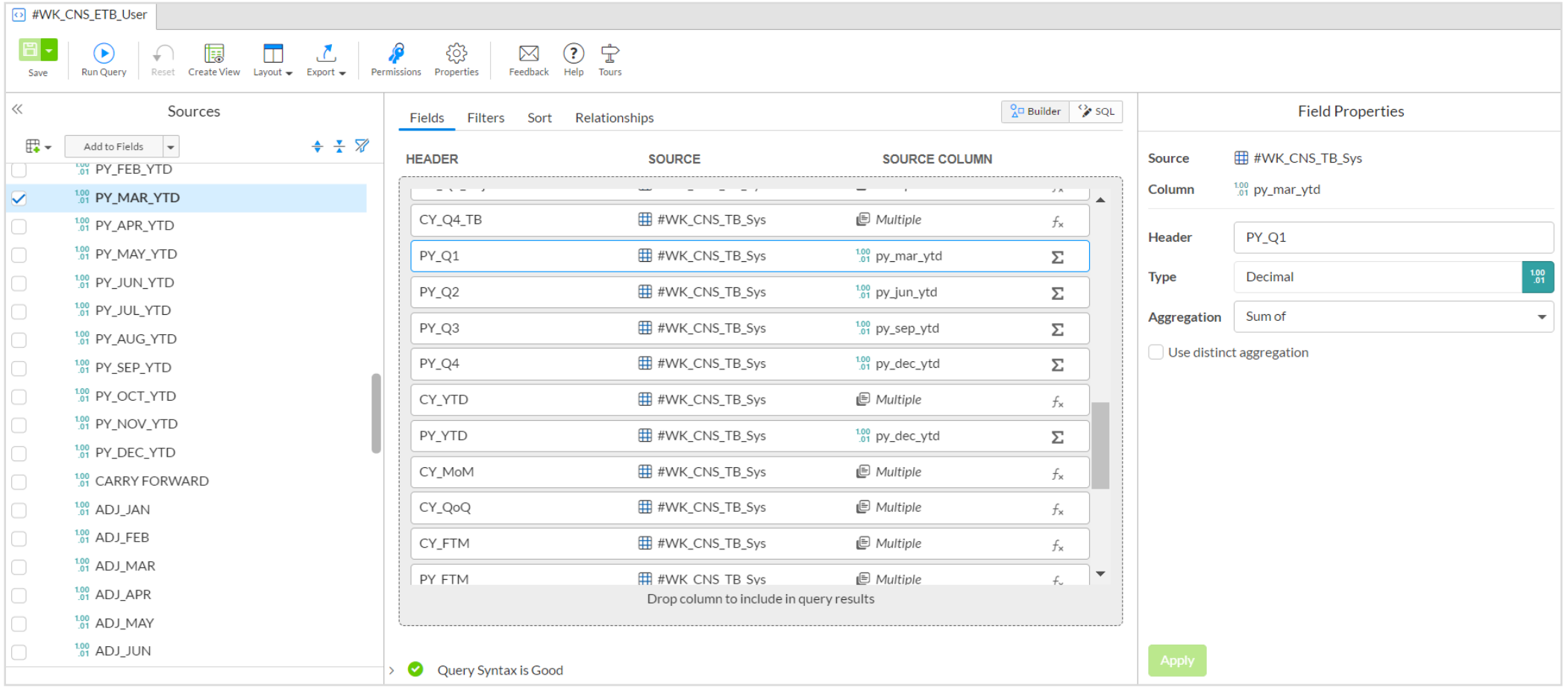
Step 23.4: Current Year QOQ



To display Current Year's Quarter on Quarter (QOQ), pull a 'Calculation' into the 'Fields' space, then with the SUM CASE formula, derive the difference in figures between quarters encapsulated with 'COALESCE'. Replicate for Budget.

Workiva

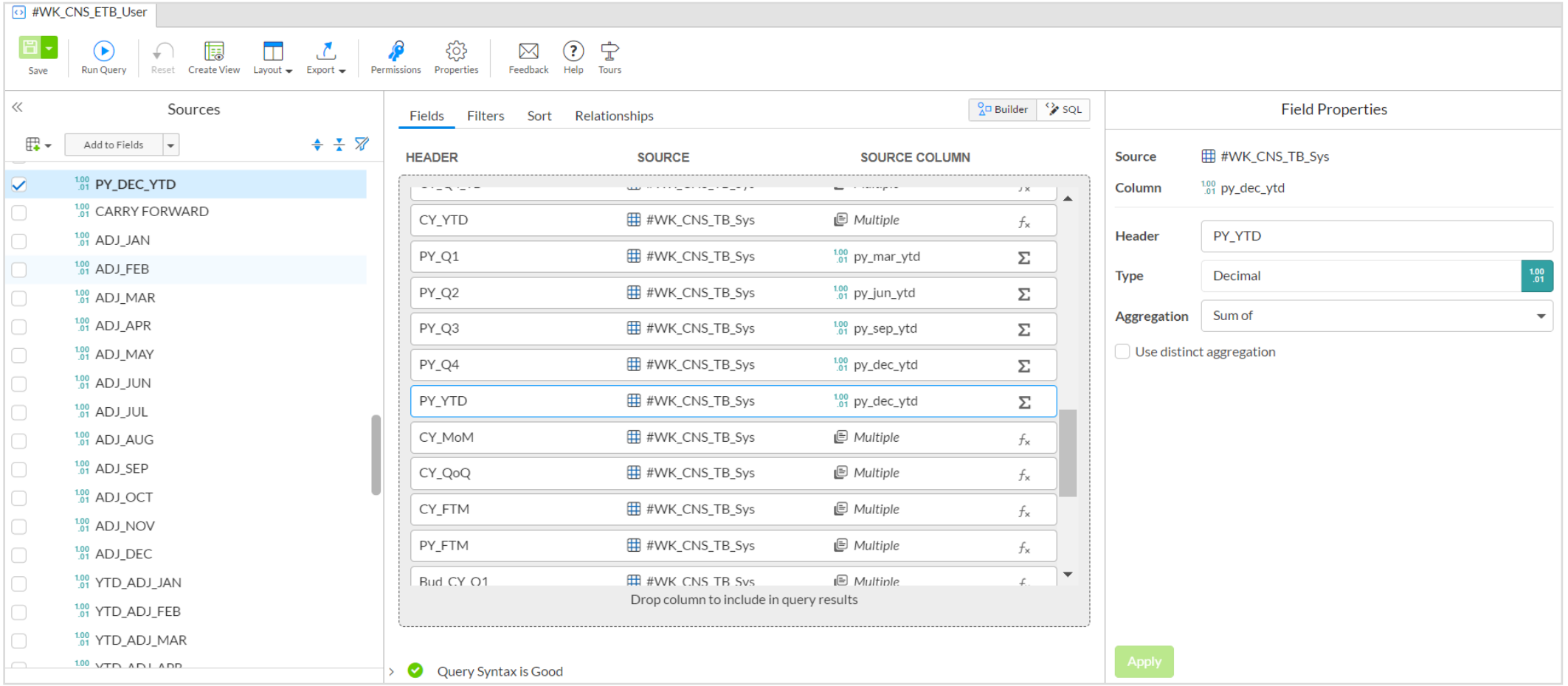
Step 23.5: Past Year's Quarter (Q) Fields



As Past Year's figures have already occurred, the Quarter (Q) Fields are simply a direct representation of the of the quarter's YTD, i.e. ('PY_MAR_YTD', 'PY_JUN_YTD', 'PY_SEP_YTD', 'PY_DEC_YTD')

Workiva

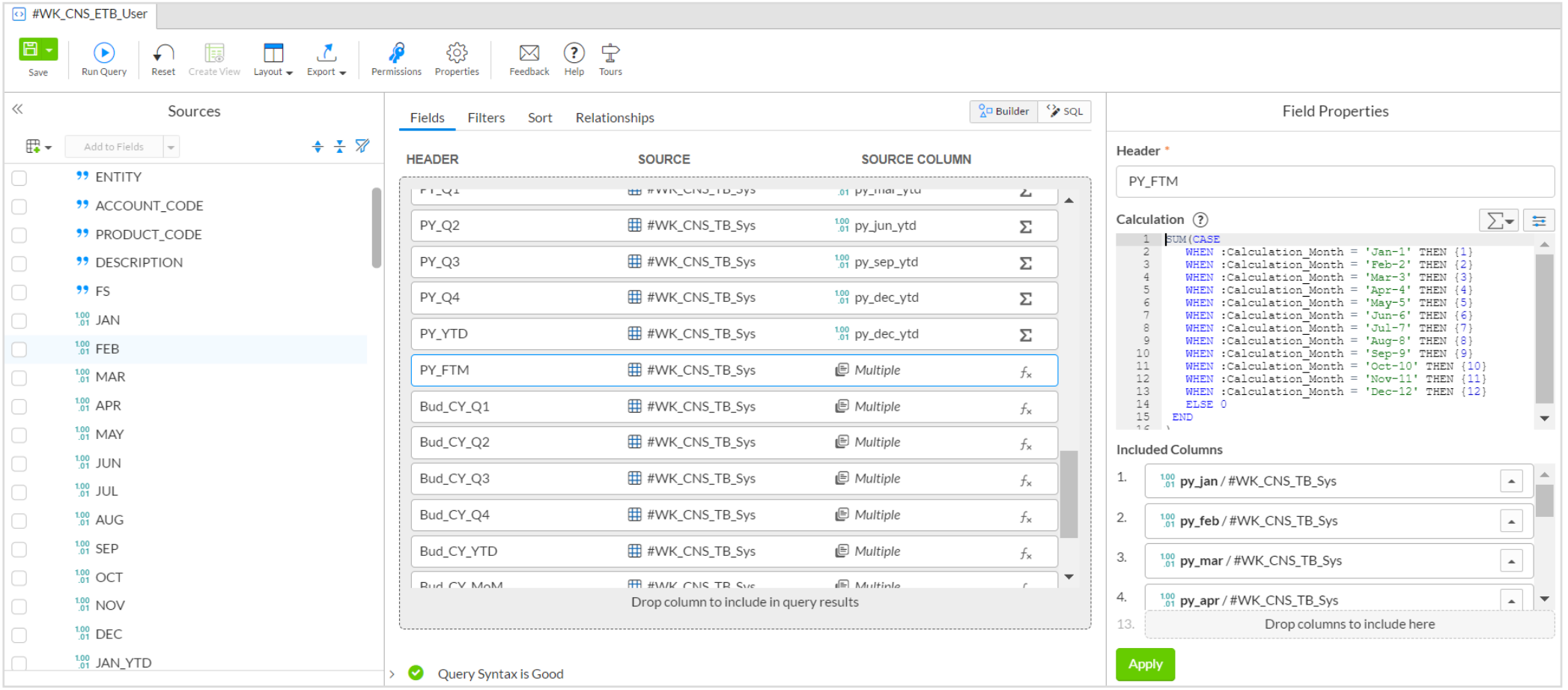
Step 24: Current Year YTD



To display Past Year's YTD, 'PY_DEC_YTD' would be an accurate representation for the year's YTD figures.

Workiva

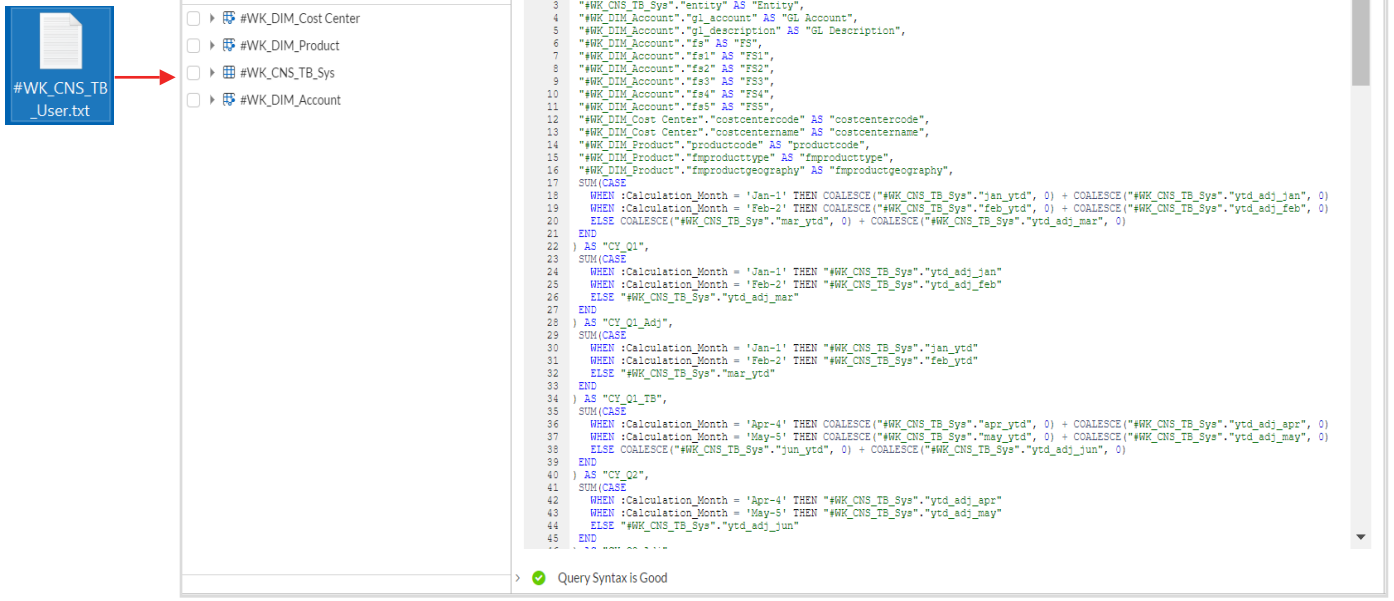
Step 24.1: Past Year's FTM



To display Past Year's FTM, pull a 'Calculation' into the 'Fields' space, then include 'PY_Jan-PY_Dec' columns along with the SUM CASE formula.

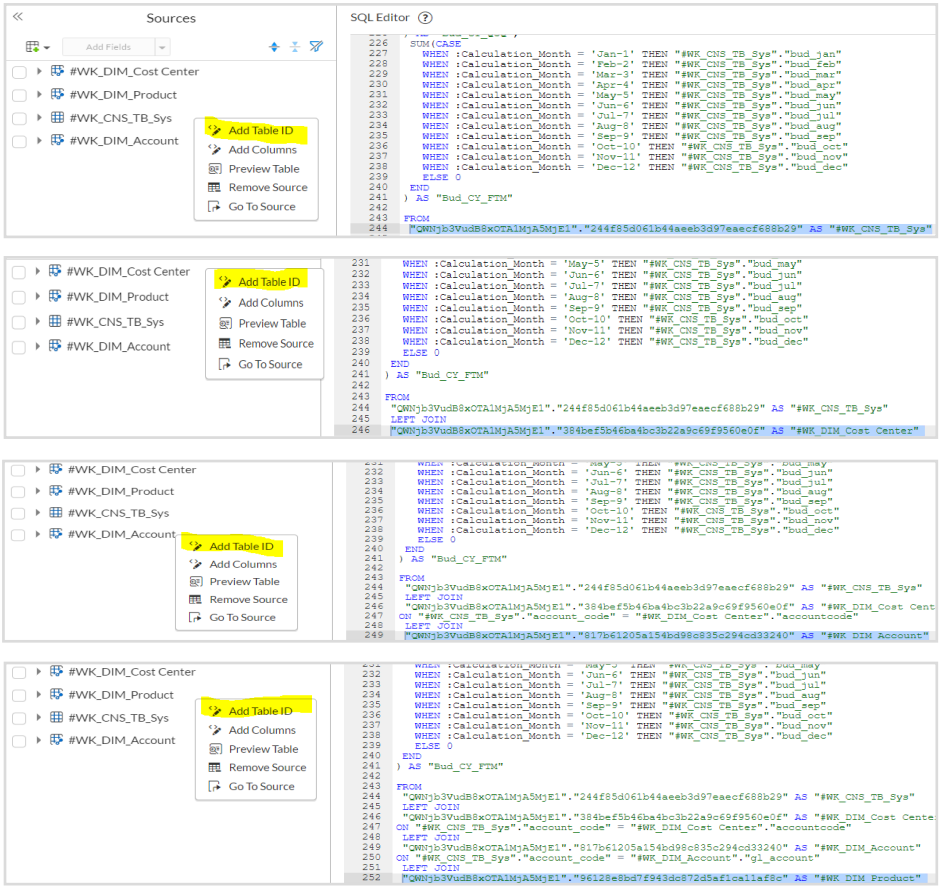
workiva

Step 24.2: Quick-Hack - #WK_CNS_TB_User



#WK_CNS_TB_User.txt

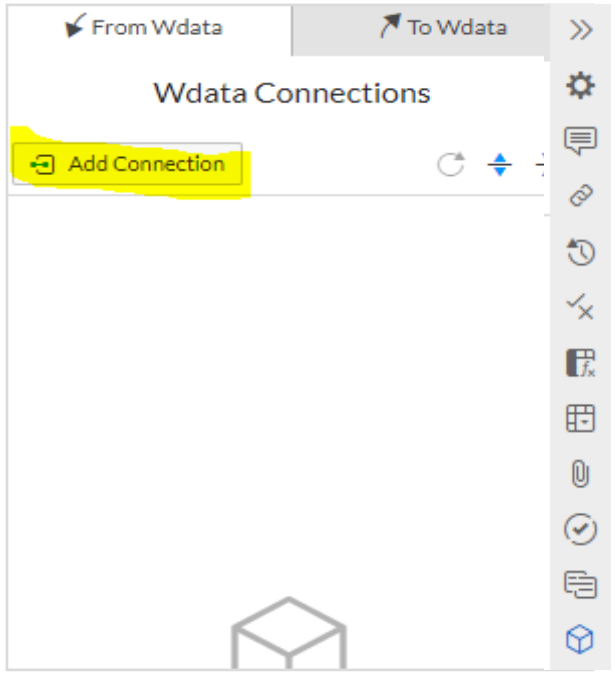
Replace Table IDs:

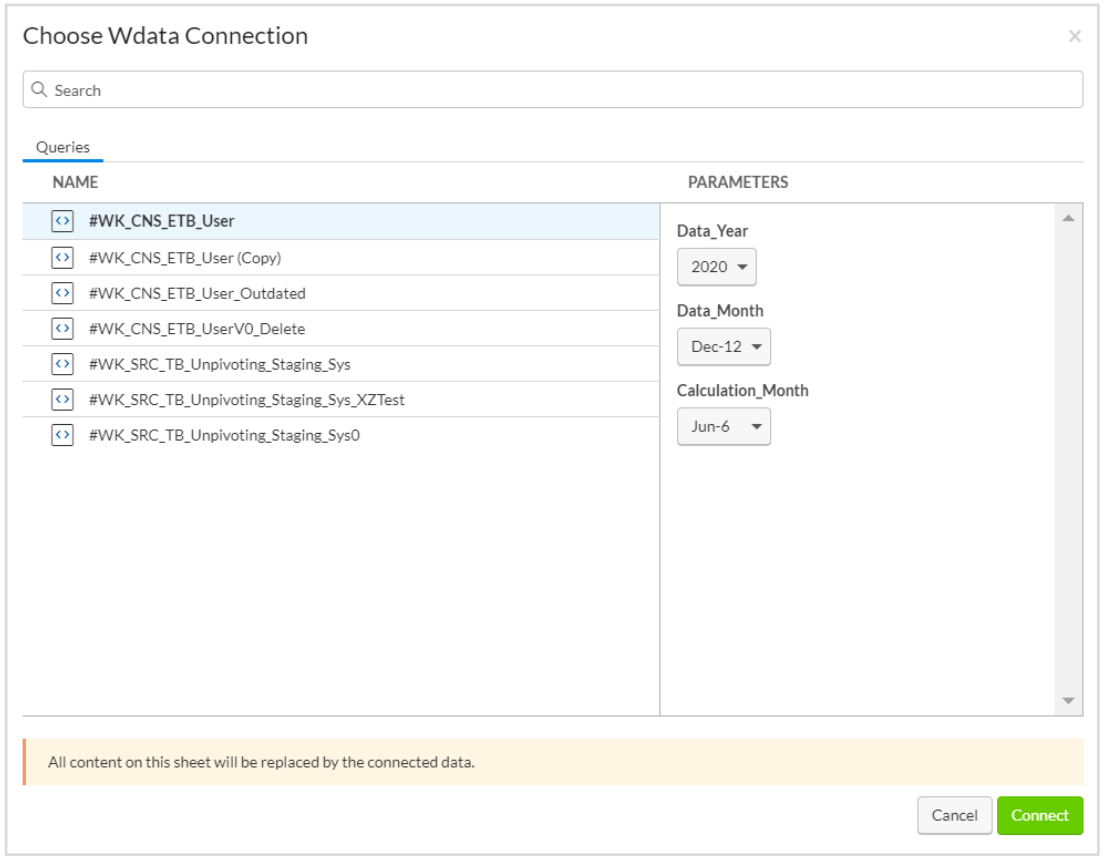


If your query doesn't work, copy paste the provided script into the 'SQL Editor', but remember to replace the 'Wdata Table ID at Line 244, 246, 249, 252. To replace the table ID, first highlight the entire line, right click on the '#WK_SRC_TB_Sys' table, and select "Add Table ID". Repeat for '#WK_DIM_Account', '#WK_DIM_Cost Center', '#WK_DIM_Account', '#WK_DIM_Product'.

Workiva

Step 25: Connecting to ETB Spreadsheet





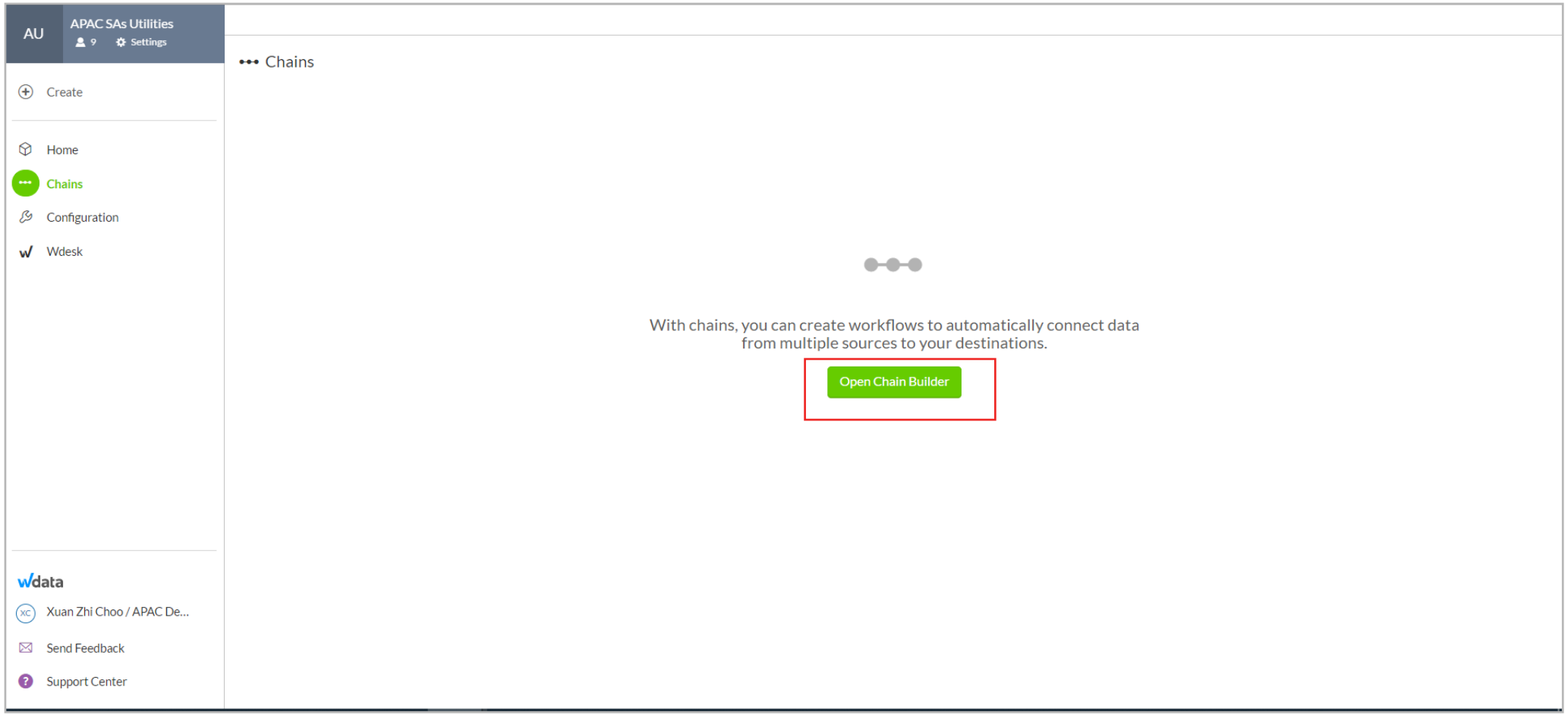
In the 'ETB' spreadsheet, go to the top right panel and click on the 'Wdata Connection' icon. Add Connection and select the ETB query (#WK_CNS_ETB_User) to be connected to this spreadsheet.

Workiva

Wdata Chains

Workiva

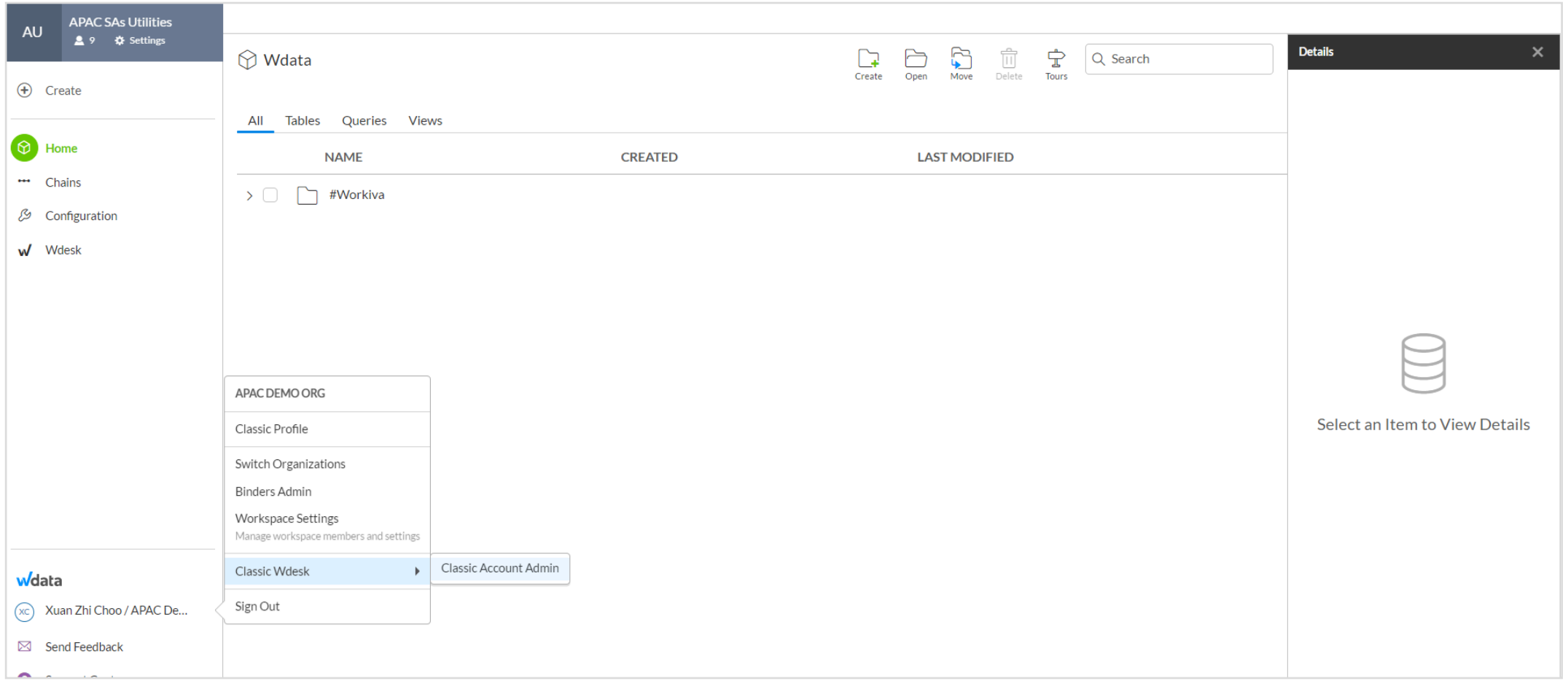
Step 1: Starting Chains



Start Chains by clicking 'Chains' on the left panel & click on 'Open Chain Builder'

workiva

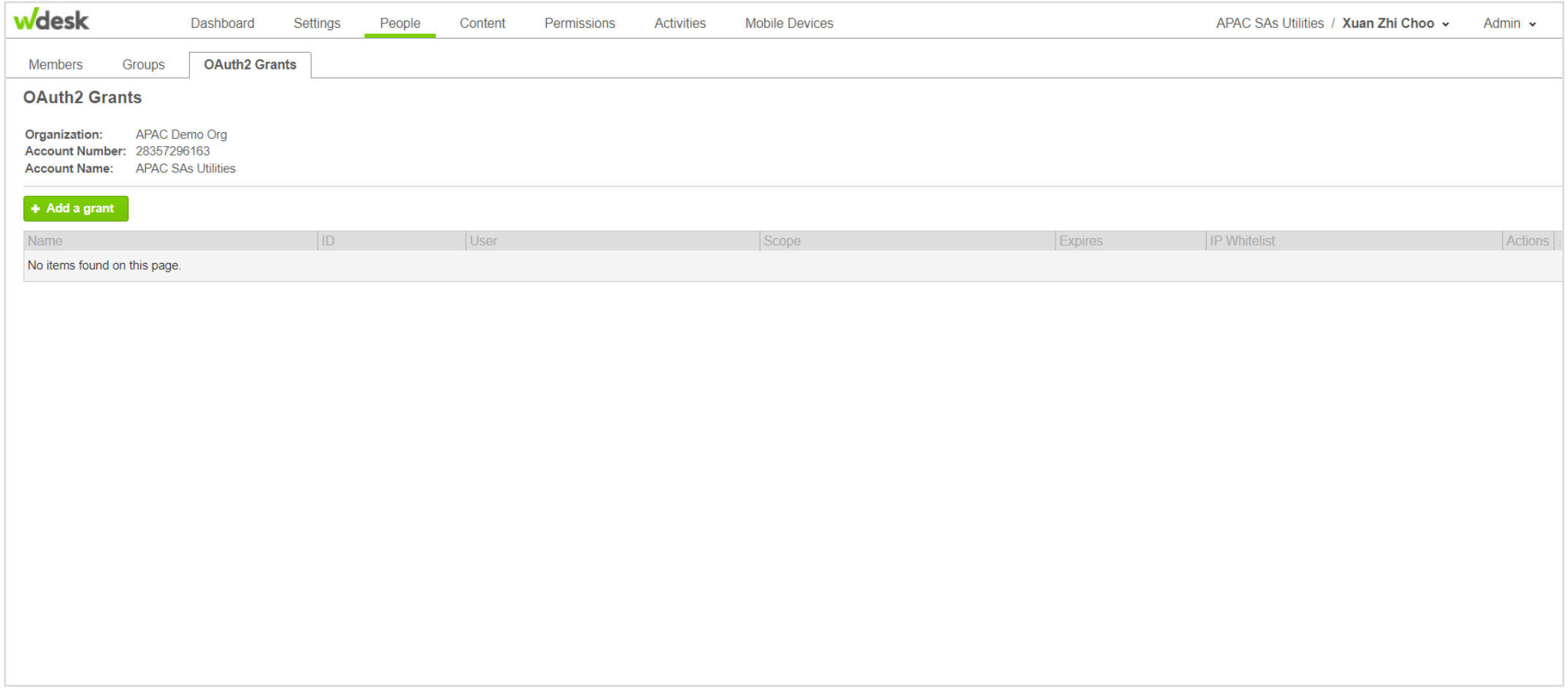
Step 2.1: Create Workiva Connection



At the Wdata homepage, click on your username on the bottom left and select 'Classic Account Admin'

workiva

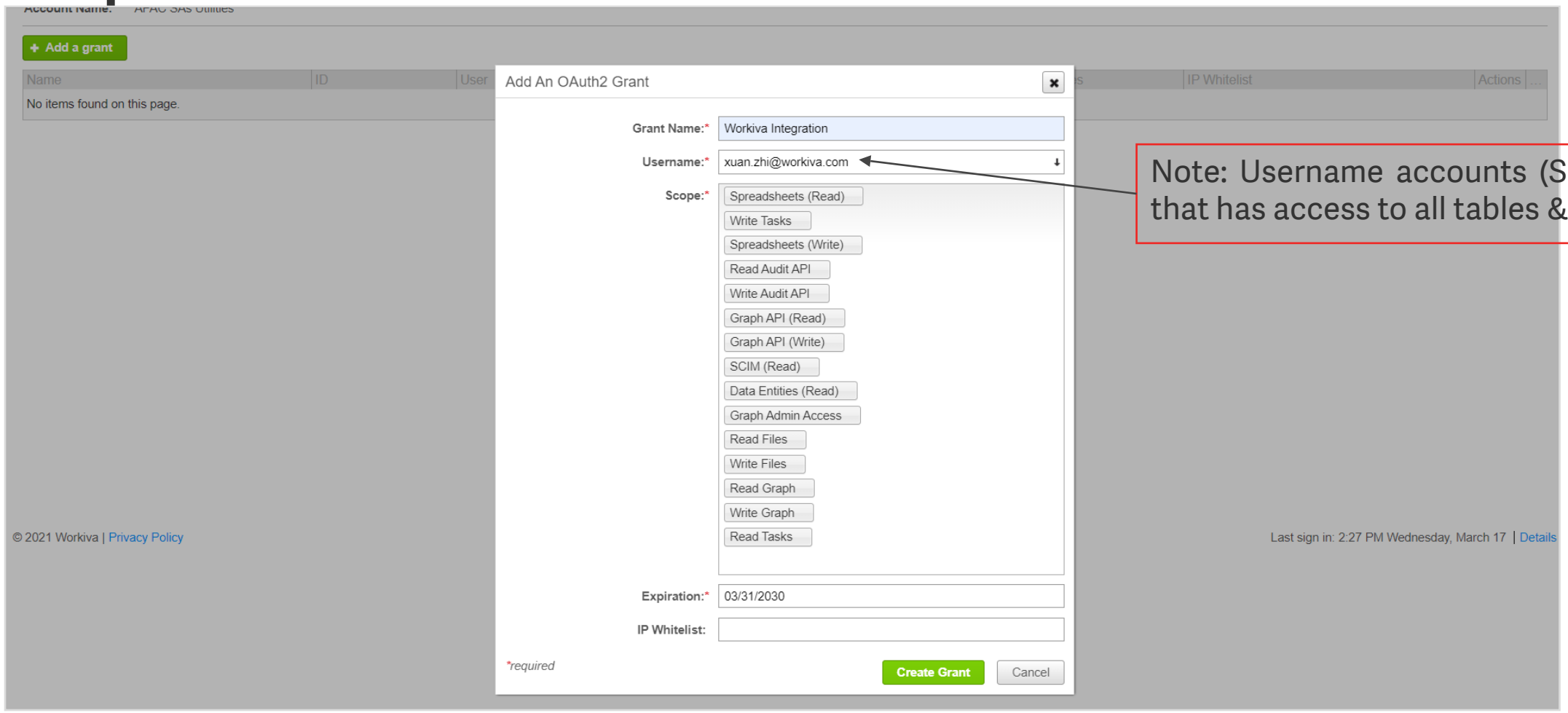
Step 2.2: Create 'OAuth2 Grant' P1



Creating an authentication grant by clicking 'Add a grant' here allows Wdata chains component to connect to Wdata & Wdesk via an API call.

workiva

Step 2.3: Create 'OAuth2 Grant' P2

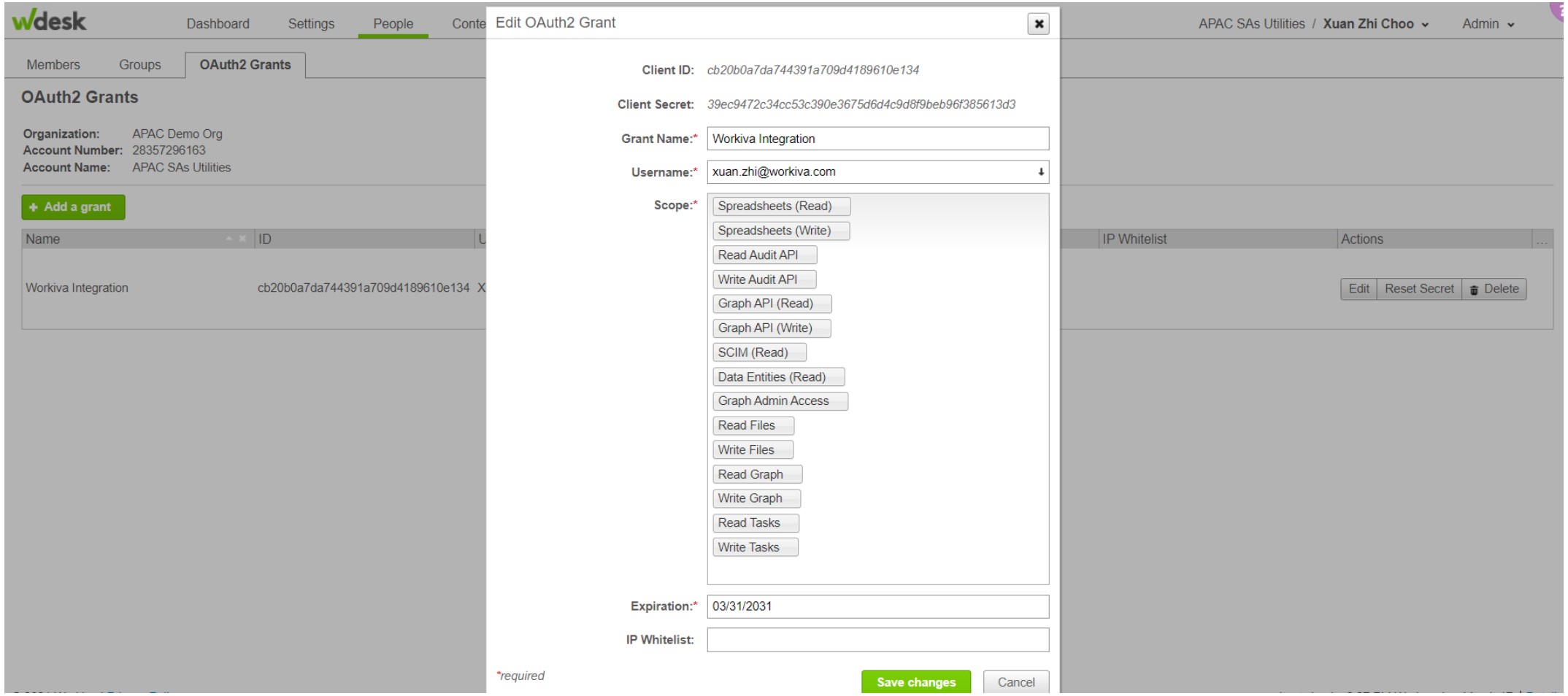


Note: Username accounts (Service Acc) that has access to all tables & queries

Once 'Create Grant' is clicked, it generates a unique 'Client ID' & 'Client Secret' that would be inputted in the chains connection to establish the connection.

