COMPLETE END TO END PROJECT OF MATILLION INVOLVING AWS S3 & SNS, SNOWFLAKE, POWER BI AND GIT.

1. What is matillion?

Matillion ETL is a **cloud-native ETL solution** that simplifies the process of collecting, preparing, and loading data. It's a drag and drop interface which helps build scalable data pipelines without the need for extensive coding.

DID YOU KNOW: Matillion named a **Challenger** in the **2023 Gartner®Magic Quadrant™ report** for Data Integration Tools.

Figure 1: Magic Quadrant for Data Integration Tools

2. What is AWS S3?

Simply say, it's a cloud based storage service.

3. What is snowflake?

Like matillion, snowflake is built for the cloud. It's a cloud-hosted, relational database that enables you to build data warehouses on demand

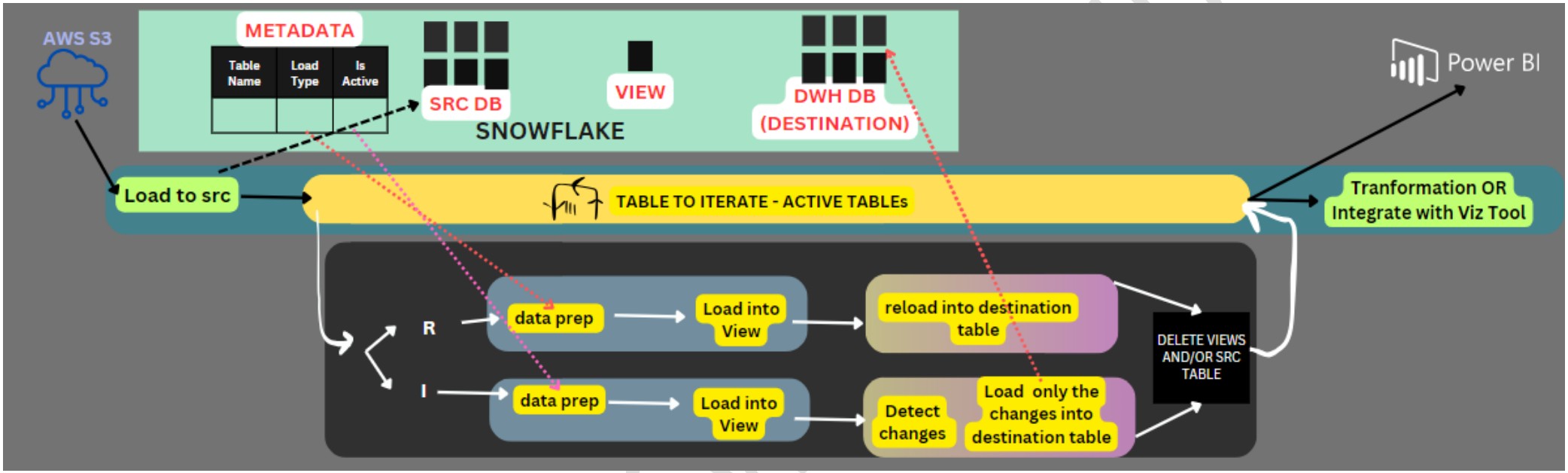
4. What is always- on tables?

The data in table is available always.

Why is it a best practice?

- Reload can take a long time for huge tables
- When reloading, only partial data will be available to integrated tools, hence incorrect results

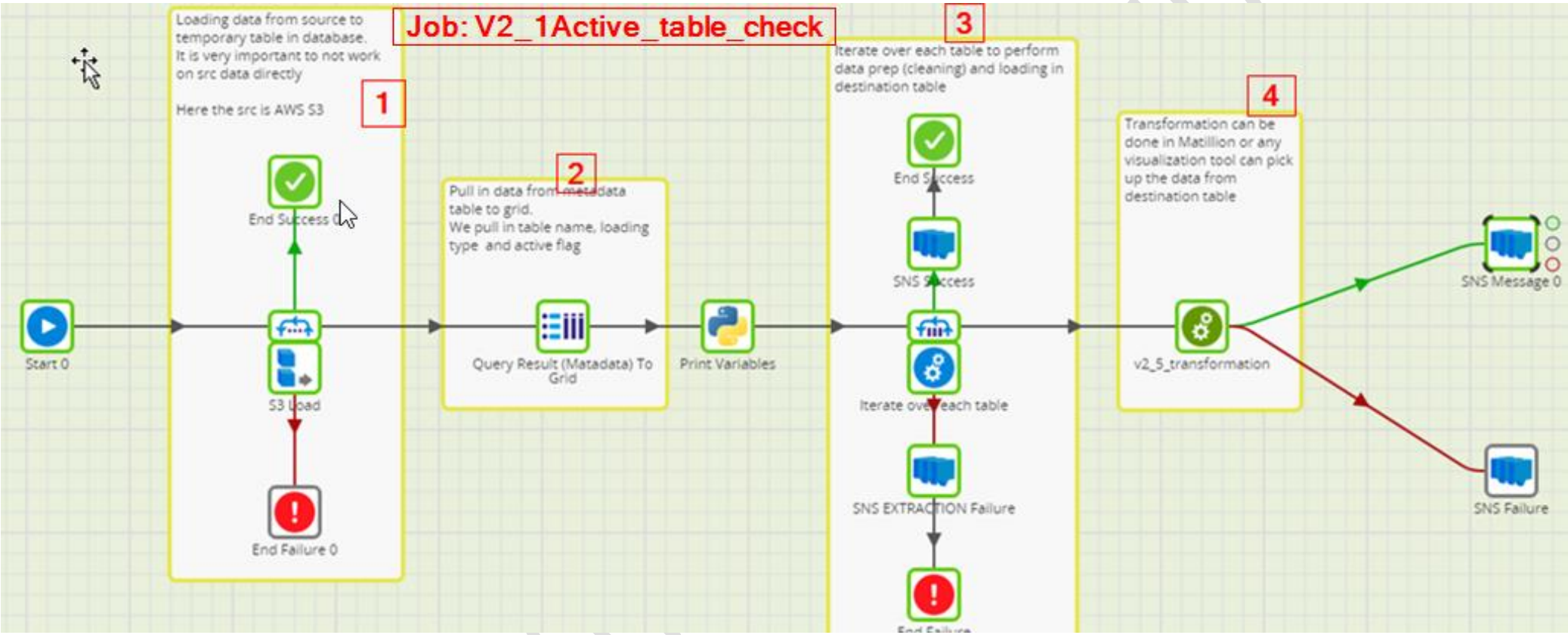
5. Architecture of today's project:



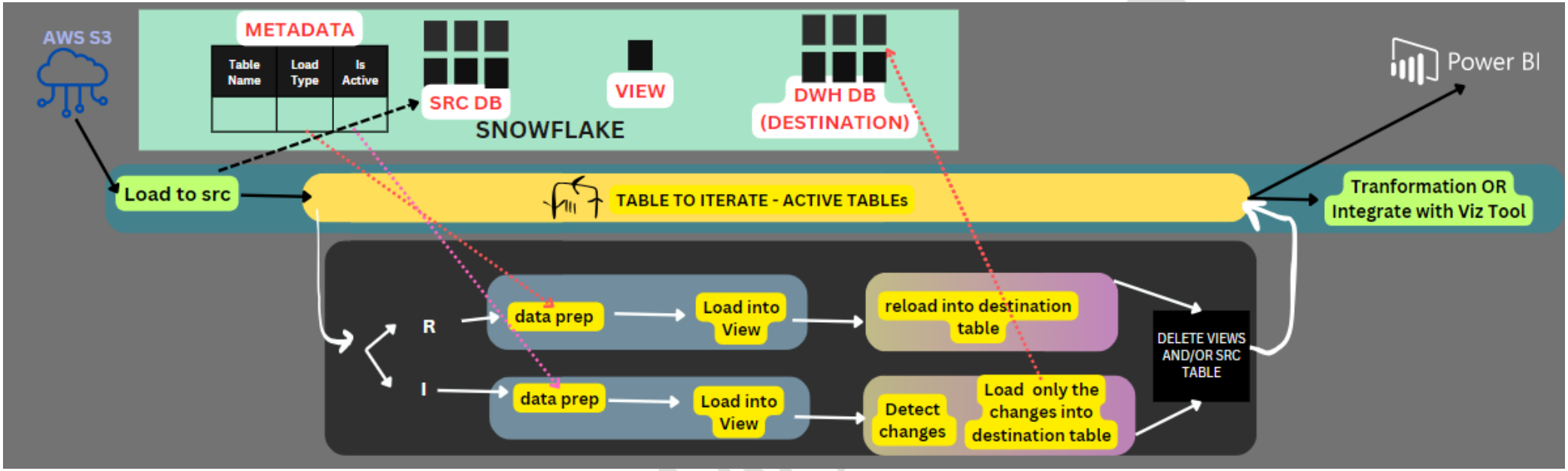
I. Prep:

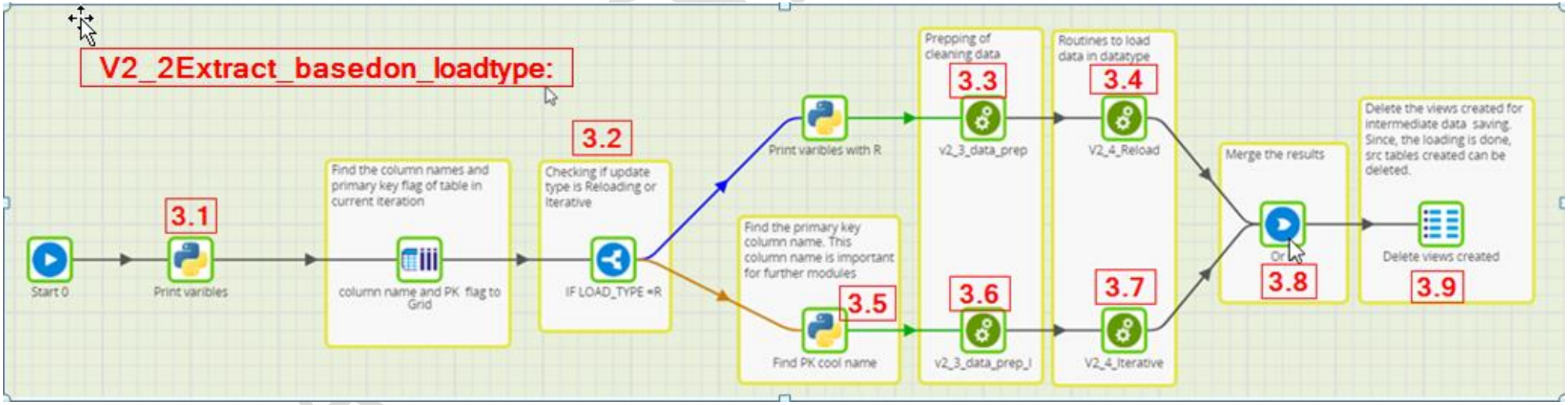
- An active AWS and Snowflake account
- Create an matillion account (ETL setup) – refer doc: Steps To Configure Matillion.docx

II. Gist of Working:

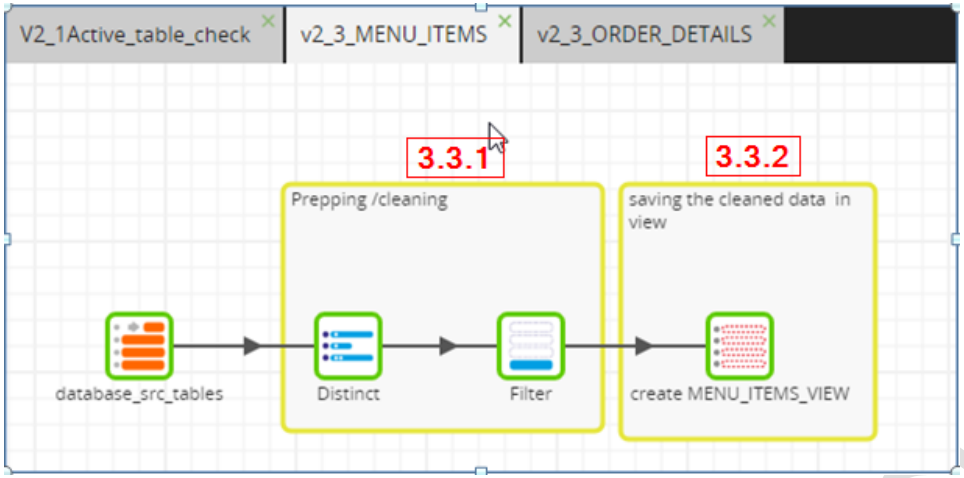


1. **Load the data from S3 to Snowflake src table [S3 LOAD]**
2. **Pull in info from metadata table into the GRID Variable [QUERY RESULT TO GRID]**
3. **Iterate all the next steps for each table [FIXED ITEARATOR + ORCHESTRATION JOB] + Notification [SNS]**
4.

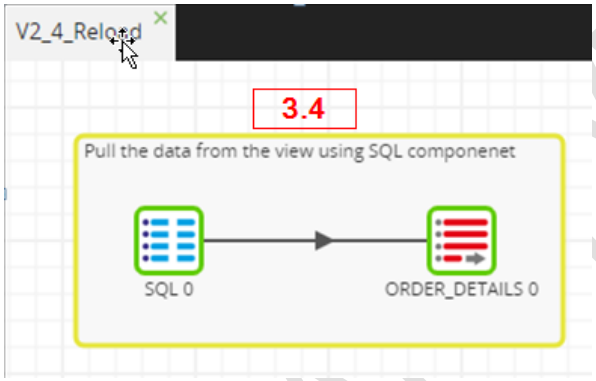

AWS S3
METADATA
Table Name
Load Type
Is Active
SRC DB
VIEW
SNOWFLAKE
DWH DB (DESTINATION)
Power BI
Load to src
TABLE TO ITERATE - ACTIVE TABLES
Tranformation OR Integrate with Viz Tool
R
data prep
Load into View
reload into destination table
I
data prep
Load into View
Detect changes
Load only the changes into destination table
DELETE VIEWS AND/OR SRC TABLE


V2_2Extract_basedon_loadtype:
Start 0
3.1
Print varibles
Find the column names and primary key flag of table in current iteration
column name and PK flag to Grid
3.2
Checking if update type is Reloading or Iterative
IF LOAD_TYPE =R
Print varibles with R
Find the primary key column name. This column name is important for further modules
3.5
Find PK cool name
Prepping of cleaning data
3.3
v2_3_data_prep
3.6
v2_3_data_prep_I
Routines to load data in datatype
3.4
V2_4_Reload
3.7
V2_4_Iterative
Merge the results
Or
3.8
Delete the views created for intermediate data saving. Since, the loading is done, src tables created can be deleted.
Delete views created
3.9

4.1. Retrieve the column name and primary key of the table [table metadata to grid]
4.2. Check if the table needs to be iterated or reloaded [IF component]
4.3. For R: Prepare /clean the data [Run transformation component]