Workiva

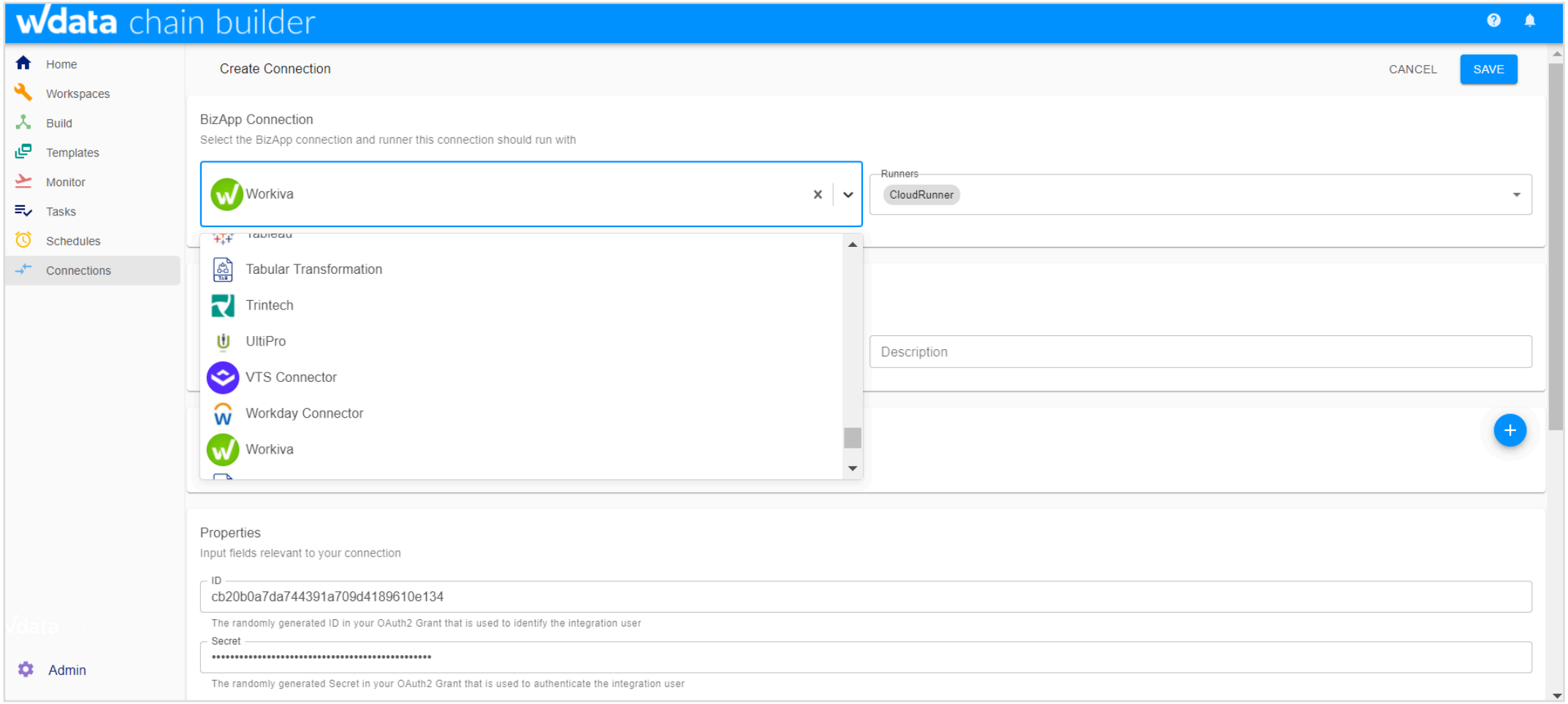
Step 2.4: Client ID & Client Secret



Copy the 'Client ID' & 'Client Secret' - paste it under the 'Properties' section as shown in "Step 2.1"

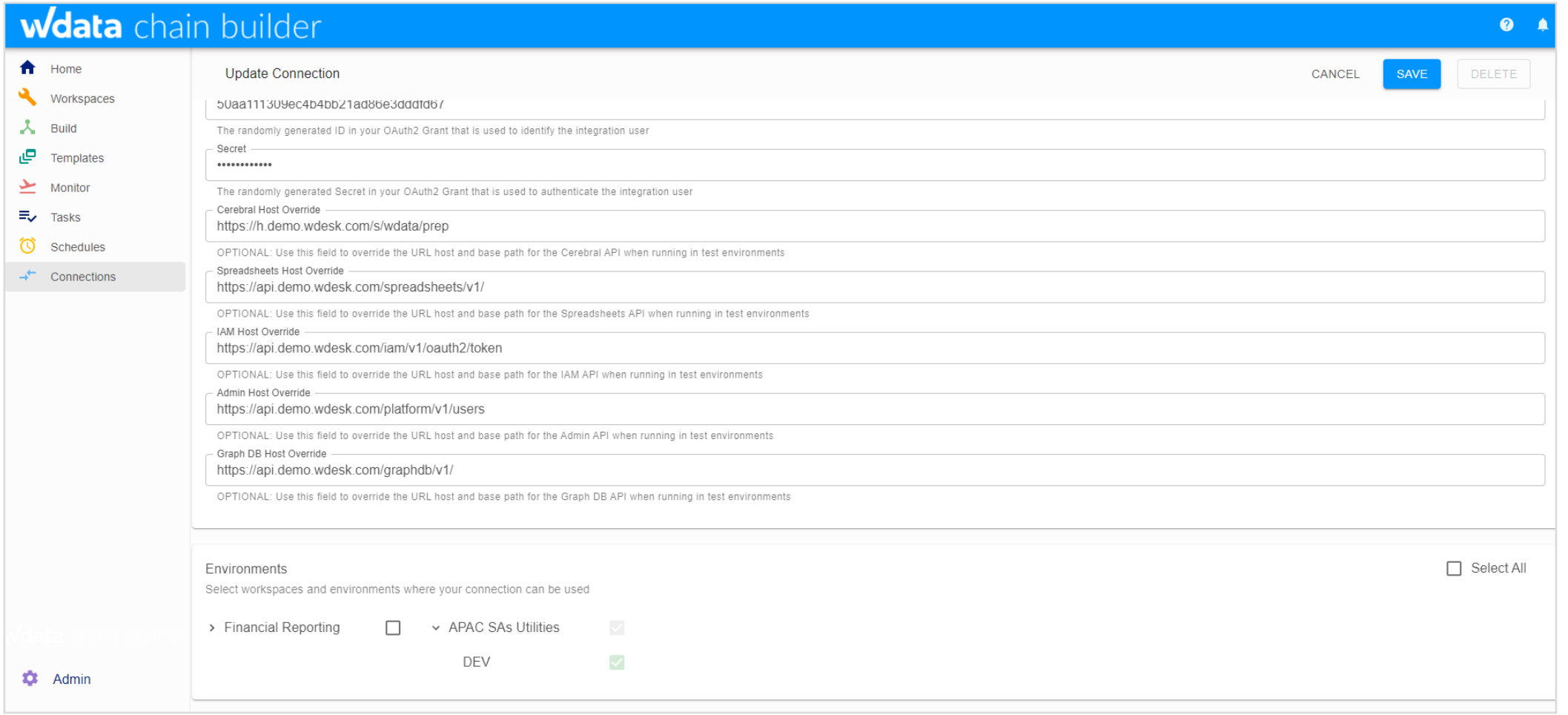
workiva

Step 2.5: Creating Workiva Connection P1



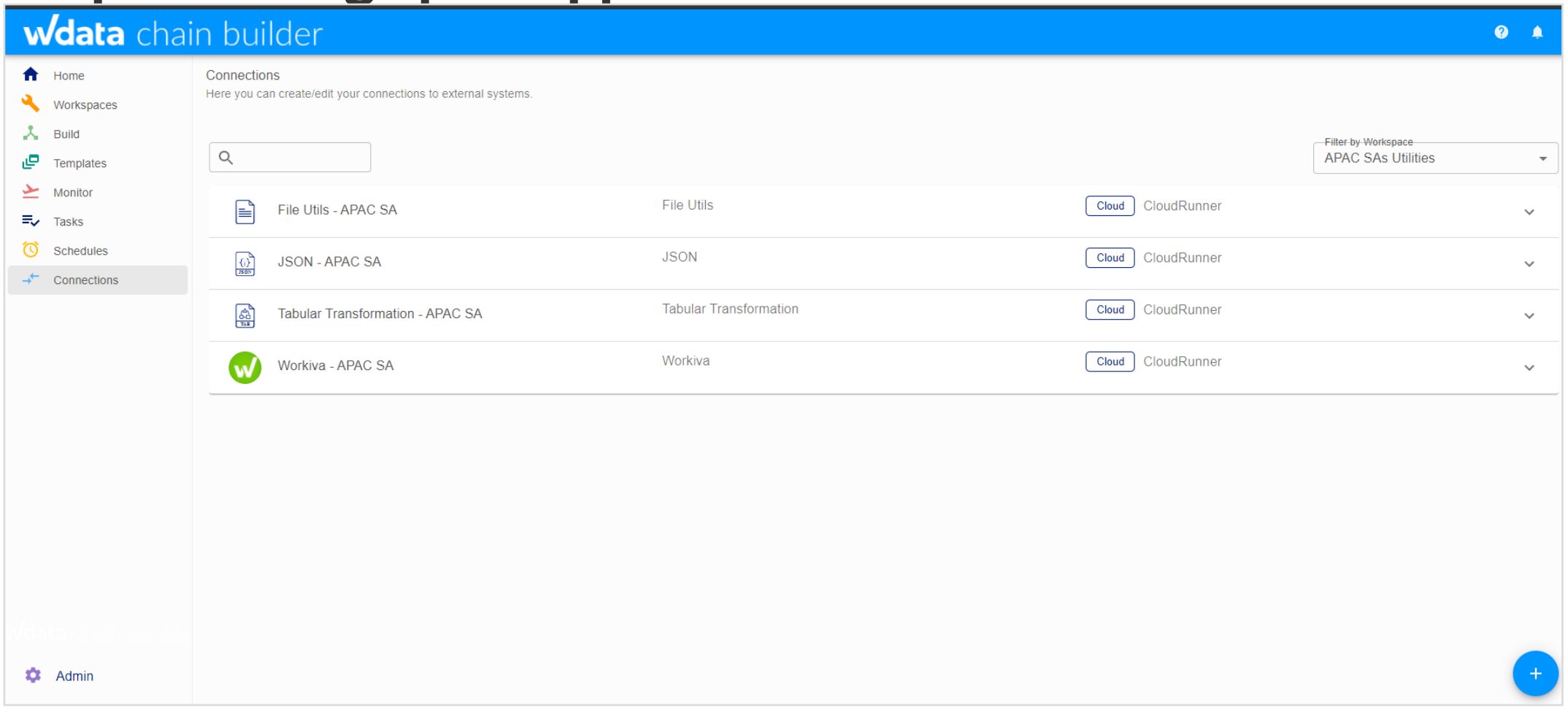
Creating a BizApp connection enables 'Chains' to talk to the Wdata & Wdesk spreadsheets. Common BizApps connections include "Workiva", "JSON", "Tabular Transformation", "File Utils".

Step 2.6: Creating Workiva Connection P2



Lastly, enable all the relevant environments that the chain would be deployed to. Hit the 'Save' button on the top right and go on to create other BizApps.

Step 3: Setting up Bizapps



Common BizApps connections include "Workiva", "JSON", "Tabular Transformation", and "File Utils".

workiva

Wdata Utility Chains

Workiva

Overview of control sheets

	A	B	C	D	E	F	G	H	I	J	K	L
1	Run_Chain	FileName	TableID	QueryID_OR_SQL	TagKey1	TagVal1	TagKey2	TagVal2	TagKey3	TagVal3	TagKey4	TagVal4
2	Yes	Dec-12_2020.csv	244f85d061b44aeeb3d97eaecf688b29	a07950871751428a9df5f53811552e21	Data_Year	2020	Data_Month	Dec-12				
3	1	2	3	4	5	6	7	8	9	10	11	12
4												

M	N	O	P	Q	R	S	T	U	V
ParamKey1	ParamVal1	ParamKey2	ParamVal2	ParamKey3	ParamVal3	ParamKey4	ParamVal4	Chain_Type	Last_Run
Data_Year	2020	Data_Month	Dec-12						
13	14	15	16	17	18	19	20		

Control sheets store the data variables that is required to run the chain (e.g. the file name of the files to be uploaded, ID of the table to which the output will be uploaded, ID of the query that should be executed, the tags and parameters used in tables and queries).

When starter chain (Chain 3) starts to run, it'll first identify the data variables in the control sheets. Note that we have 20 data variables in our example control sheets. To ease the burden of creating multiple starter chain which identifies 20 data variables, we usually create a chain template (illustrated in 'Chain 3.0: Creating Template Chain' section).

Creating Control Sheets for Training

For this training, the following Control Sheets would be created:

i. Master Controls - Controls all the other control sheets below
ii. TB EXT_SRC - Controls the External to Source chain
iii. TB SRC_STG - Controls the Source to Staging chain
iv. Refresh_Bud_Adj - Controls the Refresh Budget / Adjustment Chain

Master Controls

C2 =IF(B2="Manual","https://h.demo.wdesk.com/s/wdata/oc/app/apac-demo-org/workspace/654/environment/943/studio/chain/26795","No link, running Auto")

	A	B	C	D	E
1	Chain	Run_Mode	Manual_Chain_Link	Last_Run	
2	#WK_MASTER	Auto	No link, running Auto	On - 06/04/2021 At- 10:07 AM	
3					

Master Controls - Controls all the other control sheets below

workiva

TB EXT_SRC

A12

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Run_Chain	FileName	TableID	QueryID_OR_SQL	TagKey1	TagVal1	TagKey2	TagVal2	TagKey3	TagVal3	TagKey4	TagVal4	ParamKey1	ParamVal1
2	Yes	Act_Dec-12_2020.csv	e36418c3a92846debcebbb3f3708a555		Data_Year	2020	Data_Month	Dec-12						
3	No	Act_Nov-11_2020.csv	e36418c3a92846debcebbb3f3708a555		Data_Year	2020	Data_Month	Nov-11						
4	No	Act_Dec-12_2019.csv	e36418c3a92846debcebbb3f3708a555		Data_Year	2019	Data_Month	Dec-12						

O	P	Q	R	S	T	U	V
ParamKey2	ParamVal2	ParamKey3	ParamVal3	ParamKey4	ParamVal4	Data_Type	Last_Run
						Act	
						Act	
						Act	

TB EXT_SRC - Controls the External to Source chain

TB SRC_STG

A2 ✕ ✓ f_x Yes

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Run_Chain	FileName	TableID	QueryID_OR_SQL	TagKey1	TagVal1	TagKey2	TagVal2	TagKey3	TagVal3	TagKey4	TagVal4	ParamKey1	ParamVal1
2	Yes	Dec-12_2020.csv	244f85d061b44aeeb3d97eaecf688b29	a07950871751428a9df5f53811552e21	Data_Year	2020	Data_Month	Dec-12					Data_Year	2020
3	1	2	3	4	5	6	7	8	9	10	11	12	13	14

O	P	Q	R	S	T	U	V
ParamKey2	ParamVal2	ParamKey3	ParamVal3	ParamKey4	ParamVal4	Data_Type	Last_Run
Data_Month	Dec-12						On - 06/04/2021
15	16	17	18	19	20		

TB SRC_STG - Controls the Source to Staging chain

workiva

Refresh_Bud_Adj

B19

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Run_Chain	FileName	TableID	SheetID	TagKey1	TagVal1	TagKey2	TagVal2	TagKey3	TagVal3	TagKey4	TagVal4	ParamKey1	ParamVal1
2	Yes	Bud_Dec-12_2020.csv	244f85d061b44aeeb3d97eaecf688b29	e91d388e5e5b4f74924eb3345829c657	Data_Year	2020	Data_Month	Dec-12						
3	Yes	Adj_Dec-12_2020.csv	244f85d061b44aeeb3d97eaecf688b29	857b6c33458944b5b81c4277bcac5baa	Data_Year	2020	Data_Month	Dec-12						
4														

O	P	Q	R	S	T	U	V
ParamKey2	ParamVal2	ParamKey3	ParamVal3	ParamKey4	ParamVal4	Data_Type	Last_Run
						Bud	
						Adj	

Refresh_Bud_Adj - Controls the Refresh Budget / Adjustment Chain

Intro to Wdata Utility Chains P1

Identify Variables

- Reads Chains Control sheet and identifies the data variables:
  - Run Chain (Yes/No)
  - Wdata Table
  - Wdata Query
  - Tags
  - Parameters

Checks and Removes Redundant File(s)

- Un-import the file from the Wdata Source Table
- Deletes the outdated/ redundant file(s) which would be replaced with the new file with the same file name

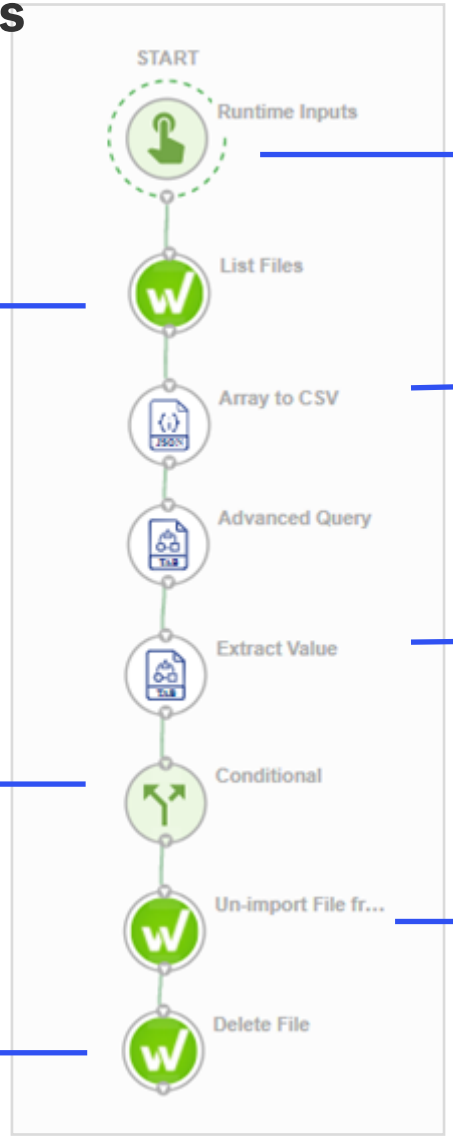
Run Query and Upload to the Wdata Table

- Runs the transformation query that transposes the data with the month data going down across to 12 columns
- Uploads to the results of the query to the Wdata Consumption table

Workiva

Chain 1: Checks and Removes Redundant File(s)

#WK_UTIL_Manage_Redundant_Files



Take the output variables from the previous chain

List the existing files in the Wdata Consumption Table

Does a file conversion to extract the ID of the files

Checks that there is a ID value

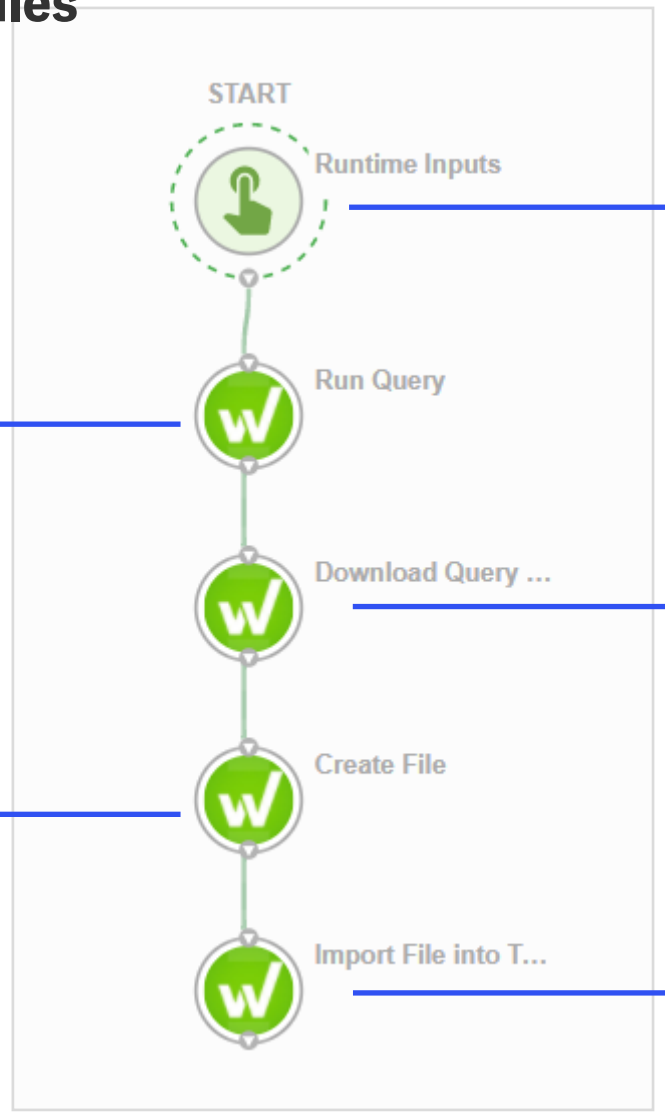
Command un-imports any exist file(s) that duplicates the one that is going to be imported

Deletes any redundant file(s) from the Wdata Consumption Table

Workiva

Chain 2: Run Query & Upload to Wdata Table

#WK_UTIL_Run_Query_Upload_Files



Triggered once the previous chain is completed

Runs the transformation query

Downloads the query results

Creates a file with the results and tags the results

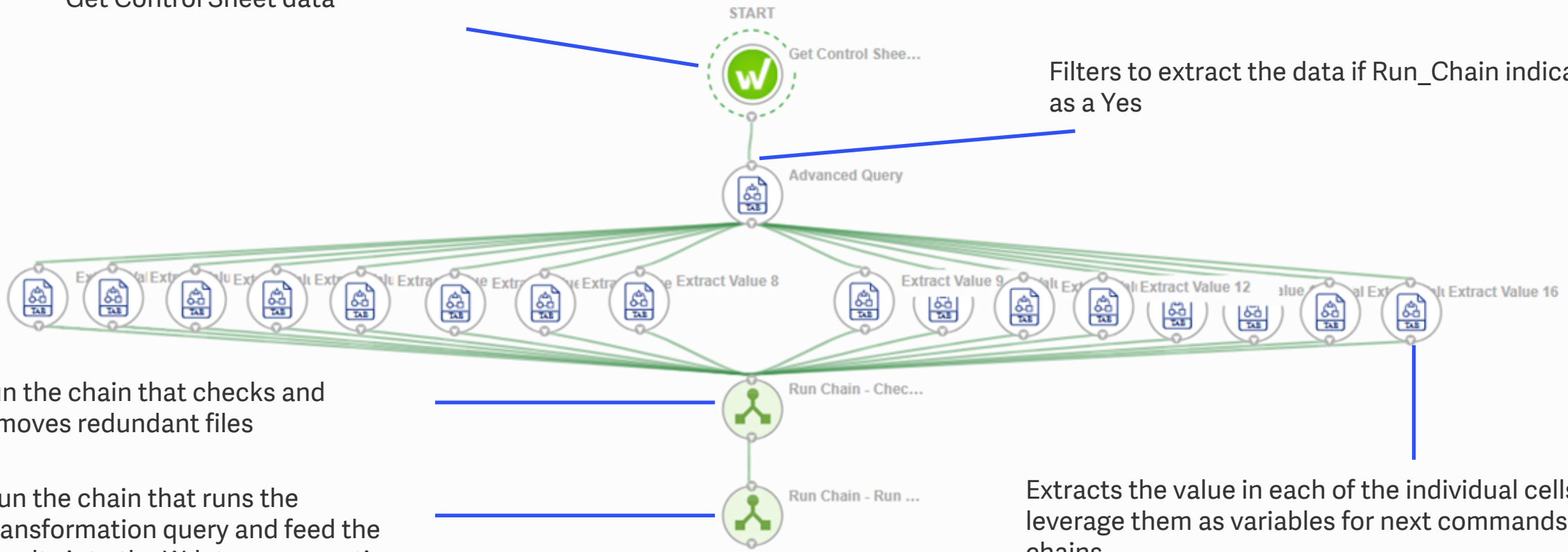
Imports the results into the Wdata Consumption Table

workiva

Chain 3: Identify Variables

#WK_TB_SRC_STG

Get Control Sheet data



Filters to extract the data if Run_Chain indicated as a Yes

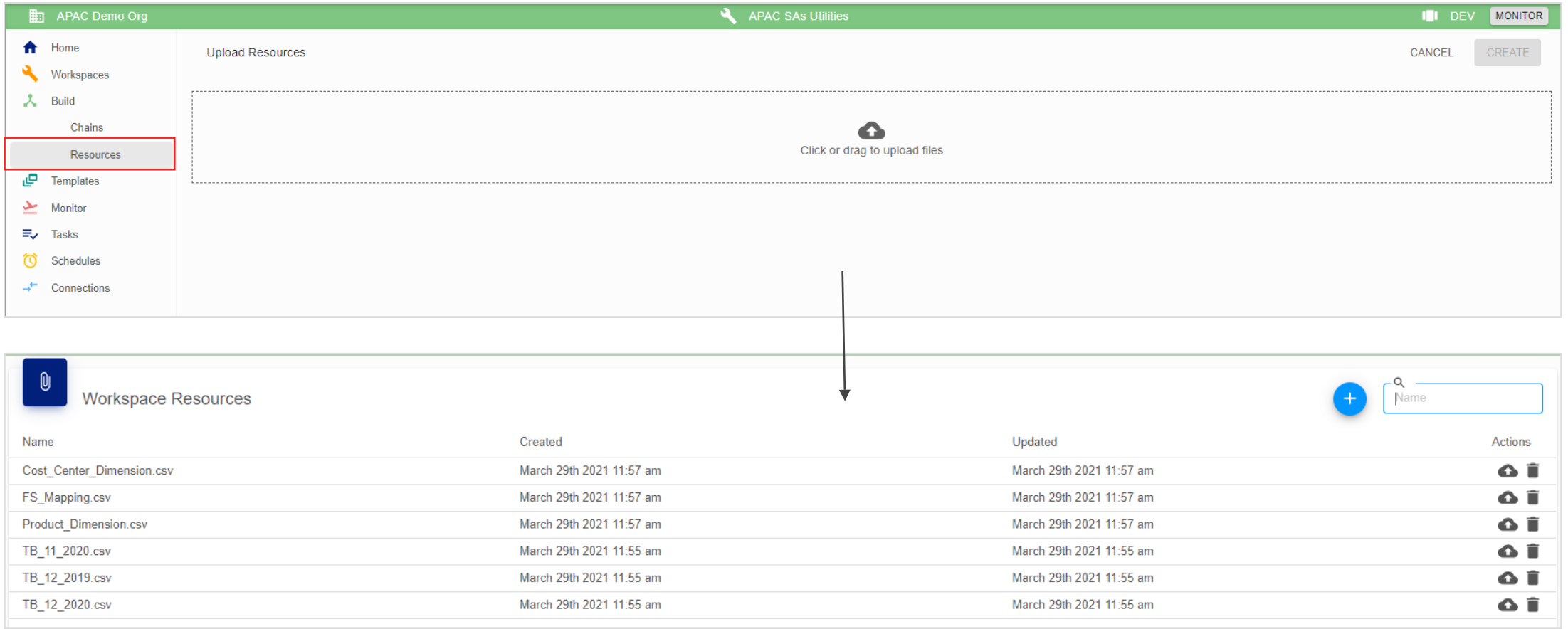
Run the chain that checks and removes redundant files

Run the chain that runs the transformation query and feed the results into the Wdata consumption table

Extracts the value in each of the individual cells and leverage them as variables for next commands in the chains

workiva

Upload Resources

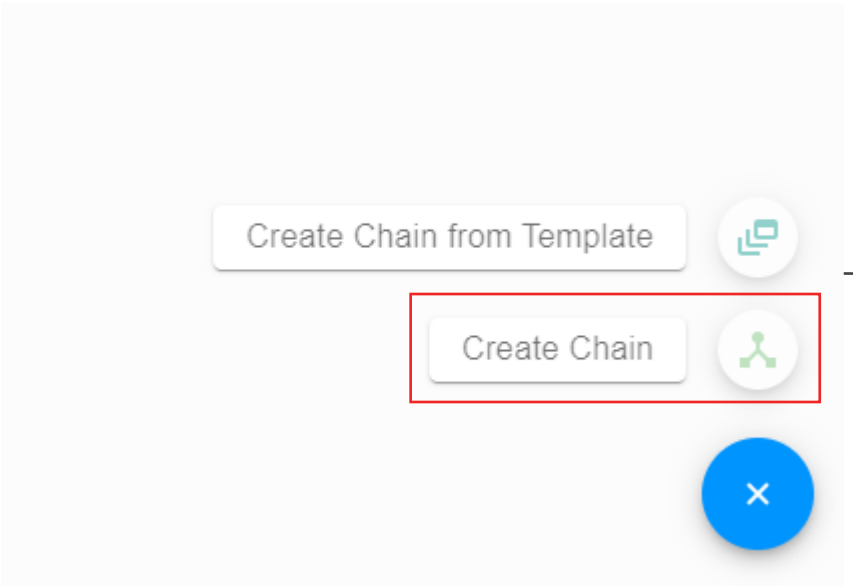


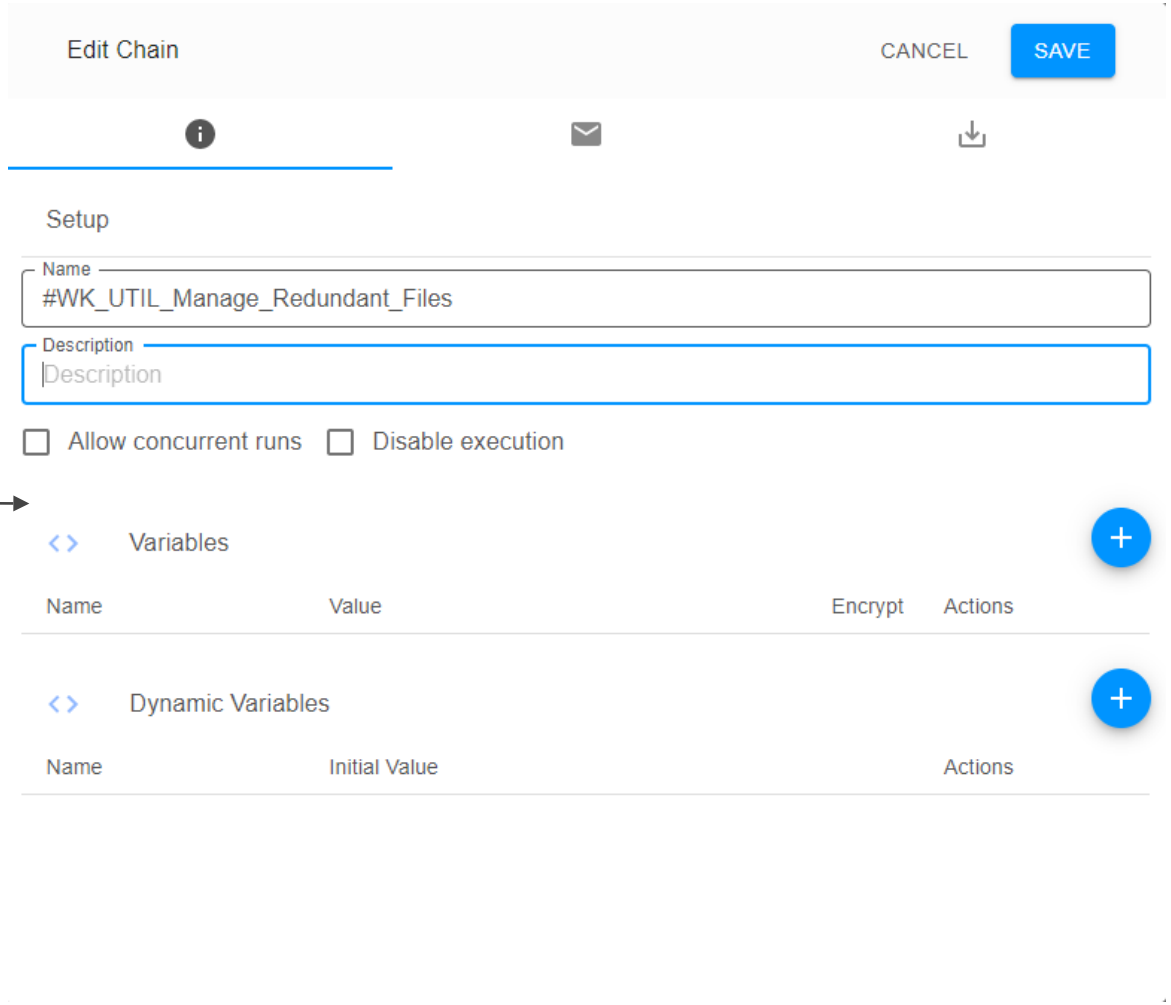
Click on "Build">"Resources" and upload the TB data and dimension tables as shown above. This would serve as a starting point for chains.

workiva

Chain 1: Manage Redundant Files (UTIL)

Step 1: Create New Chain

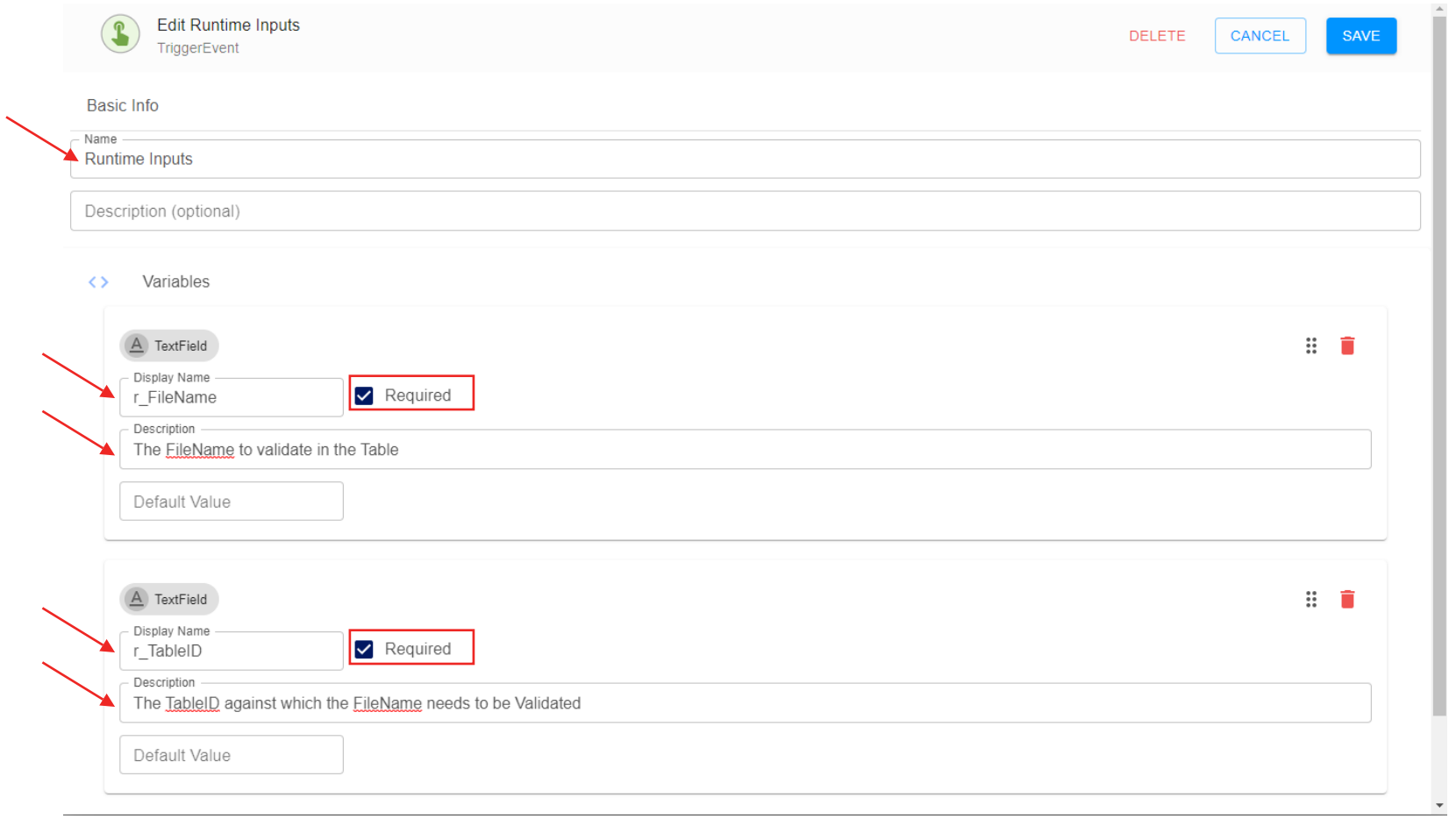




In "Build">"Chains" page, click the blue "+" button on the lower right corner to "Create Chain". Name the new chain as "#WK_UTIL_Manage_Redundant_Files" and hit save.

Workiva

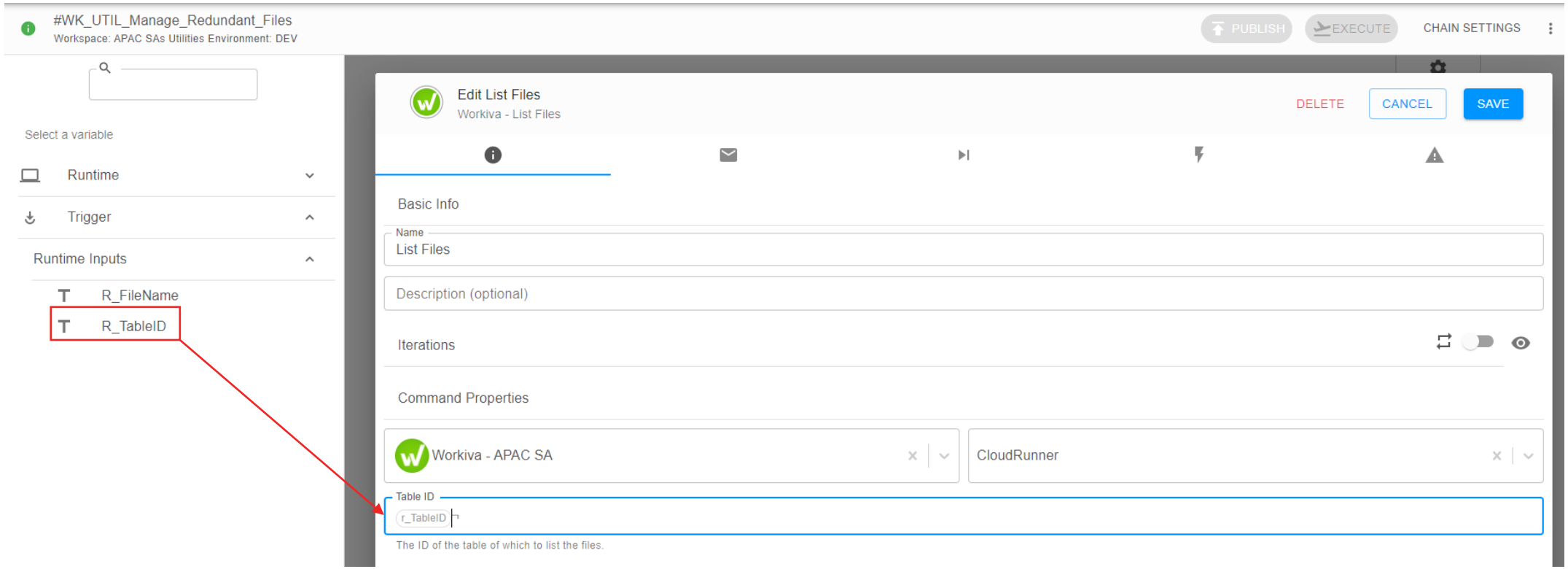
Step 2: Create List Files Node P1



Create a "Runtime Inputs" node and drag the node in the "Start" circle. Fill in the node information as shown above. Specifically, create two Text Field variables: "r_FileName" and "r_TableID".

Workiva

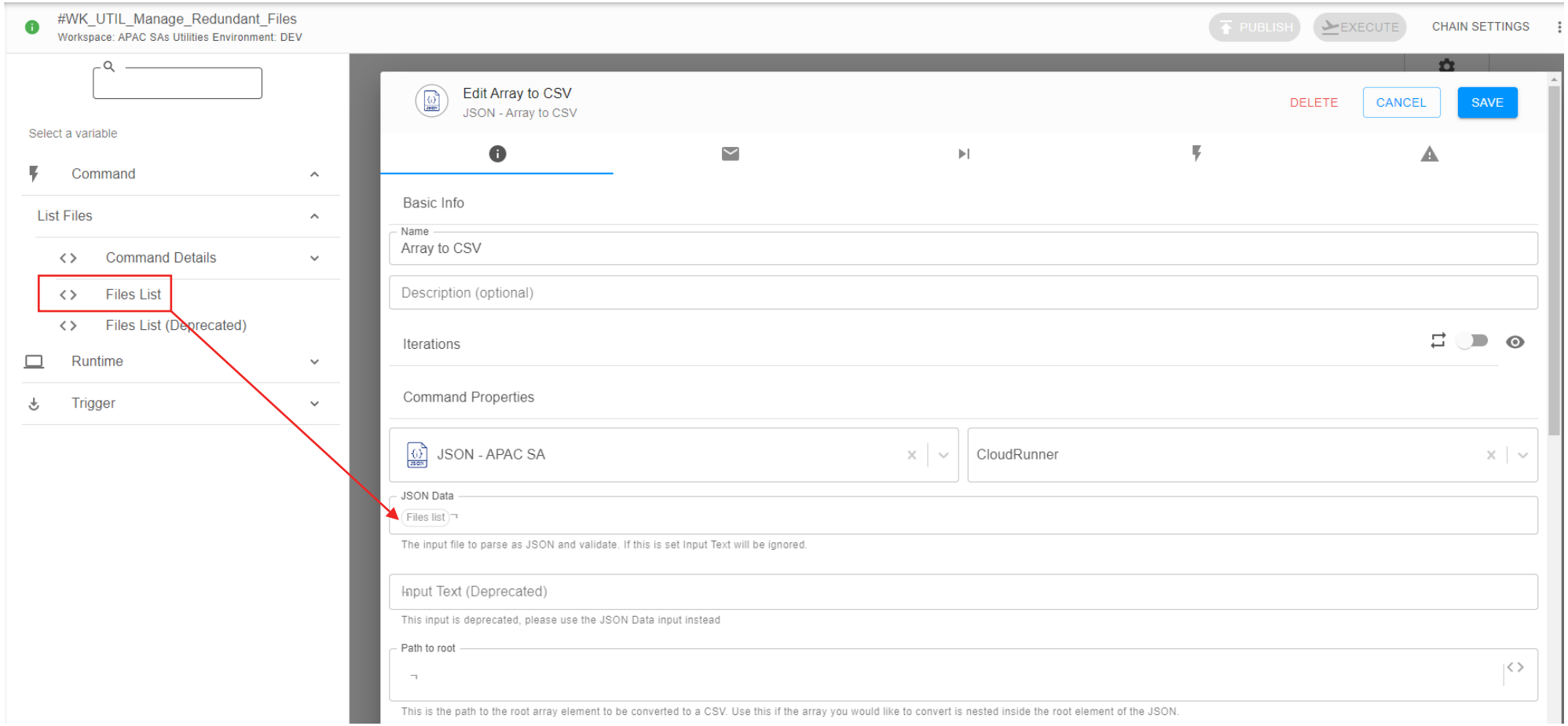
Step 3: Create List Files Node P2



Create a "Workvia">"List Files" node and link the node with the Runtime Inputs node. Fill in the node information as shown above. Specifically for Table ID, choose the variable "r_TableID" under "Runtime Inputs" from the left panel.

workiva

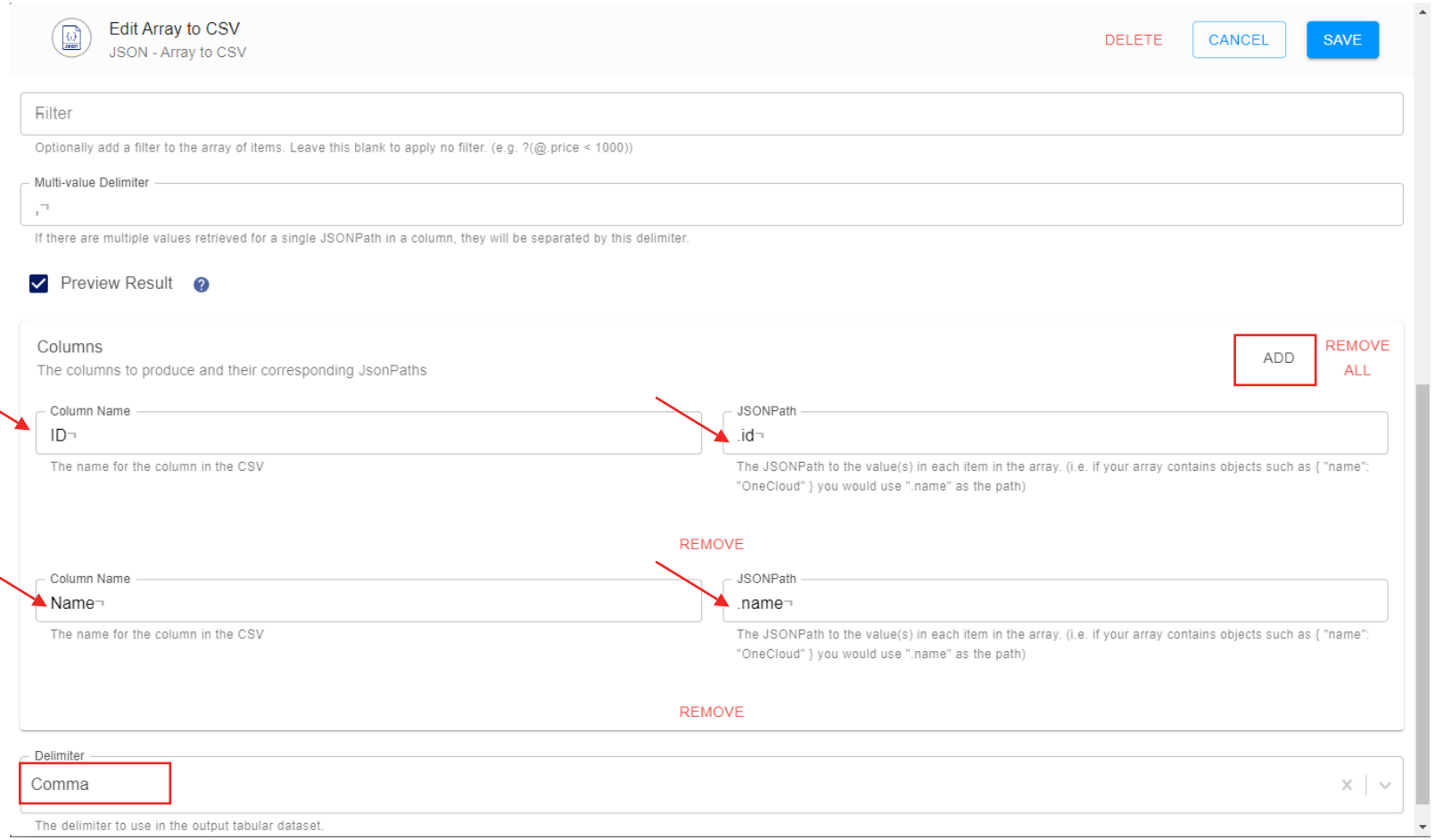
Step 4: Create Array to CSV node P1



Create a "JSON">"Array to CSV" node and link the node with the List Files node. Fill in the node information as shown above. Specifically for JSON Data, choose the variable "Files List" under "List Files" from the left panel.

Workiva

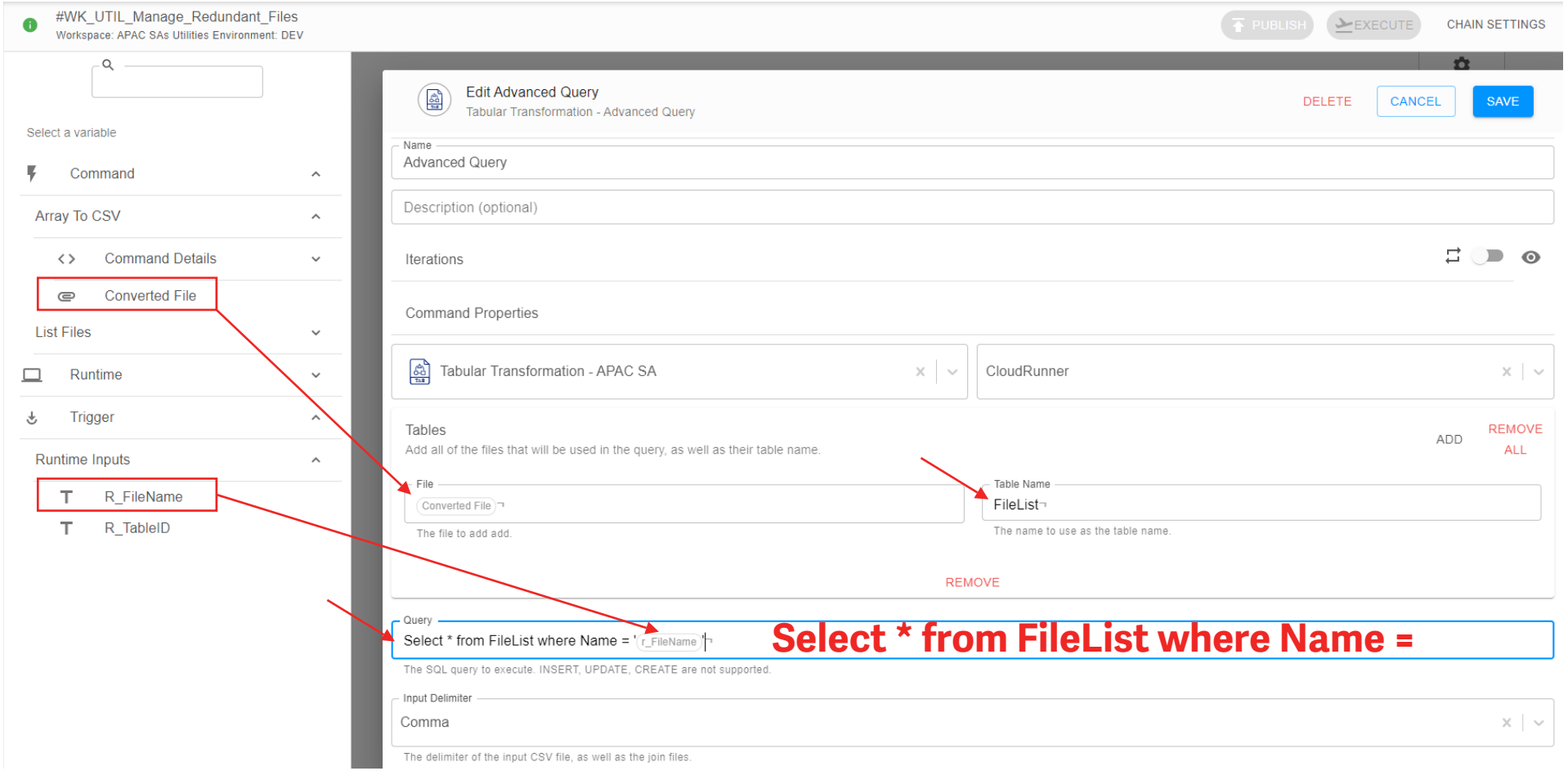
Step 4.1: Create Array to CSV node P2



Continue to fill in the node information as shown above. Specifically, add two sets of fields in the Columns section. Also, make sure the "Comma" is chosen as the Delimiter.

Workiva

Step 5: Create Advanced Query Node P1



Create a "Tabular Transformation">"Advanced Query" node and link the node with the Array to CSV node. Fill in the node information as shown above. Specifically for File, choose the variable "Converted File" under "Array to CSV" from the left panel. Also, replicate the query as shown above and use the variable "r_FileName" in the query.

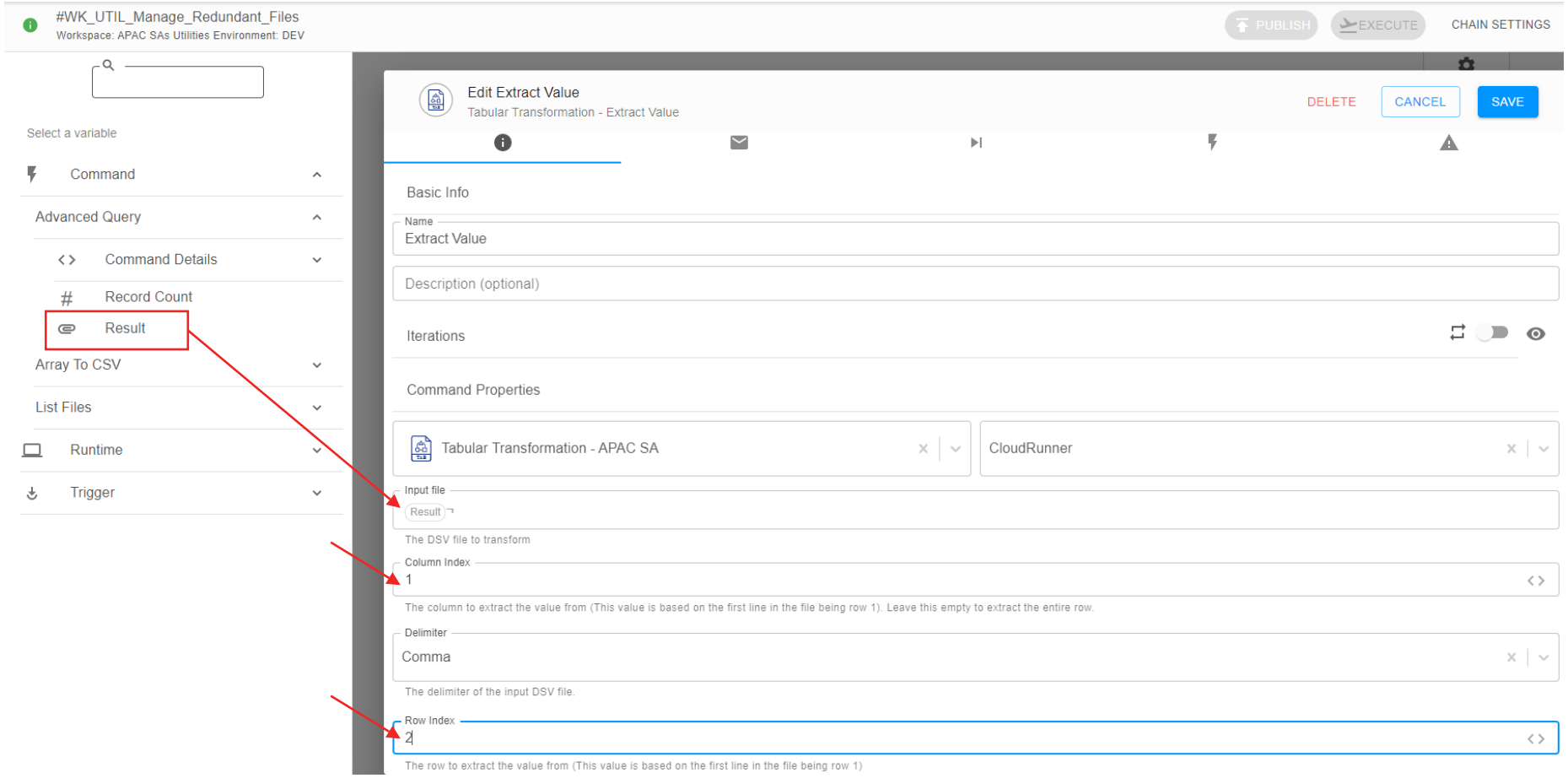
Workiva

Step 5.1: Create Advanced Query Node P2

Make sure that "Comma" is chosen for both the Input Delimiter and Output Delimiter. Check the box for preview results.

Workiva

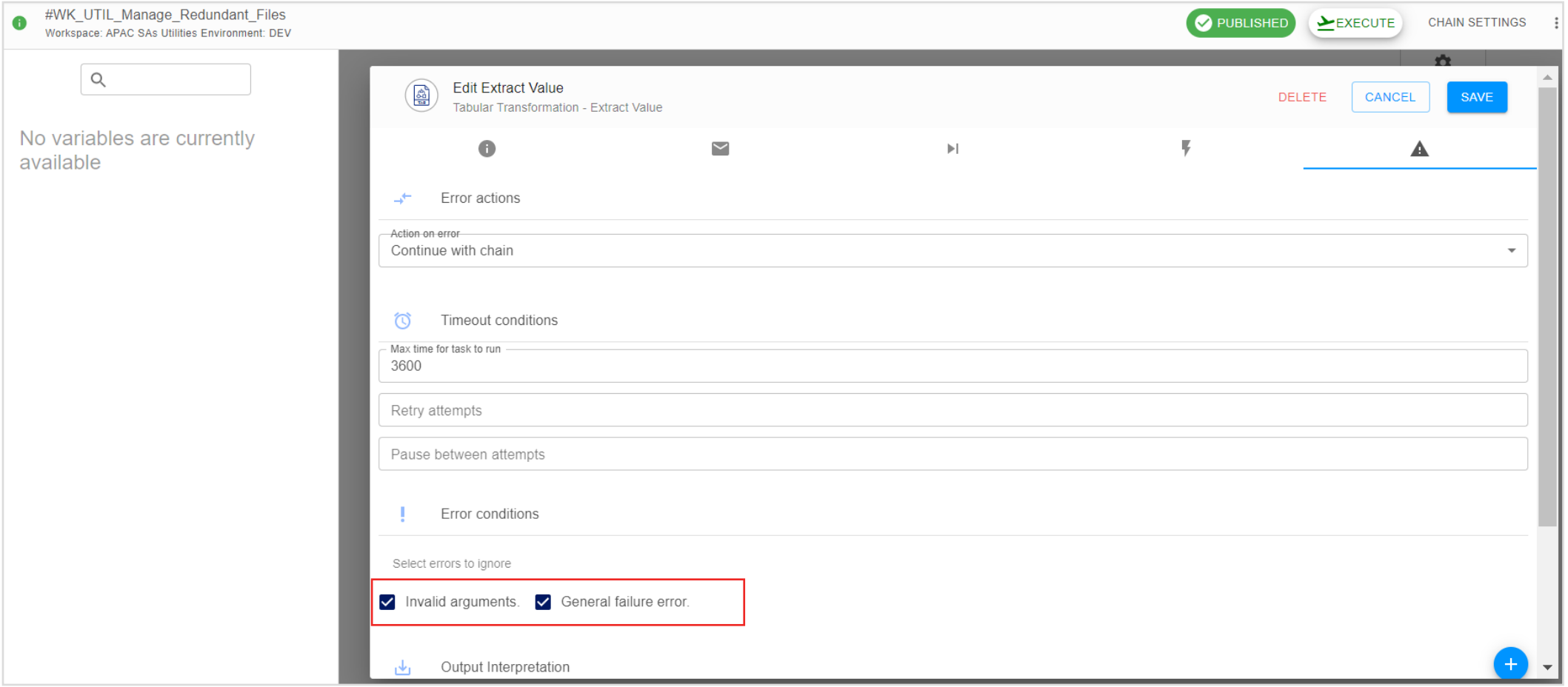
Step 6: Create Extract Value node



Create a "Tabular Transformation">"Extract Value" node and link the node with the Advanced Query node. Fill in the node information as shown above. Specifically for Input File, choose the variable "Result" under "Advanced Query" from the left panel. Make sure the Column Index is set as 1 and Row Index is set as 2.

Workiva

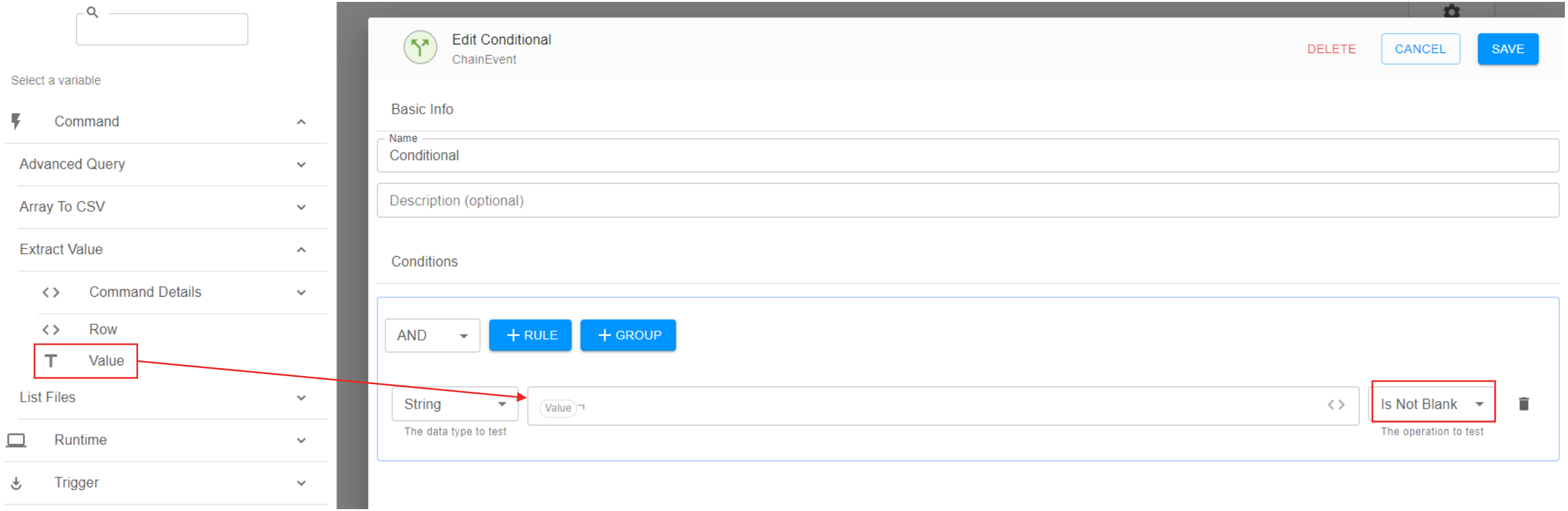
Step 6.1: Error Handling



For error handling purpose, in the "Extract Value" node, click on the 'Triangle' tab and tick both 'Invalid arguments' & 'General failure error'. This would ensure the chain runs regardless.

Workiva

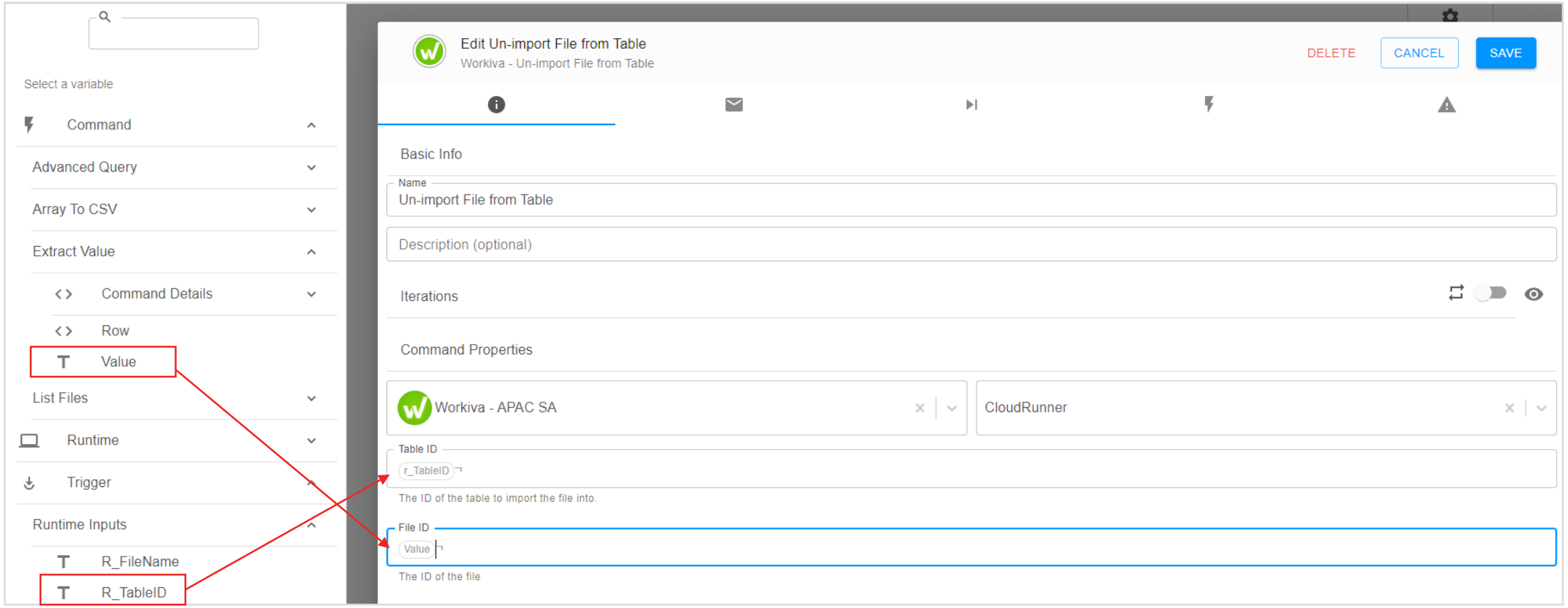
Step 7: Create Conditional node



Create a "Conditional" node and link the node with the Extract Value node. Fill in the node information as shown above. Specifically for String Value, choose the variable "Value" under "Extract Value" from the left panel. Make sure to choose "is not blank" in the operation to test.

workiva

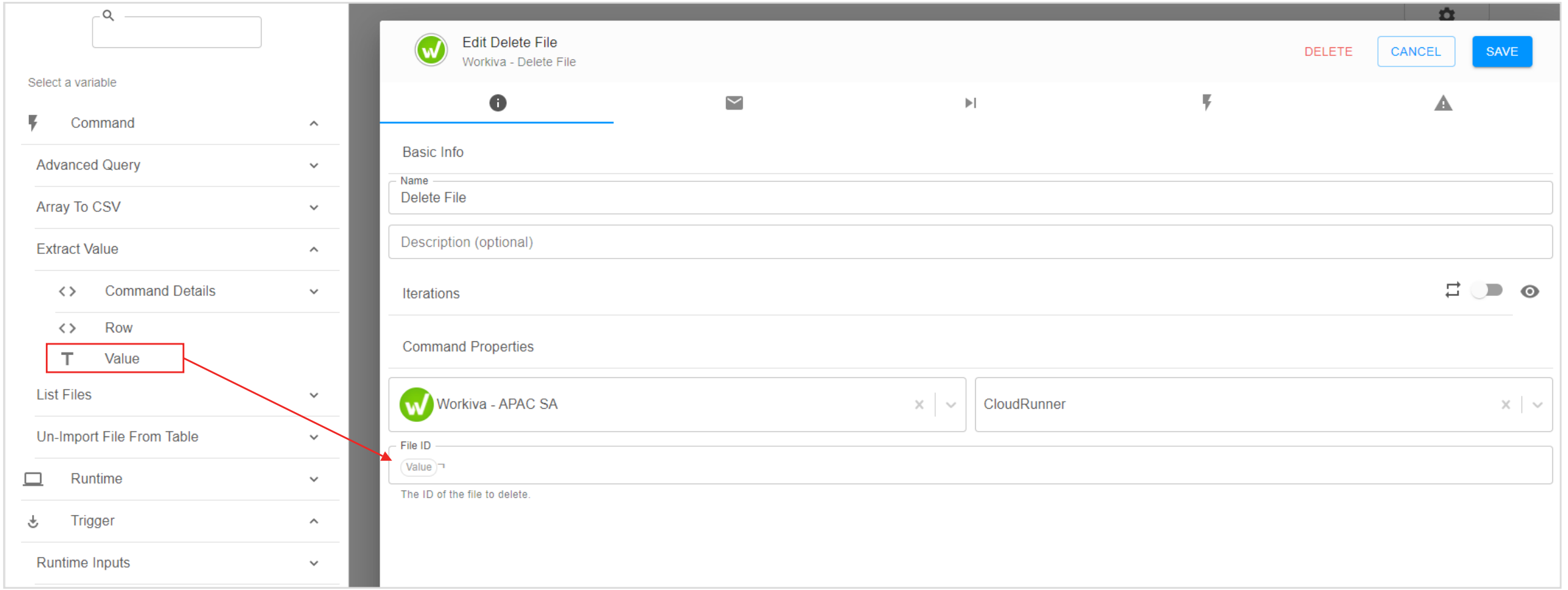
Step 8: Create Un-import File from Table node



Create a "Workiva">"Un-import File from Table" node and link the node with the Conditional node. Fill in the node information as shown above. Specifically for Table ID, choose the variable "r_TableID" under "Runtime Inputs". For File ID, choose the variable "Value" under "Extract Value" from the left panel.

workiva

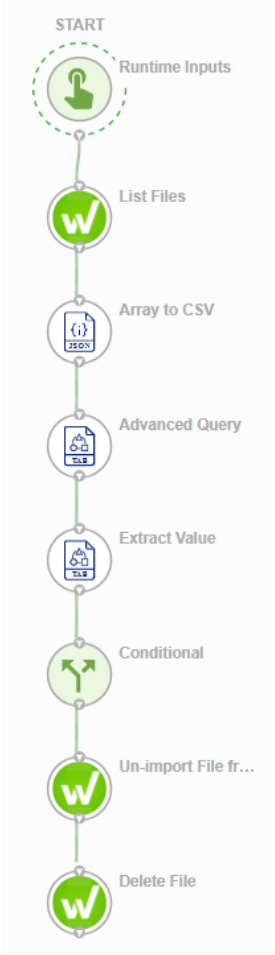
Step 9: Create Delete File node



Create a "Workiva">"Delte File" node and link the node with the Un-import File node. Fill in the node information as shown above. Specifically for Table ID, choose the variable "Value" under "Extract Value" on the left panel.

workiva

Step 10: Complete Chain 1 - Manage Redundant Files

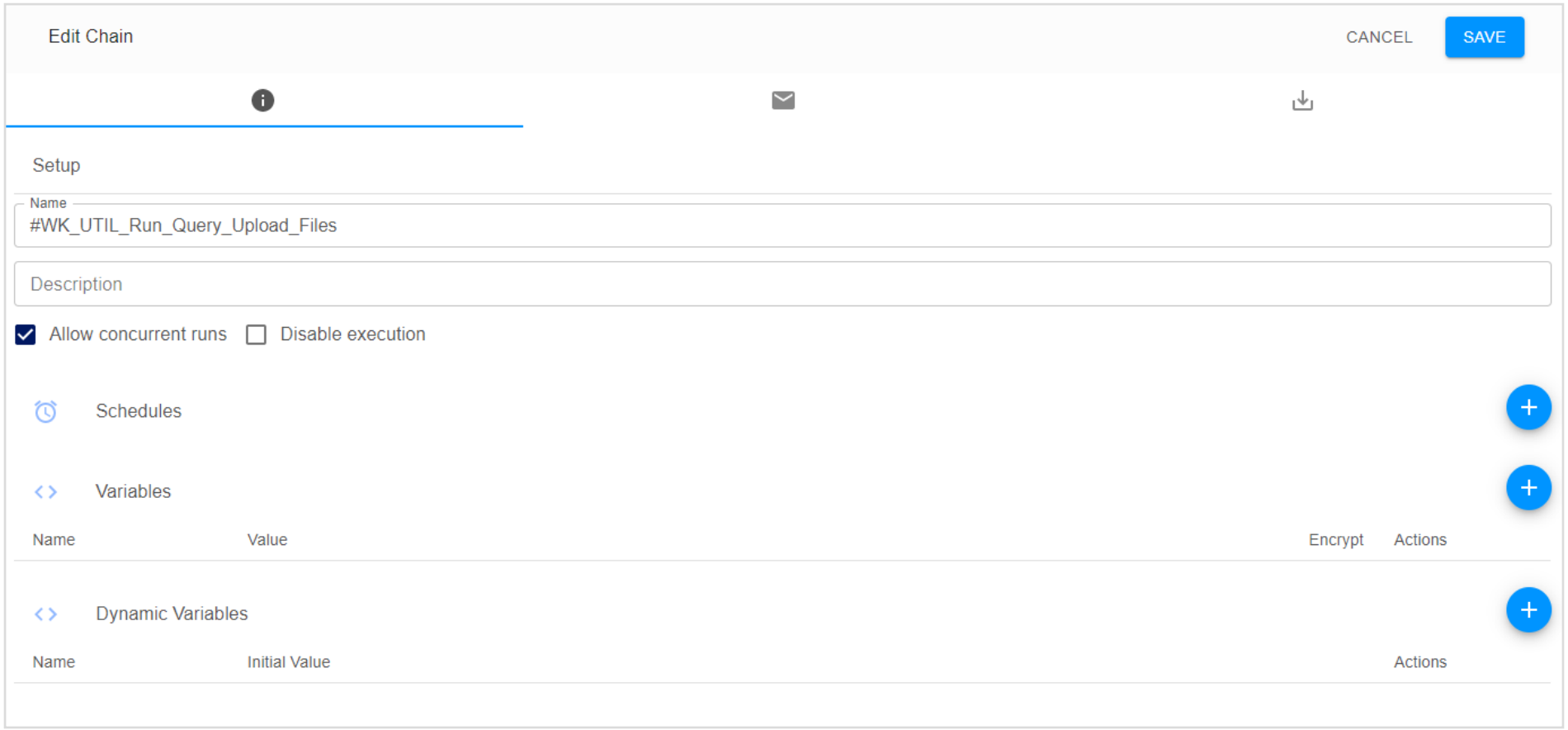


Double check the structure of the chain. Remember to hit Publish.

Workiva

Chain 2: Run Query & Upload Files (UTIL)

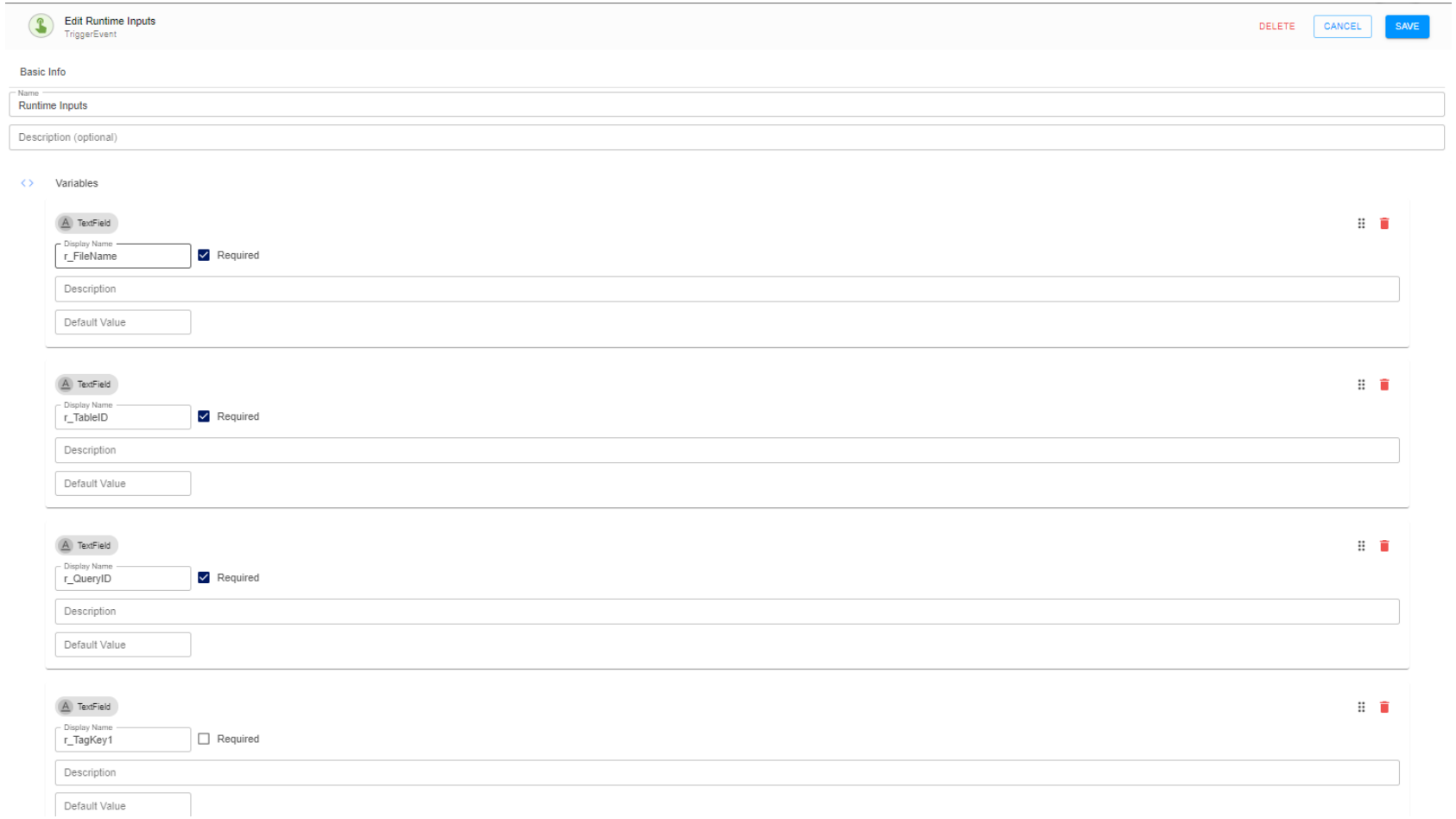
Step 1: Create New Chain



Create a new chain "#WK_UTIL_Run_Query_Upload_Files" as shown above and hit save.

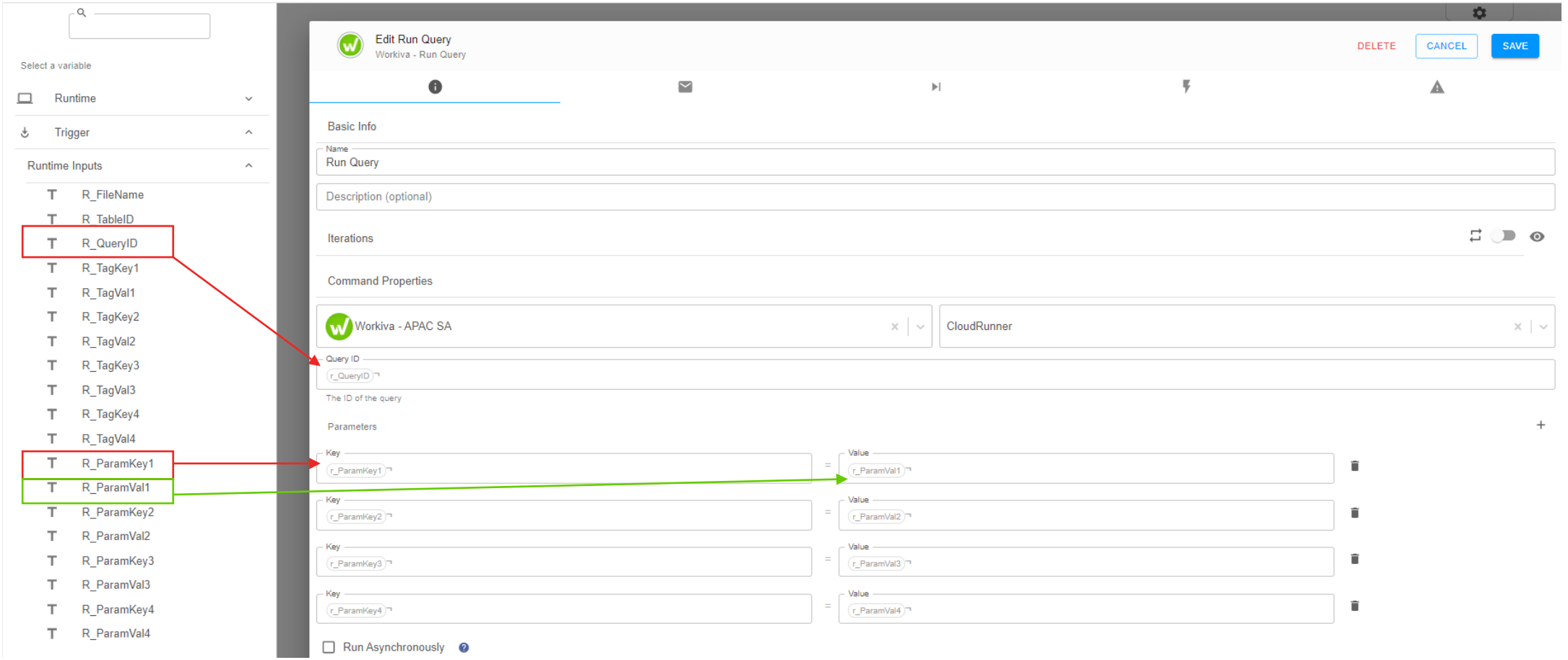
workiva

Step 2: Create Runtime Inputs Node



Create a "Runtime Inputs" node and drag the node in the "Start" circle. Fill in the node information as shown above. Specifically, create 19 Text Field variables: "r_FileName", "r_TableID", "r_QueryID", "r_TagKey1", "r_TagVal1", "r_TagKey2", "r_TagVal2", "r_TagKey3", "r_TagVal3", "r_TagKey4", "r_TagVal4", "r_ParamKey1", "r_ParamVal1", "r_ParamKey2", "r_ParamVal2", "r_ParamKey3", "r_ParamVal3", "r_ParamKey4" and "r_ParamVal4". Check "Required" for the first three variables.

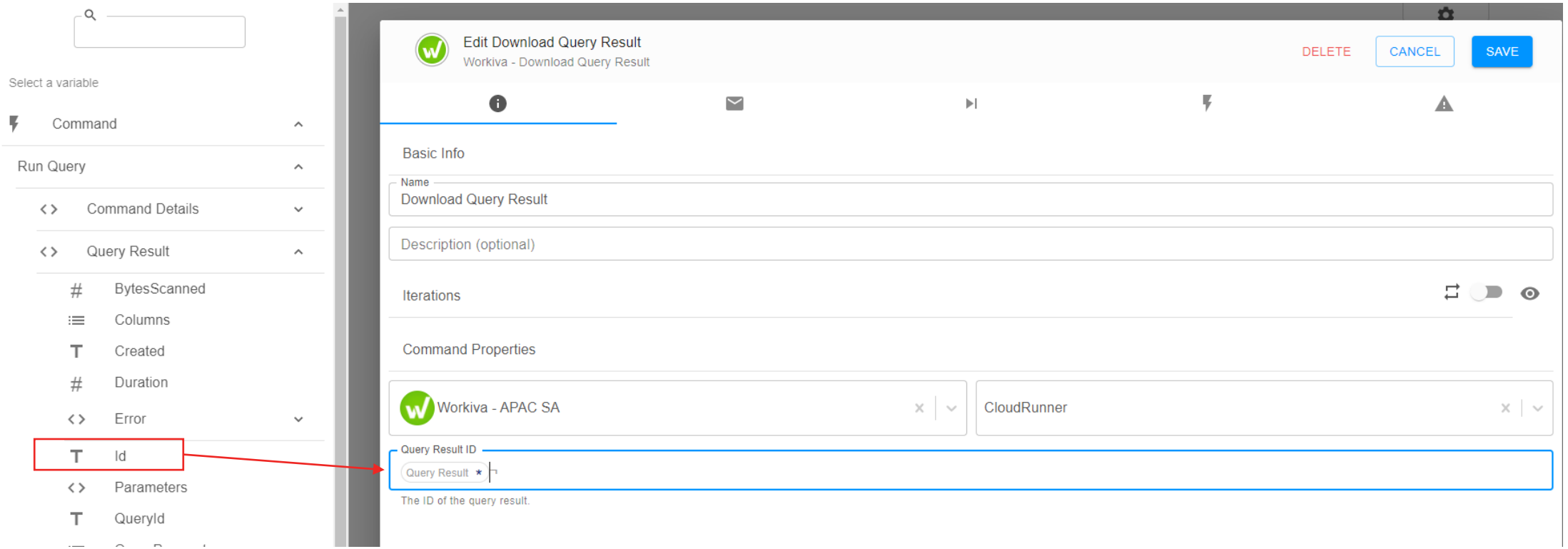
Step 3: Create Run Query Node



Create a "Workiva">"Run Query" node and link the node with the Runtime Inputs node. Fill in the node information as shown above. Specifically, create 4 sets of parameters and select the corresponding Parameter Key and Parameter Value as shown above.

Workiva

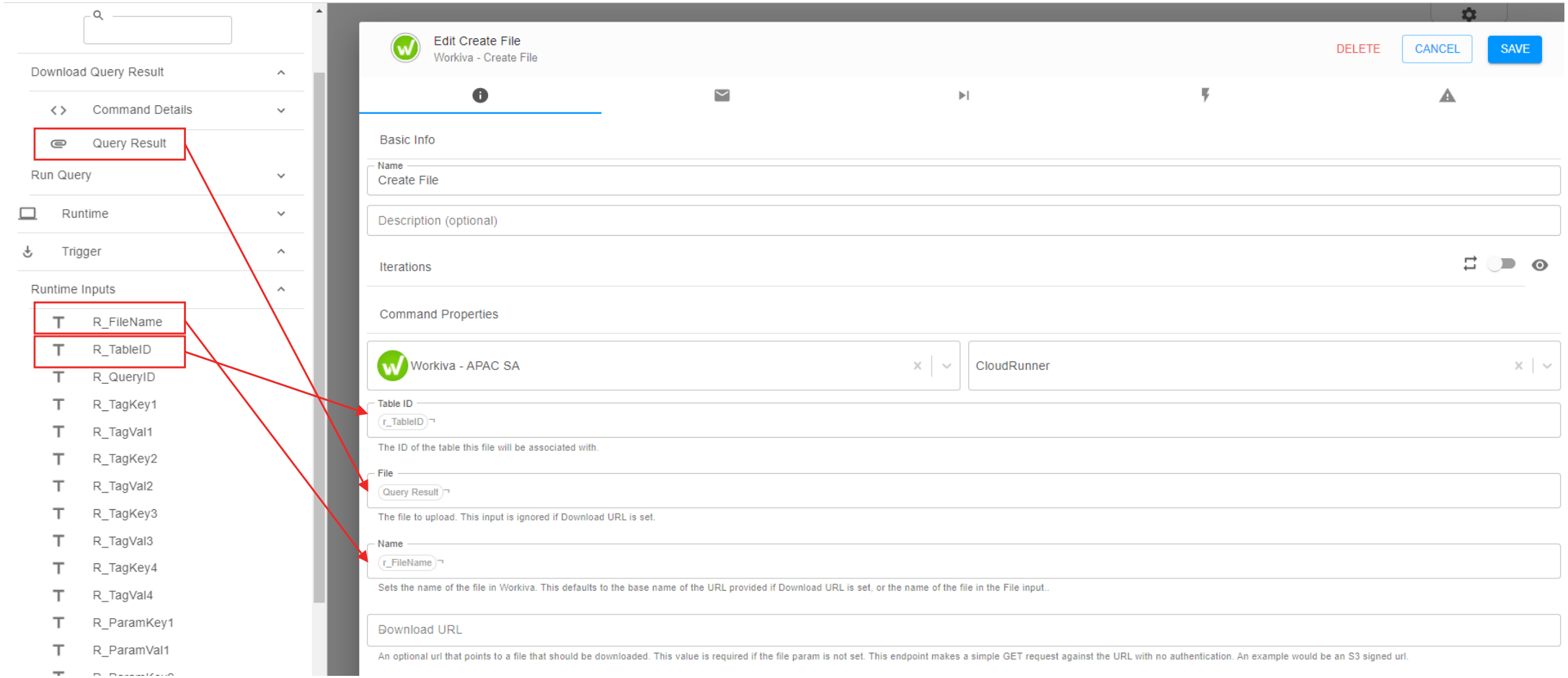
Step 4: Create Download Query Result Node



Create a "Workiva">"Download Query Result" node and link the node with the Run Query node. Fill in the node information. Specifically, for Query Result ID choose the variable "Id" under "Run Query">"Query Result" from the left panel.

workiva

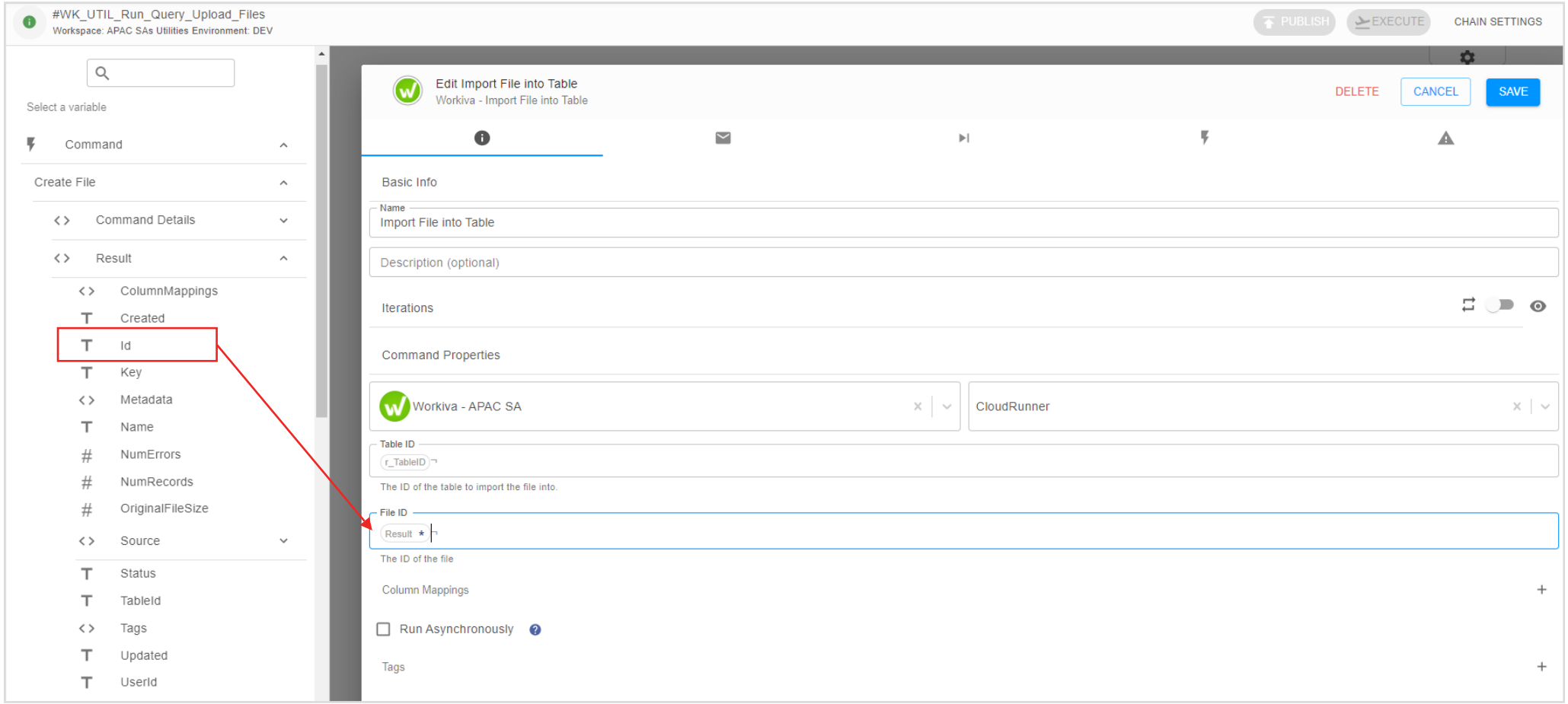
Step 5: Create Create File Node



Create a "Workiva">"Create File" node and link the node with the Download Query Result node. Fill in the node information as shown above.

workiva

Step 6: Create Import File into Table Node P1

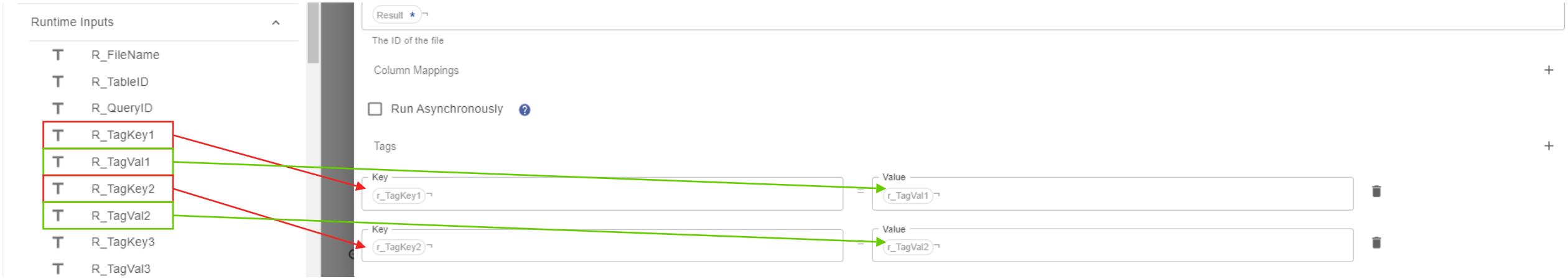


Create a "Workiva">"Create File" node and link the node with the Download Query Result node. Fill in the node information as shown above. Specifically for File ID, choose the variable "Id" under "Create File">"Result" from the left panel. For Table ID, choose the variable "r_TableID" under "Trigger">"Runtime Inputs".