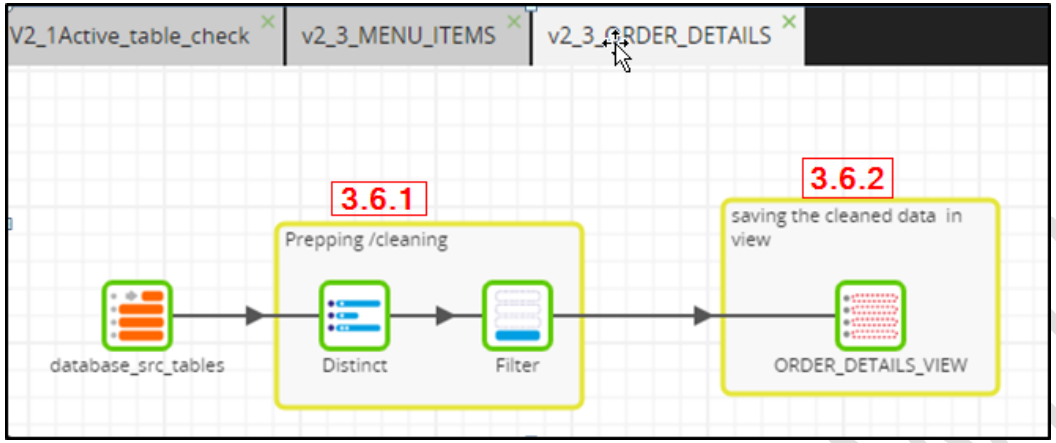


4.3.1. Cleaning [DISTINCT + FILTER NULLS]
4.3.2. Save it in a view [CREATE VIEW]

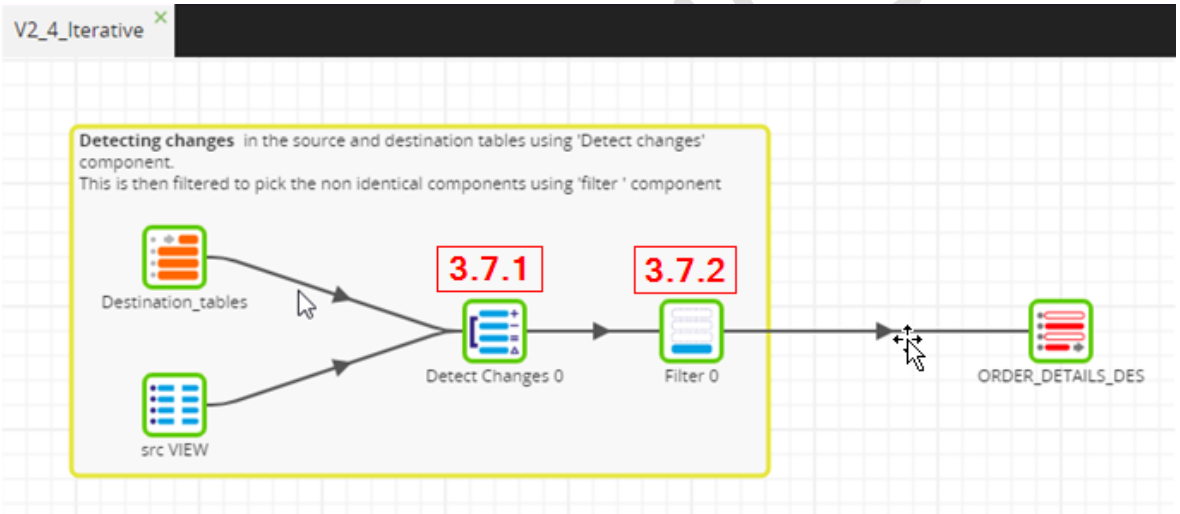


4.4. For R : Load the data from the VIEW into the destination database [RUN transformation component]
4.5. For I: Identify the name of the primary key using the flag [Python script]

4.6. For I: Prepare /clean the data [Run transformation component]



4.6.1. Cleaning [DISTINCT + FILTER NULLS]
4.6.2. Save it in a view [CREATE VIEW]



4.7. For I: Load the data from the VIEW into the destination database [RUN transformation component]

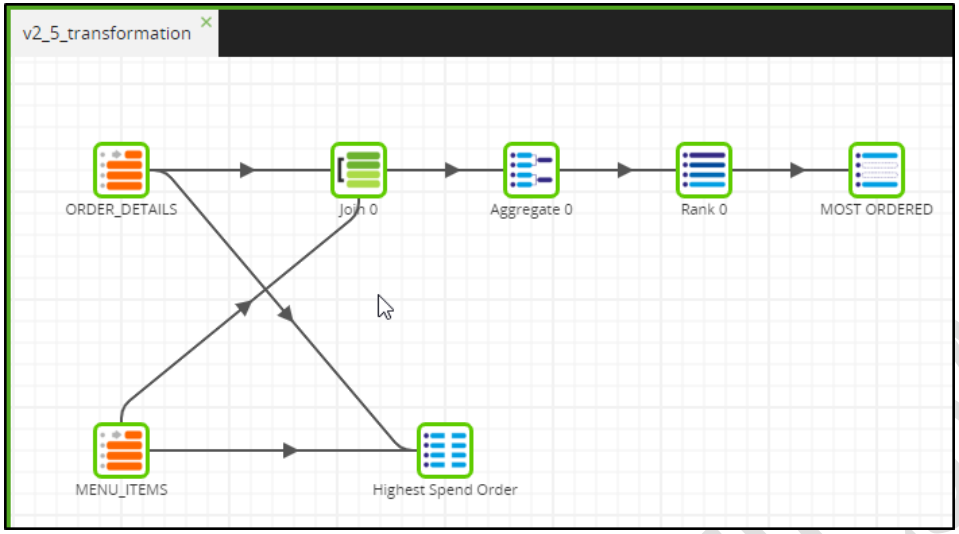
4.7.1. Detect changes [Detect changes]

4.7.2. Filter rows that need to be Upserted [filter]

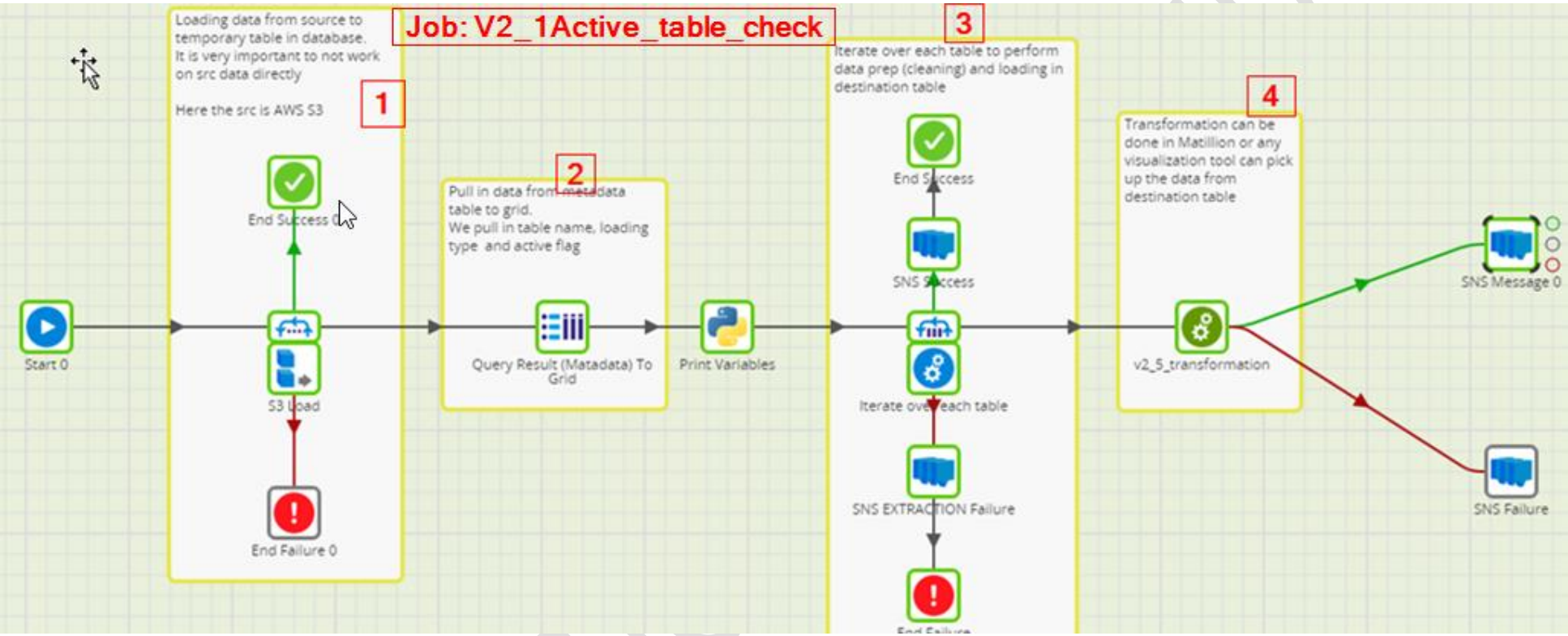
4.8. Irrespective of R or I , the pipeline needs to be merged [OR]

4.9. Since the cleaning and loading is completed, Unnecessary data/ structure is removed [SQL SCRIPT]

5. You can perform transformations to retrieve value /KPI OR use the data as input to any visualization tool. [RUN transformation component]



III. DETAILED:



Create an orchestration job V2_1Active_table_check

1. Prep:

- First Create following Job Variables:

Manage Job Variables

Name	Type	Behaviour	Visibility	Value	Description
dictionary_name	Text	Shared	Public	Mat_Tableinput	
file_name	Text	Shared	Public	MENU_ITEMS	
JV_DB	Text	Shared	Public	RESTAURANTSALES	
JV_LOADTYPE	Text	Shared	Public	I	
JV_SCHEMA	Text	Shared	Public	RESTAURANT_DB	
JV_TABLE_NAME	Text	Shared	Public	ORDER_DETAILS	

Text Mode:

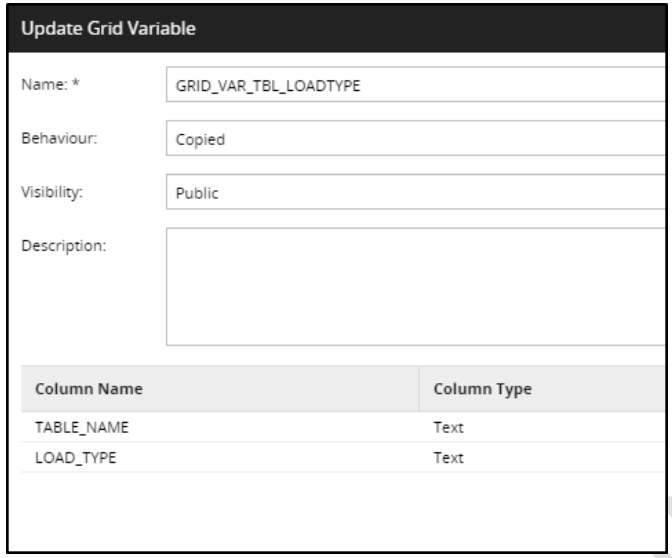
dictionary_name Text Shared Public Mat_Tableinput
file_name Text Shared Public MENU_ITEMS
JV_DB Text Shared Public RESTAURANTSALES
JV_LOADTYPE Text Shared Public I
JV_SCHEMA Text Shared Public RESTAURANT_DB
JV_TABLE_NAME Text Shared Public ORDER_DETAILS

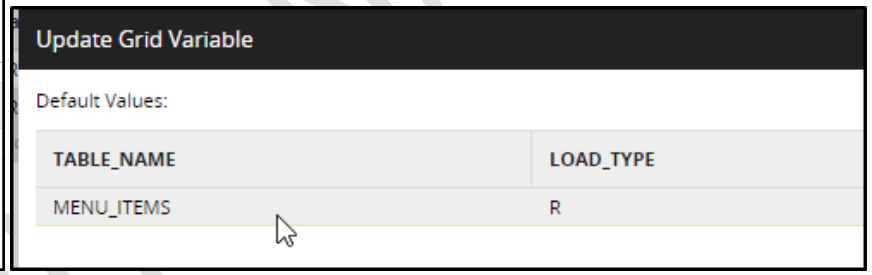
- Create Grid variables:

Manage Grid Variables

Name	Behaviour	Visibility	Columns	Values	
GRID_INTIAL_TABLE_NAME	Shared	Public			
GRID_VAR_TBL_LOADTYPE	Copied	Public			

GRID_VAR_TBL_LOADTYPE





TEXT Mode:

TABLE_NAME Text

LOAD_TYPE Text

- Configuring SNS
  - AWS SNS > Create SNS topic > Configure subscription

1. Load the data from S3 to Snowflake src table [S3 LOAD + File Iterator]

- **<u>properties of S3 load:</u>**

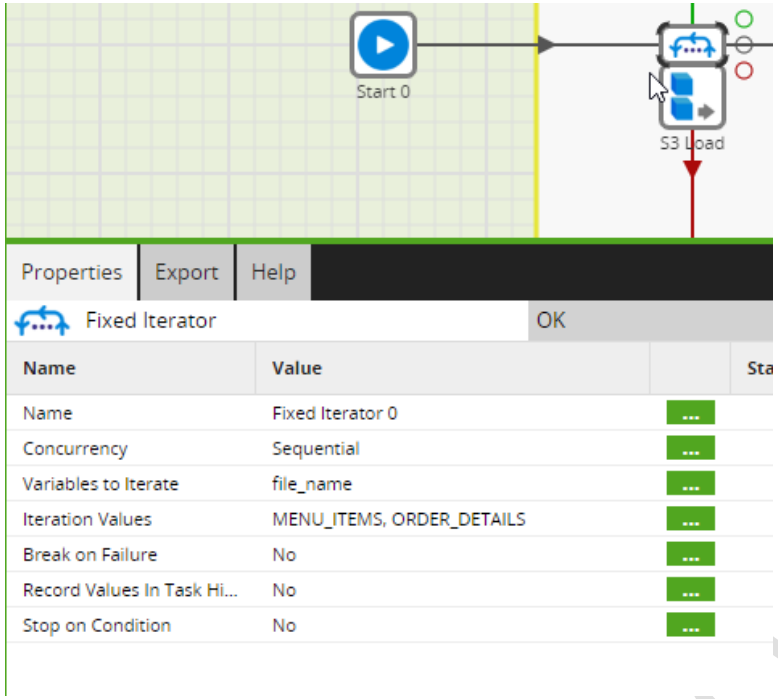
S3 Load

Name	Value
Name	S3 Load
Stage	Matillion_Stage
Pattern	${file_name}.csv
Warehouse	MATILLION_PRG
Database	${JV_DB}
Schema	${JV_SCHEMA}
Target Table	${file_name.replace(/"/g,'')}
Load Columns	
Format	[Custom]
File Type	CSV
Compression	AUTO
Record Delimiter	
Field Delimiter	,
Skip Header	1

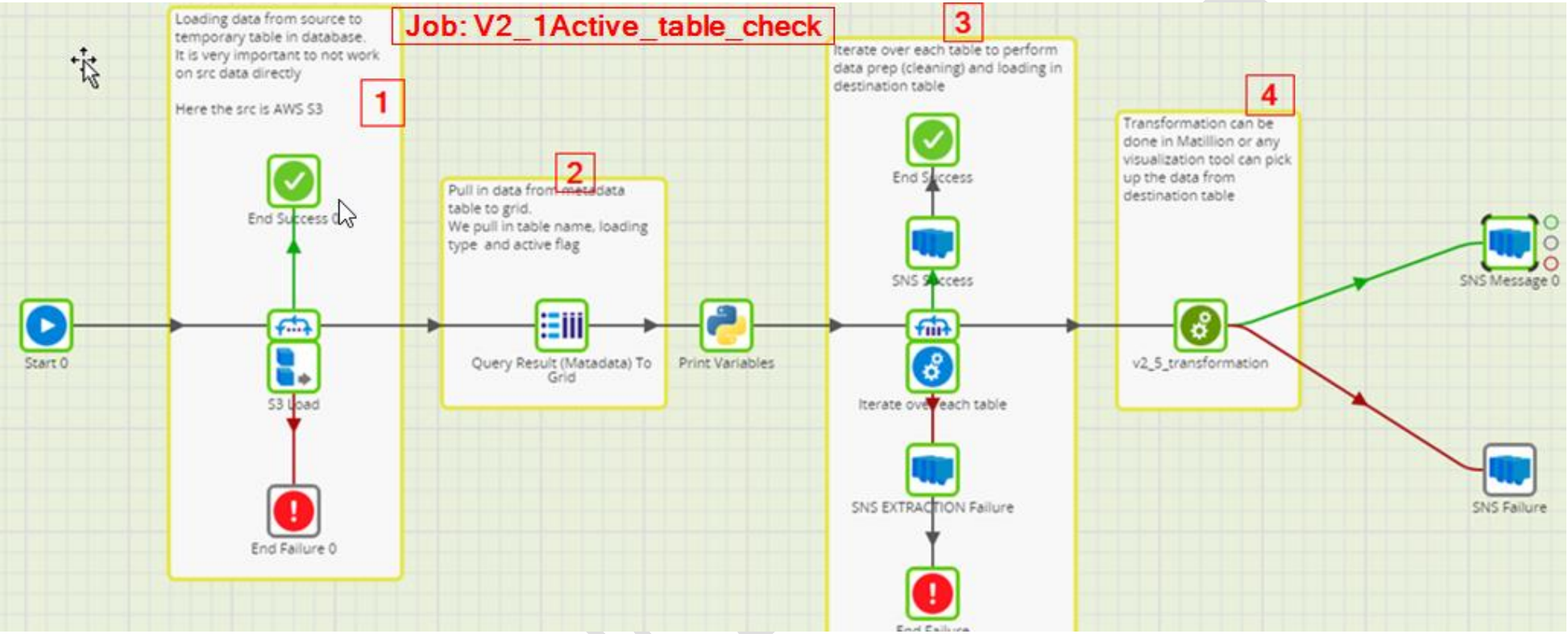
Name	
Stage	Matillion_Stage
Pattern	${file_name}.csv
Warehouse	MATILLION_PRG
Database	${JV_DB}
Schema	${JV_SCHEMA}
Target Table	${file_name.replace(/"/g,'')}