Workiva

Step 6.1: Create Import File into Table Node P2

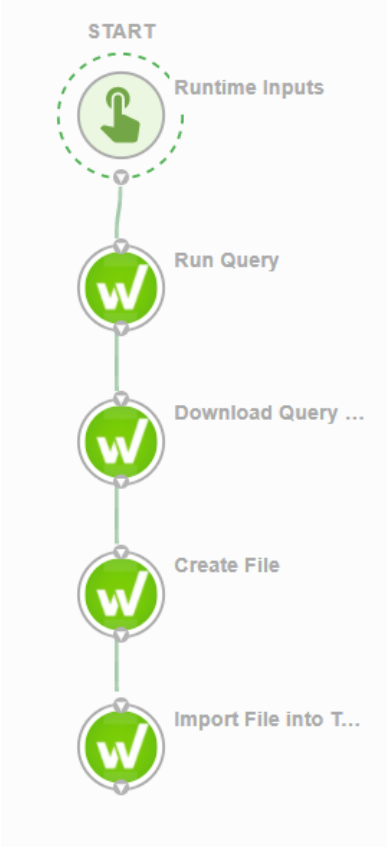
Stay in the previous command



Add two tags in the bottom and fill in the tag information as shown above. In here, we are adding tags "Data_Year" and "Data_Month".

workiva

Step 7: Complete Chain 2 - Run Query & Upload Files

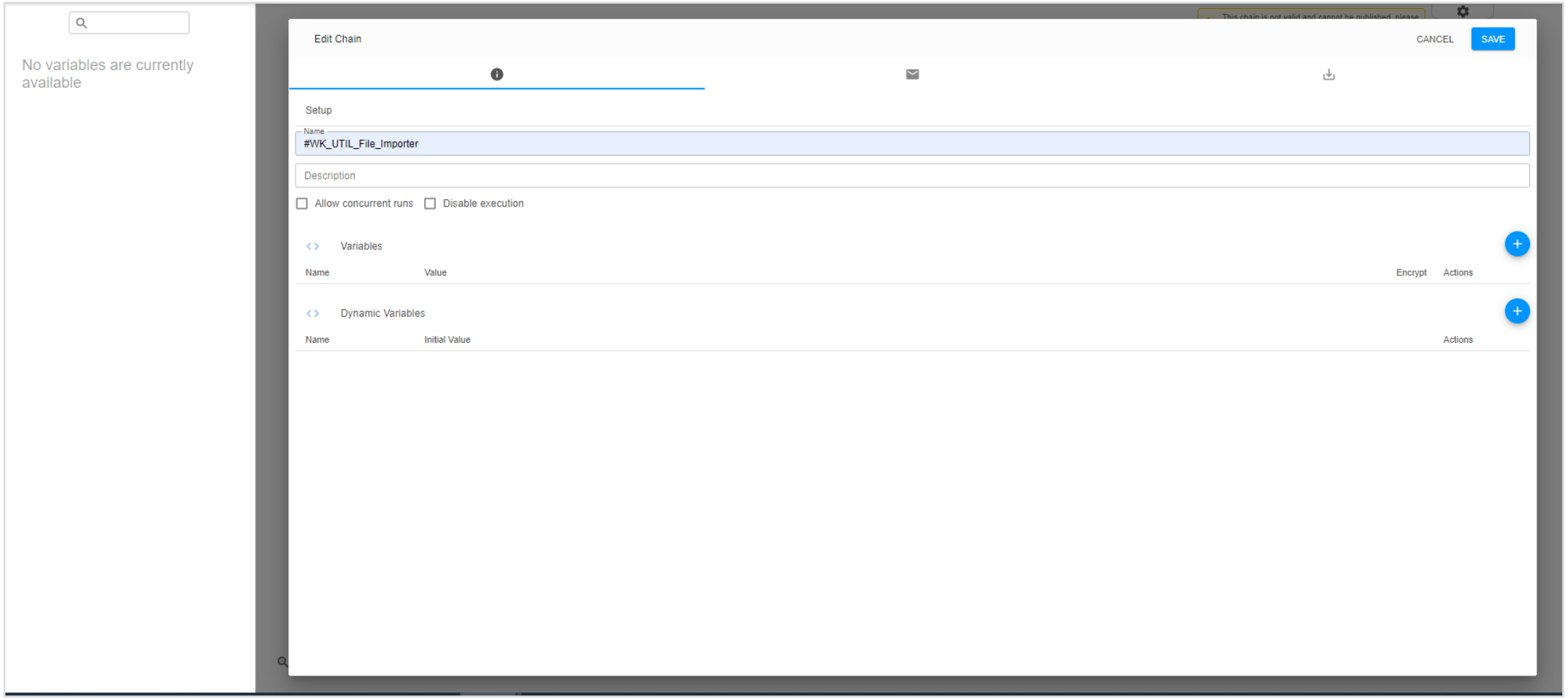


Double check the structure of the chain. Remember to hit Publish.

Workiva

Chain 3: File Importer (UTIL)

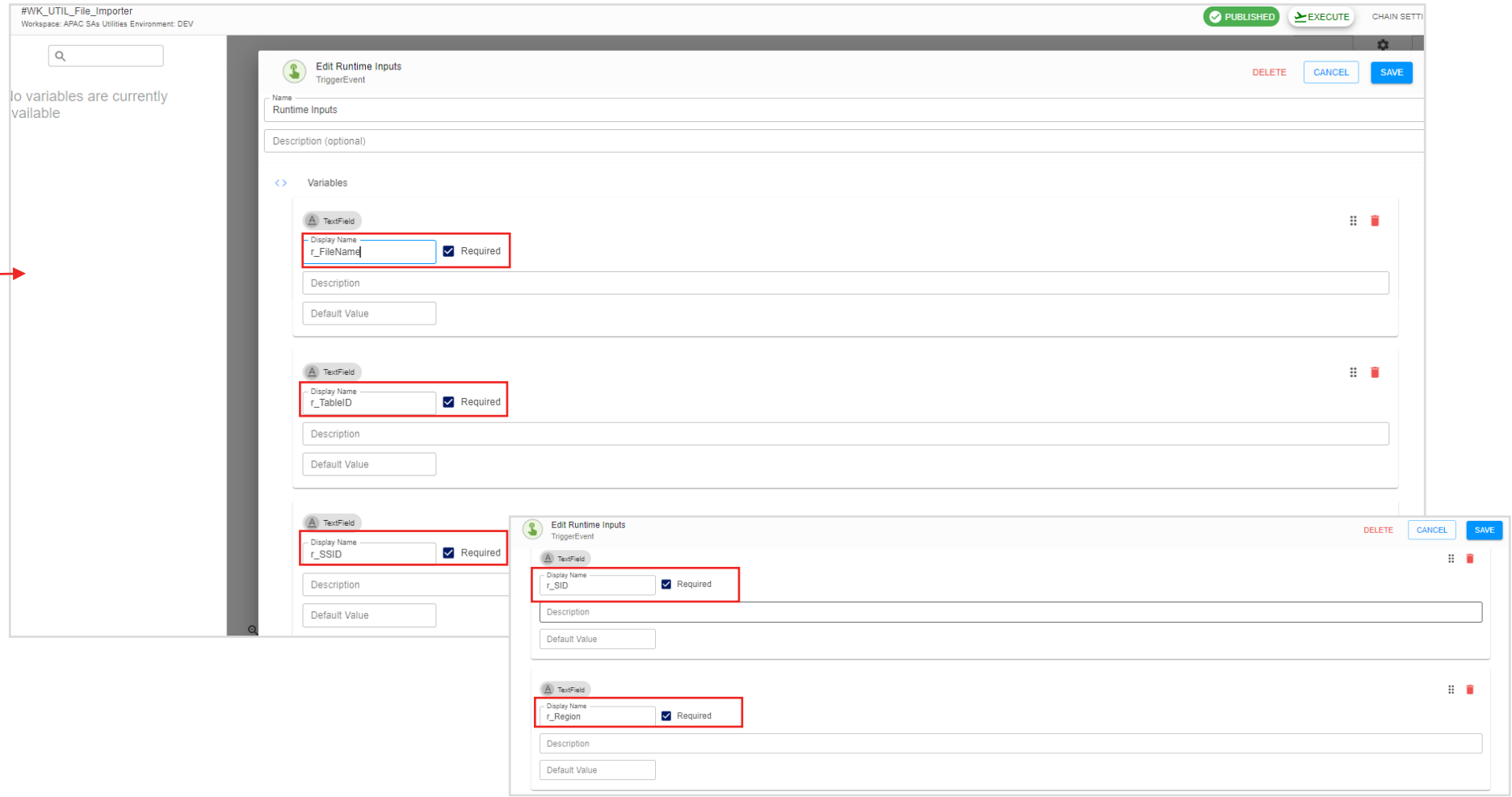
Step 1: Create New Chain



Create a new chain "#WK_UTIL_File_Importer" as shown above and hit save.

workiva

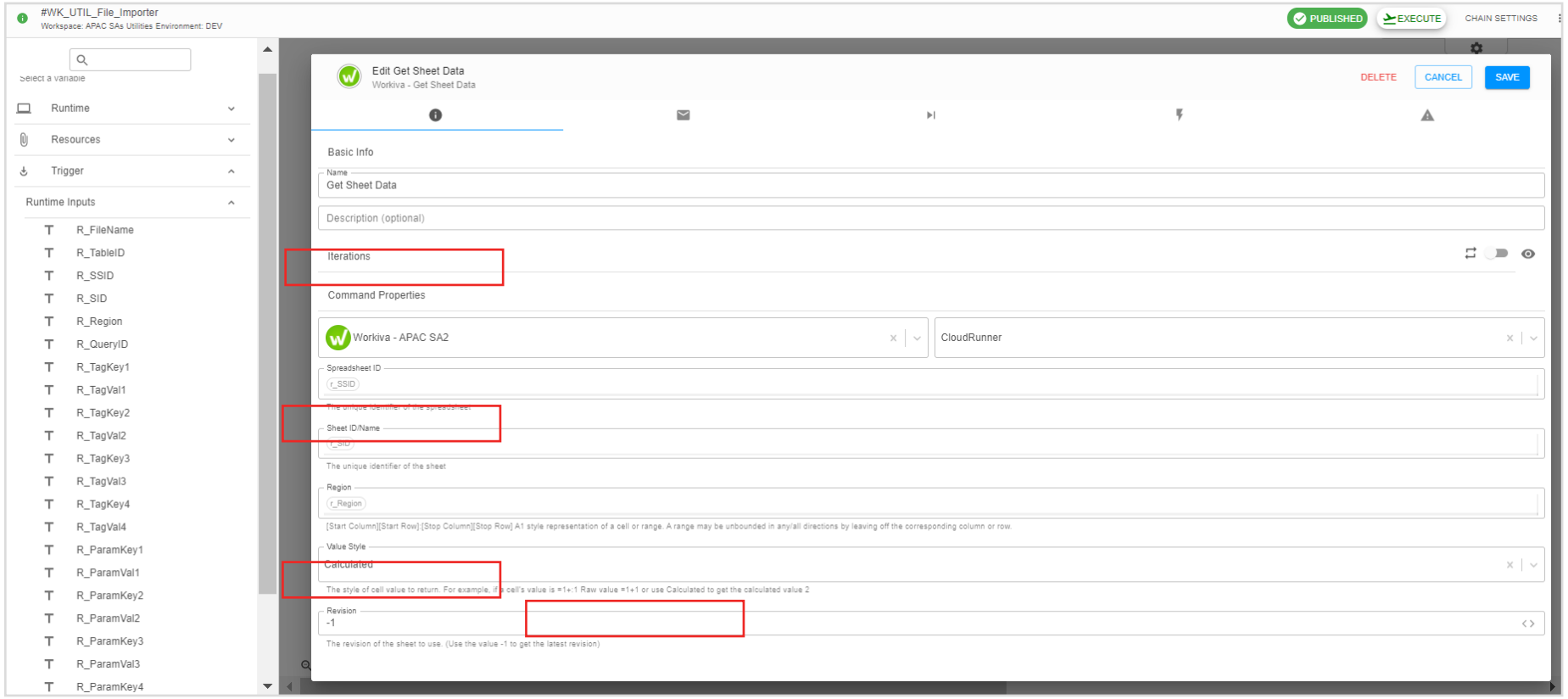
Step 2: Runtime Inputs



Input a 'Runtime Inputs' node and indicate 5 fields as "Required". The five fields are: 'r_FileName, r_TableID, r_SSID, r_SID, r_Region'

Workiva

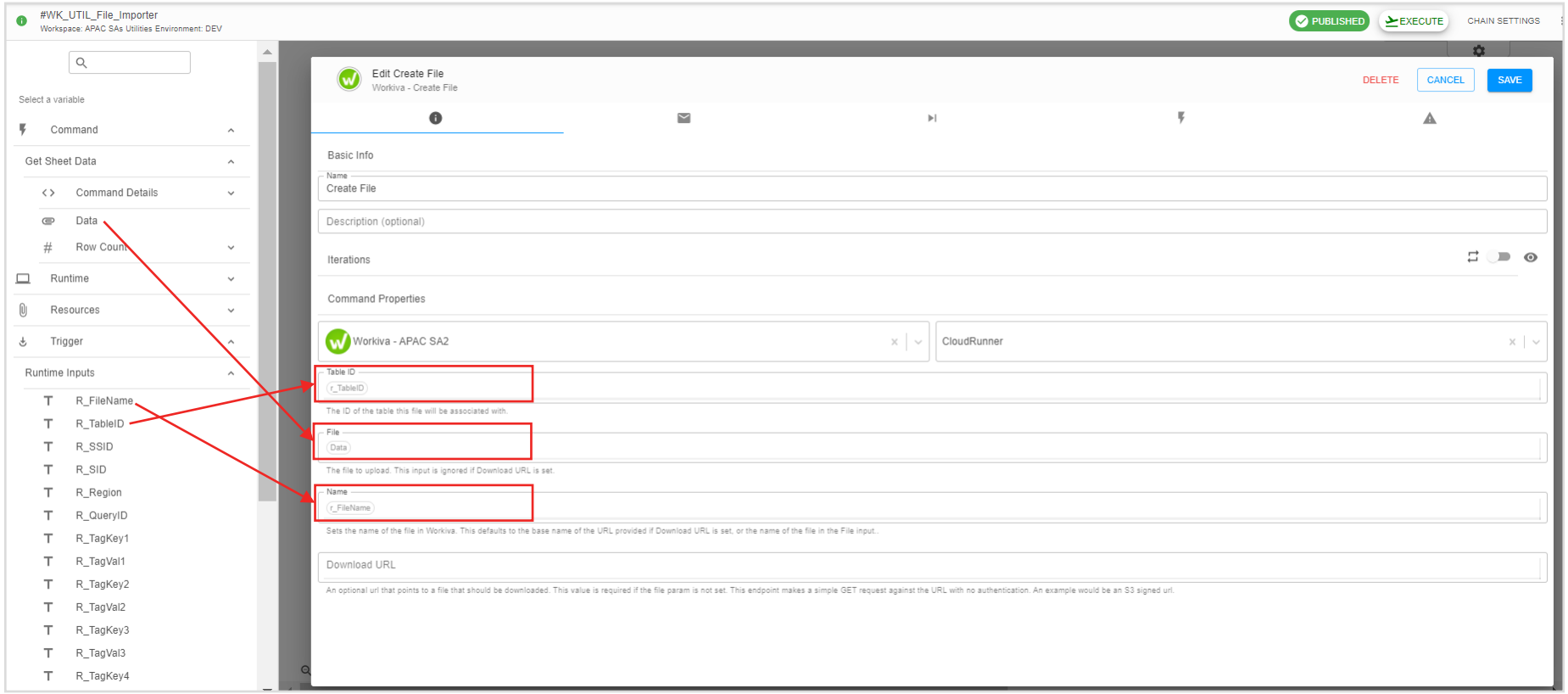
Step 3: Get Sheet Data



Input a 'Runtime Inputs' node and indicate 5 fields as "Required". The five fields are: 'r_FileName, r_TableID, r_SSID, r_SID, r_Region'

Workiva

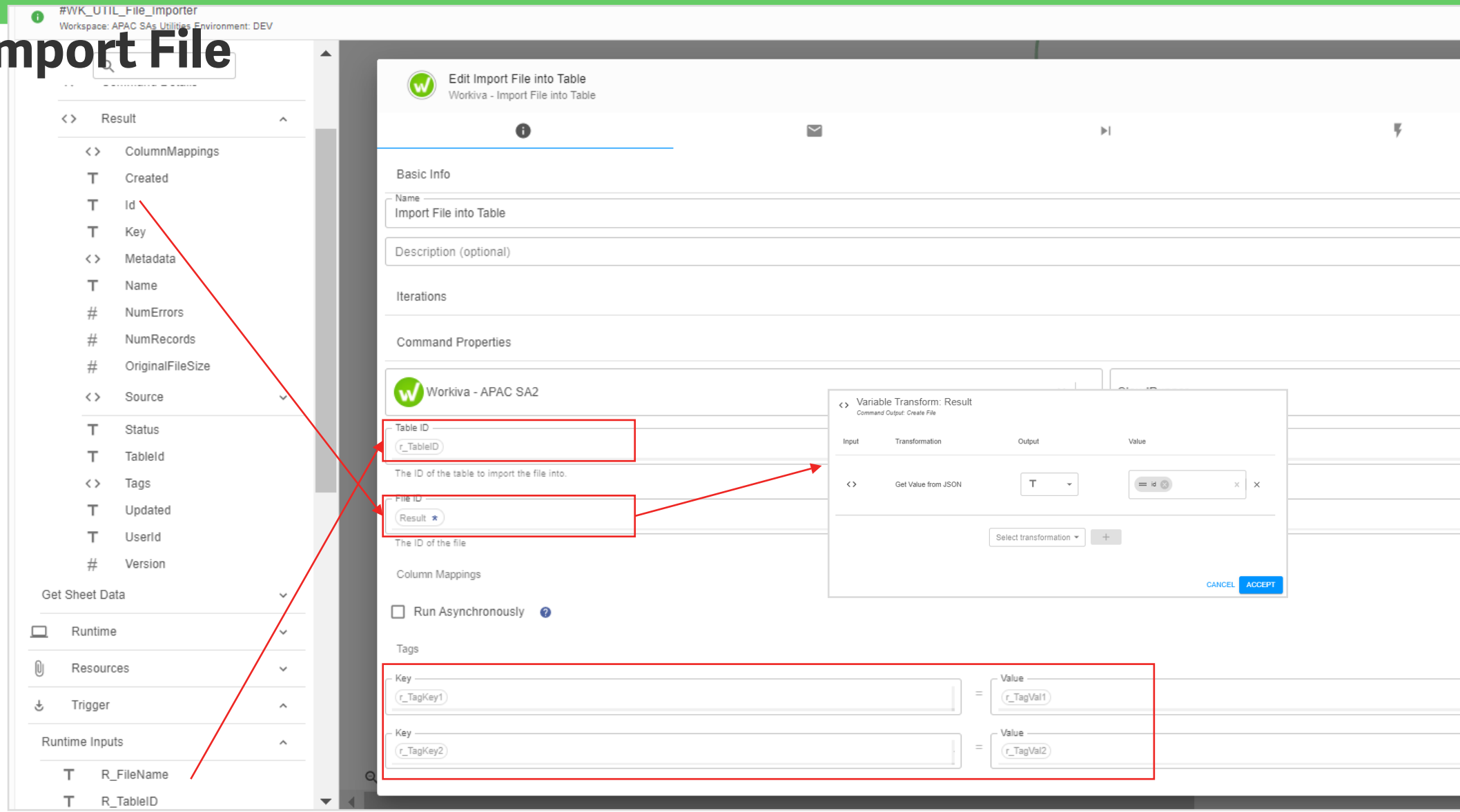
Step 4: Create File



Connect a 'Create File' node and input the following input variables from the previous 'Get Sheet Data' & 'Runtime Inputs'

Workiva

Step 5: Import File

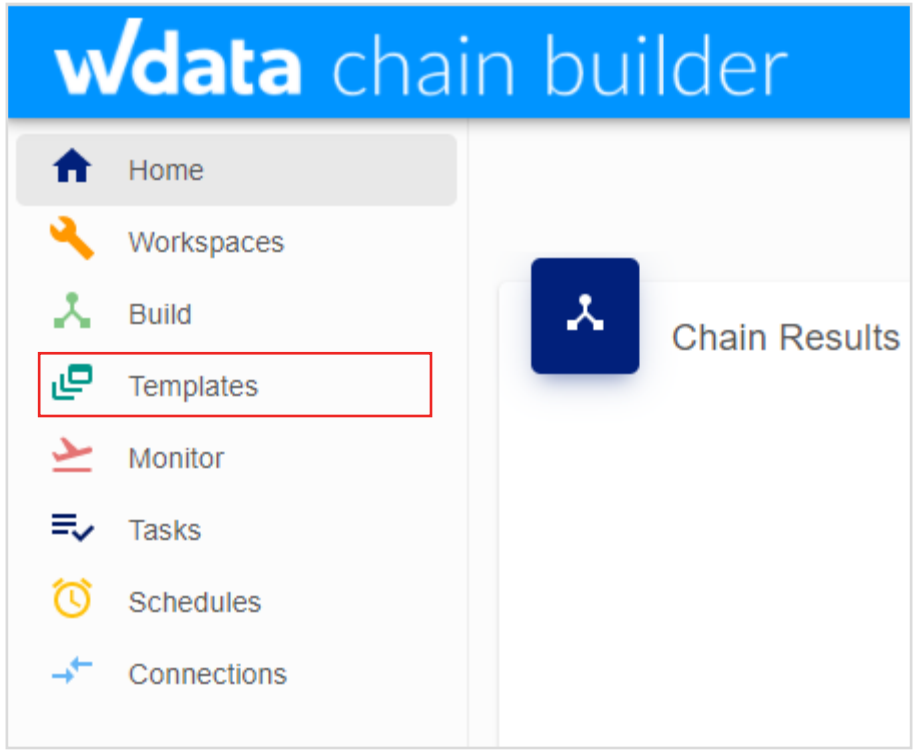


Connect a 'Import File' node and input the following input variables from the previous 'Create File' & 'Runtime Inputs'

Workiva

Chain 4: Creating Template Chain

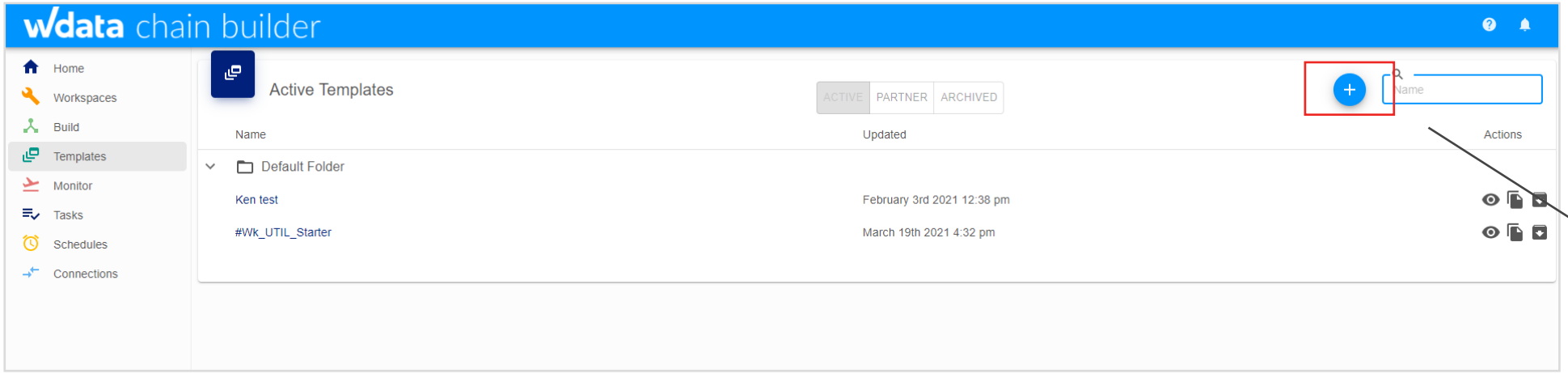
Step 1: Go to Templates

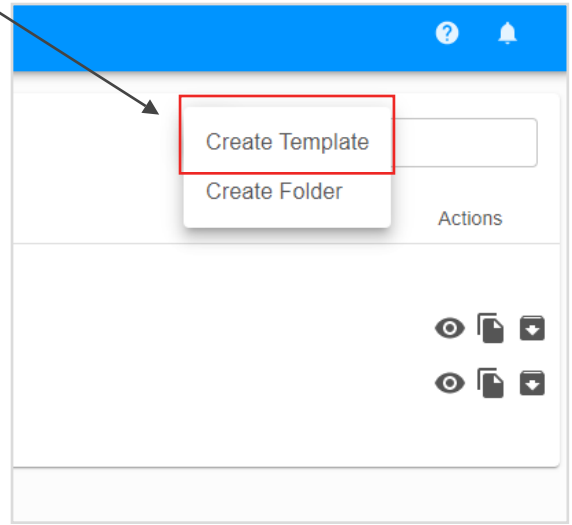


Open the Chain Builder and click "Templates" on the left panel.

Workiva

Step 2: Create Template

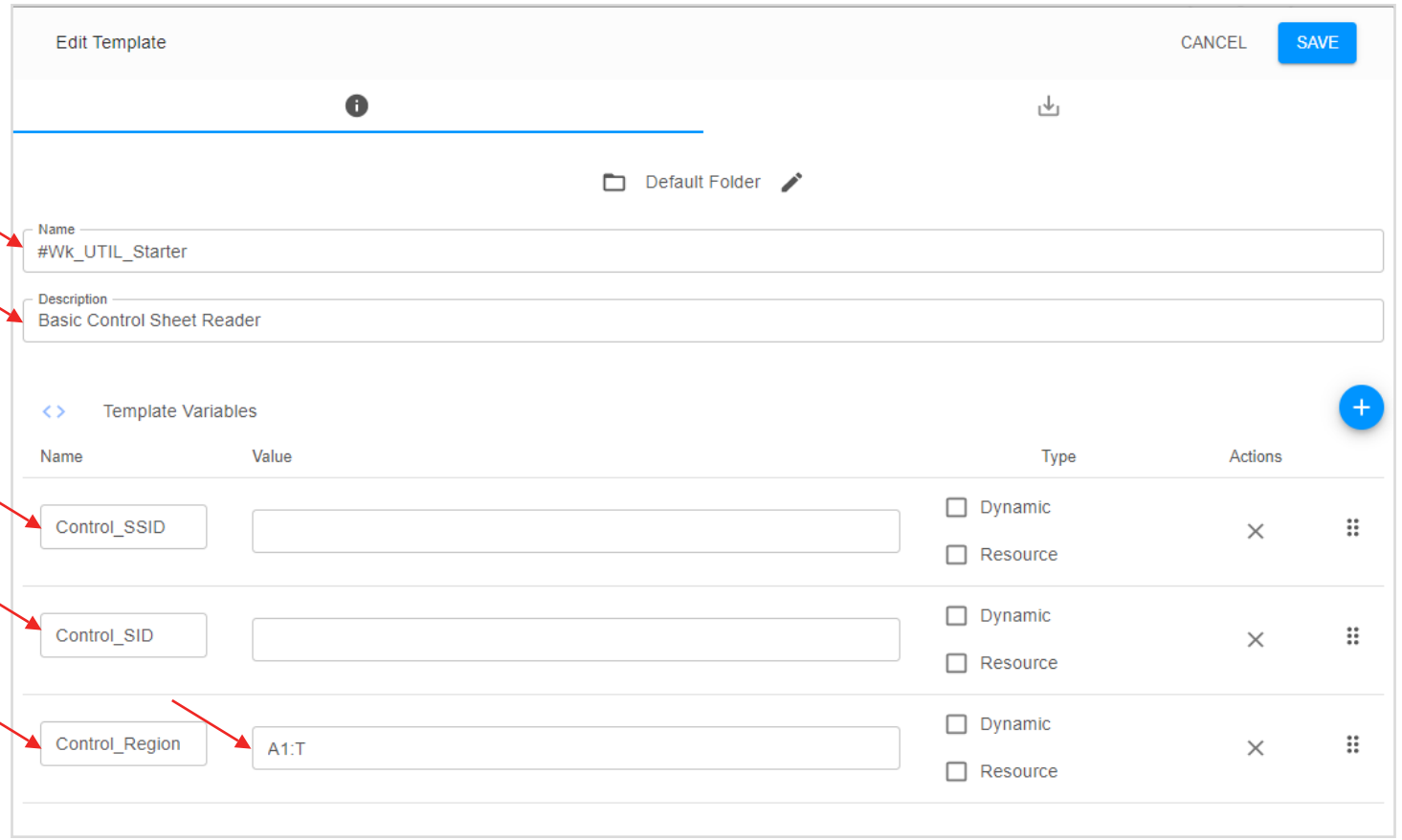




In the Templates homepage, click the blue "+" button on the right and click "Create Template".

Workiva

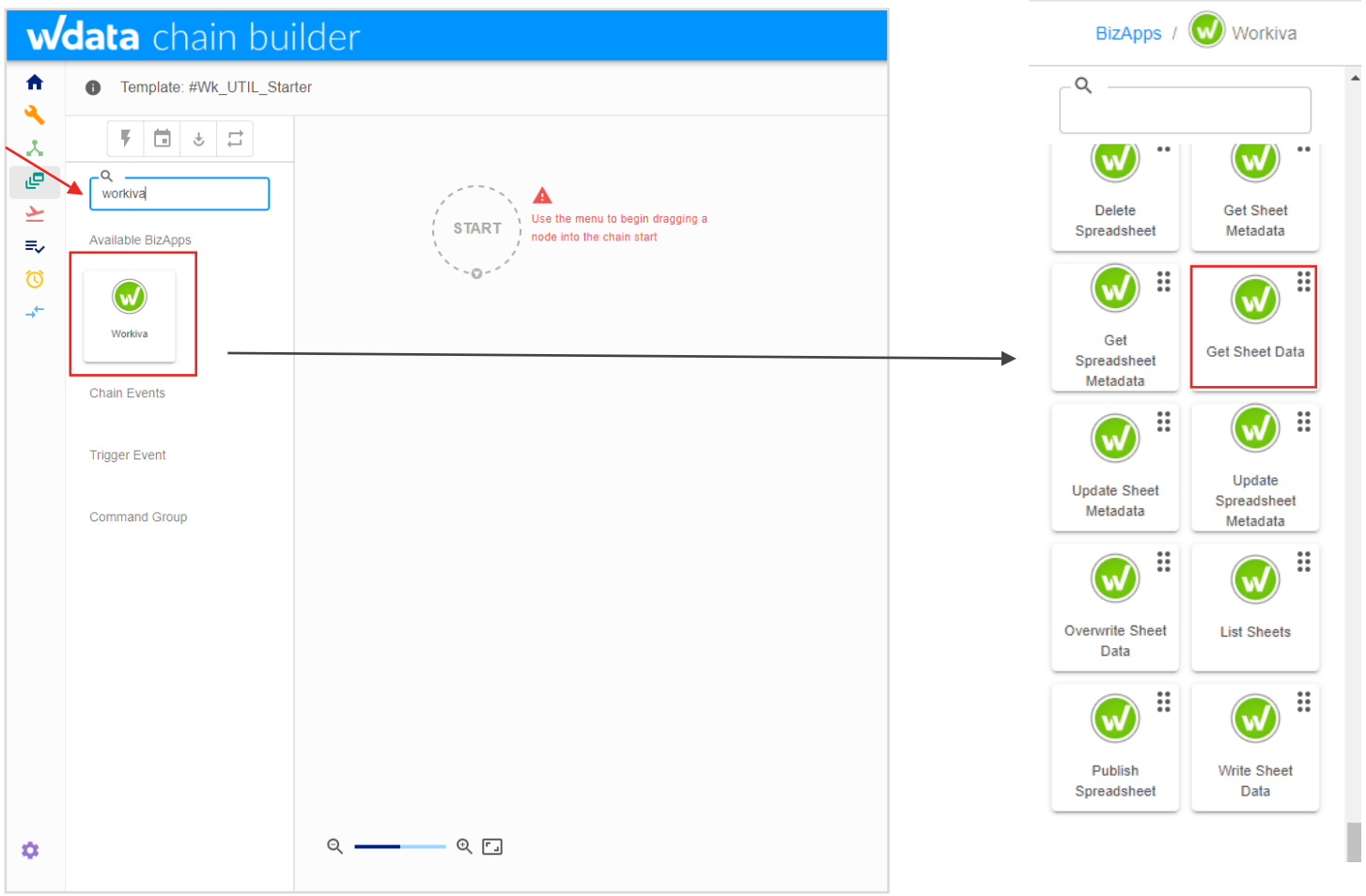
Step 3: Edit Template Settings



Edit the template settings by filling in the above information in the fields as indicated by the red arrows.

workiva

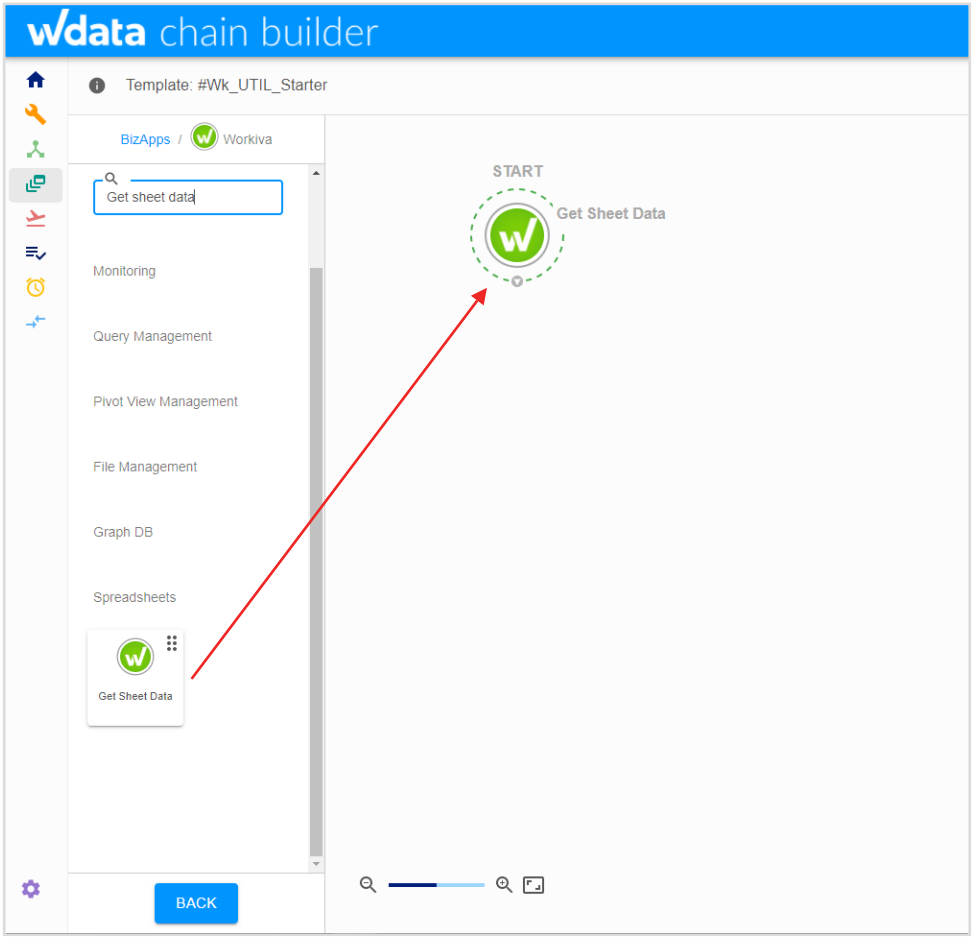
Step 4: Create the "Get Sheet Data" Node P1



In the blank template, go to the BizApp search bar and search for "Workiva". Click into Workiva icon and find the node "Get Sheet Data".

workiva

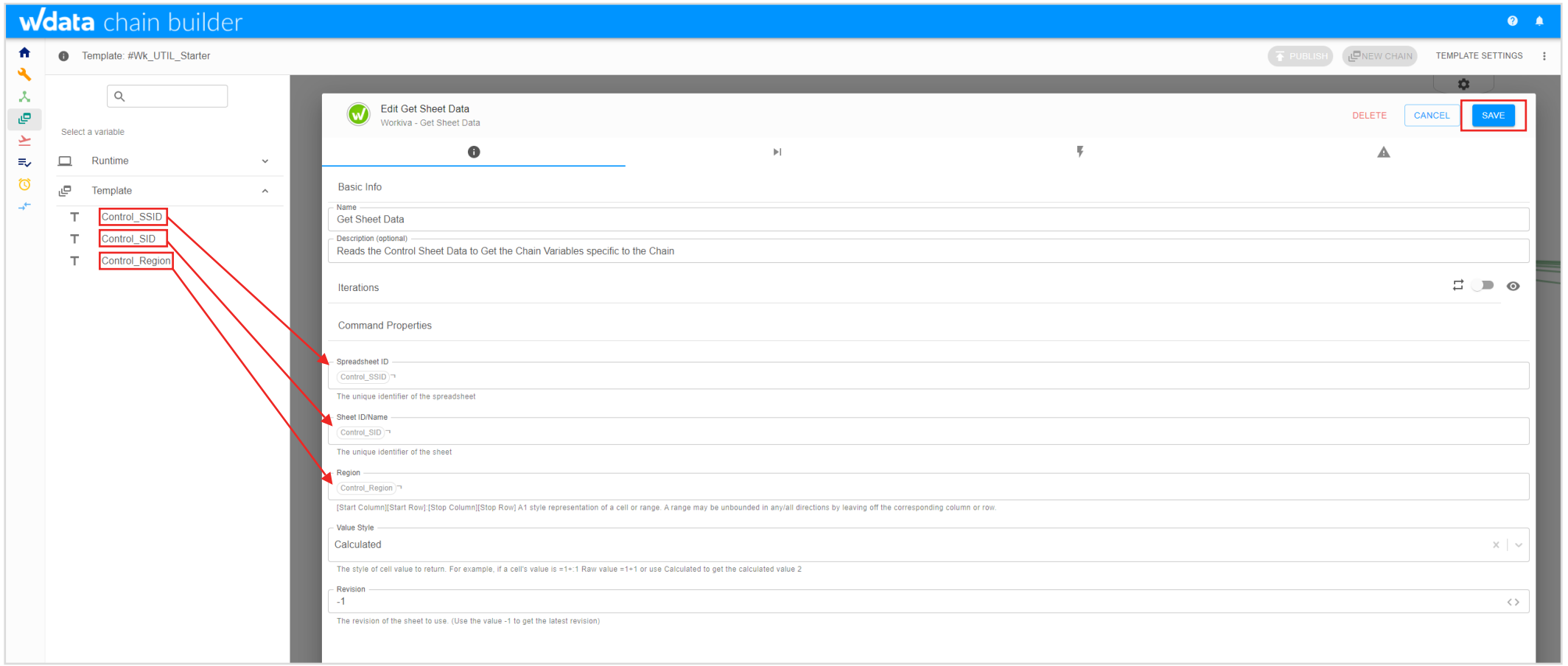
Step 4.1: Create the "Get Sheet Data" Node P2



Drag the node from the BizApp panel to the "Start" circle and double click on the node to edit.

Workiva

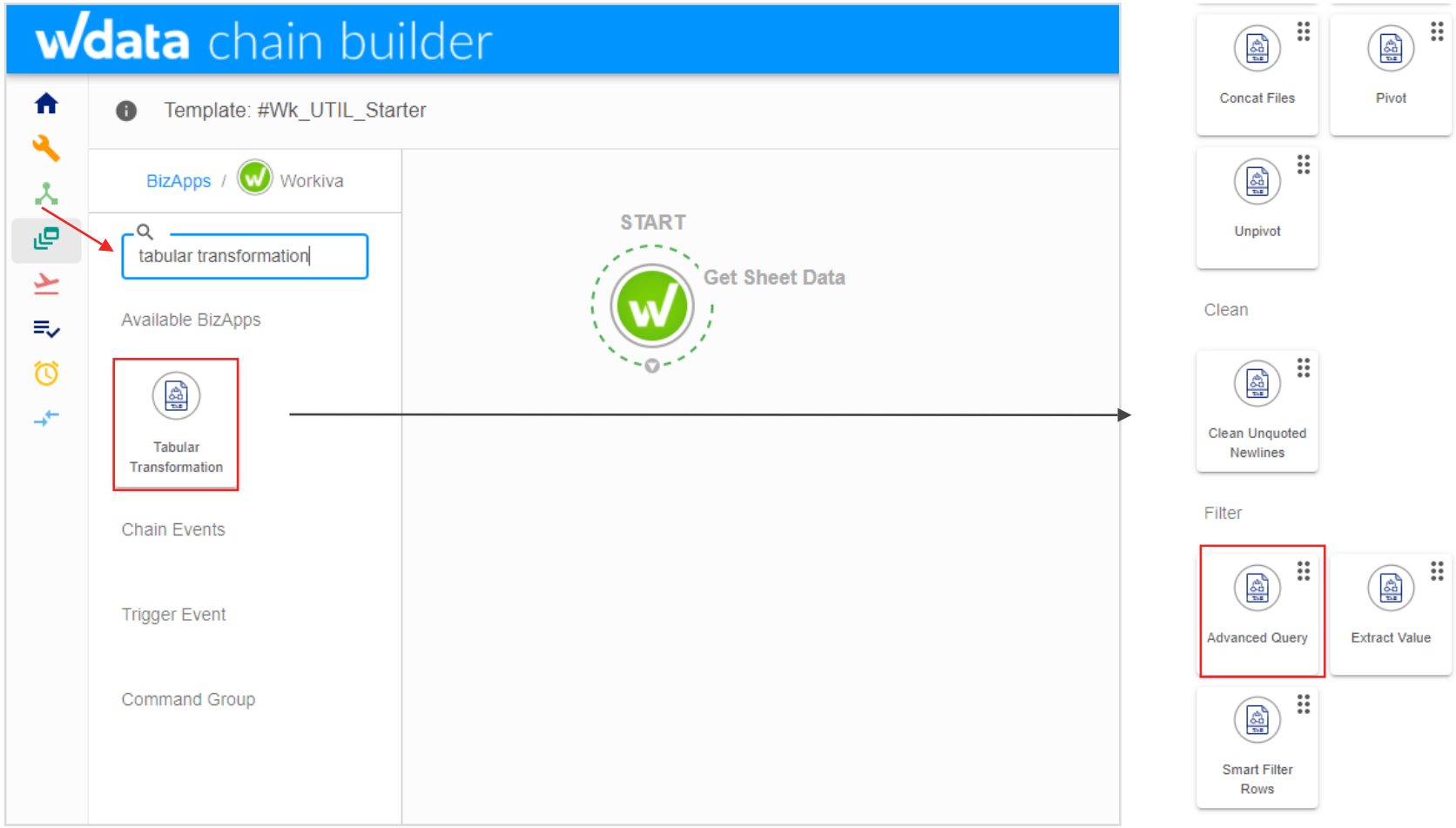
Step 4.2: Edit "Get Sheet Data" Node



Input the node information as shown above. For the Spreadsheet ID, Sheet ID/Name and Region fields, go to the left panel and select the corresponding variables under 'Template'. Then, click the "Save" button on the upper right corner. The first node is now created.

Workiva

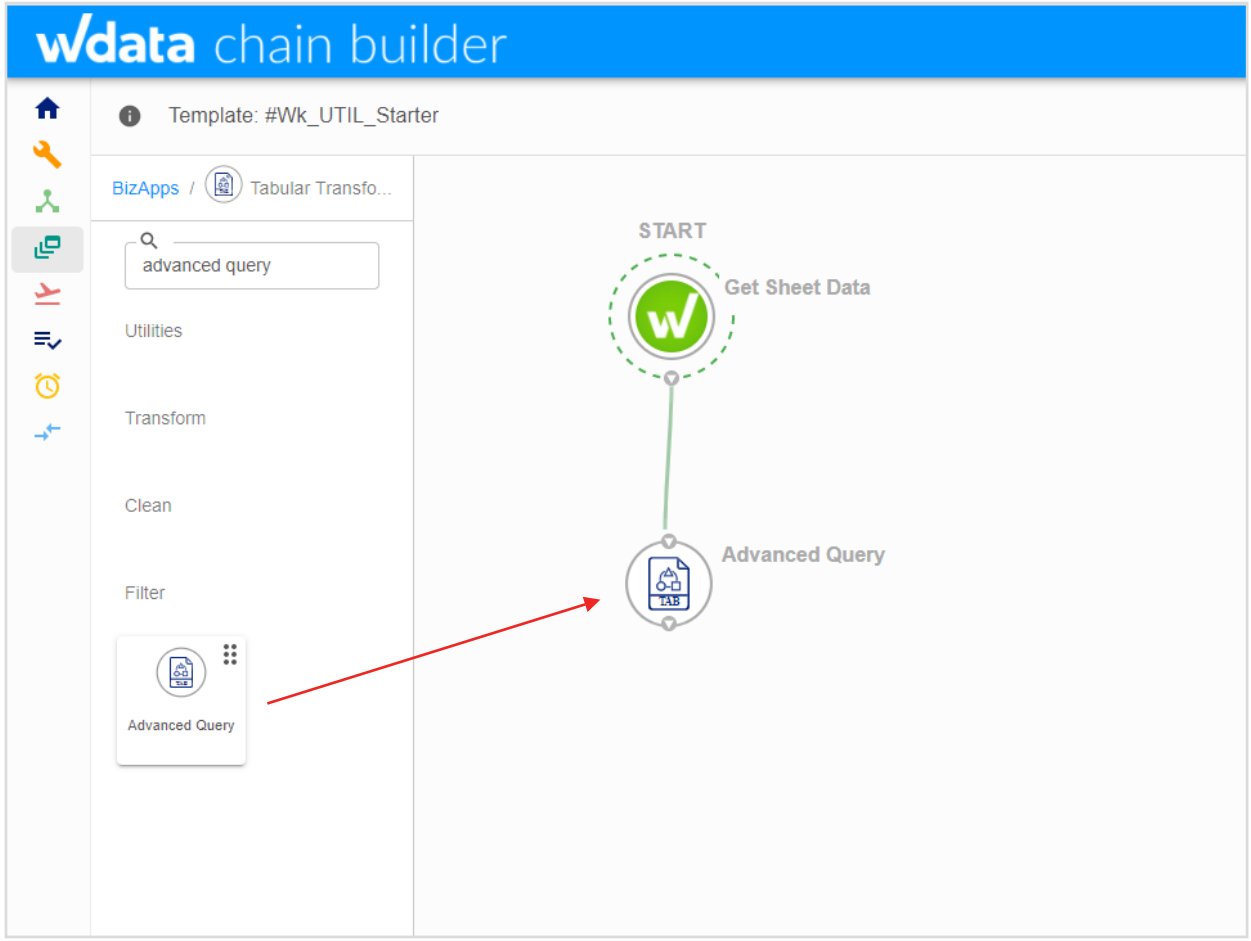
Step 5: Create the "Advanced Query" Node P1



Create the "Advanced Query" node by searching "Tabular Transformation" > "Advanced Query" in the BizApp search bar.

Workiva

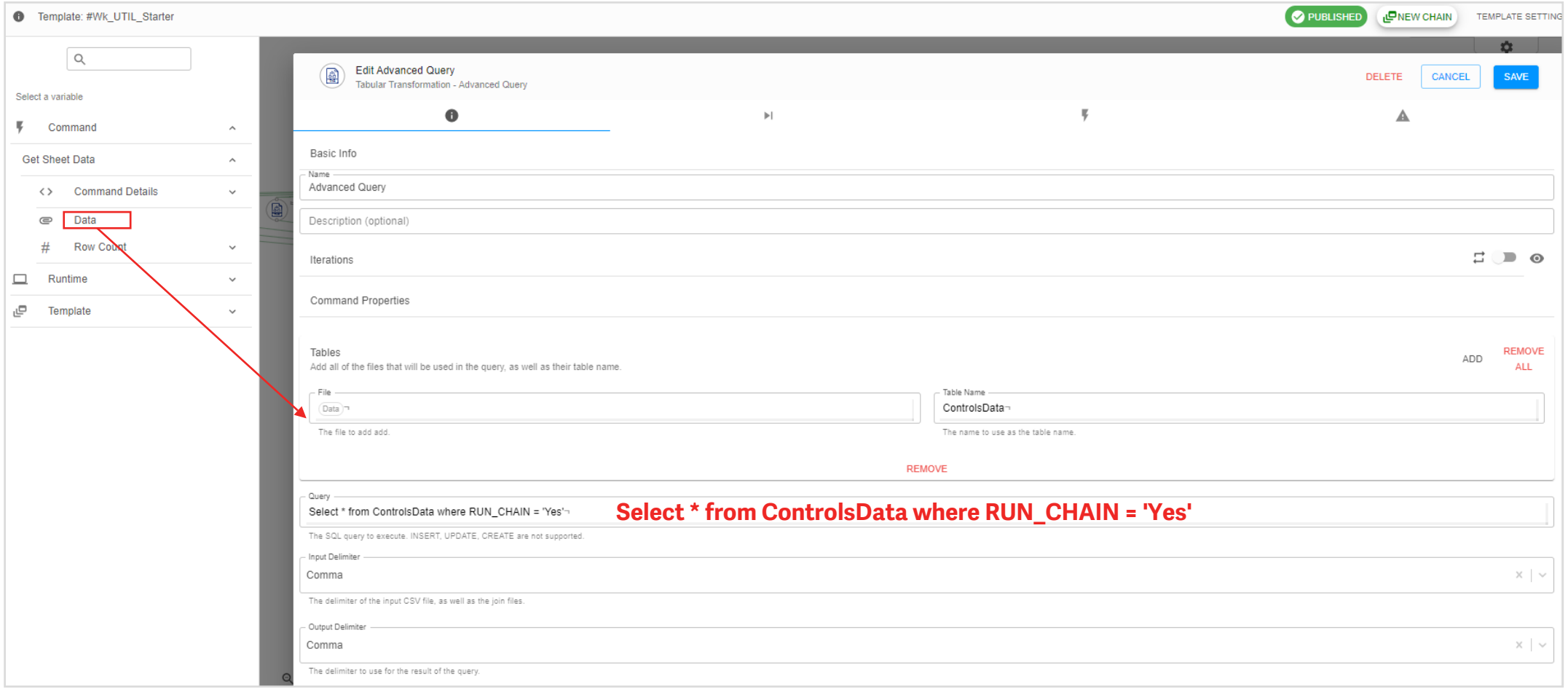
Step 5.1: Create the "Advanced Query" Node P2



Drag the node from the BizApp panel to the blank space in the template. Then, connect the "Get Sheet Data" node to the "Advanced Query" node. Double click on the "Advanced Query" node to edit.

Workiva

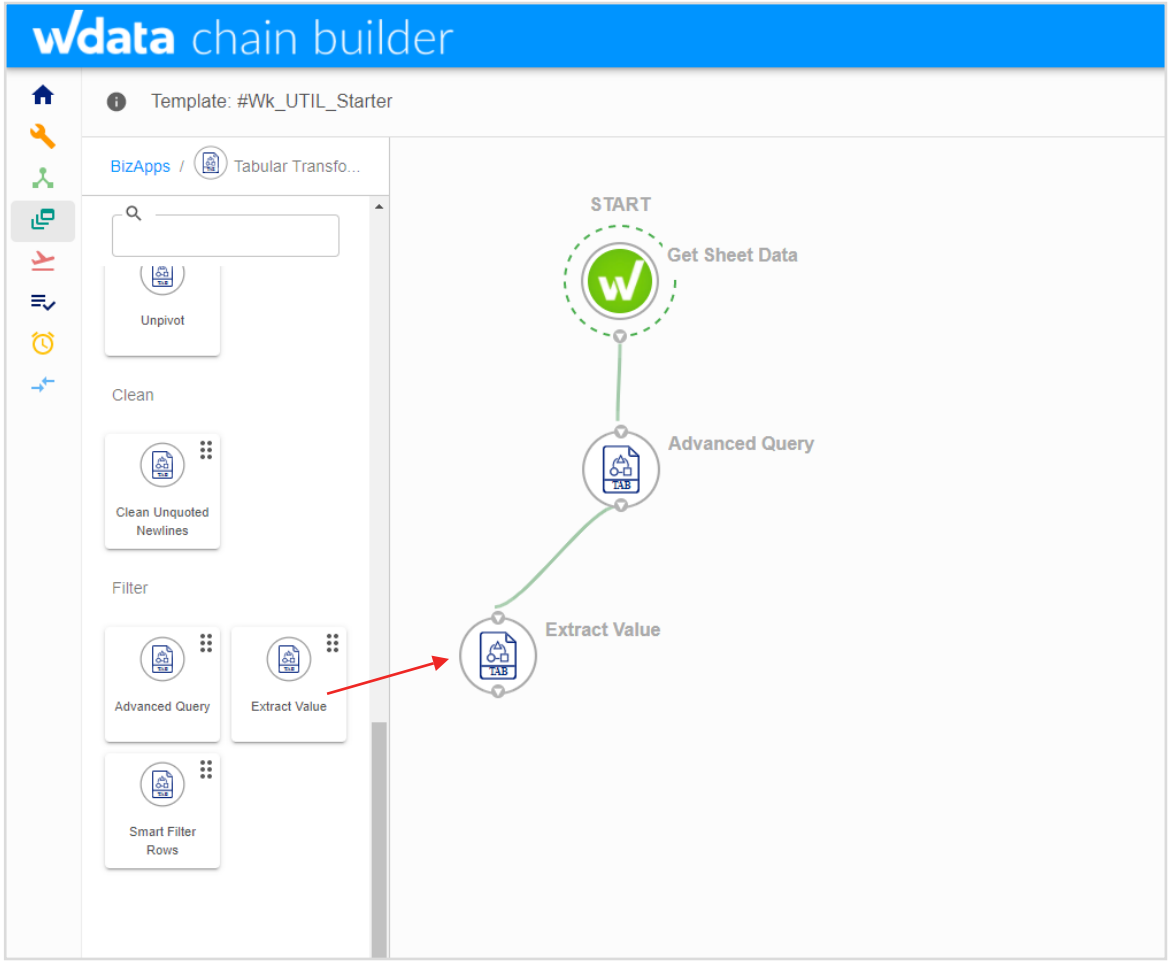
Step 5.2: Edit "Advanced Query" Node



Input the node information as shown above. For the File field, go to the left panel and select the variable 'Data' under 'Get Sheet Data'. Then, click the "Save" button.

workiva

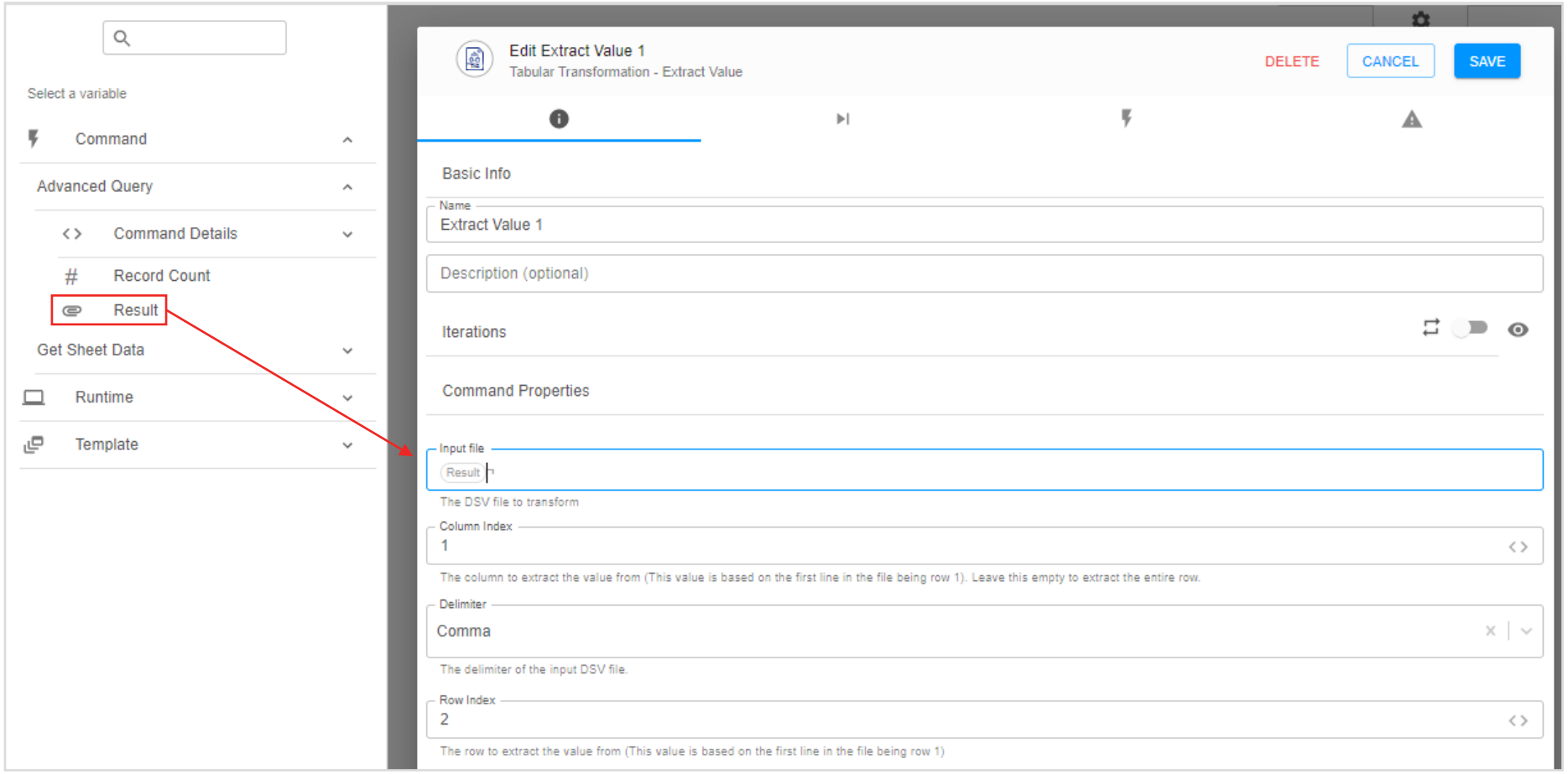
Step 6: Create the "Extract Value" Nodes



Create the "Extract Value" node by searching "Tabular Transformation" > "Extract Value" in the BizApp search bar. Remember to connect the "Advanced Query" node to the "Extract Value" node.

workiva

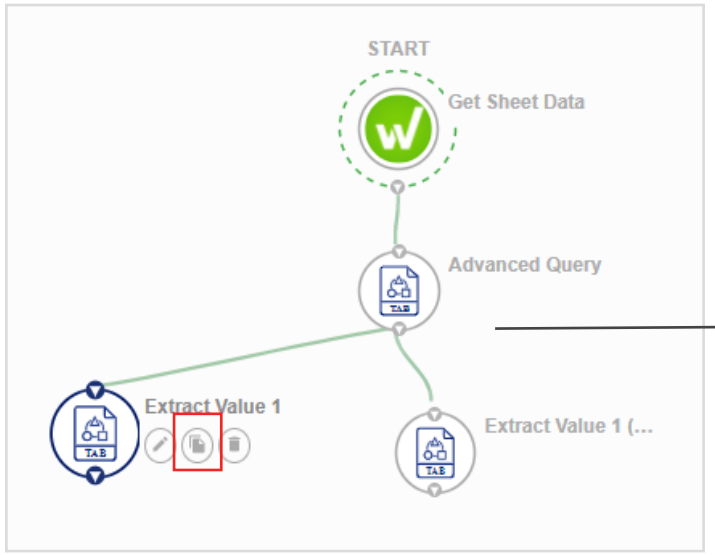
Step 6.1: Edit "Extract Value" Nodes

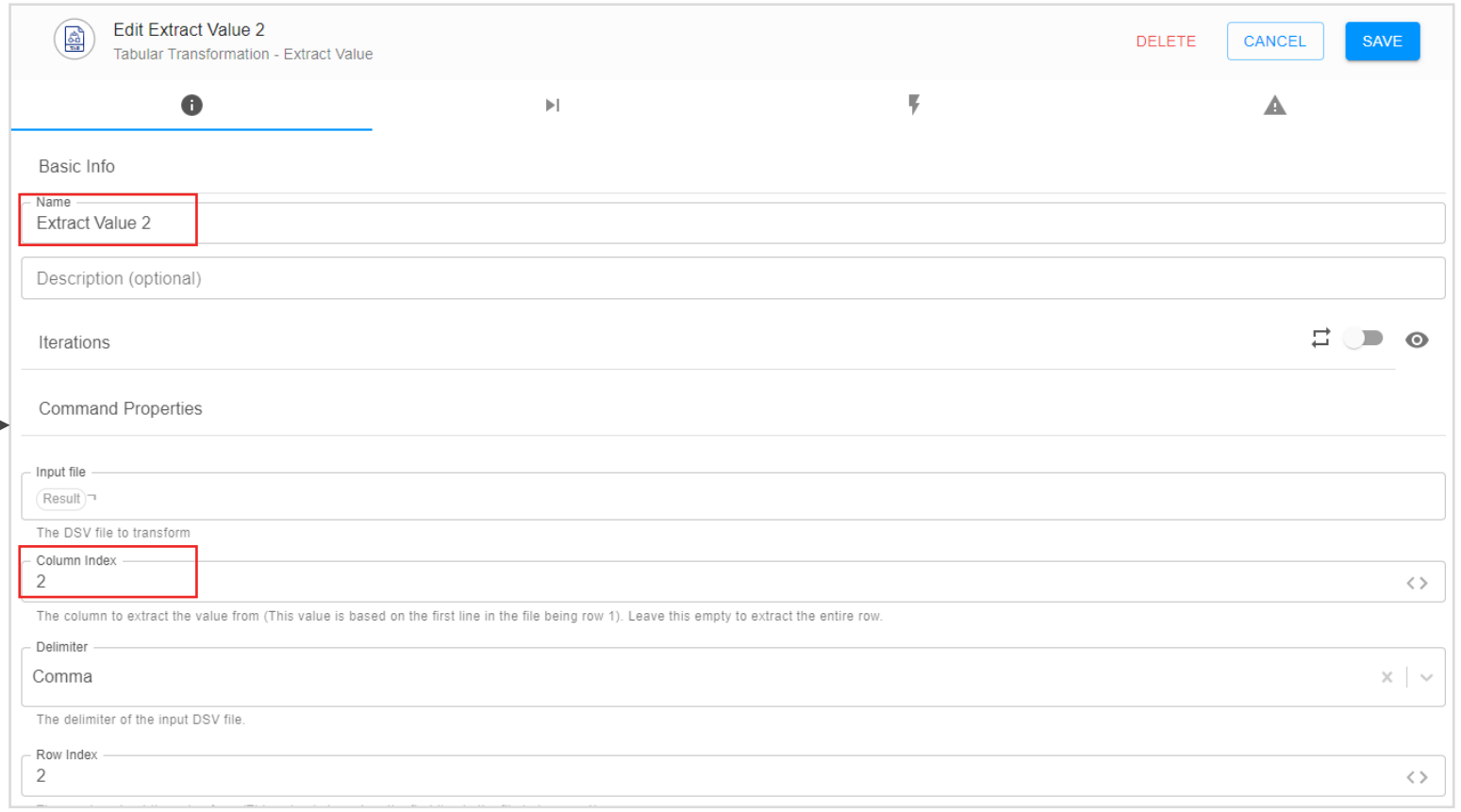


Input the node information as shown above. For the Input File, go to the left panel and select the variable 'Result' under 'Advanced Query'. By doing this, we are taking the output of the previous node as the input file of this node. Then click "Save".

Workiva

Step 6.2: Duplicate "Extract Value" Nodes

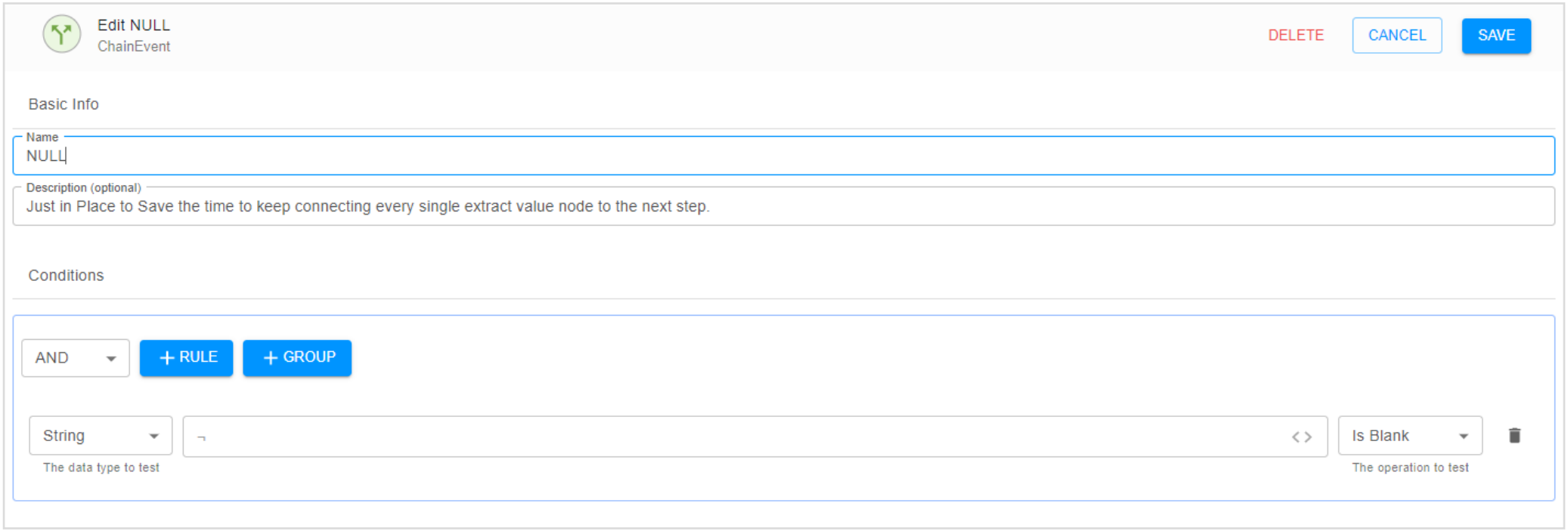




Next, duplicate the "Extract Value" node by clicking on the duplicate icon. A copy of the node will automatically pop out. Link the copy with the Advanced Query node and edit the information of the new node.
Note that each node extracts information from each column on the control sheet. As such, you would need to create 20 "Extract Value" nodes to extract information from 20 columns in the control sheet. Remember to name the nodes and fill in the Column Index following a numerical order. (i.e. name your second node "Extract Value 2". This node extracts the information in the second column of the control sheet, which is File Name)

Workiva

Step 6.3: Create Chain Event Node



Create a "Conditional" node and link the node with all of the Extract Value node. Fill in the node information as shown above. This node is just in place to hold the connections from all of the Extract Value nodes.

Workiva

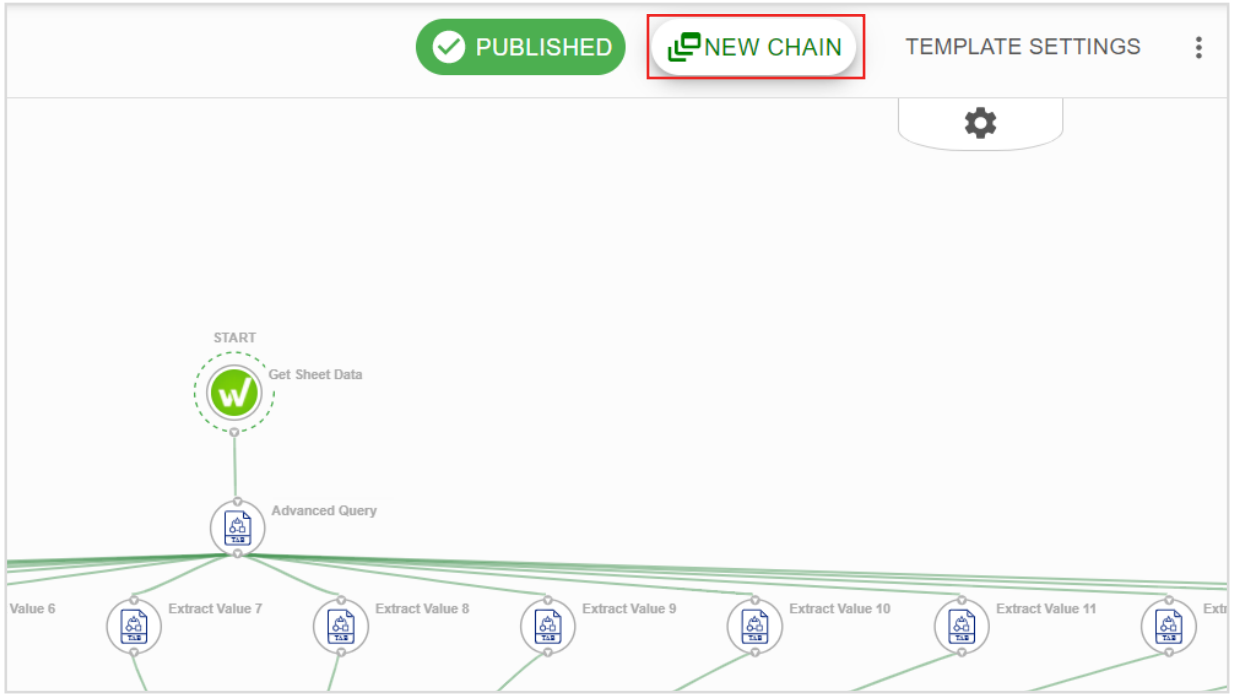
Step 6.4: Complete the Template Chain

Your chain is now completed. Hit "Publish" on the upper right corner. (Tips: if the nodes are all over the place, the Layout Graph function could re-arrange the chain structure automatically)

Workiva

Chain 4.1: External to Source

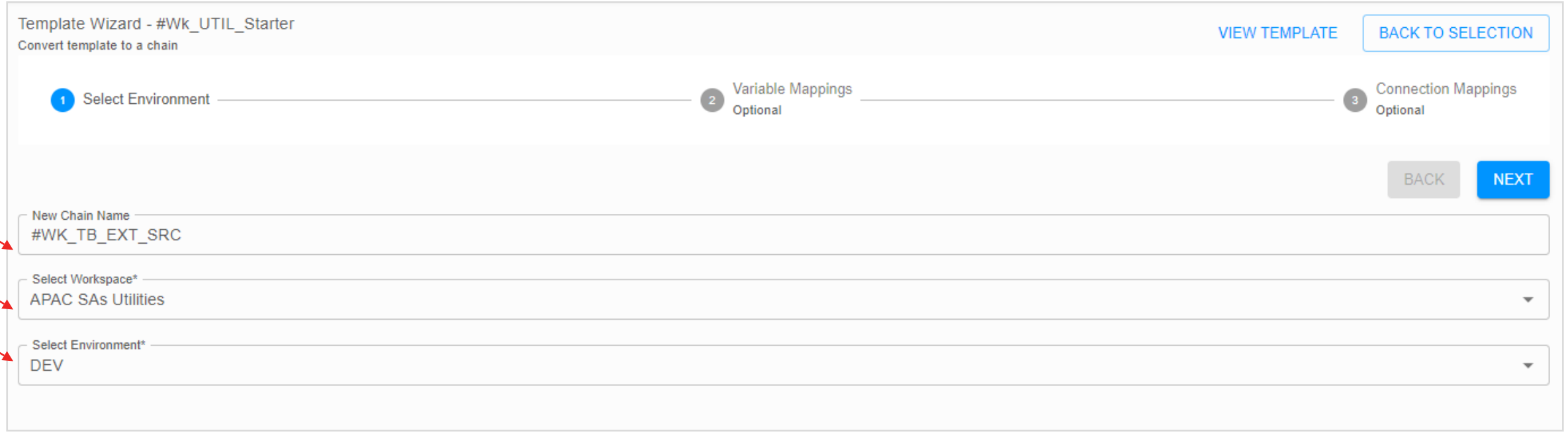
Step 1: Create New Chain from Template P1



In the template, click "New Chain" on the upper right corner. The External to Source Chain helps us import the data from the external systems to the source table.

workiva

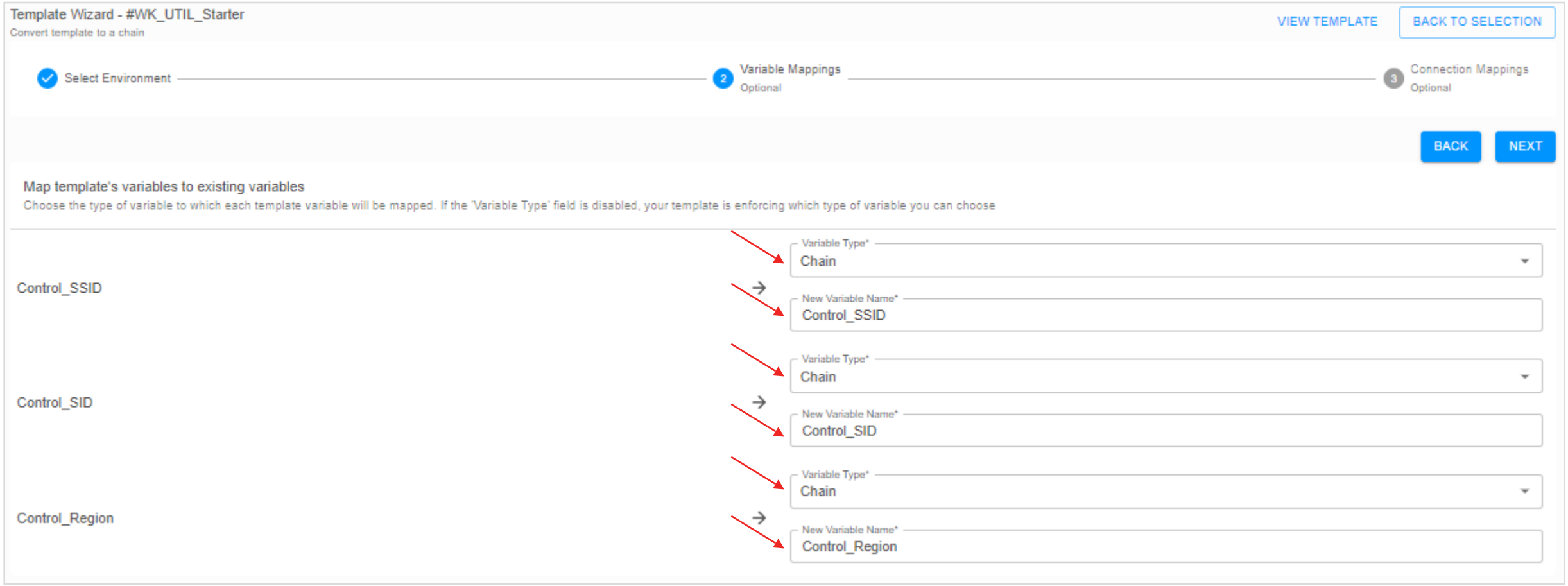
Step 1.1: Create New Chain from Template P2



Name the new chain as "#WK_TB_EXT_SRC". Select the corresponding workspace and choose "DEV" as the environment. Hit Next.

Workiva

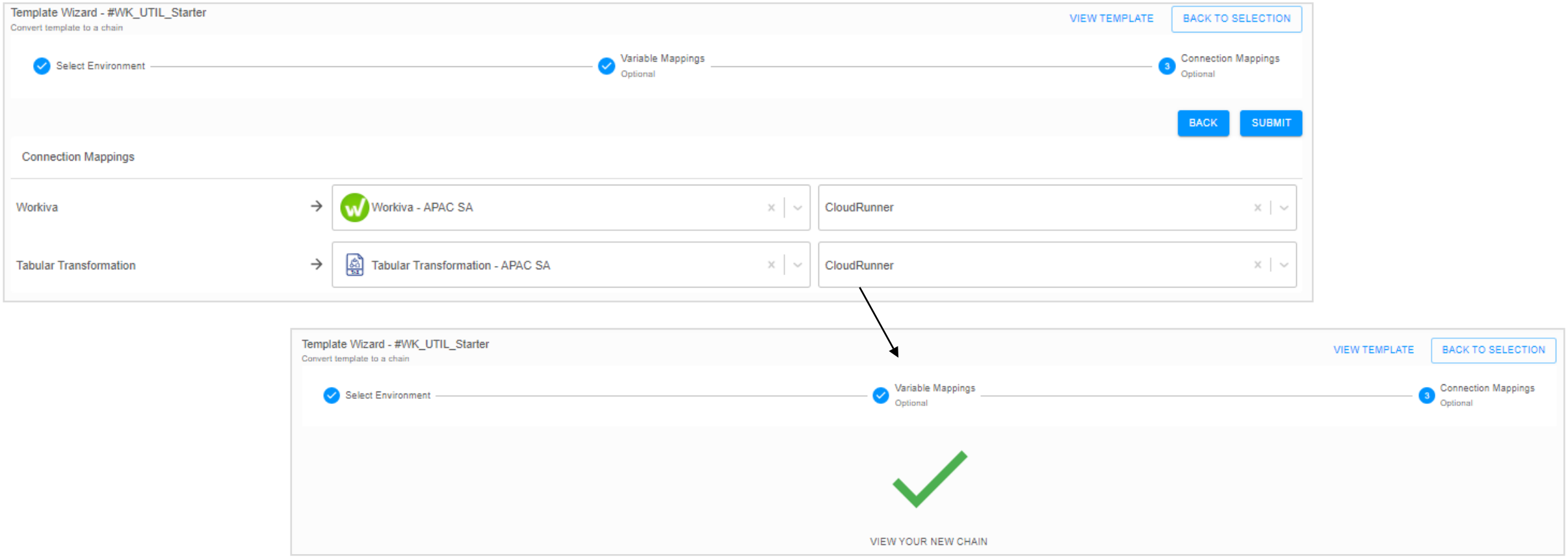
Step 1.2: Create New Chain from Template P3



Choose "Chain" as the Variable Type and input the Variable Names again as shown above.

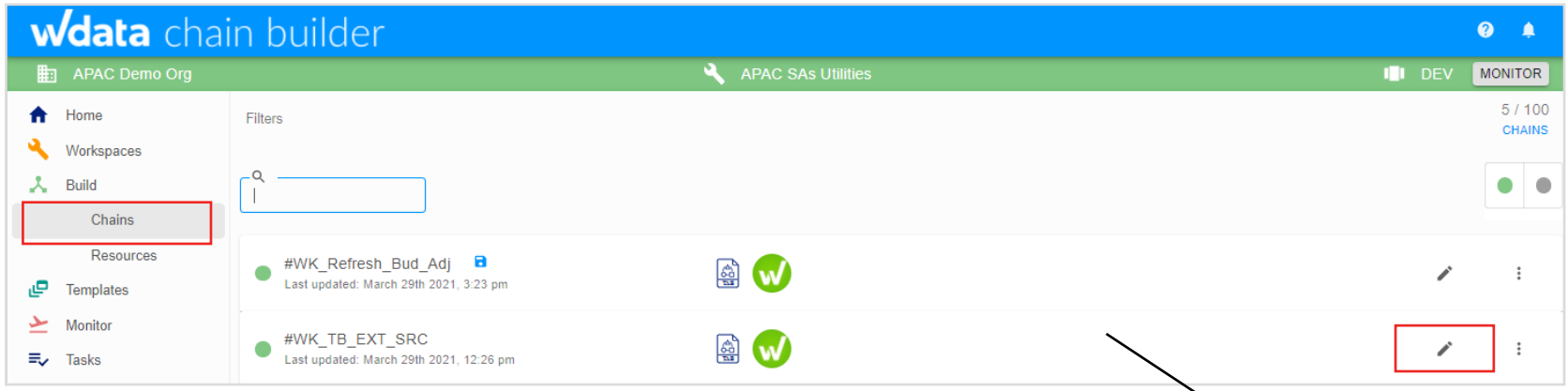
workiva

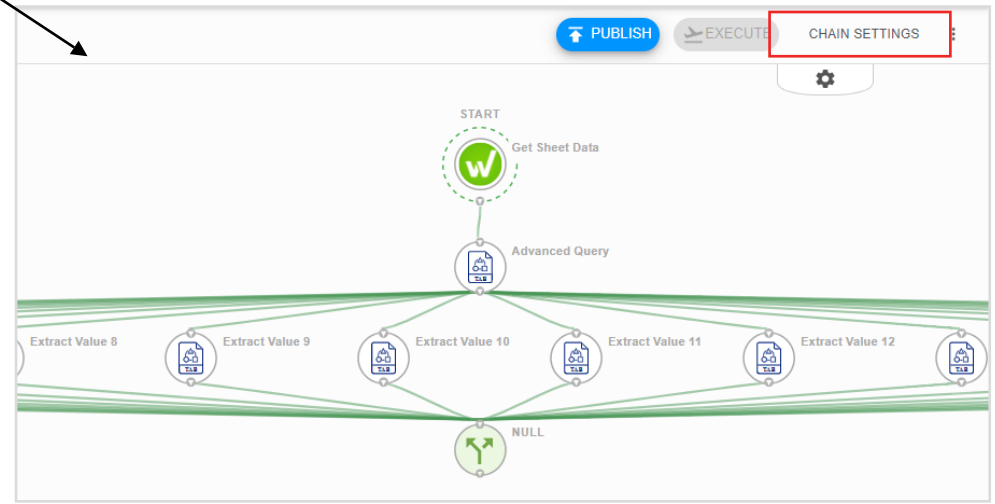
Step 1.3: Create New Chain from Template P4



Select the above connections as shown above. Click submit. The new chain is now set up.

Step 2: Edit Chain Settings P1

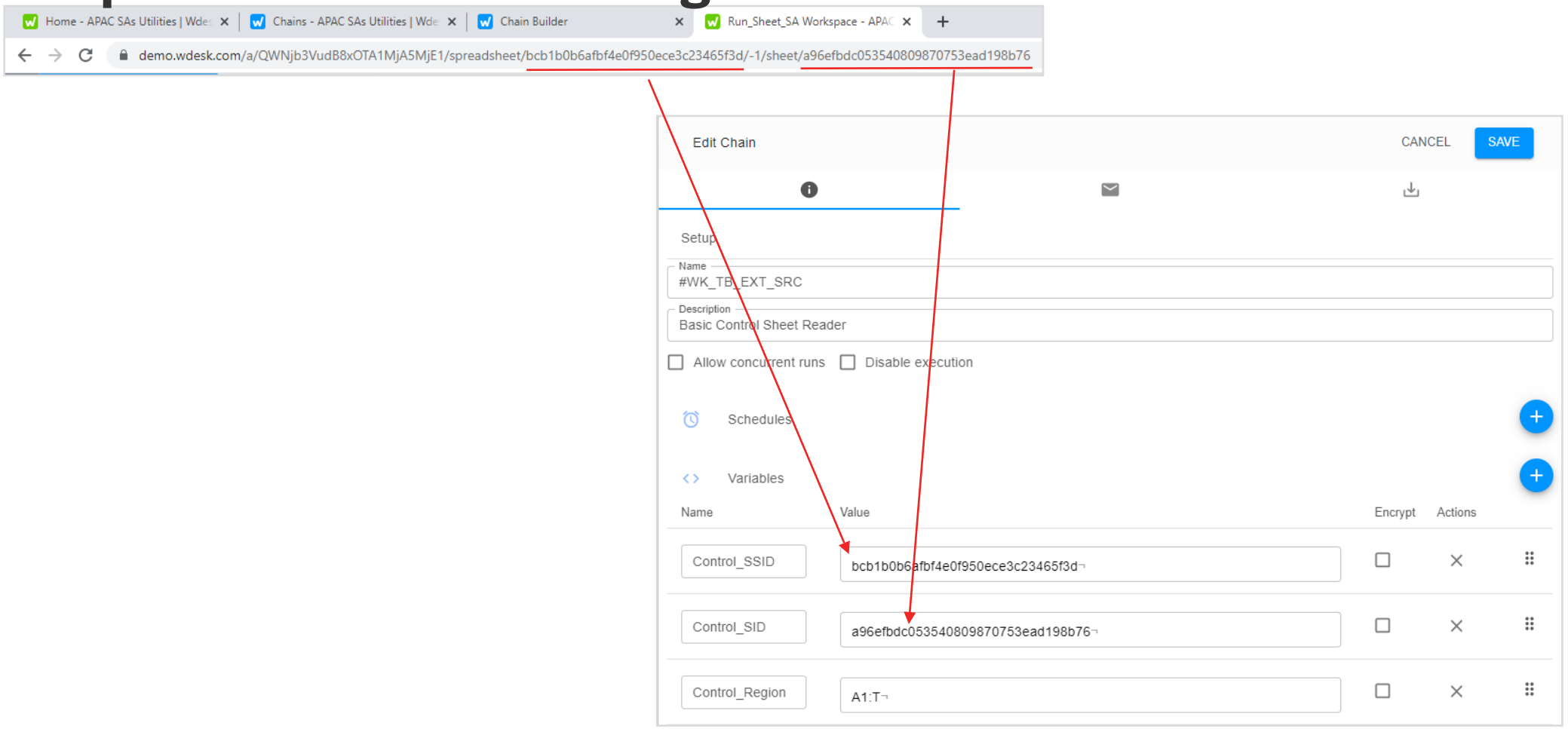




Go to "Build">"Chains" on the left panel and click into the edit icon. Then, choose "Chain Settings" on the upper right corner to edit the settings.