NOTE: Please set ‘Force Load ‘ to True

Properties of Fixed Iterator:



Variables to Iterate	file_name
Iteration Values	MENU_ITEMS, ORDER_DETAILS



2. Pull in info from metadata table into the GRID Variable [QUERY RESULT TO GRID]

Properties:

Query Result To Grid | OK

Name	Value		Status
Name	Query Result (Matadata) To Grid	...	
Basic / Advanced	Advanced	...	
SQL Query	/* Note: this query will be run as a subque...	...	
Grid Variable	GRID_VAR_TBL_LOADTYPE	...	
Grid Variable Mapping	TABLE_NAME, LOAD_TYPE	...	

SQL QUERY :

```
SELECT
UPPER(TABLE_NAME) TABLE_NAME,UPPER(LOAD_TYPE) LOAD_TYPE,IS_ACTIVE
FROM
RESTAURANTSALES.RESTAURANT_METADATA.TABLE_LOAD_METADATA
WHERE IS_ACTIVE =1
```

Python script – just to display variables (Also helps in displaying message in the logs

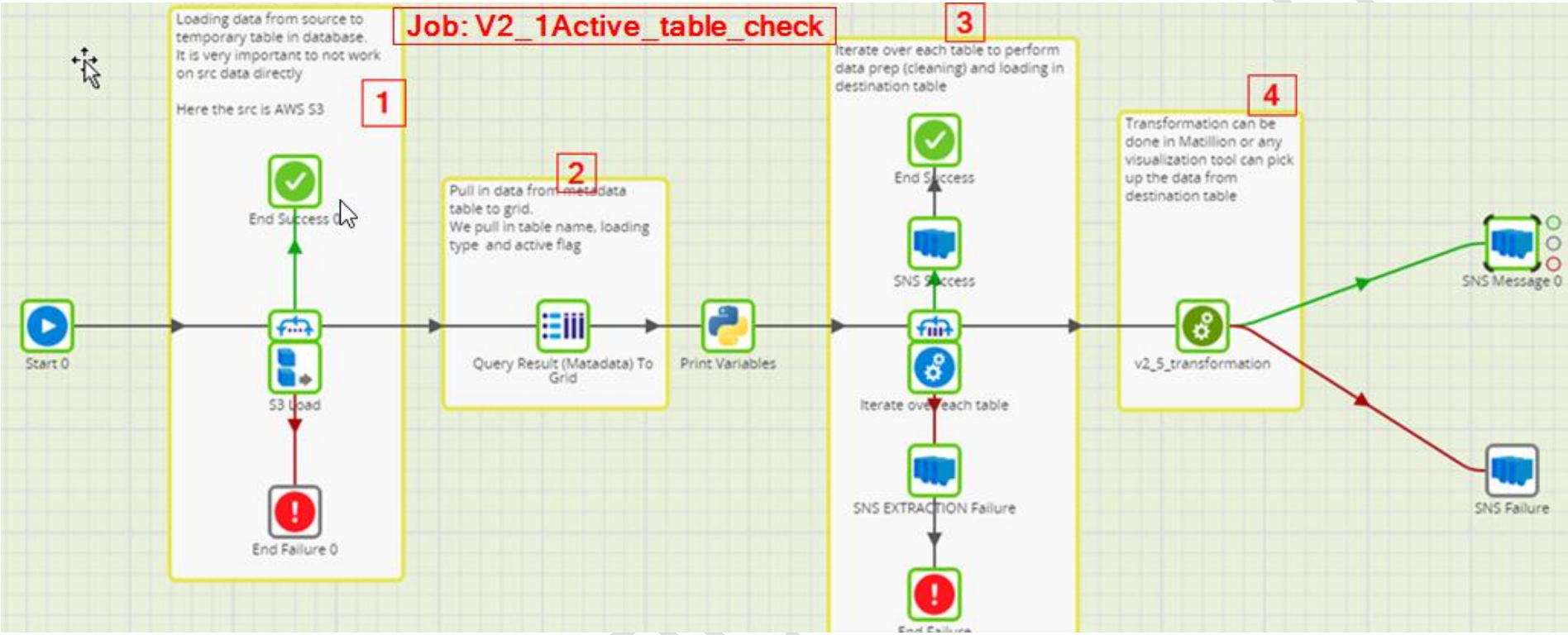
Script:

```
print (context.getGridVariable('GRID_VAR_TBL_LOADTYPE'))
print(JV_TABLE_NAME)
print (JV_LOADTYPE)
```

Logs:

V2_1Active_table_check	Print Variables	0.2s	23:18:26	23:18:26	23:18:26	[['MENU_ITEMS', 'R'], ['ORDER_DETAILS'

3. Iterate all the next steps for each table [FIXED ITEARATOR + ORCHESTRATION JOB] + Notification [SNS]



Points to ponder: Why we have added iterator here? Since we are iterating to another transformation job, why don't we do it in the same job?

Run Orchestration: V2_2Extract_basedon_loadtype2

Run Orchestration OK

Name	Value
Name	Iterate over each table
Orchestration Job	V2_2Extract_basedon_loadtype2
Set Scalar Variables	JV_DB, ${JV_DB}, JV_TABLE_NAME, ${JV_T...
Set Grid Variables	

First configure variables:

Job Variables:

Manage Job Variables

	Name	Type	Behaviour	Visibility	Value
+	JV_DB	Text	Shared	Public	RESTAURANTSALES
+	JV_LOADTYPE	Text	Shared	Public	I
+	JV_PRIMARYKEY	Text	Shared	Public	P
+	JV_SCHEMA	Text	Shared	Public	RESTAURANT_DB
+	JV_TABLE_NAME	Text	Shared	Public	ORDER_DETAILS
+	SCALAR_VAR_BATCH_ID	Text	Shared	Public	000

Text mode:

JV_DB Text Shared Public RESTAURANTSALES
JV_LOADTYPE Text Shared Public I
JV_PRIMARYKEY Text Shared Public P
JV_SCHEMA Text Shared Public RESTAURANT_DB
JV_TABLE_NAME Text Shared Public ORDER_DETAILS

—----------

Set Scalar Variables

	Variable	Value
	JV_DB	${JV_DB}
	JV_TABLE_NAME	${JV_TABLE_NAME}
	JV_SCHEMA	${JV_SCHEMA}
	JV_LOADTYPE	${JV_LOADTYPE}

Text Mode:

JV_DB ${JV_DB}
JV_TABLE_NAME ${JV_TABLE_NAME}

JV_SCHEMA ${JV_SCHEMA}
JV_LOADTYPE ${JV_LOADTYPE}

Points to ponder? What is set scalar variables?