Workiva

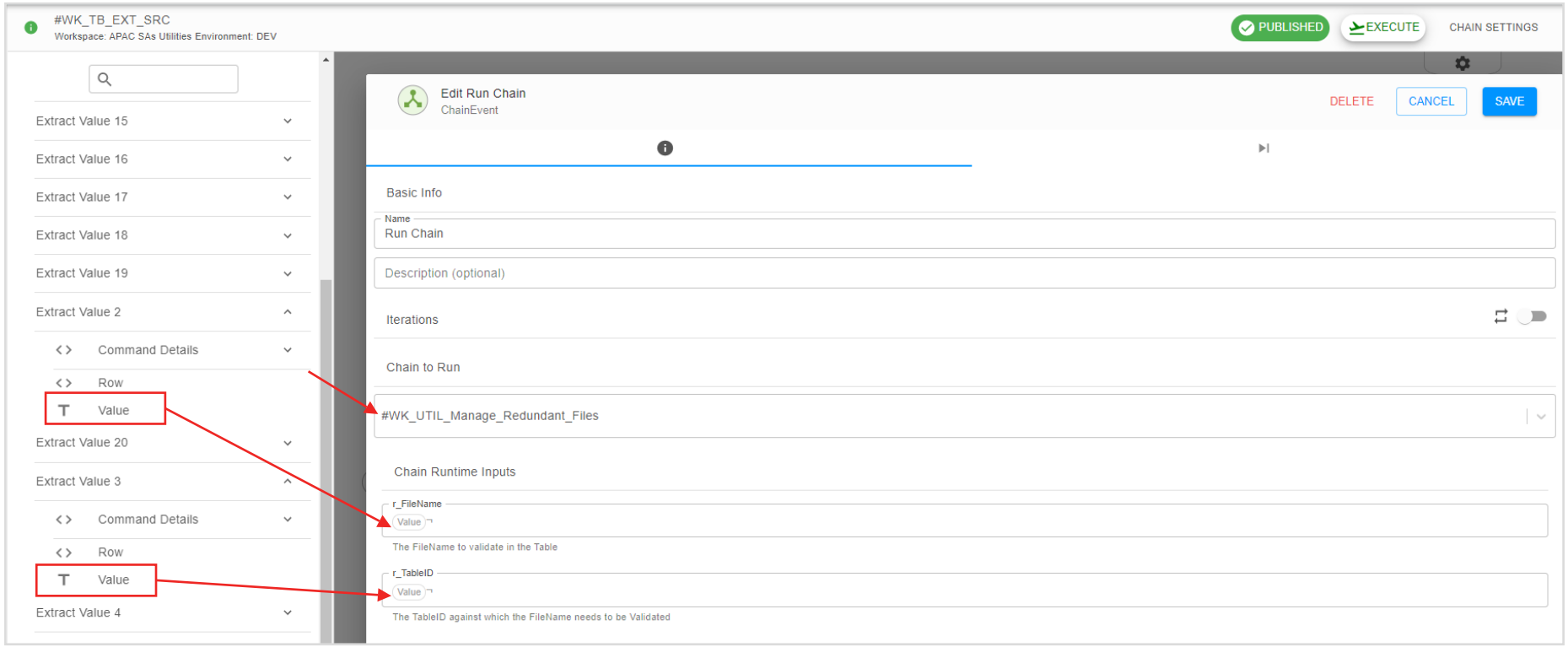
Step 2.1: Edit Chain Settings P2



Fill in the sheet ID and spreadsheet ID of the control sheet. To find those IDs, go to your control sheet to check the URL. The alphanumeric string after "/spreadsheet/" is the spreadsheet ID and the string after "/sheet/" is the sheet ID.

Workiva

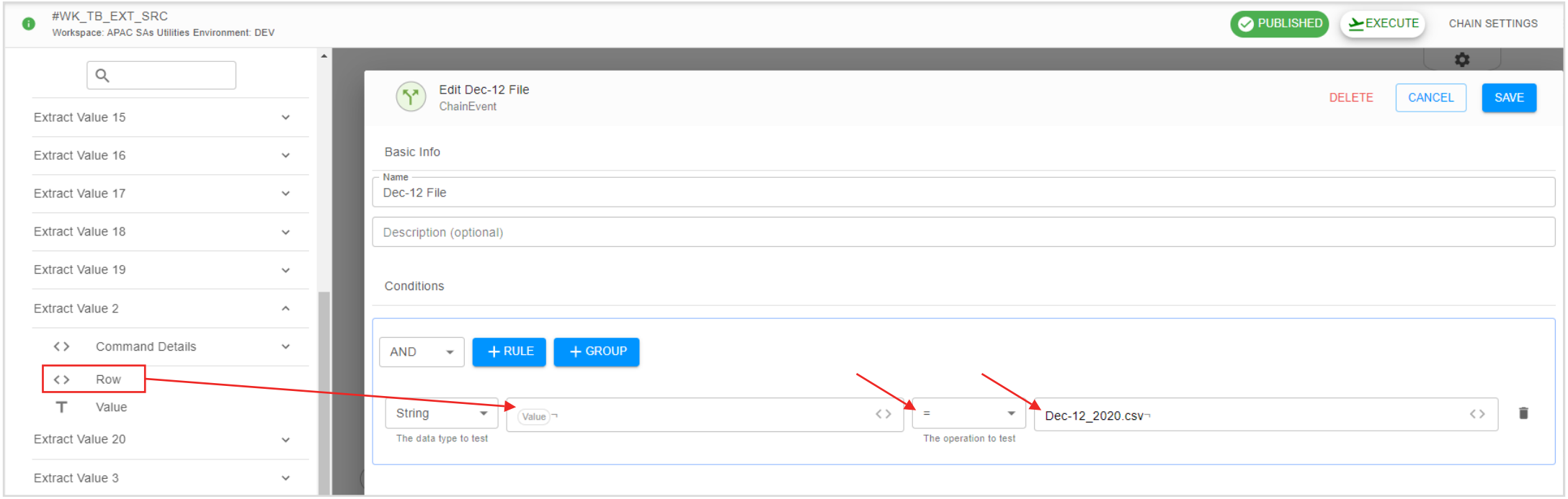
Step 3: Create Run Chain Node



Create a "Run Chain" node and link the node with the Conditional node. Fill in the node information as shown above. For r_FileName, choose the variable "Value" under "Extract Value 2" and for r_TableID, choose the variable "Value" under "Extract Value 3". (the Extract Value 2 node extracts the FileName column and Extract Value 3 node extracts the TableID from the control sheet)

Workiva

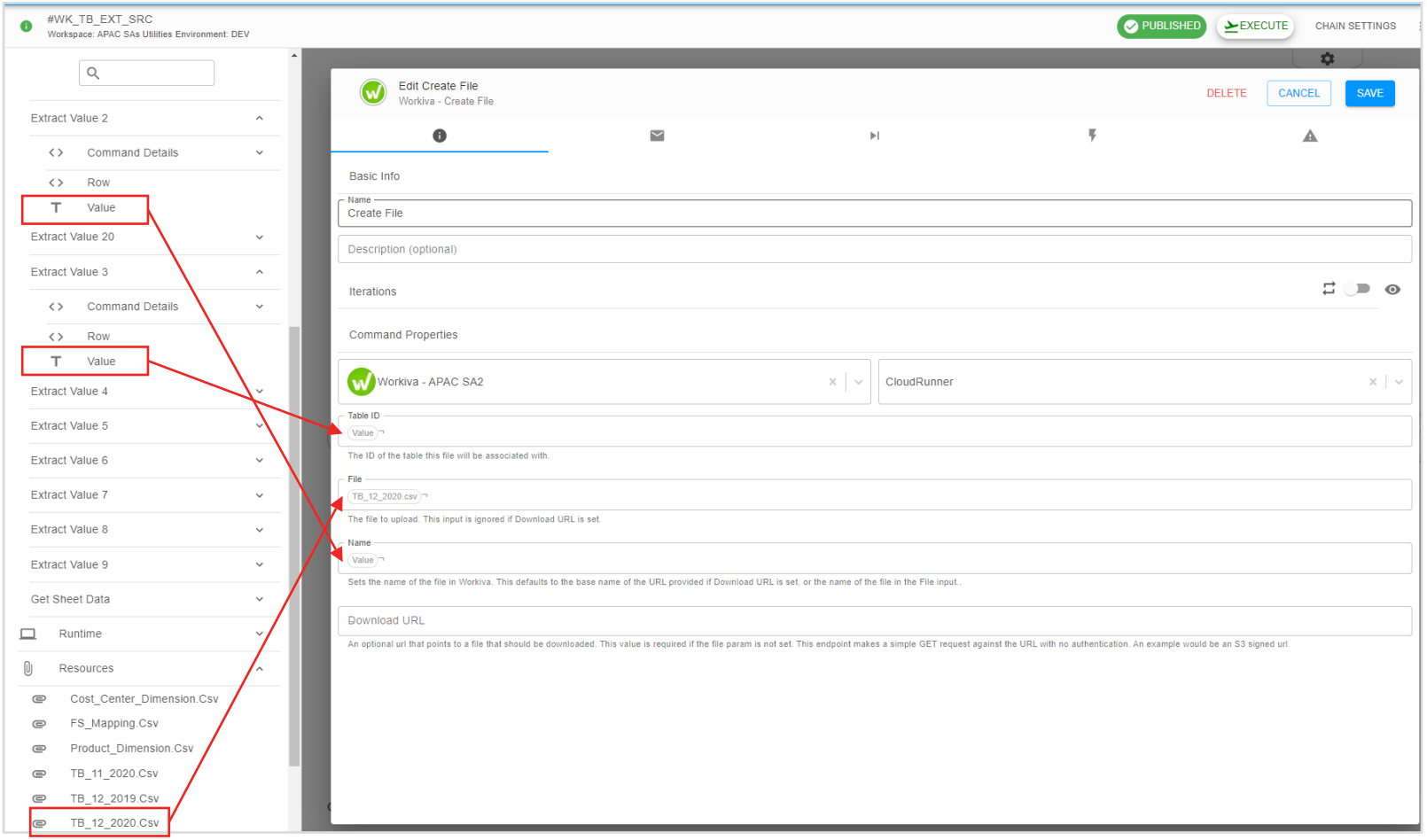
Step 4: Create Conditional Node - Edit Dec-12 File



Create a "Conditional" node and link the node with the Run Chain node. Fill in the node information as shown above. Please note that for value, choose the variable "Value" under "Extract Value 2". Note that we will later create other 2 conditional nodes to account for the situation in which we upload the November 2020 and December 2019 files.

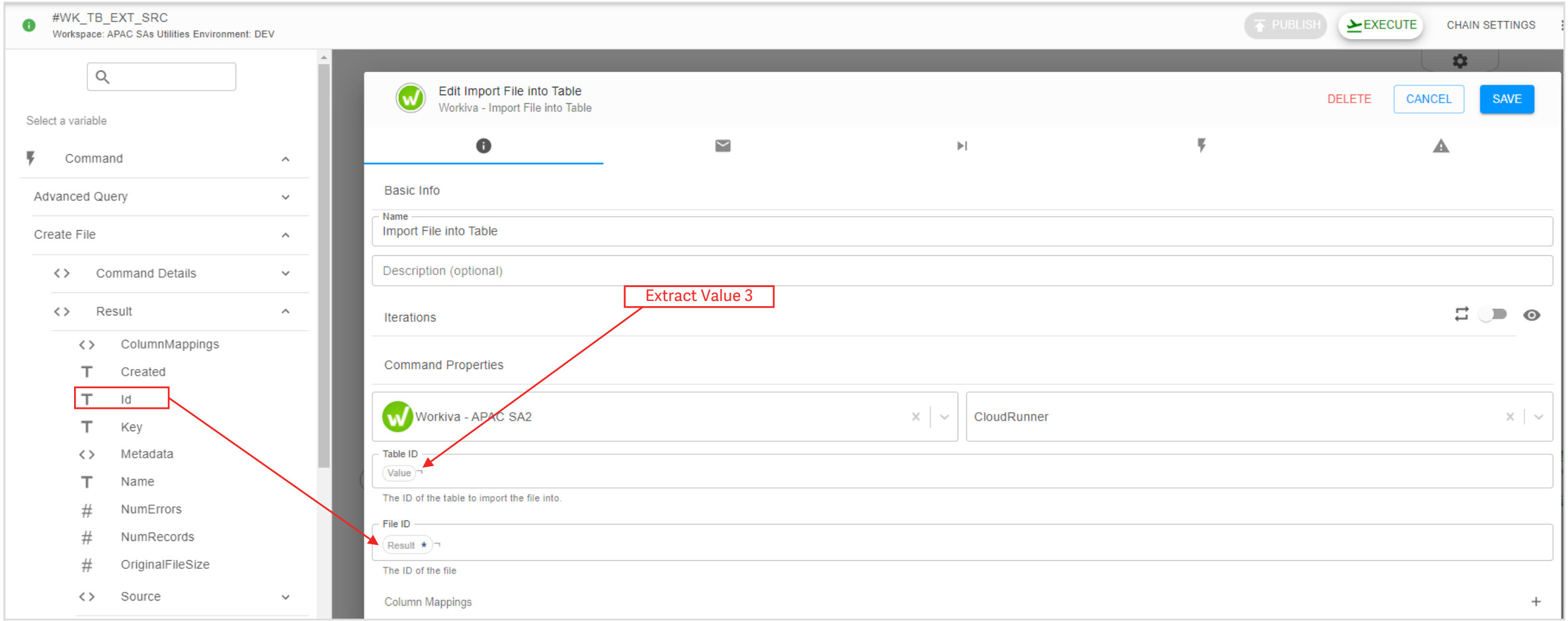
Workiva

Step 5: Create Create File Node



Create a "Workiva">"Create File" node and link the node with the previous Conditional node. Fill in the node information as shown above.

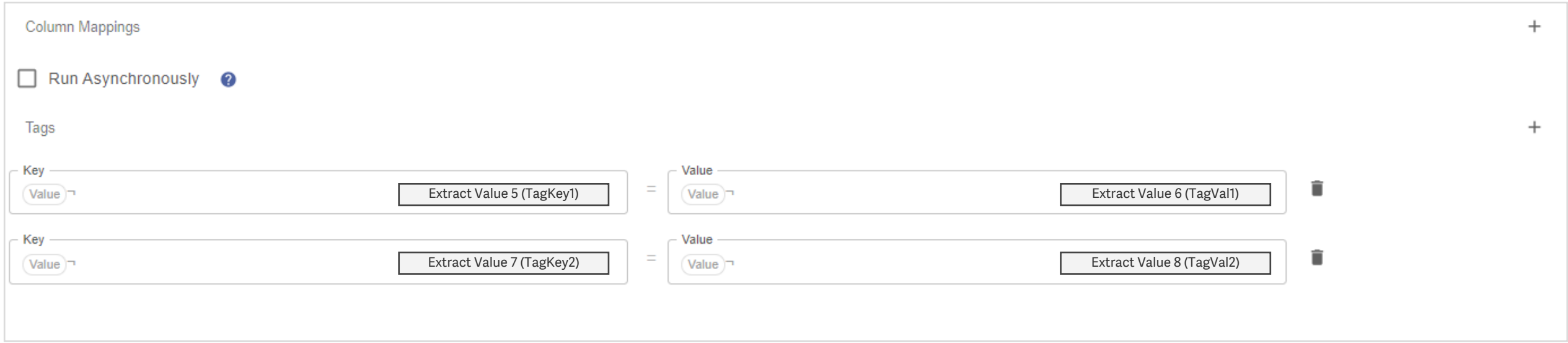
Step 6: Create Import File into Table Node P1



Create a "Workiva">"Import File into Table" node and link the node with the previous Conditional node. Fill in the node information as shown above. For TableID, choose the variable "Value" under "Extract Value 3"

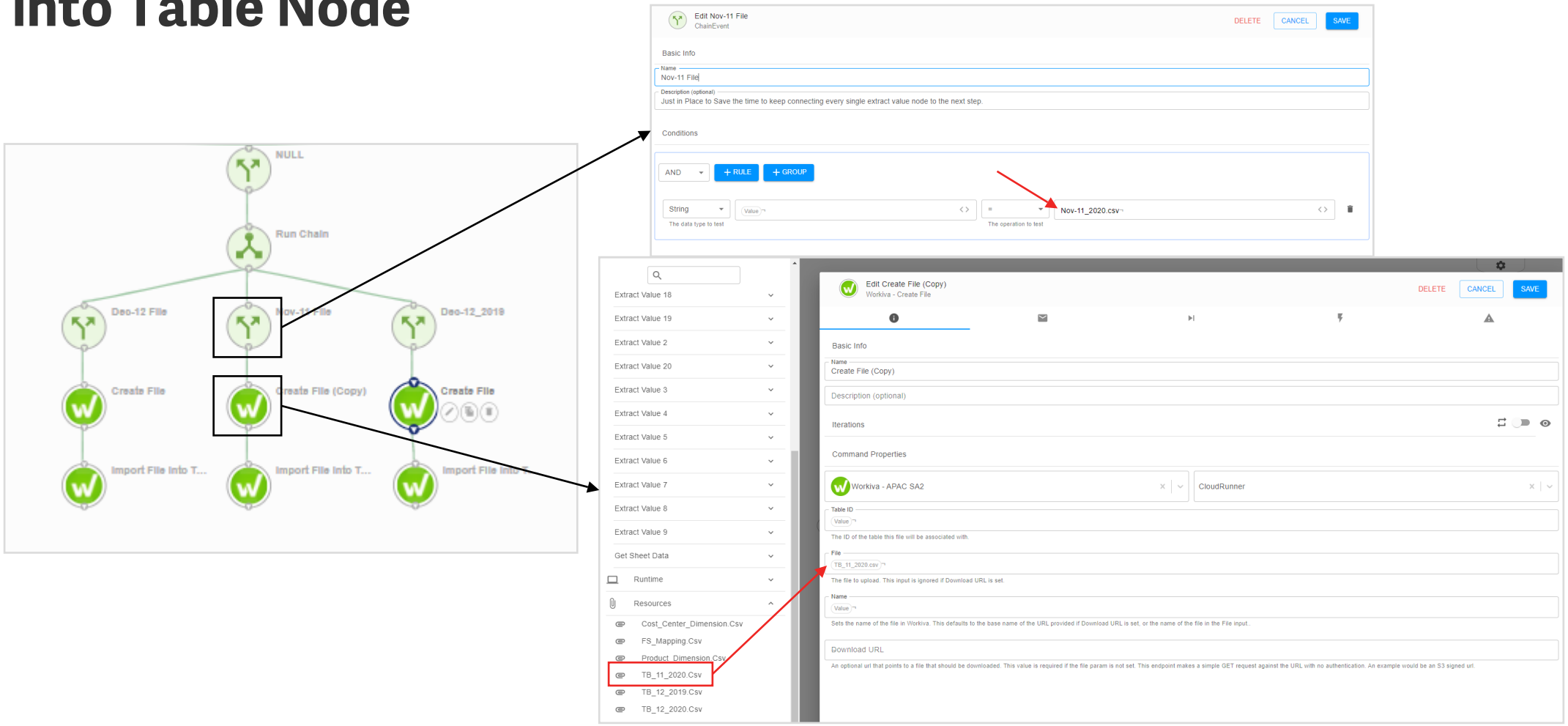
workiva

Step 6.1: Create Import File into Table Node P2



Add two tags in the bottom and fill in the tag information as shown above. Select variables "Value" under "Extract Value 5" and "Extract Value 7" as tag keys and "Extract Value 6" and "Extract Value 8" as tag values.

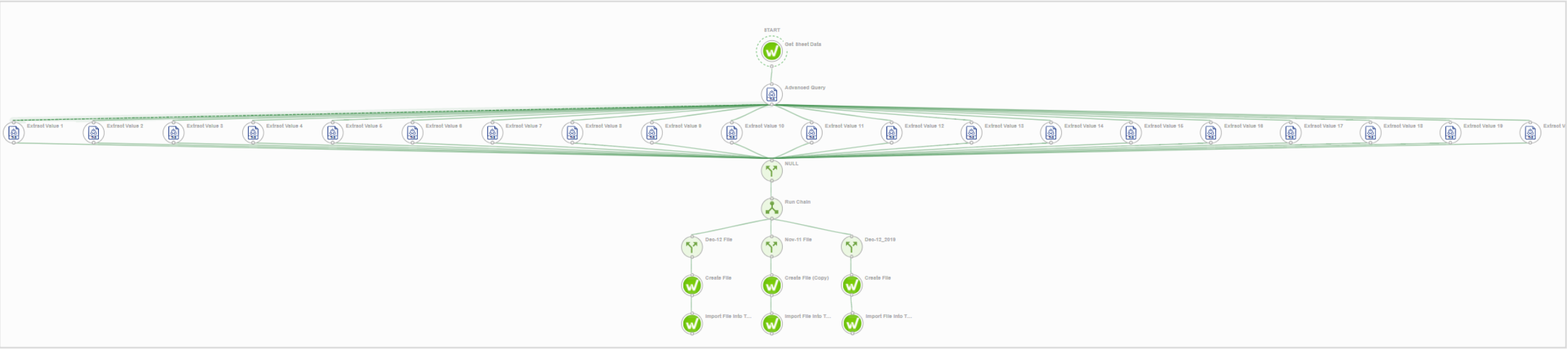
Step 7: Duplicate Conditional Node, Create File Node and Import File into Table Node



Duplicate two sets of Conditional Node, Create File Node and Import File into Table Node and connect the nodes as shown above. For each set, edit the csv file names in the first two nodes as indicated in red.

Workiva

Step 8: Complete Chain 3.1 - External to Source

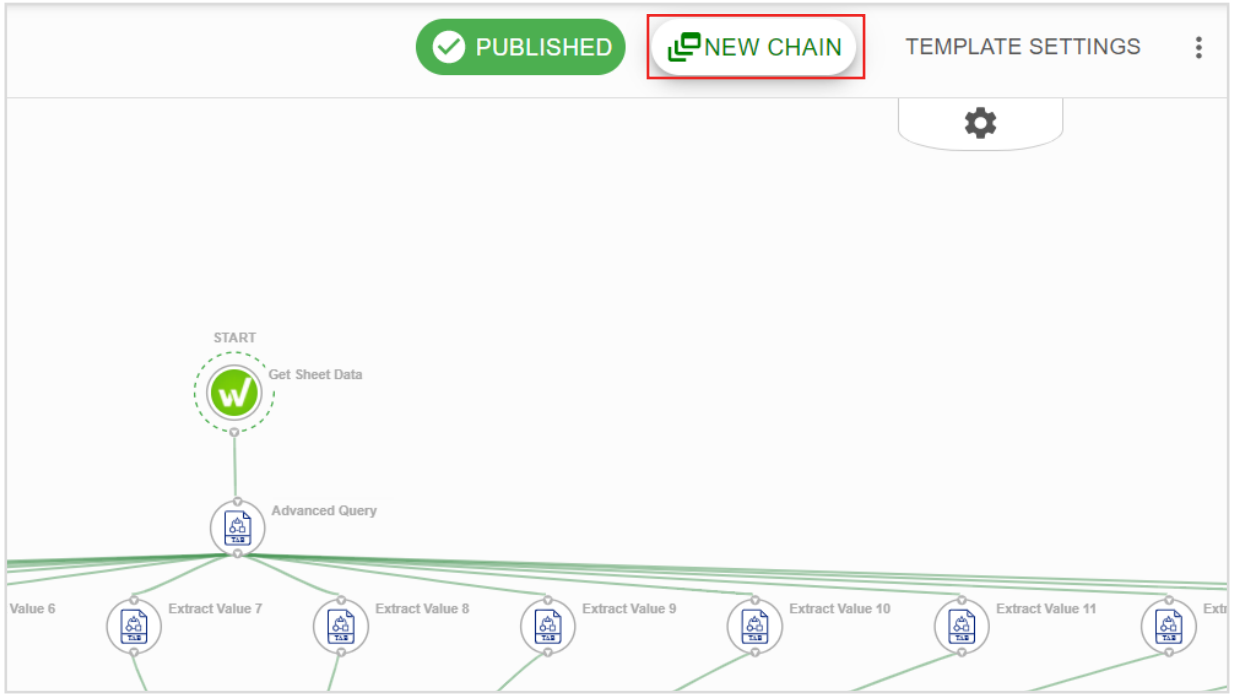


Double check the structure of the chain. Remember to hit Publish.

Workiva

Chain 4.2: Source to Staging

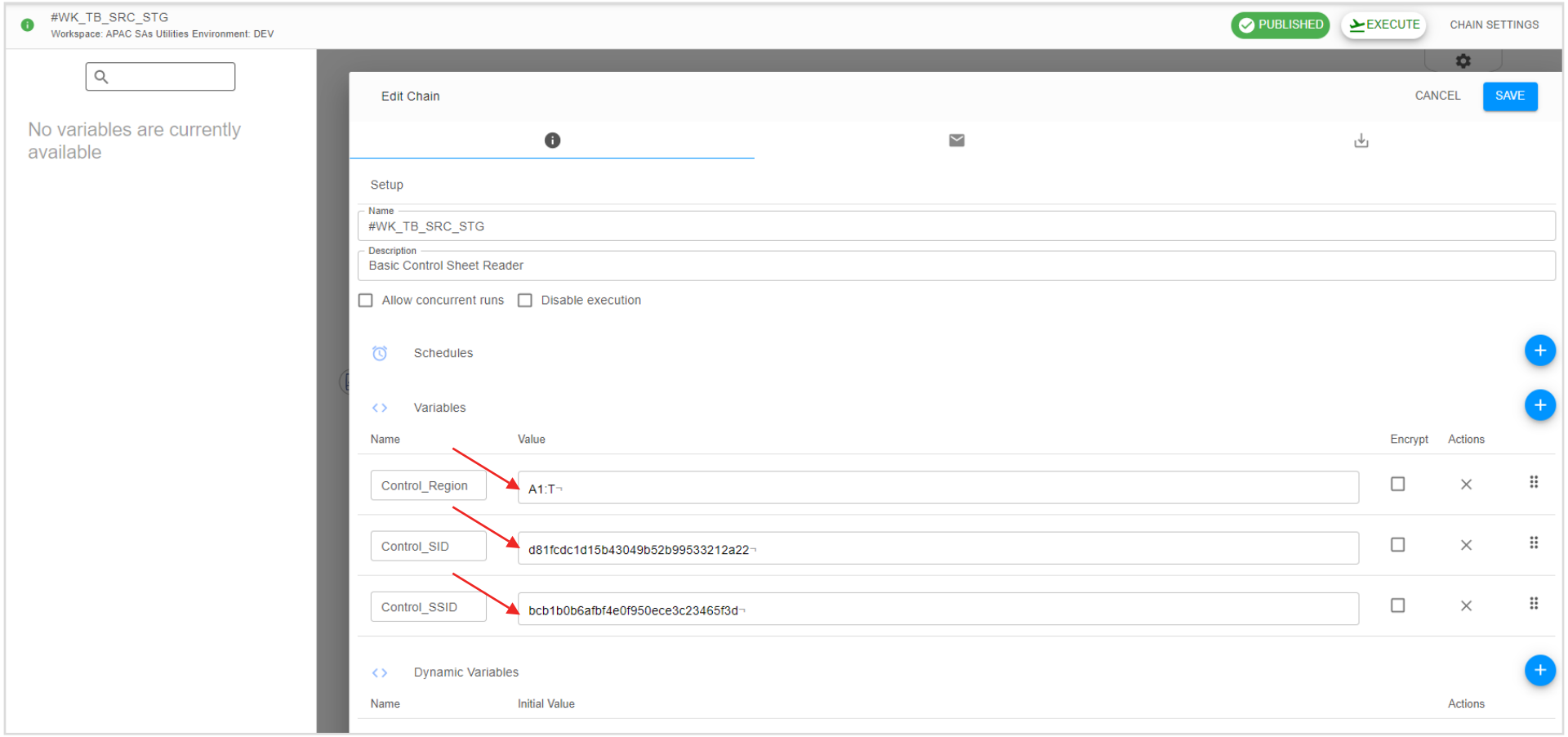
Step 1: Create New Chain from Template P1



In the template, click "New Chain" on the upper right corner. The External to Source Chain helps us import the data from the external systems to the source table.

workiva

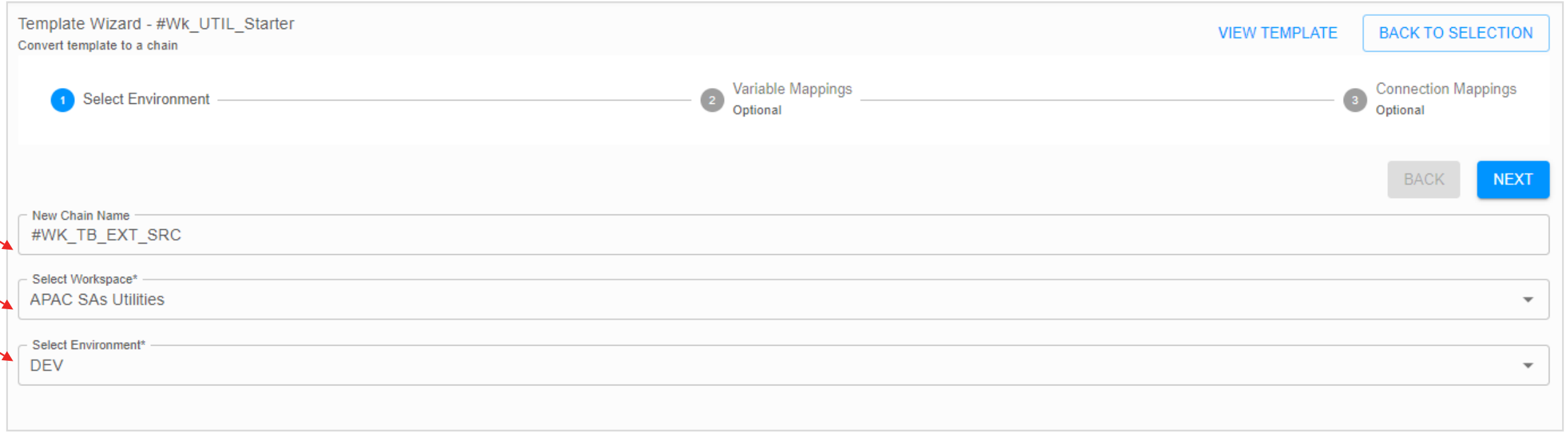
Step 1: Edit Chain Settings



Repeat step 1 (including sub-steps) and build a new chain "#WK_TB_SRC_STG" from the template. Make sure to fill in the correct Control Sheet ID and Control Spreadsheet ID.

Workiva

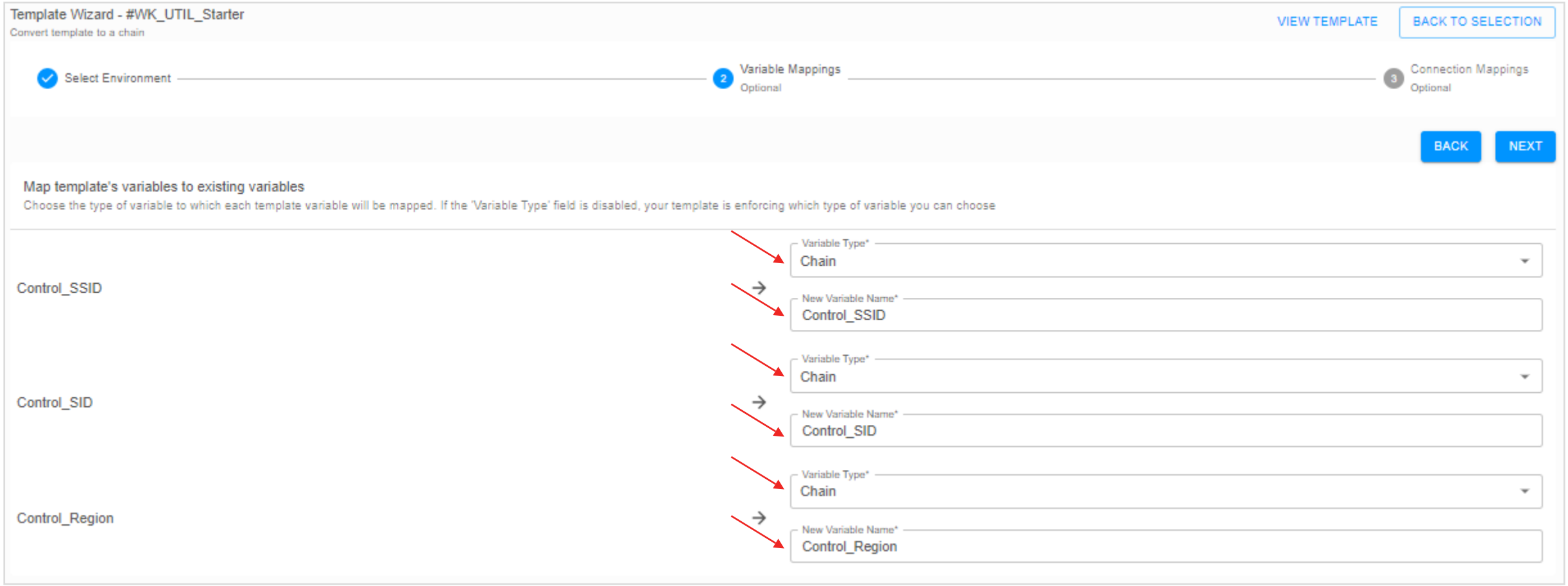
Step 1.1: Create New Chain from Template P2



Name the new chain as "#WK_TB_EXT_SRC". Select the corresponding workspace and choose "DEV" as the environment. Hit Next.

Workiva

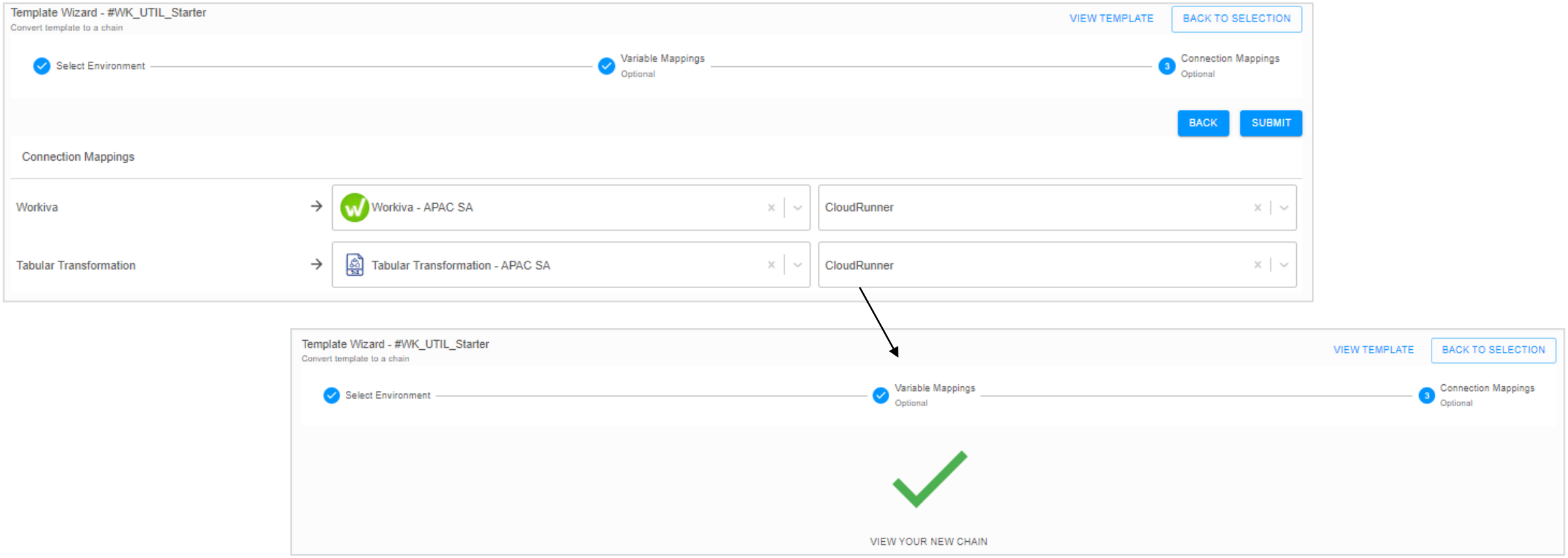
Step 1.2: Create New Chain from Template P3



Choose "Chain" as the Variable Type and input the Variable Names again as shown above.

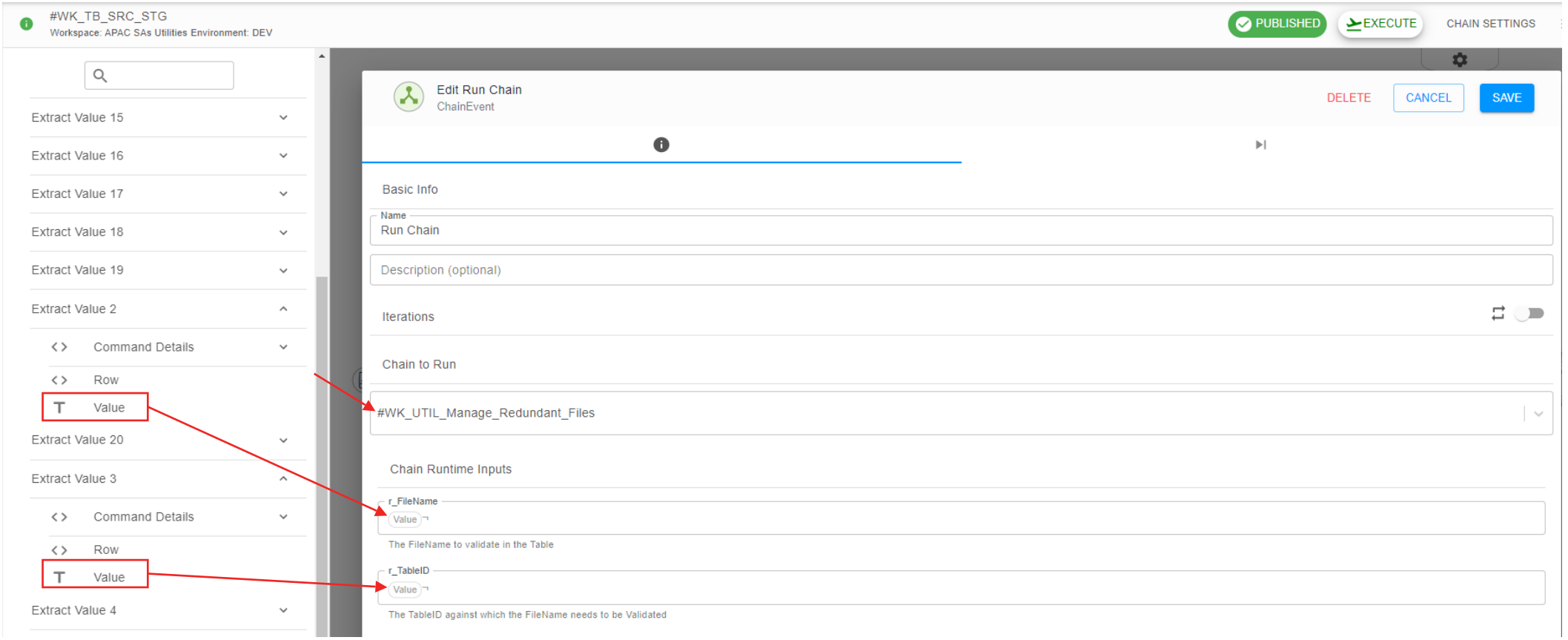
workiva

Step 1.3: Create New Chain from Template P4



Select the above connections as shown above. Click submit. The new chain is now set up.

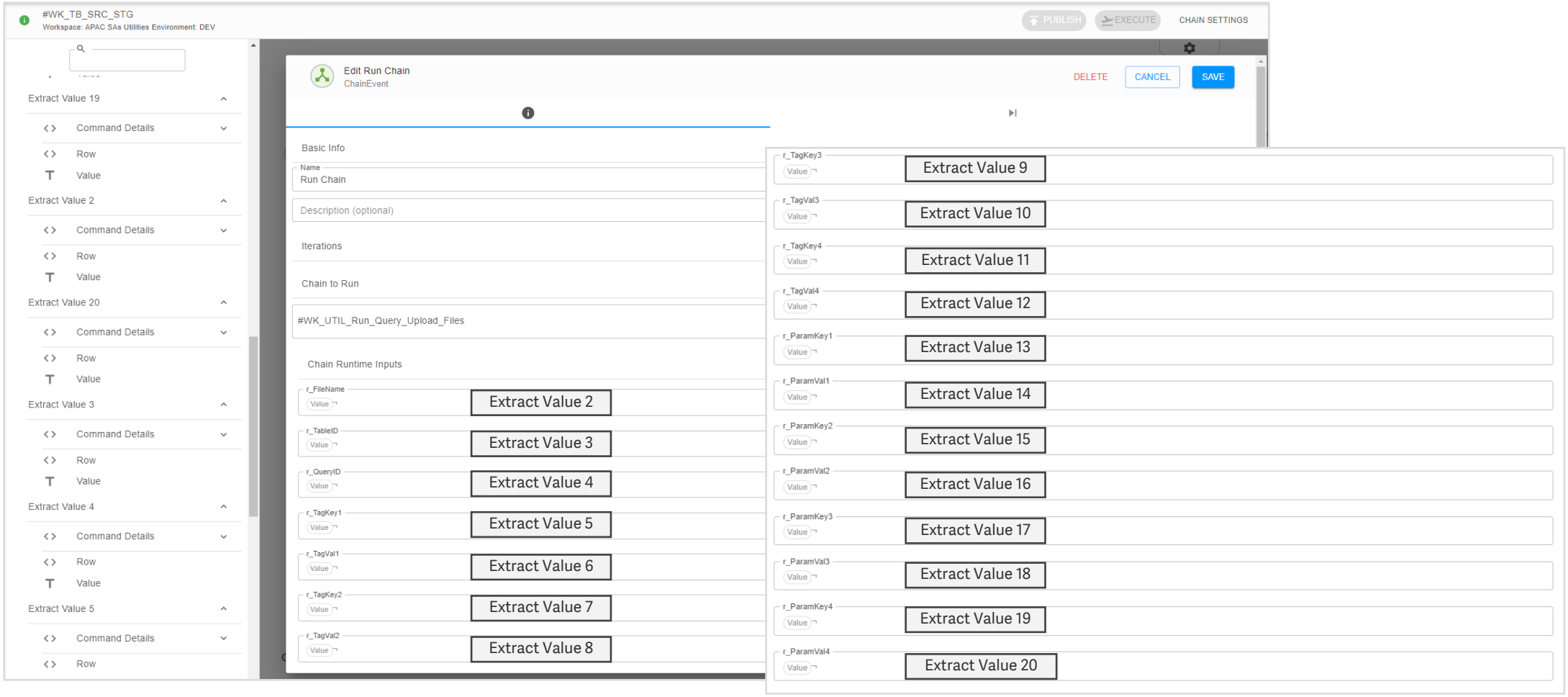
Step 2: Create Run Chain Node P1



Repeat step 3 in Chain 1.1 to create a "Run Chain" node and link the node with the conditional node.

Workiva

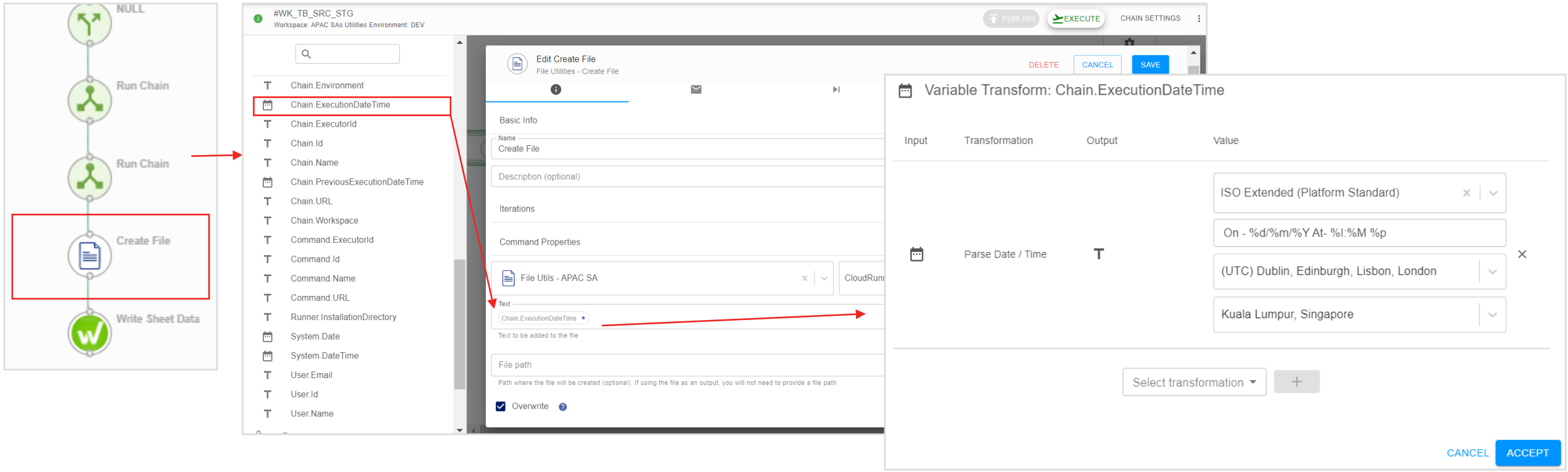
Step 2.1: Create Run Chain Node P2



Create another "Run Chain" node and link the node with the previous Run Chain node. Fill in the node information as shown above. Select the variable "Value" under the Extract Value node that corresponds to each field.

Workiva

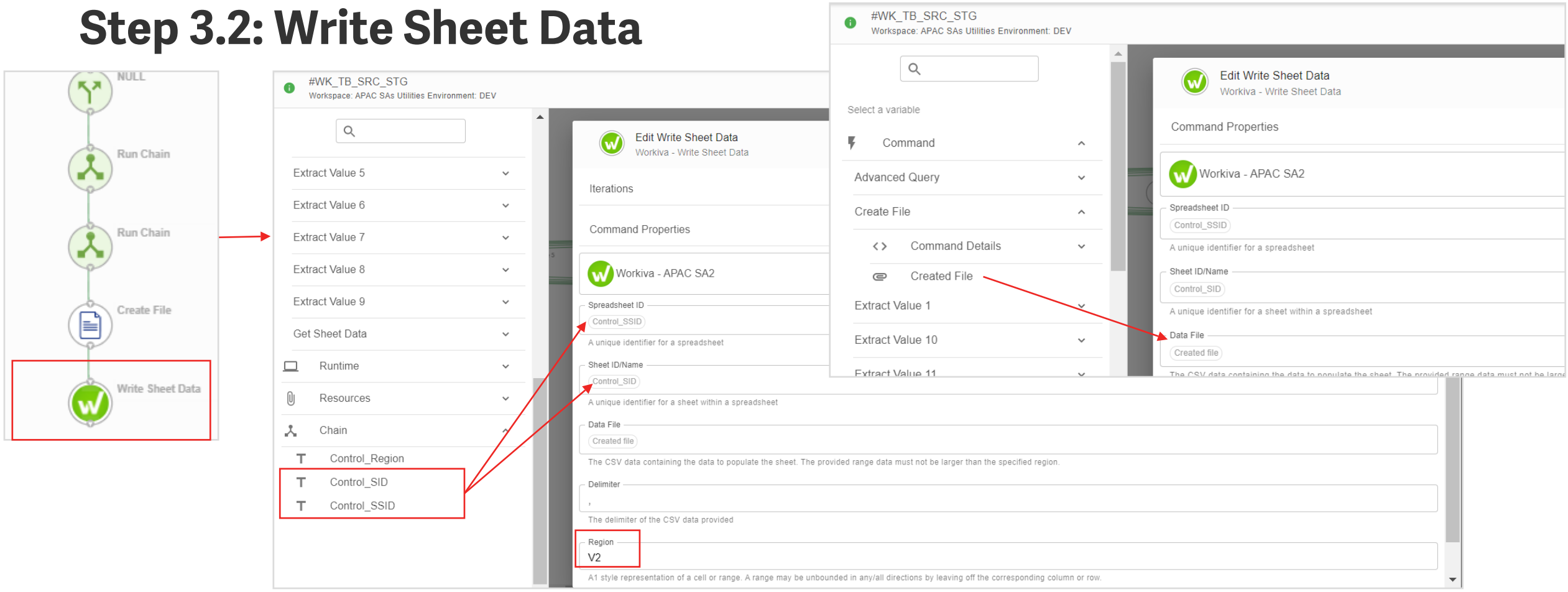
Step 3.1: Logging Time Stamp



In the 'Text', select "Chain.ExecutionDateTime". Click on the "Chain.ExecutionDateTime", under 'Select Transformation', select "Parse Date / Time", and configure according to the screenshot above. Note, the input below 'ISO Extended (Platform Standard)' is, copy the following: On - %d/%m/%Y At- %I:%M %p

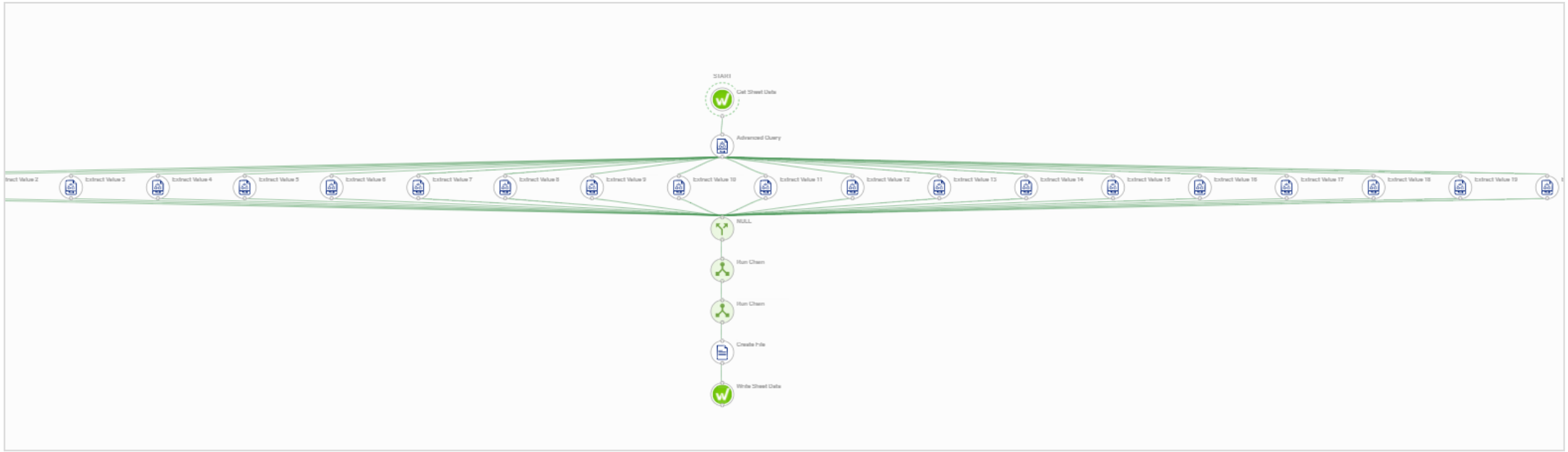
On - %d/%m/%Y At- %I:%M %p

Step 3.2: Write Sheet Data



In the 'Write Sheet Data', input the Spreadsheet, Sheet ID, Created File from the Chain and 'Create File' variables.
In the Region, indicate "V2" as this is the cell where we would want the timestamp log to be inputted in the Master Control Sheet.

Step 4: Completed Chain 3.2 - Source to Staging

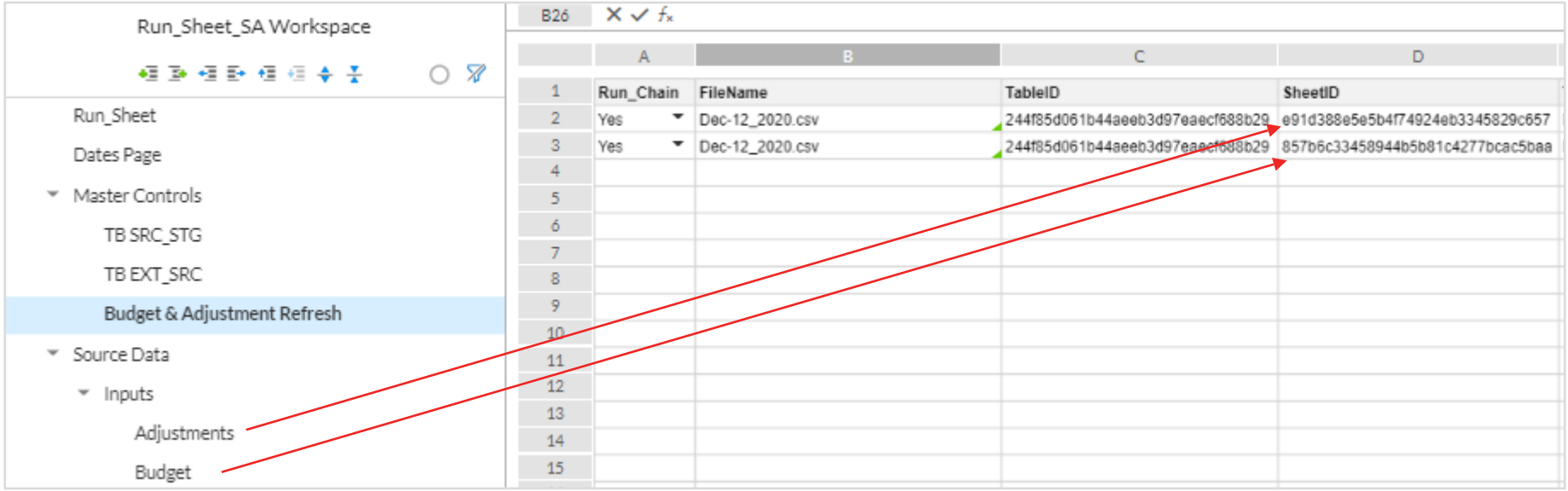


Double check the structure of the chain. Remember to hit Publish.

workiva

Chain 5: Budget & Adjustment Refresh

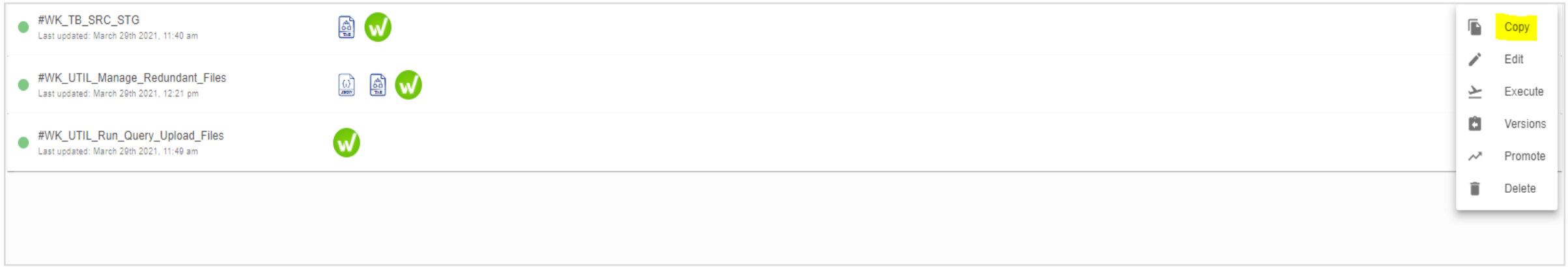
Step 1: Budget & Adjustment Refresh sheet



Create a 'Budget & Adjustment Refresh' control sheet, TableID indicated at the consumption table '#WK_CNS_TB_Sys'; SheetID indicated in 'Adjustment & Budget' sheets

Workiva

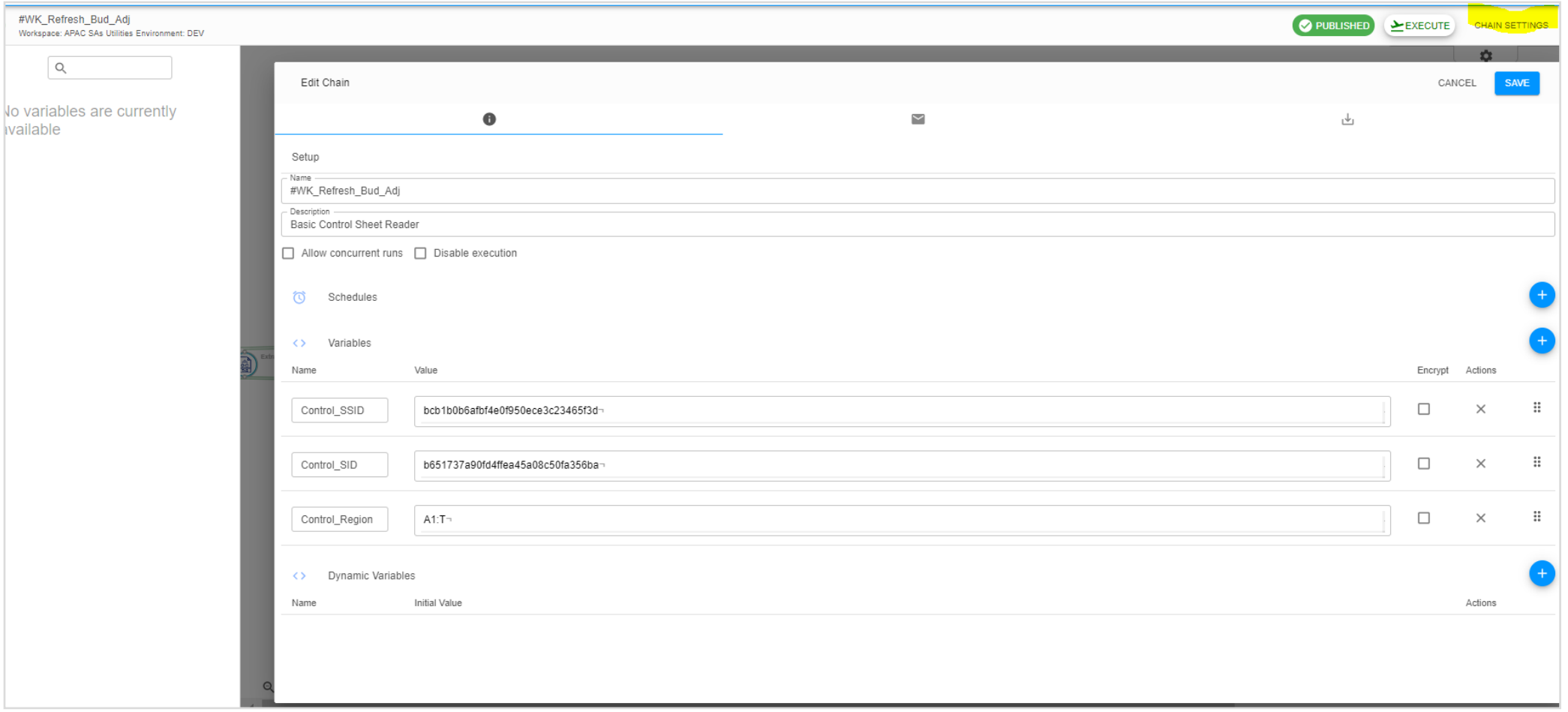
Step 2: Duplicate the "#WK_TB_SRC_STG" chain



Click on the 'three dots' on the right of the chain, click on 'Copy'. Rename the chain to '#WK_Refresh_Bud_Adj'

Workiva

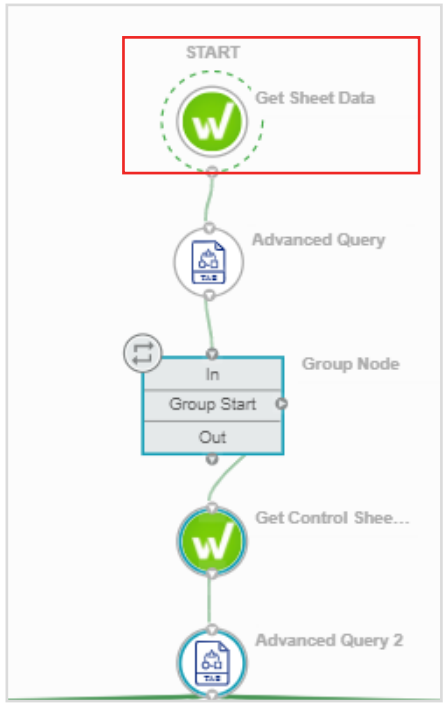
Step 3: Updating the 'Control_SID' Variable in Chain Settings

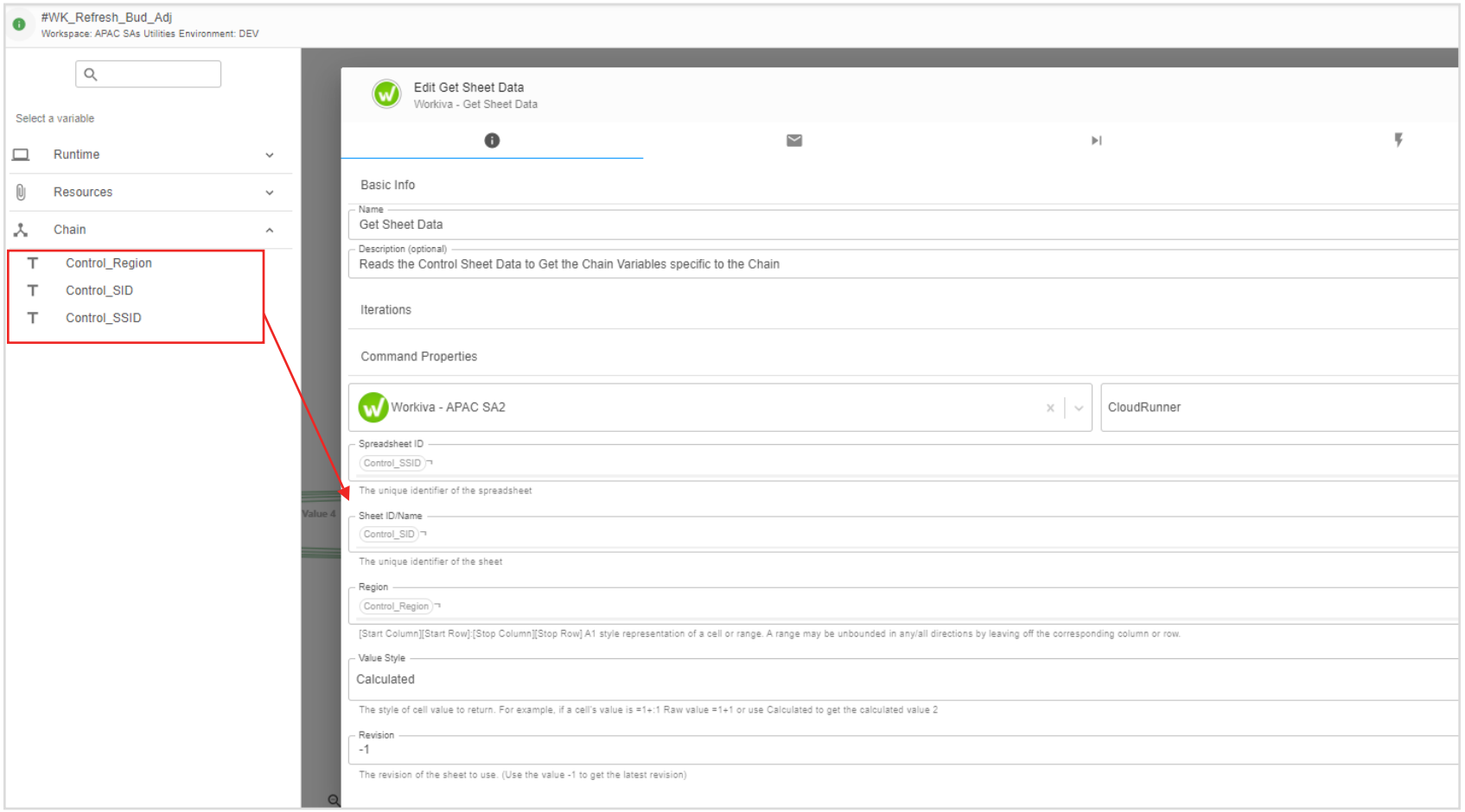


Navigate to 'Chain Settings', update the "Control_SID" to the control sheet that controls 'Budget' & 'Adjustment' spreadsheets.

Workiva

Step 4: Updating Get Sheet Data

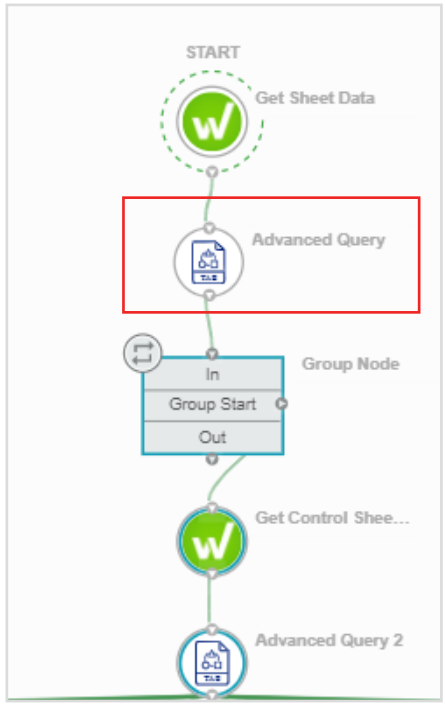


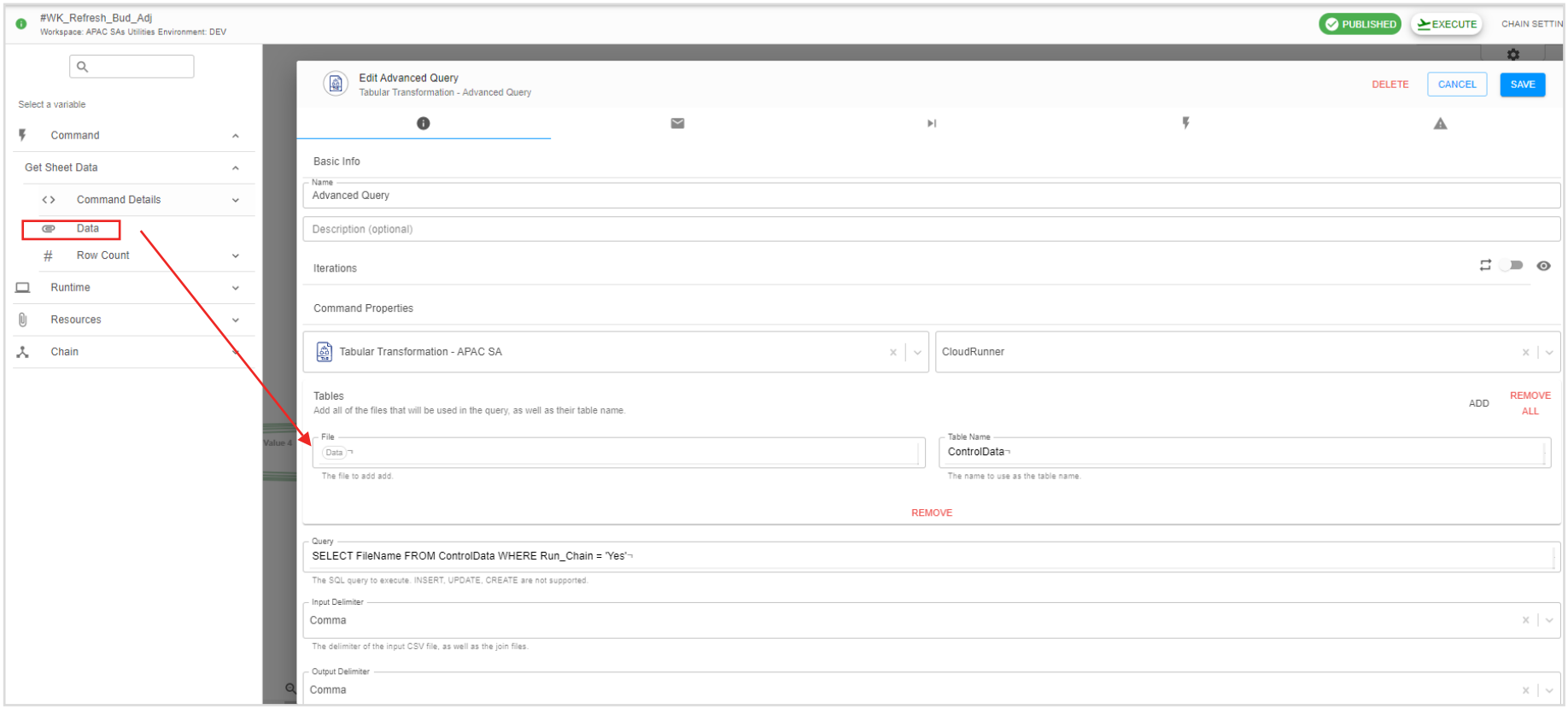


Naturally, the Spreadsheet ID, Sheet ID, Region would be inherited as the chain was copied, but check that they are correctly pointed to the variables defined in the 'Chain Settings' listed in the previous step

Workiva

Step 5: Adding an 'Advanced Query' after the 'Get Sheet Data'



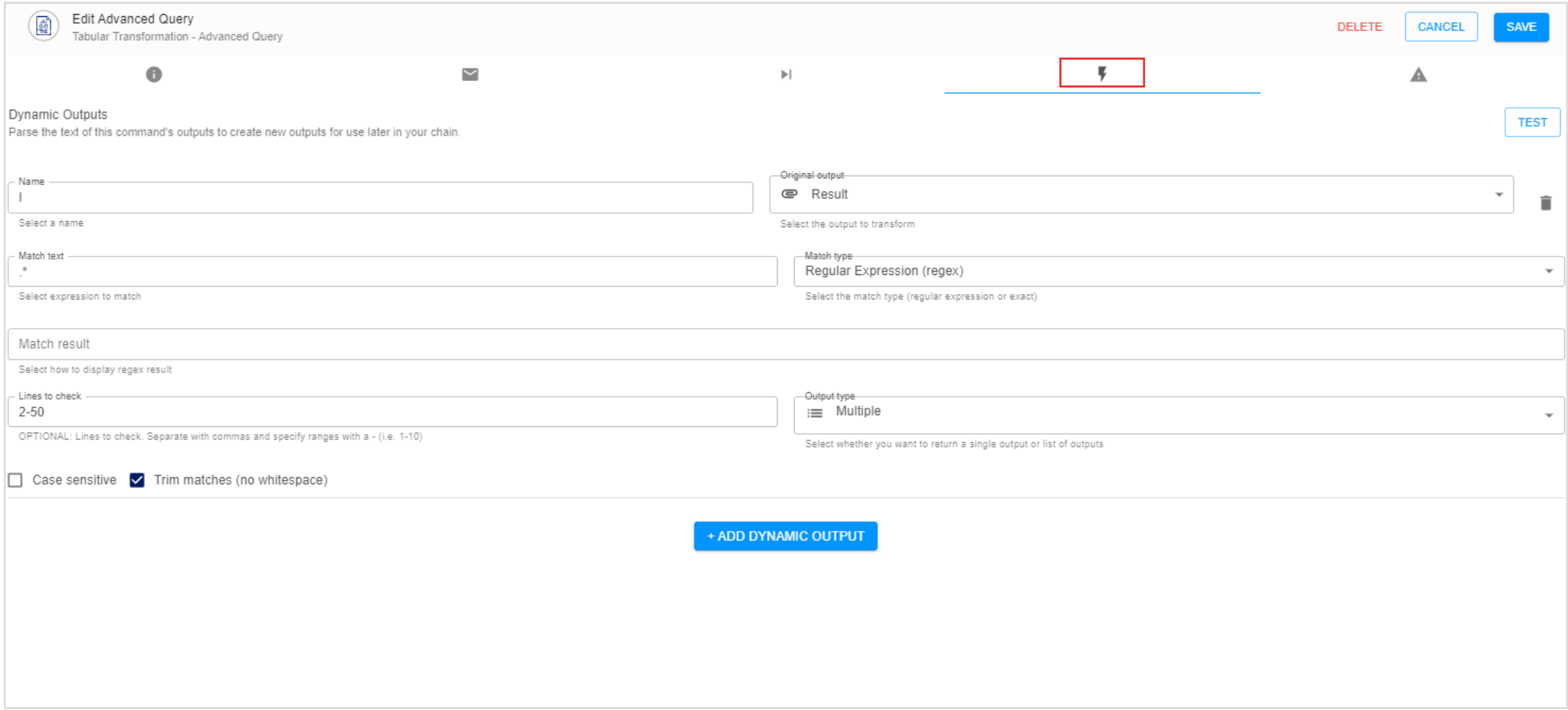


In edit of 'Advanced Query', edit the query to " SELECT FileName FROM ControlData WHERE Run_Chain = 'Yes' "

Select FileName from ControlsData where RUN_CHAIN = 'Yes'

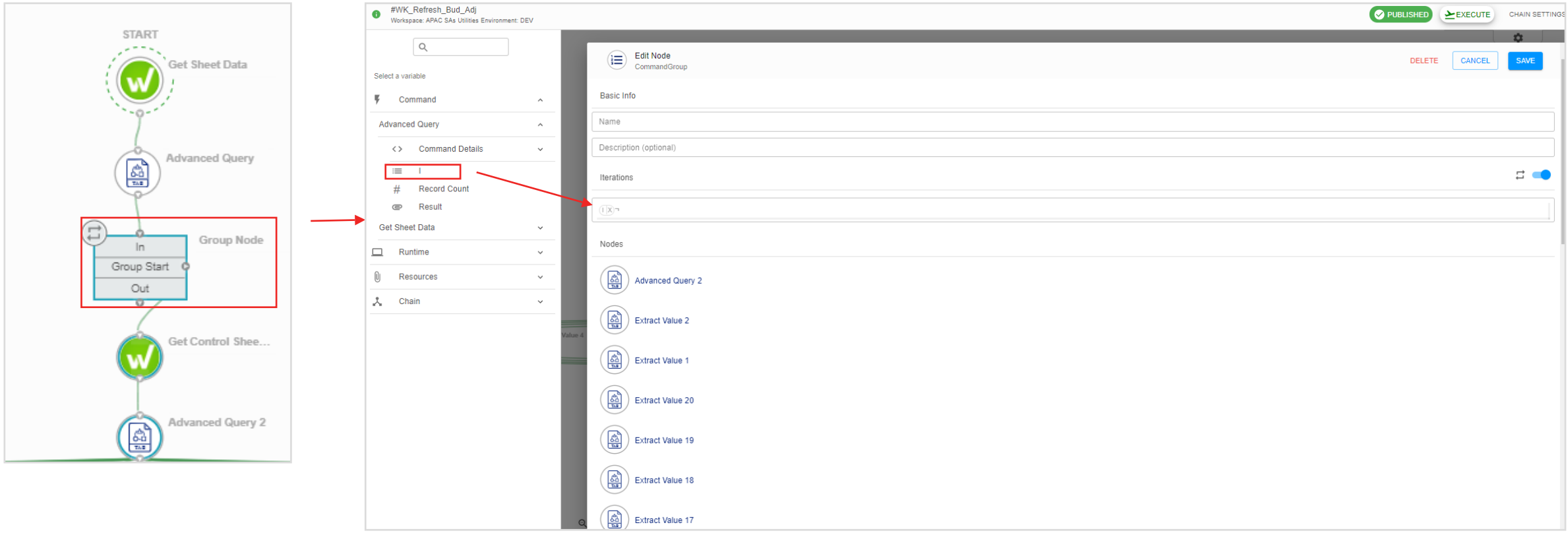
Workiva

Step 5.1: Iteration: Adding Dynamic Output



Click on the 'Lightning' icon indicated in the red box to access the 'Dynamic Outputs'. Populate the values as illustrated. The objective of populating the values here is to create an Iterator, which would be used in the Command Group (next step).

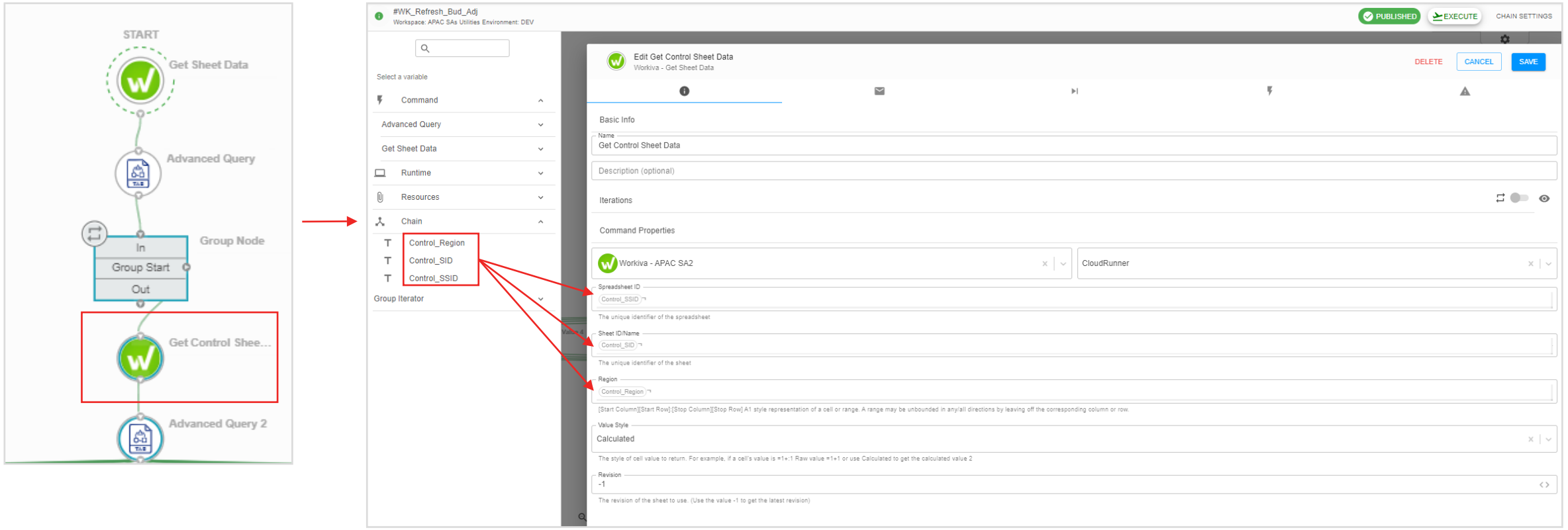
Step 6: Adding the 'Command Group' node



First, search 'Command Group' and join it after the 'Advanced Query' node. Then, in edit mode, enable 'Iterations' and input the 'I' Dynamic Output value that was defined in previous step

Workiva

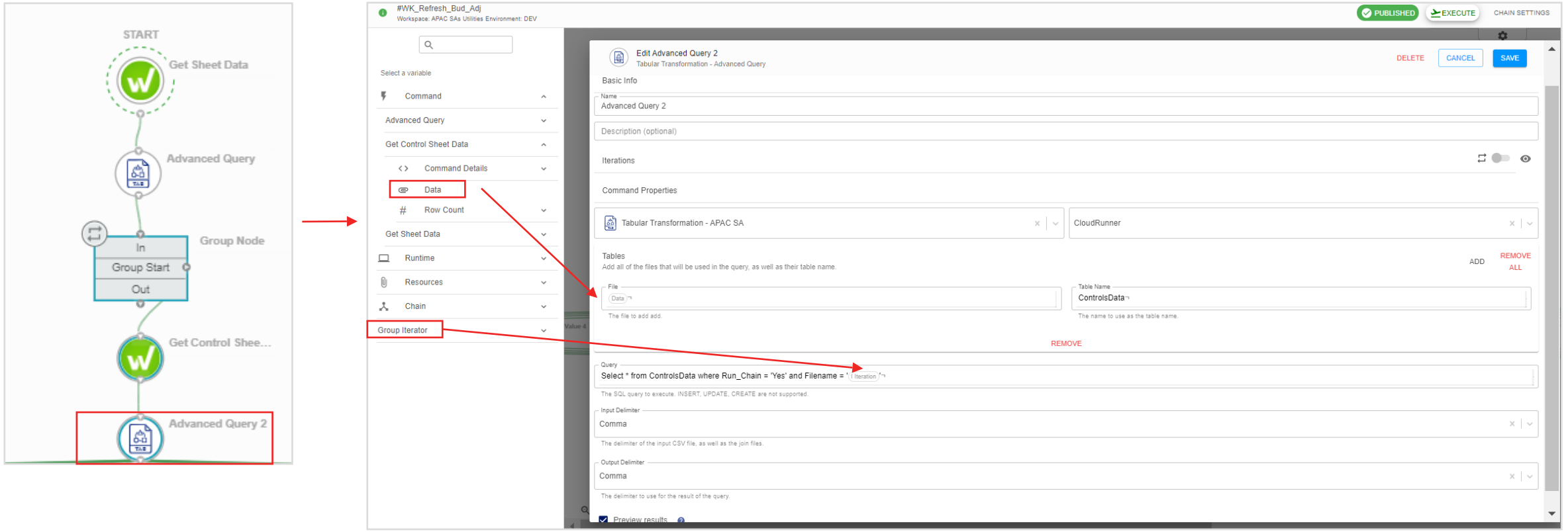
Step 7: Editing the second 'Get Sheet Data' node



The second 'Get Sheet Data' node is joined to the start of the 'Command Group' node. With the iteration inherited from previous step, this node will read the fields in the Control Sheet and extract the values accordingly.