Iterator:

Grid Iterator OK

Name	Value
Name	Grid Iterator 0
Grid Variable	GRID_VAR_TBL_LOADTYPE
Grid Variable Ma...	TABLE_NAME, JV_TABLE_NAME, LOAD_TYPE, JV_LOADTYPE
Break on Failure	No
Concurrency	Sequential
Record Values In ...	Yes
Stop on Condition	No

Grid Variable Mapping

Grid Column	Variable Name
TABLE_NAME	JV_TABLE_NAME
LOAD_TYPE	JV_LOADTYPE

Text Mode:
TABLE_NAME JV_TABLE_NAME
LOAD_TYPE JV_LOADTYPE

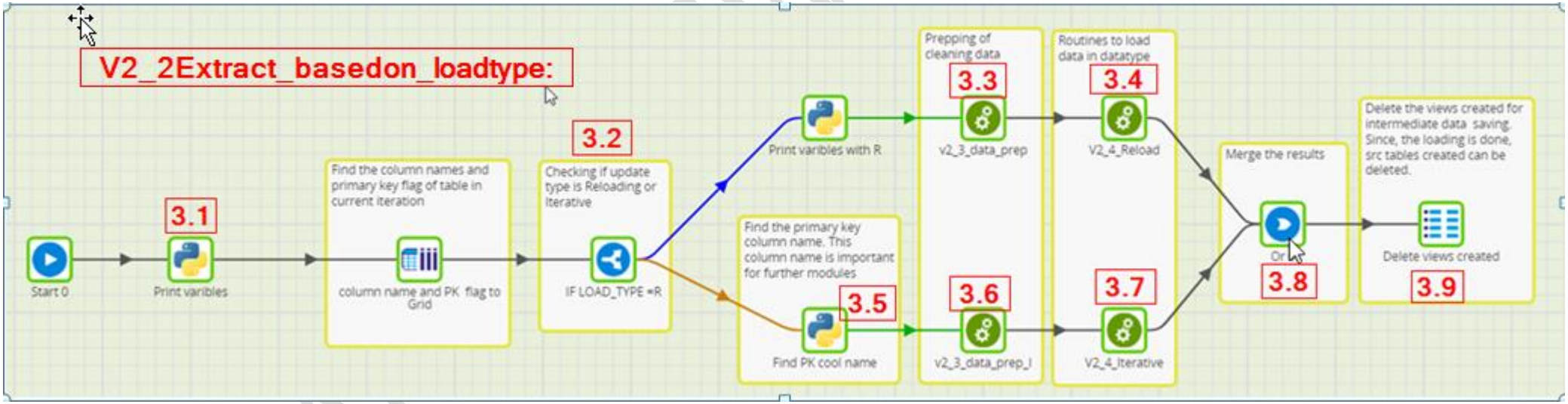
SNS Message:

SNS Message OK

Name	Value
Name	SNS Success
AWS Region	eu-north-1
Topic Name	Matillion_Status
Subject	AWS Notification Message
Message	EXTRACTION AND CLEANING SUCCESSFULLY C...

Message for success:	Message for failure:
EXTRACTION AND CLEANING SUCCESSFULLY COMPLETED JOB ID:${job_id} JOB NAME: ${job_name} MESSAGE:${component_message}	Needs attention!!! Job failed :-(JOB ID:${job_id} JOB NAME: ${job_name} MESSAGE:${component_message} DETAILED ERROR:${detailed_error}

End success and end failure:



Create another Orchestration job: **V2_2Extract_basedon_loadtype2**

Manage Grid Variables

Name	Behaviour	Visibility
DES_COLUMNS	Shared	Public

Update Grid Variable

Name: * DES_COLUMNS

Behaviour: Shared

Visibility: Public

Description: This will have metadata of the destination columns

Column Name	Column Type
COl_name	Text
Primary_key	Text

Text Mode:
COl_name Text
Primary_key Text

Update Grid Variable

Default Values:

COl_name	Primary_key
ORDER_DETAILS_ID	Yes
ORDER_ID	No
ORDER_DATE	No
ORDER_TIME	No

Text Mode:
ORDER_DETAILS_ID Yes
ORDER_ID No
ORDER_DATE No
ORDER_TIME No

3.1. Retrieve the column name and primary key of the table [table metadata to grid]

Python Script — OK

Name	Value
Name	Print varibles
Script	print (JV_DB) print (JV_SCHEMA) print (JV_TABL...
Interpreter	Python 3
Timeout	360

```
print (JV_DB)
print (JV_SCHEMA)
print (JV_TABLE_NAME)
```

Table Metadata To Grid:

Table Metadata To Grid

Name	Value
Name	column name and PK flag to Grid
Database	${JV_DB}
Schema	${JV_SCHEMA}
Grid Variable	DES_COLUMNS
Grid Variable Map...	Name, Primary Key
Target Table	${JV_TABLE_NAME}

3.2. Check if the table needs to be iterated or reloaded [IF component]

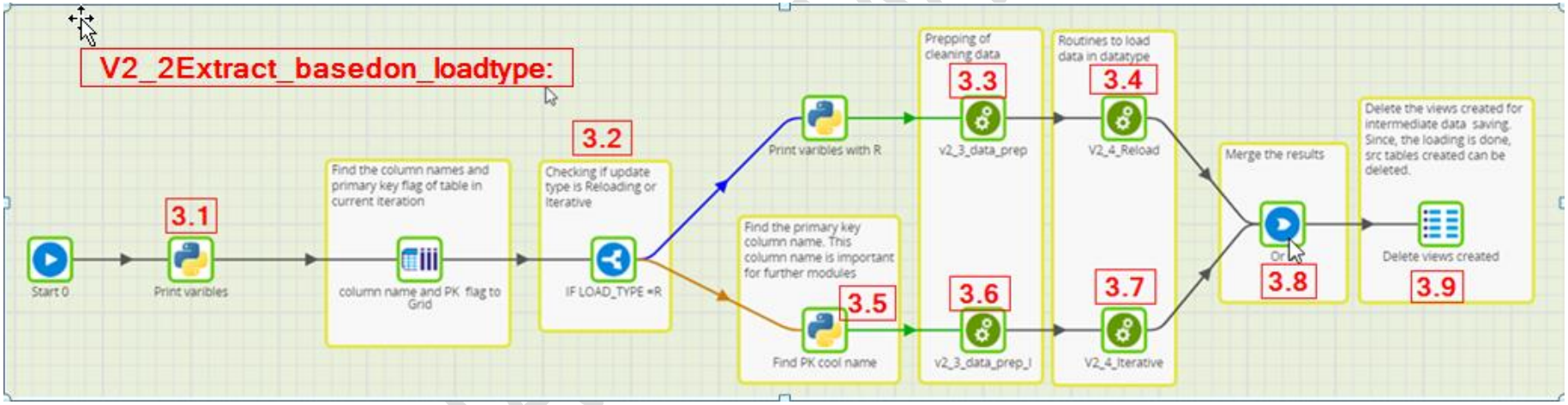
If

Name	Value
Name	IF LOAD_TYPE =R
Mode	Simple
Condition	JV_LOADTYPE, Is, Equal to, R
Combine Conditions	And

Condition

	Input Variable	Qualifier	Comparator	Value
	JV_LOADTYPE	Is	Equal to	R

3.3. For R: Prepare /clean the data [Run transformation component]



Python component:

print (JV_DB)

print (JV_SCHEMA)

print (JV_TABLE_NAME)

print(context.getGridVariable('DES_COLUMNS'))

—----------

Create a Transformation job – V2_3_MENU_ITEMS

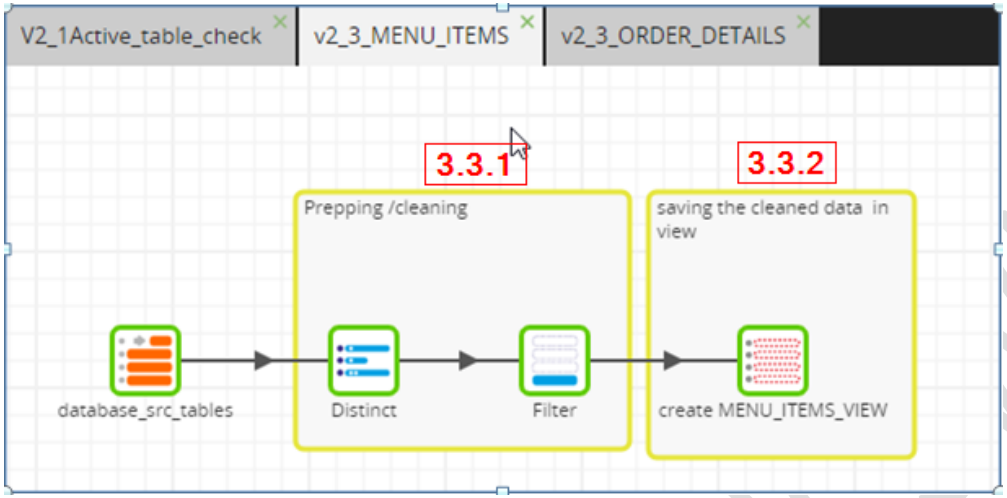


Table Input OK

Name	Value
Name	database_src_tables
Database	RESTAURANTSALES
Schema	RESTAURANT_DB
Target Table	MENU_ITEMS
Column Na...	MENU_ITEM_ID, ITEM_NAME, CATEGORY, PRICE
Offset	