Workiva

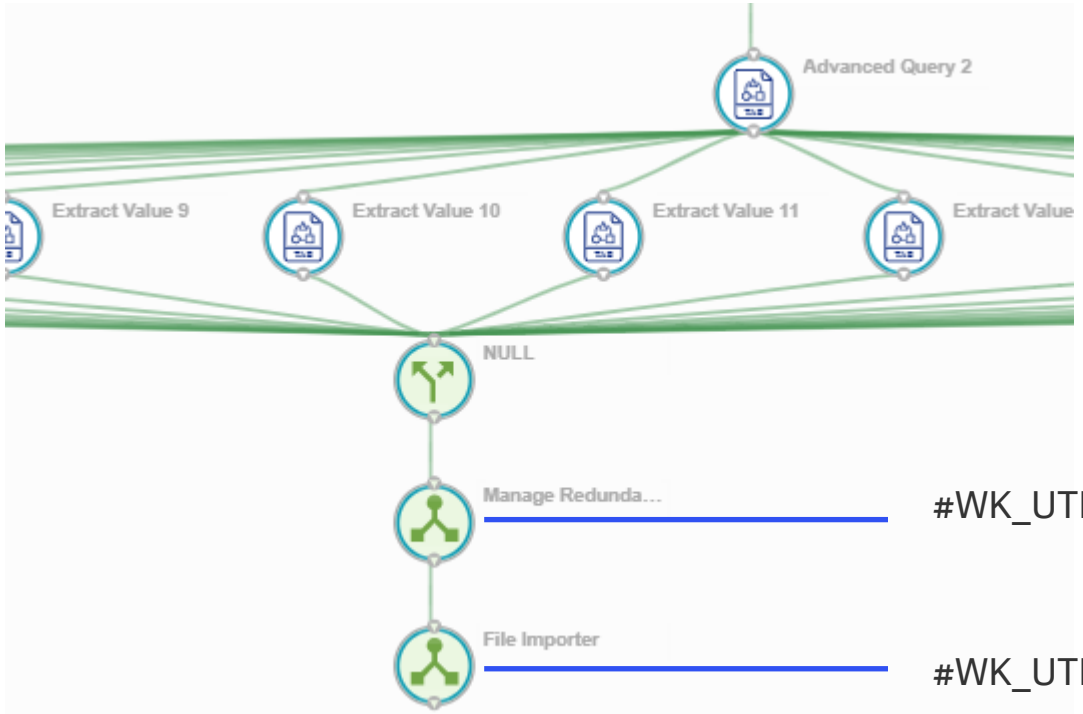
Step 8: Editing the second 'Advanced Query' node



The second 'Advanced Query' node extracts the Data from the second 'Get Sheet Data' node (effectively the control sheet values via the iterators). Here we input the query to identify Run_Chain = 'Yes' & the inherited Filename(s) from the Iterator.

```
Select * from ControlsData where Run_Chain = 'Yes' and Filename =
```

Step 9: Second Part of the Chain



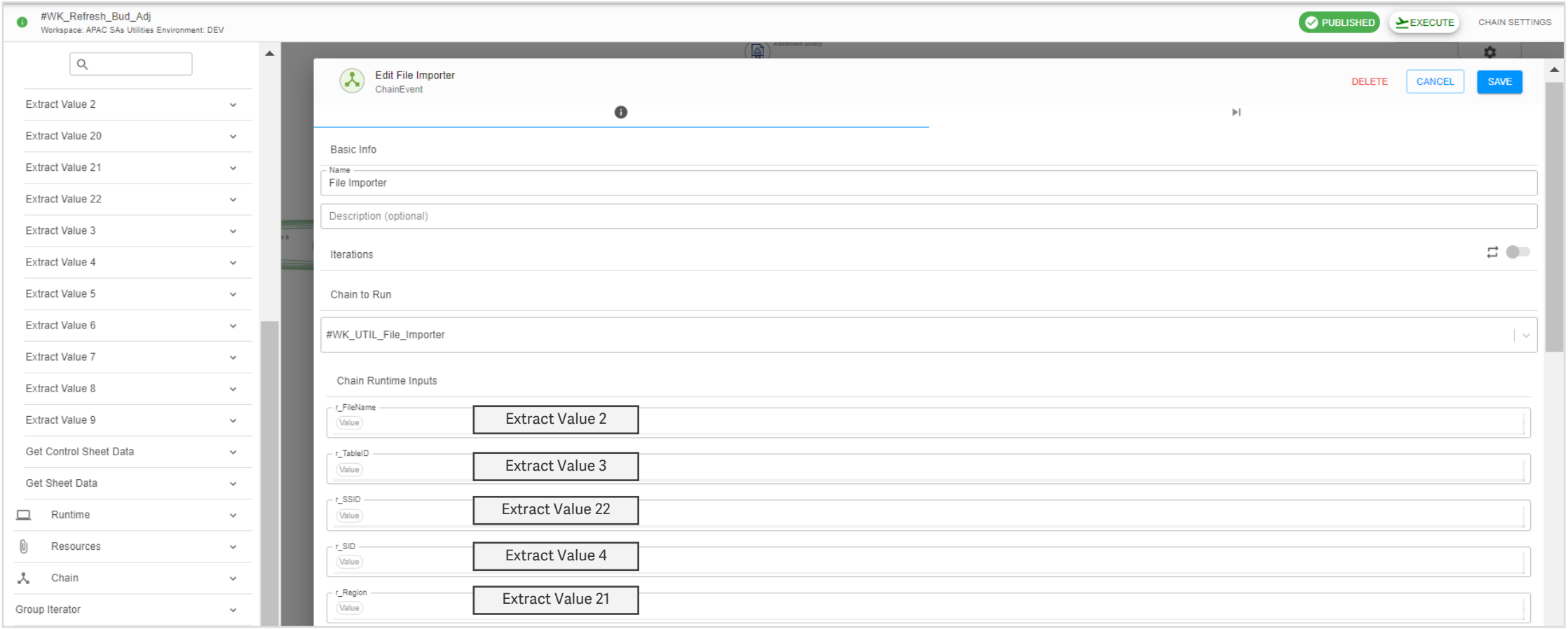
#WK_UTIL_Manage_Redundant_Files Chain

#WK_UTIL_File_Importer

At the bottom part of the the chain, we would be running two UTIL chains - '#WK_UTIL_Manage_Redundant_Files' & '#WK_UTIL_File_Importer'

Workiva

Step 9.1: Configuring the Runtime Inputs for File Importer

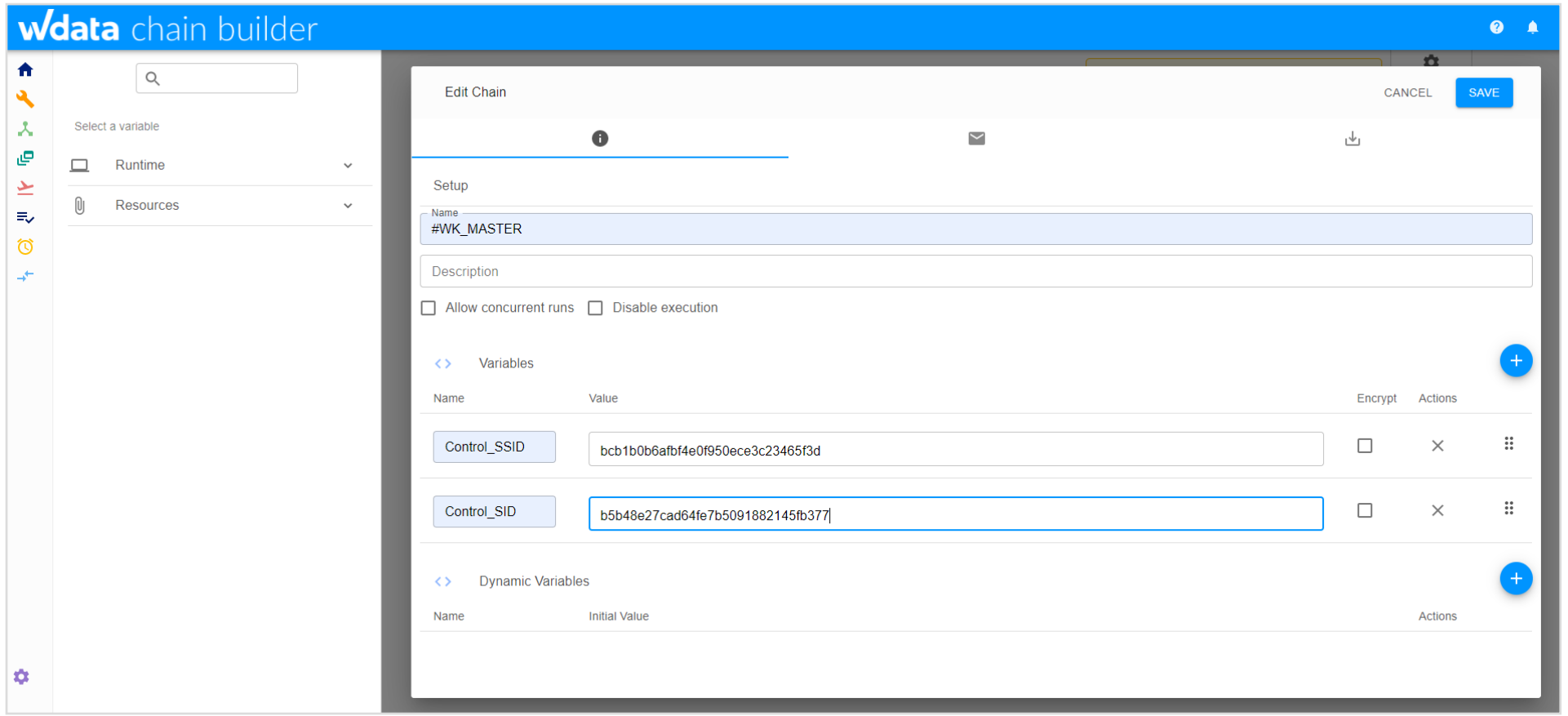


Input the Runtime input values, that is required to run the UTIL Chain - '#WL_UTIL_File_Importer'; mainly the "FileName, TableID, SSID, SID, Region"

Workiva

Chain 6: Master Chain

Step 1: Create New Chain



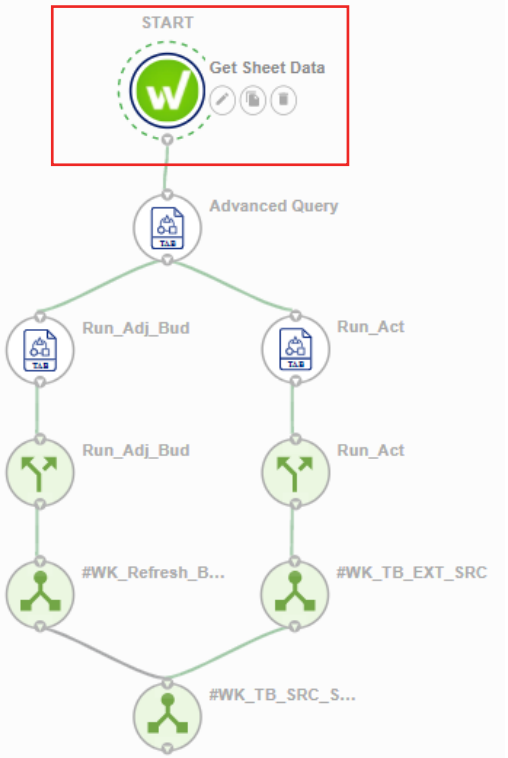
Create a New Chain (not from template). Name is "#WK_MASTER", and input the following variables:

i. Control_SSID - Master Control's spreadsheet ID

ii. Control_SID - Master Control's sheet ID

Workiva

Step 2: Get Sheet Data



#WK_MASTER
Workspace: APAC SAs Utilities Environment: DEV

PUBLISHED | EXECUTE | CHAIN SETTINGS

Select a variable

Runtime

Resources

Chain

Control_SSID

Control_SID

Edit Get Sheet Data
Workiva - Get Sheet Data

DELETE | CANCEL | SAVE

Get Sheet Data

Description (optional)

Iterations

Command Properties

Workiva - APAC SA2

CloudRunner

Spreadsheet ID
Control_SSID
The unique identifier of the spreadsheet

Sheet ID/Name
Control_SID
The unique identifier of the sheet

Region
A1:F2
[Start Column][Start Row]:[Stop Column][Stop Row] A1 style representation of a cell or range. A range may be unbounded in any/all directions by leaving off the corresponding column or row.

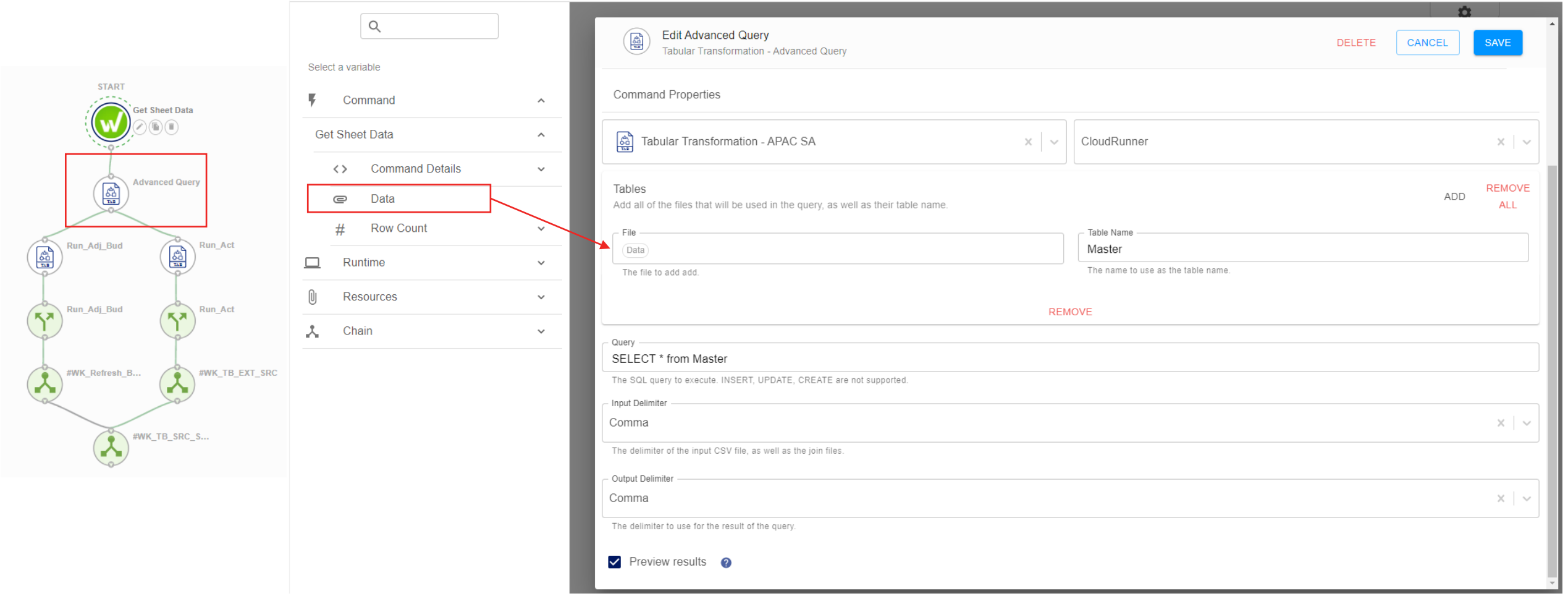
Value Style
Calculated
The style of cell value to return. For example, if a cell's value is =1+:1 Raw value =1+1 or use Calculated to get the calculated value 2

Revision
-1
The revision of the sheet to use. (Use the value -1 to get the latest revision)

Start with a 'Get Sheet Data', input the Spreadsheet & Sheet ID from the Chain Variables

workiva

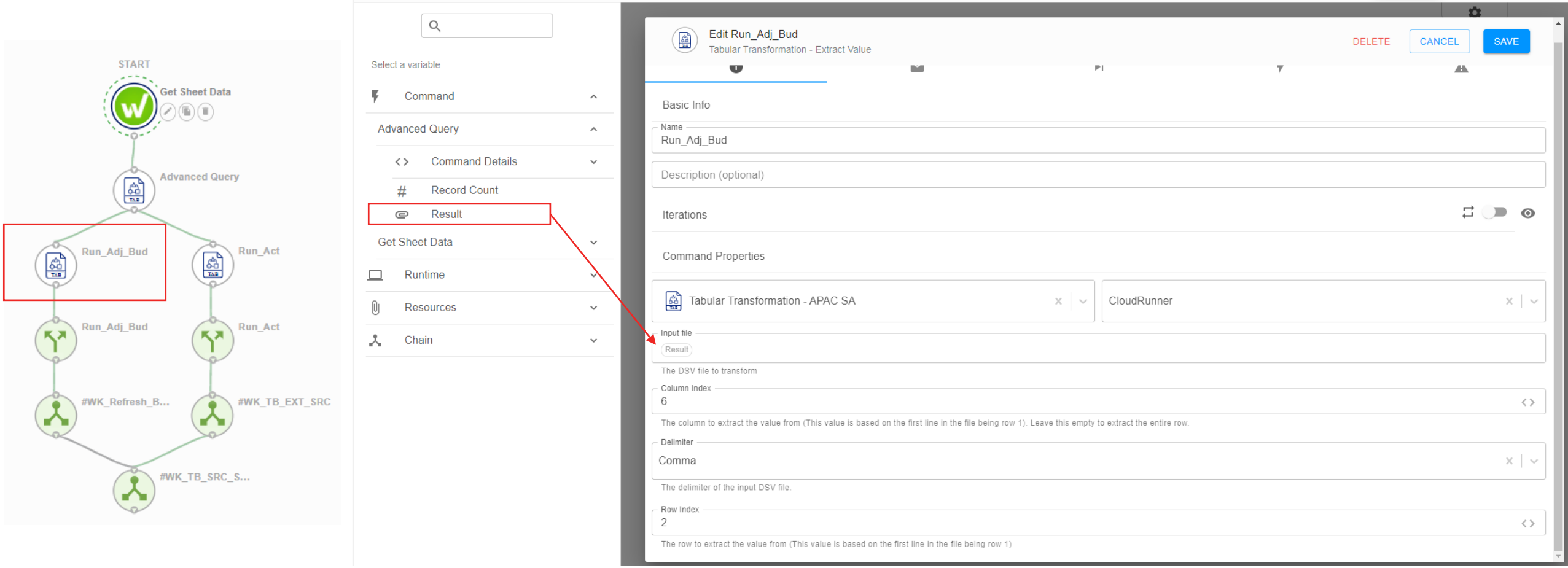
Step 3: Advanced Query



In 'Advance Query', input the "Data" file from the previous 'Get Sheet Data' node. Under 'Query', input: "**SELECT * from Master**"

Workiva

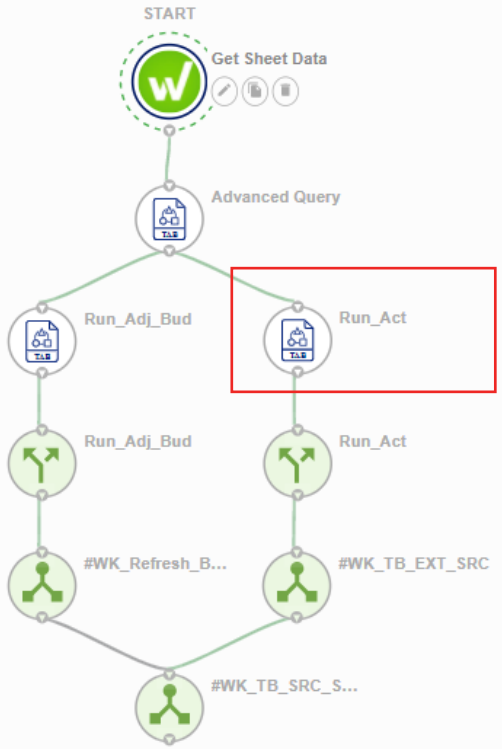
Step 4: Extract Value Run_Adj_Bud

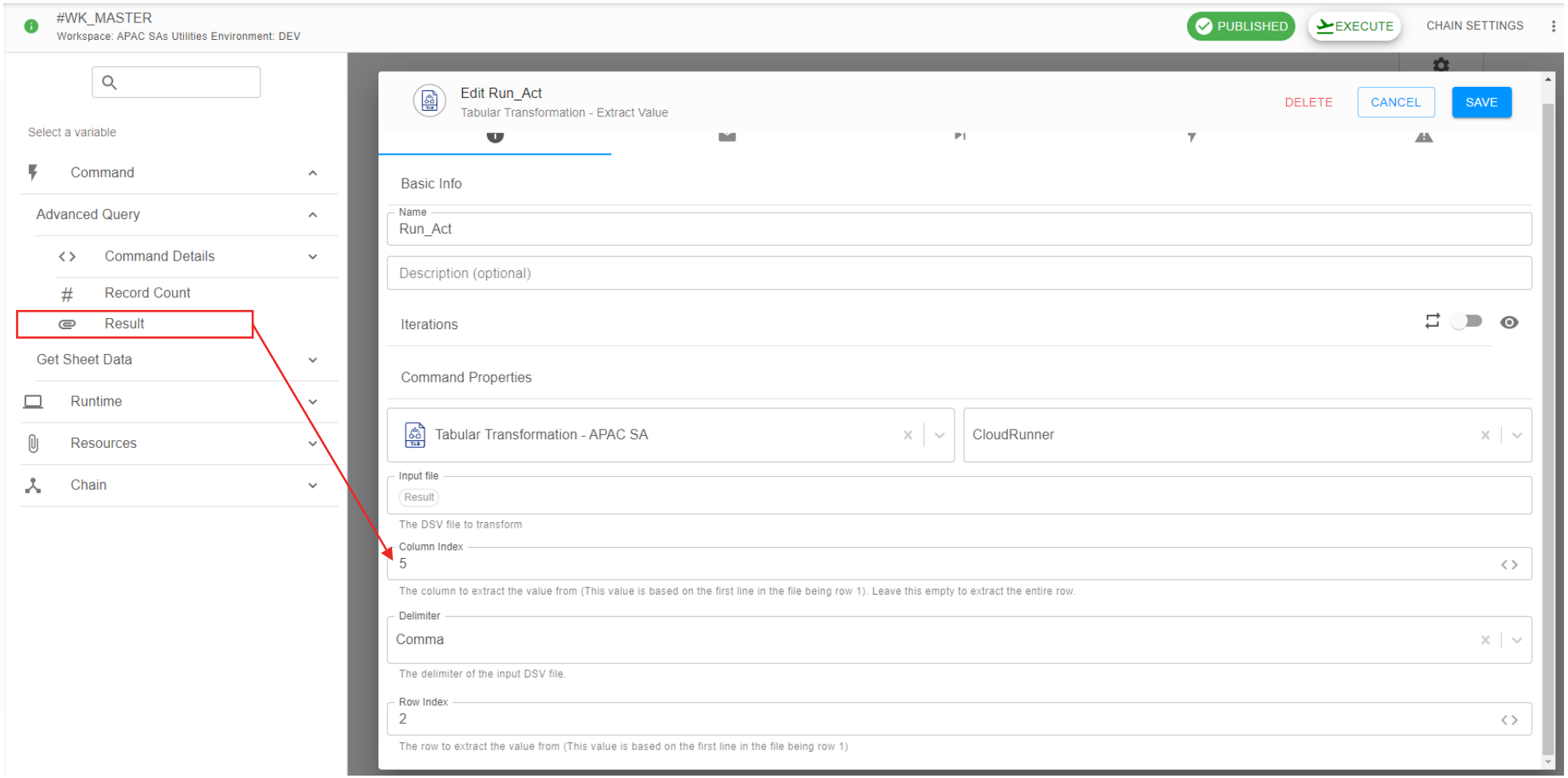


In 'Extract Value', input the "Result" file from the previous 'Advance Query' node. Input "6" / "2" under 'Column Index / 'Row Index'

Workiva

Step 5: Extract Value Run_Act

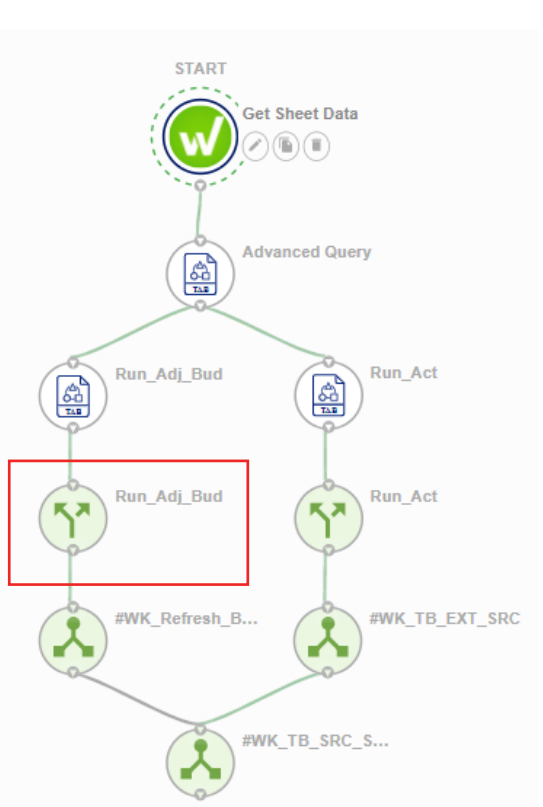


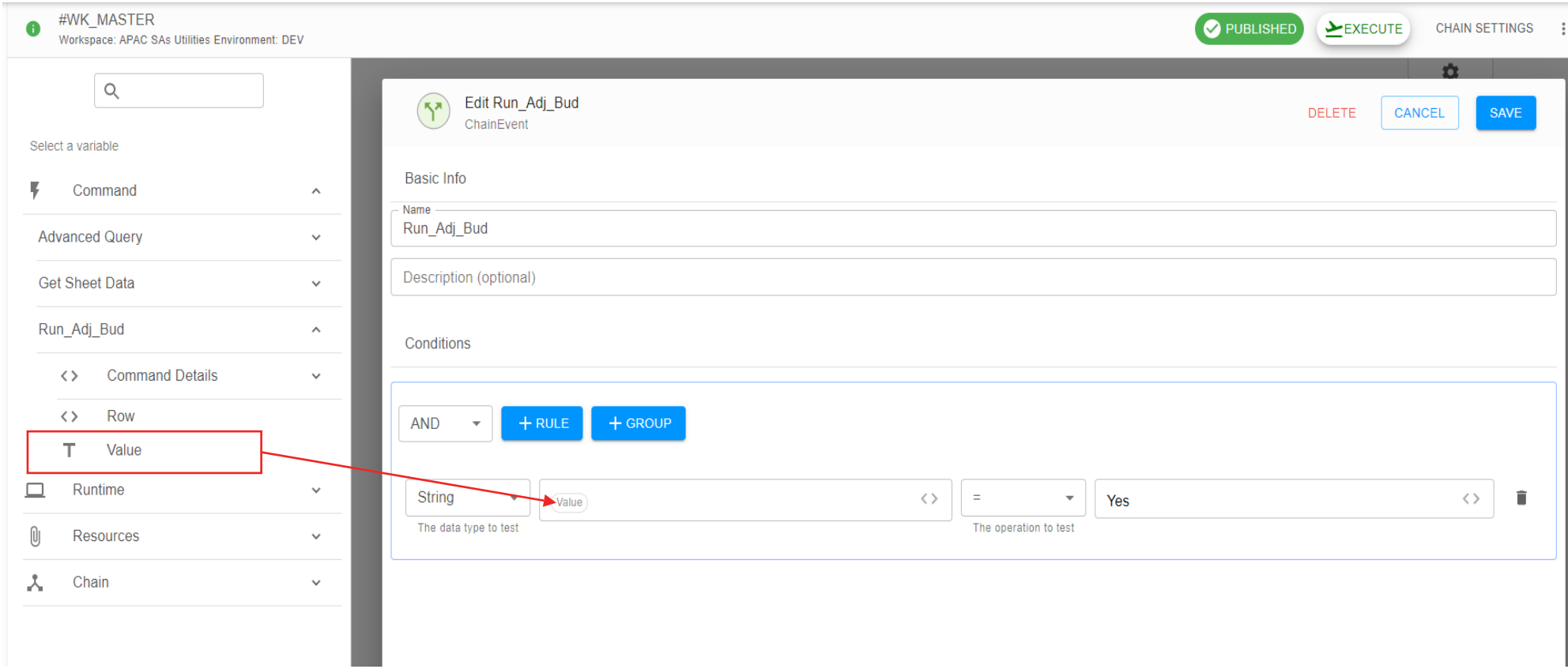


In 'Extract Value', input the "Result" file from the previous 'Advance Query' node. Input "5" / "2" under 'Column Index / 'Row Index'

Workiva

Step 6: Conditional Run_Adj_Bud

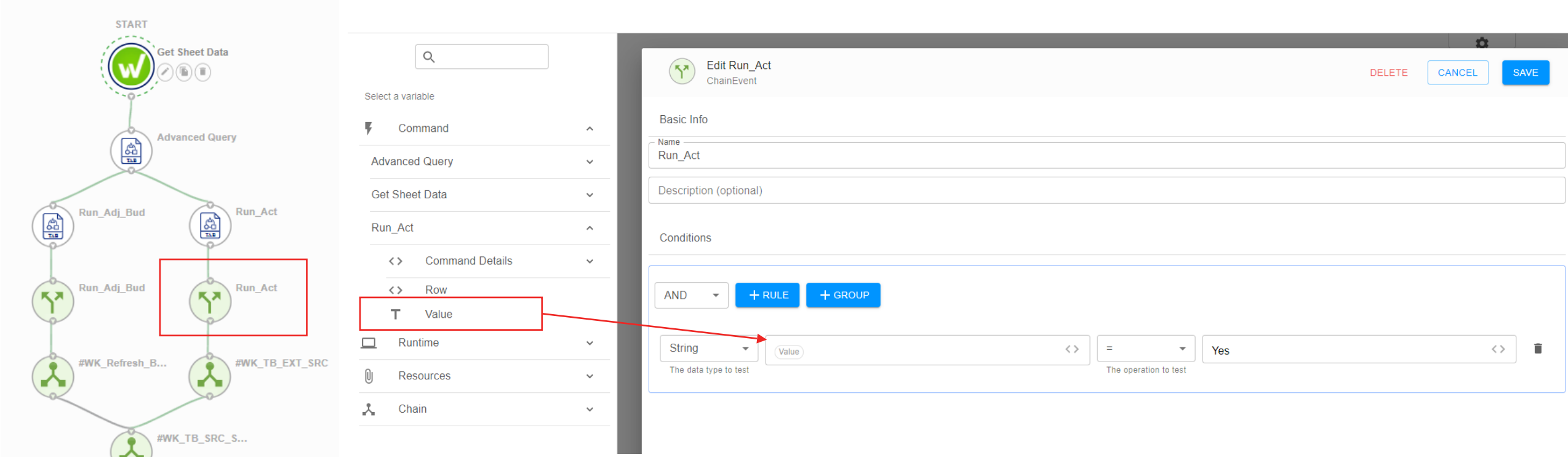




In 'Conditions', input the "Value" from the previous 'Extract Value' node. Modify the Condition to "=" to Yes (value in the Run_Adj_Bud columns in the Master Control Sheet).

Workiva

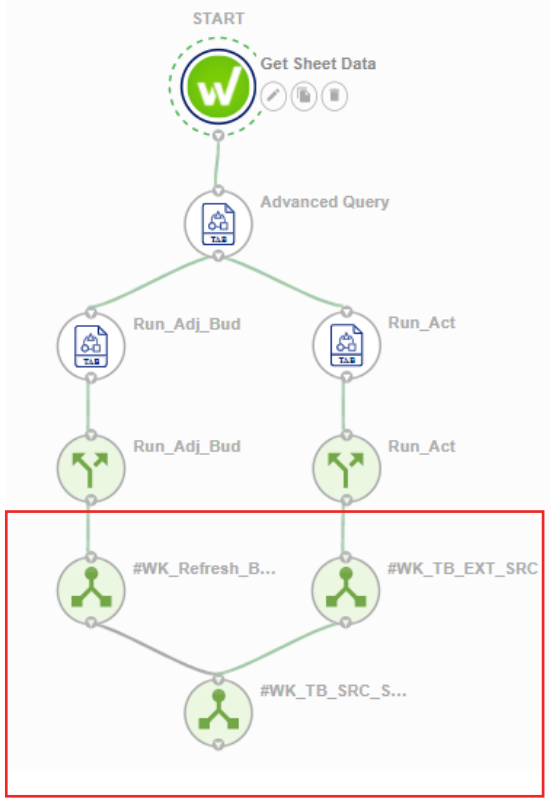
Step 7: Conditional Run_Act

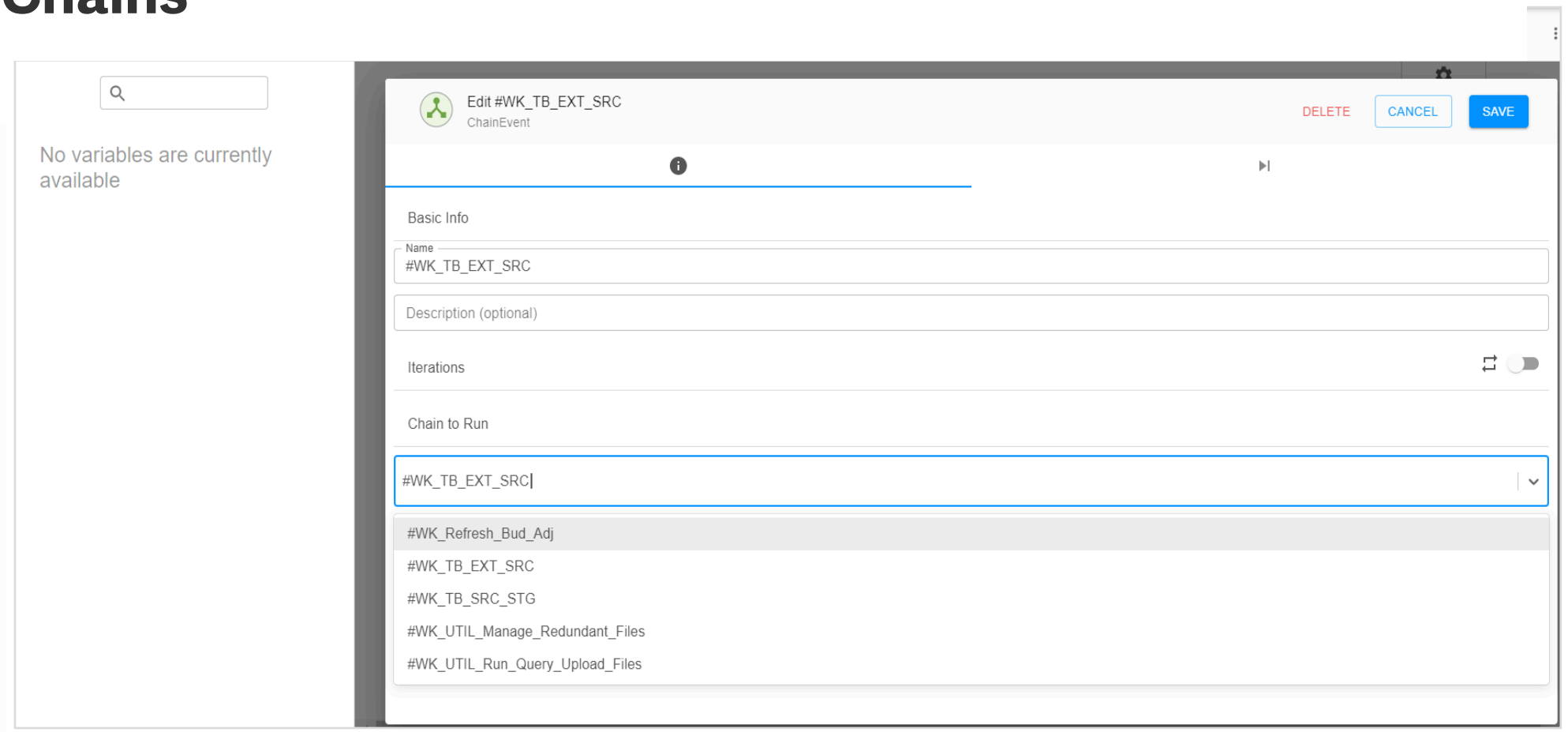


In 'Conditions', input the "Value" from the previous 'Extract Value' node. Modify the Condition to "=" to Yes (value in the Run_Act columns in the Master Control Sheet).

workiva

Step 8: Run Chains



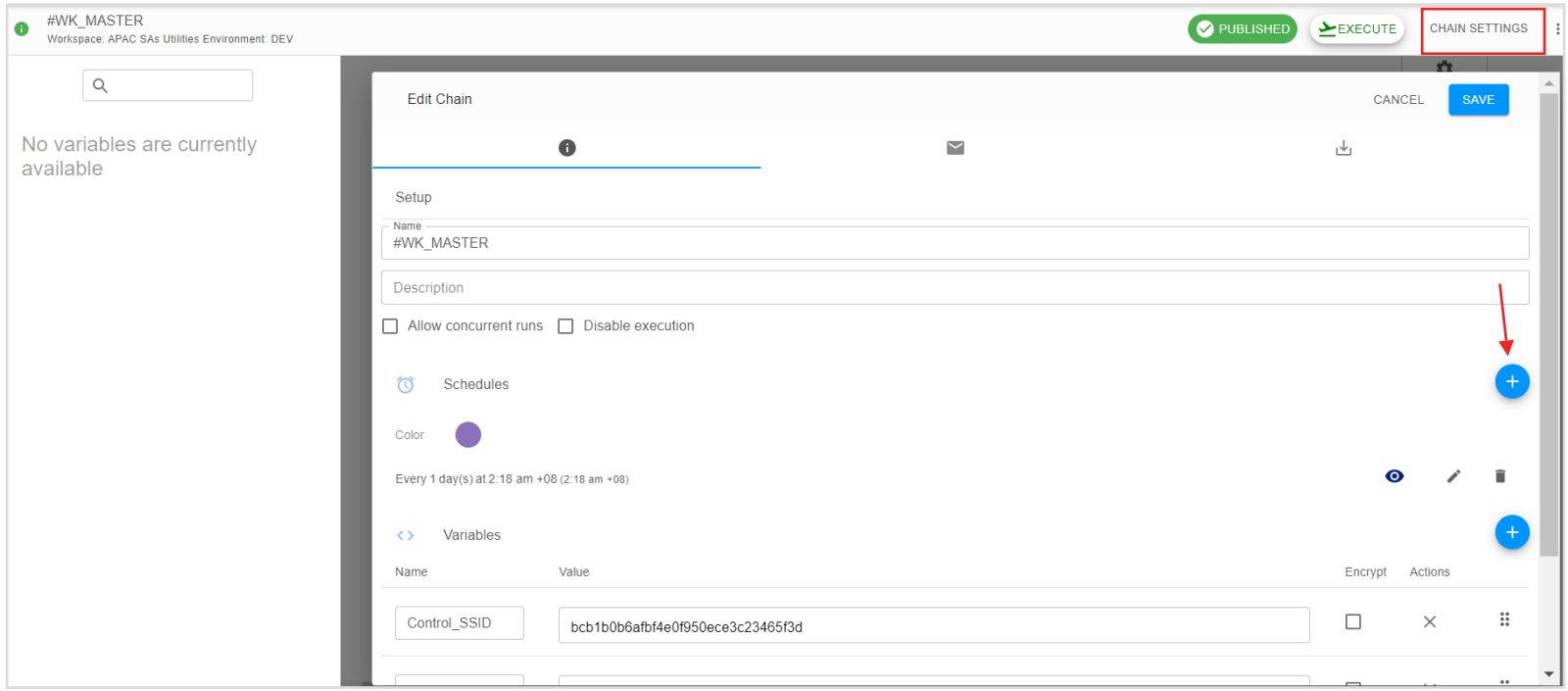


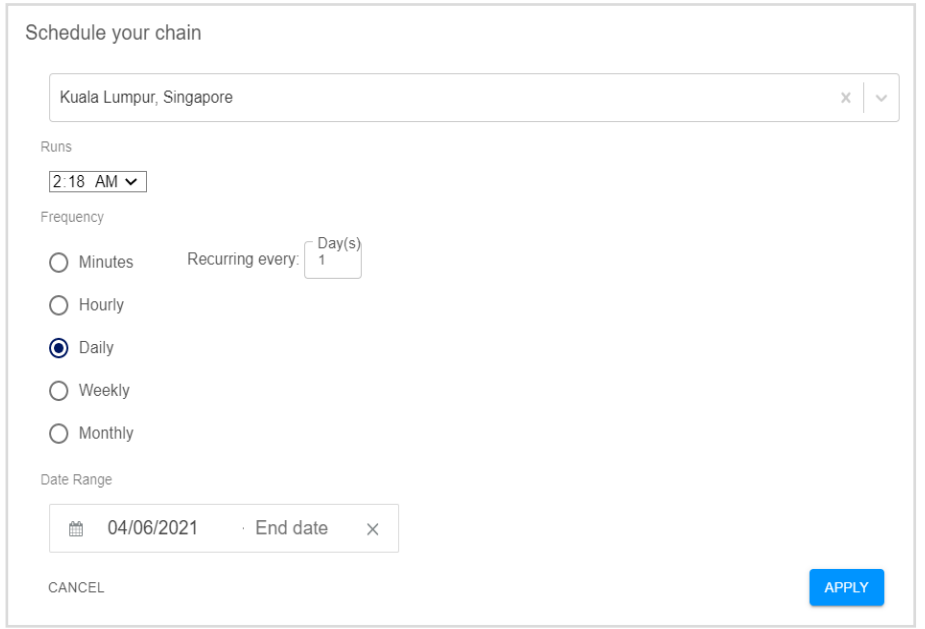
Place the listed chain in the Run Chain nodes following the order of the image to the left of the slide

i. #WK_TB_EXT_SRC
ii. #WK_Refresh_Bud_Adj
iii. #WK_TB_SRC_STG

Workiva

Step 9: Schedule Timer

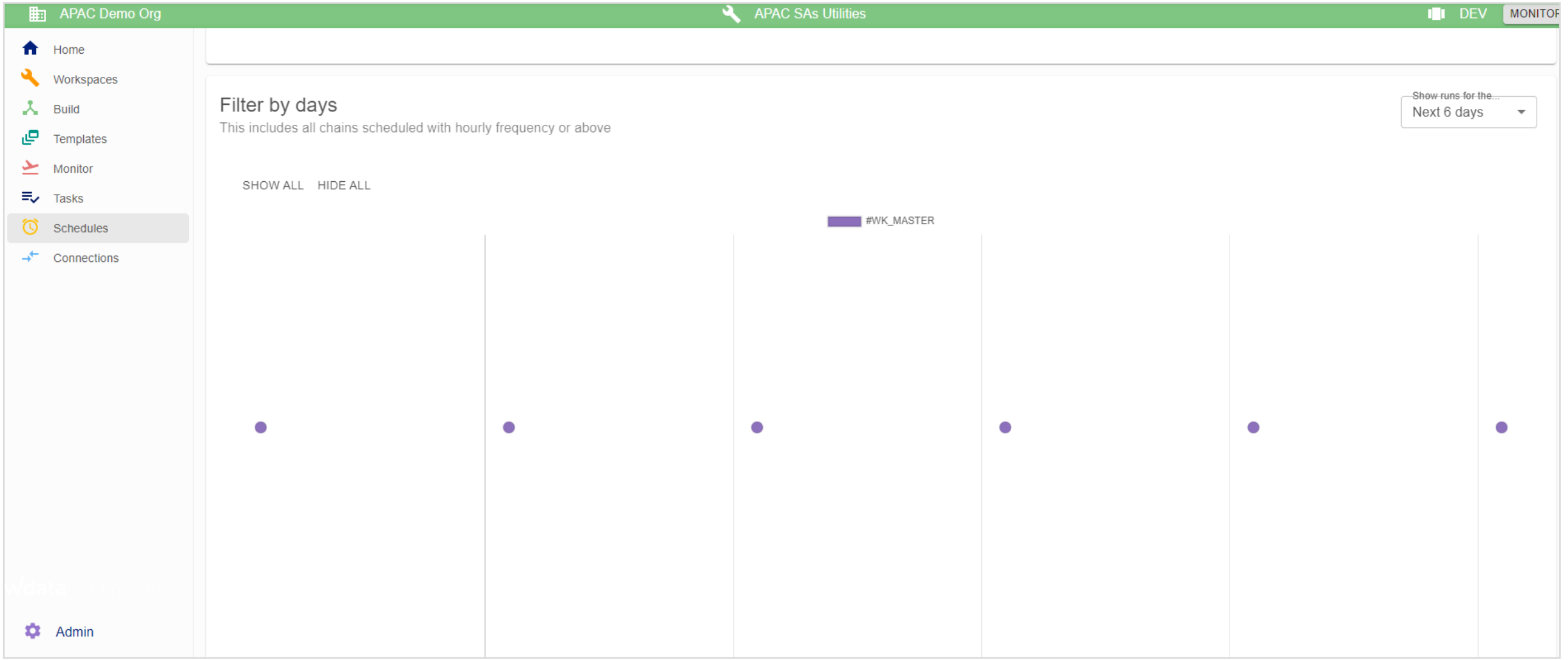




Access 'Chain Settings', add a 'Schedule', set it to run at 0200 Daily. This schedule job would run the Master Chain Daily at 2am.

Workiva

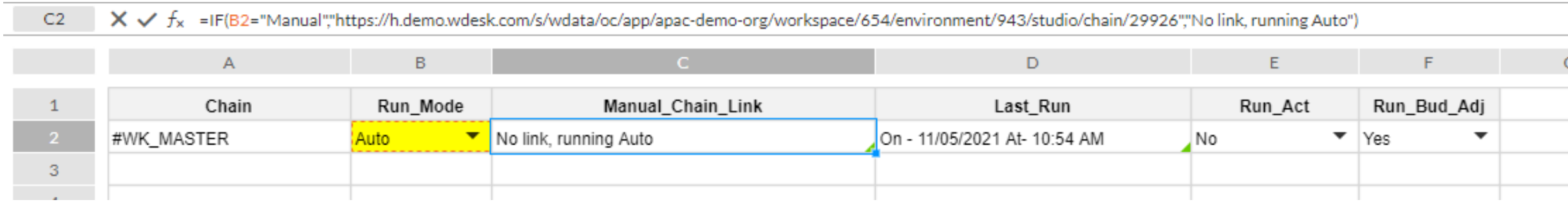
Step 9.1: Schedule for Timer Batch Job



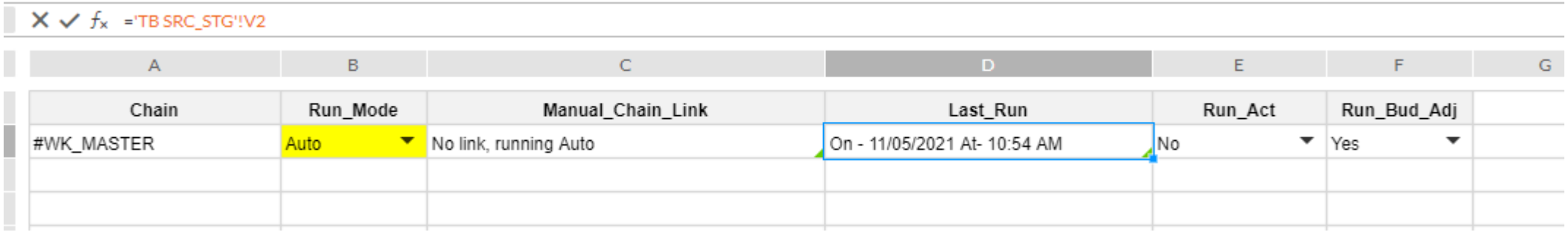
In the 'Schedules' you would be able to see the schedules of the timer batch jobs that would run daily.

Workiva

Step 10: Master Controls Sheet



Create a 'Master' control sheet, with the following columns. Set 'Run_Mode' to "Auto" and input the formula in Cell C2: "=IF(B2="Manual","{URL of the Master Chain}","No link, running Auto")"



In Cell D2, link it to Cell V2 from the 'SRC_STG' Controls Sheet.

workiva