3.3.1. Cleaning [DISTINCT + FILTER NULLS]

Distinct:

Distinct OK

Name	Value
Name	Distinct
Columns	MENU_ITEM_ID, ITEM_NAME, CATEGORY, PRICE

Filter:

Filter OK

Name	Value
Name	Filter
Filter Conditions	ITEM_NAME, Not, Null, CATEGORY, Not, Null, P...
Combine Conditions	AND

Filter Condition (Text Mode)

ITEM_NAME Not Null

CATEGORY Not Null

PRICE Not Null

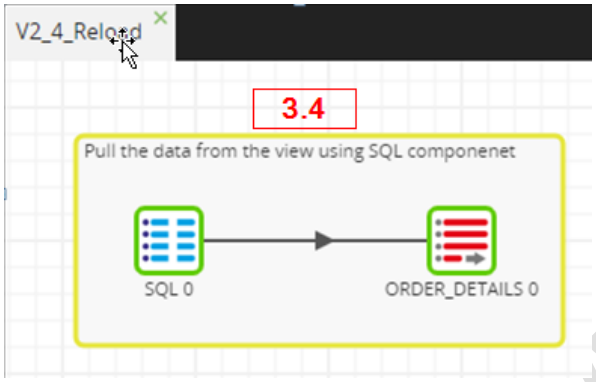
3.3.2. Save it in a view [CREATE VIEW]

Create view:

Create View

Name	Value
Name	create MENU_ITEMS_VIEW
Database	RESTAURANTSALES
Schema	RESTAURANT_DB_DES
View Name	MENU_ITEMS_VIEW
Secure View	No
View Type	Standard

Create a Transformation job – V2_4_Reload



- Configure job variables:
  JV_DBText Shared Public RESTAURANTSALES
  JV_SCHEMA Text Shared Public RESTAURANT_DB
  JV_TABLE_NAME Text Shared Public MENU_ITEMS

- configure grid variable
  DES_COLUMNS grid variables

Manage Grid Variables

	Name	Behaviour	Visibility	Columns	Values	
+	DES_COLUMNS	Shared	Public			

Update Grid Variable

Name: * DES_COLUMNS

Behaviour: Shared

Visibility: Public

Description:

Column Name	Column Type
COl_name	Text
Primary_key	Text

TEST mode:

COl_name Text
Primary_key Text

Update Grid Variable

Default Values:

COl_name	Primary_key
MENU_ITEM_ID	Yes
ITEM_NAME	No
CATEGORY	No
PRICE	No

TExt mode:

MENU_ITEM_ID Yes
ITEM_NAME No
CATEGORY No
PRICE No

- SQL:

SQL

Name	Value
Name	SQL 0
SQL Query	SELECT * FROM RESTAURANTSALE...

SELECT * FROM RESTAURANTSALES.RESTAURANT_DB_DES.${JV_TABLE_NAME}_VIEW

Table output: - write an input data flow out to an existing output table.

Table Output

Name	Value
Name	ORDER_DETAILS 0
Warehouse	COMPUTE_WH
Database	${JV_DB}
Schema	${JV_SCHEMA}_DES
Target Table	${JV_TABLE_NAME}_DES
Fix Data Ty...	No
Column M...	DES_COLUMNS, COl_name, COl...
Order By	
Truncate	Truncate

Column Mapping

Grid Variable: DES_COLUMNS

Column Mapping

Input Column: COl_name

Output Column: COl_name

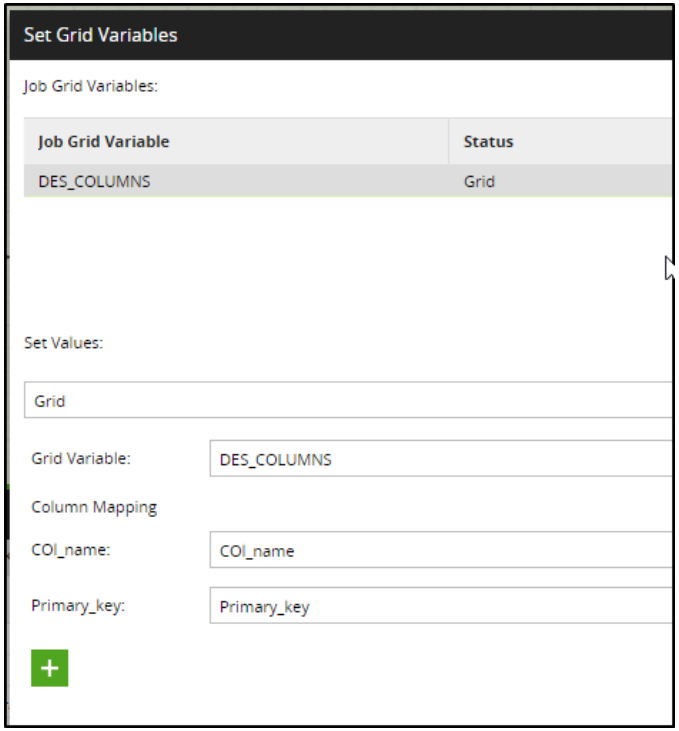
Run Transformation OK

Name	Value
Name	v2_3_data_prep
Transforma...	v2_3_${JV_TABLE_NAME}
Set Scalar V...	JV_DB, ${JV_DB}, JV_SCHEMA, ${JV_SCHEMA}, JV_TABLE_NAME, ${JV_TABLE_NAME}
Set Grid Var...	DES_COLUMNS, grid

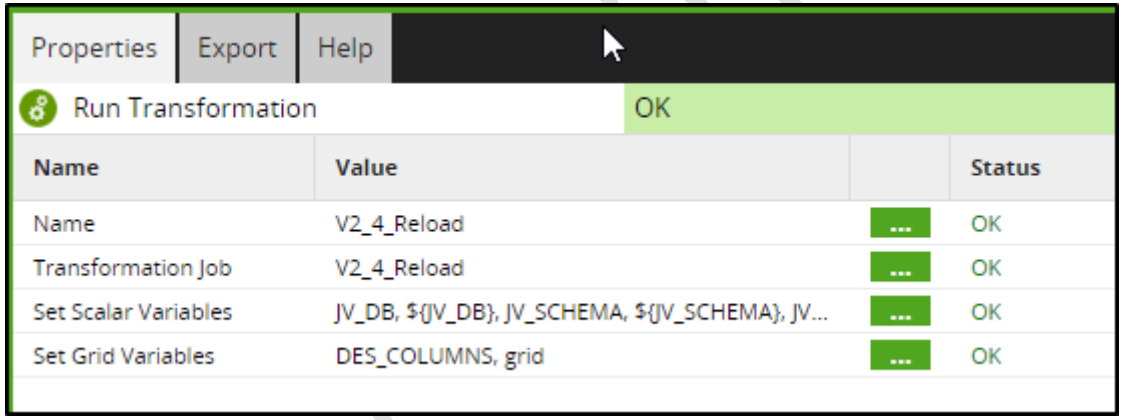
v2_3_${JV_TABLE_NAME}

Text Mode:

JV_DB ${JV_DB}
JV_SCHEMA ${JV_SCHEMA}
JV_TABLE_NAME ${JV_TABLE_NAME}



3.4. For R : Load the data from the VIEW into the destination database [RUN transformation component]

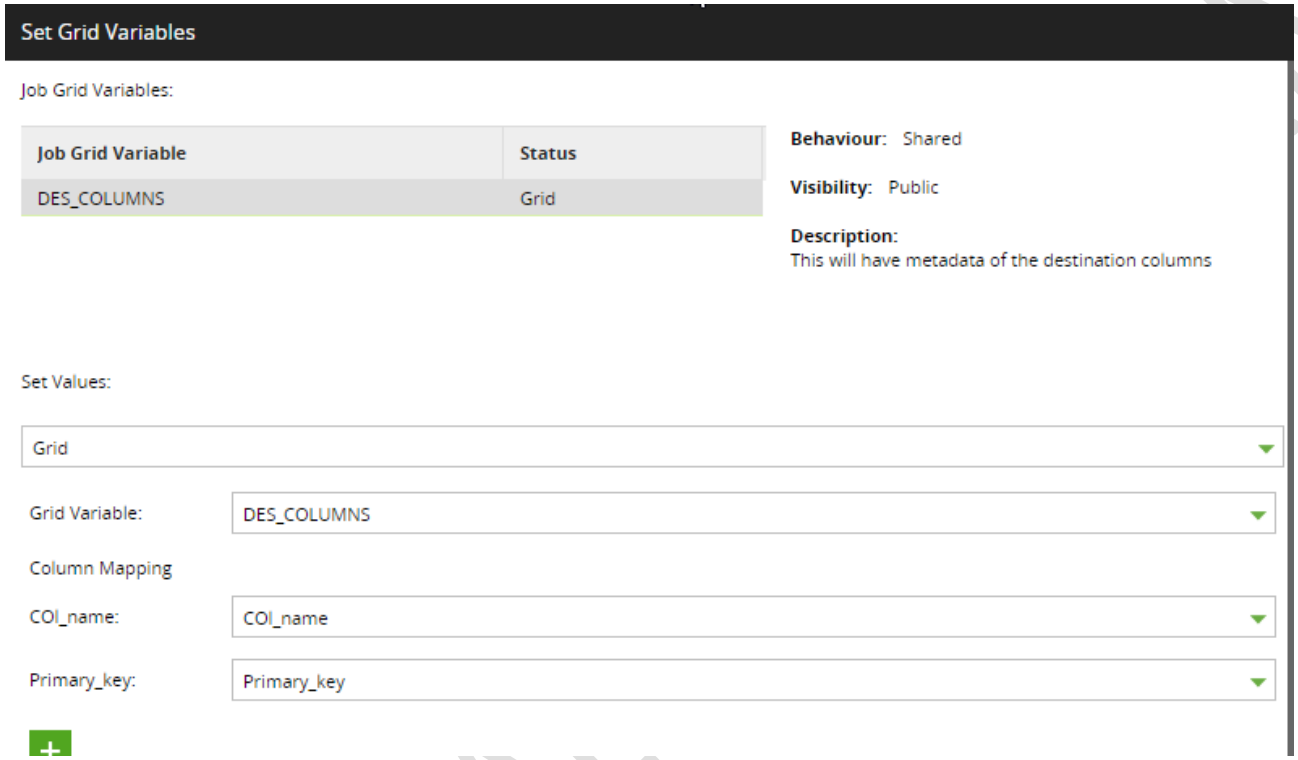


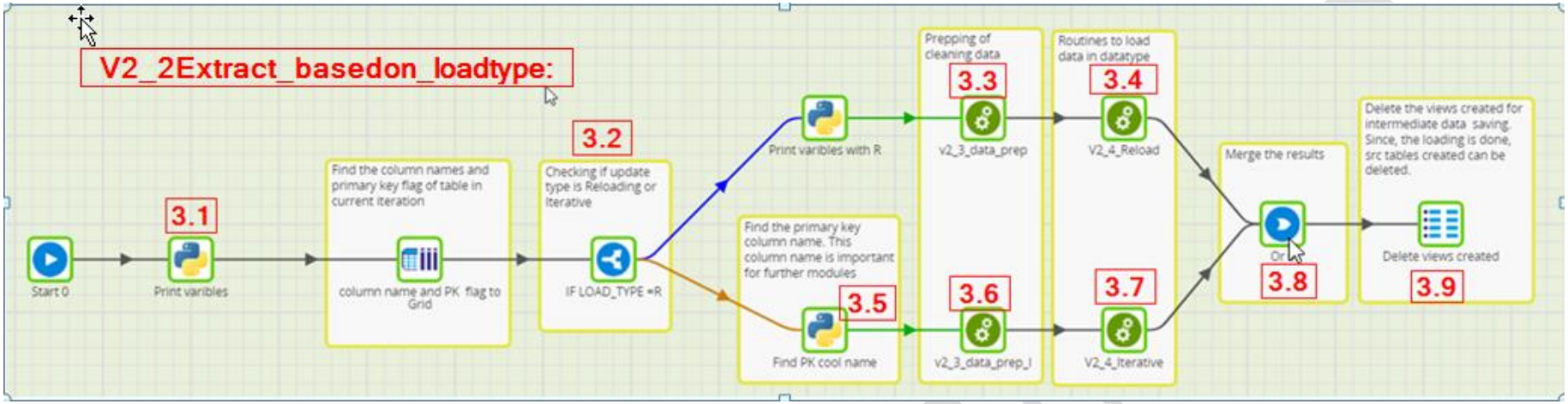
set scalar variable:

JV_DB ${JV_DB}

JV_SCHEMA ${JV_SCHEMA}

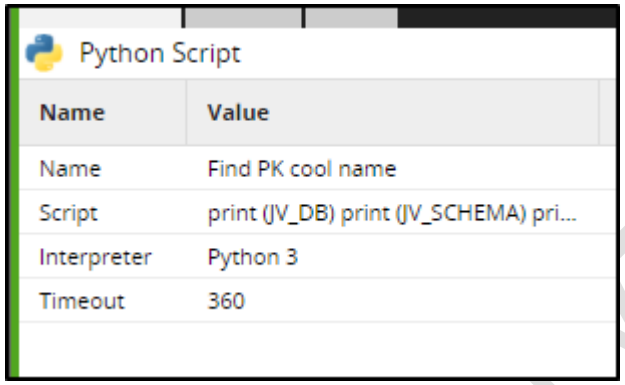
JV_TABLE_NAME ${JV_TABLE_NAME}





3.5. For I: Identify the name of the primary key using the flag [Python script]

Python script:



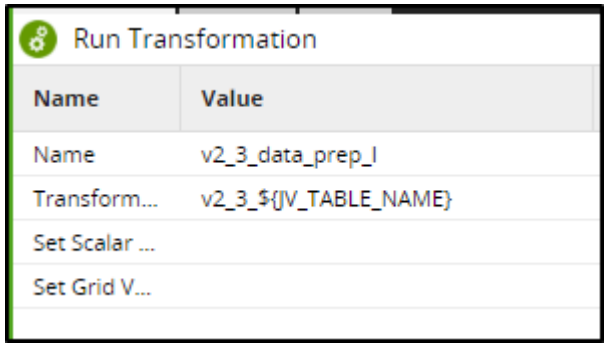
Python Script

Name	Value
Name	Find PK cool name
Script	print (JV_DB) print (JV_SCHEMA) pri...
Interpreter	Python 3
Timeout	360

```
for i in context.getGridVariable('DES_COLUMNS'):
 if i[1]=='Yes':
  context.updateVariable('JV_PRIMARYKEY', i[0])
print(JV_PRIMARYKEY)
```

3.6. For I: Prepare /clean the data [Run transformation component]

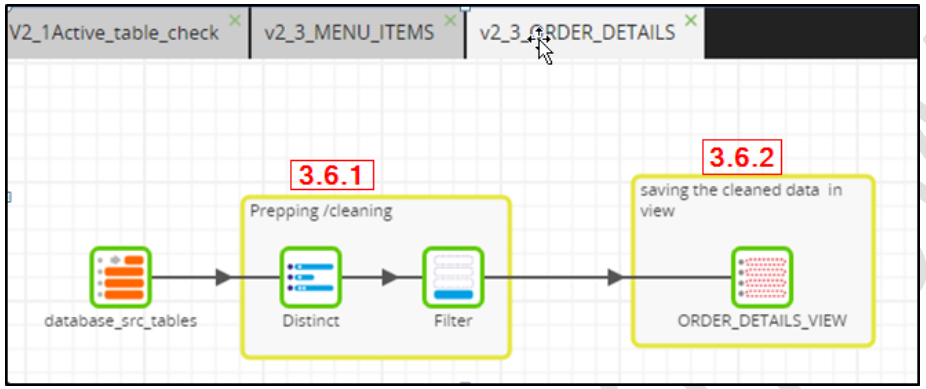
Run transformation component:



Run Transformation

Name	Value
Name	v2_3_data_prep_I
Transform...	v2_3_${JV_TABLE_NAME}
Set Scalar ...	
Set Grid V...	

Create a Transformation job – V2_3_ORDER_DETAILS



adding variables:

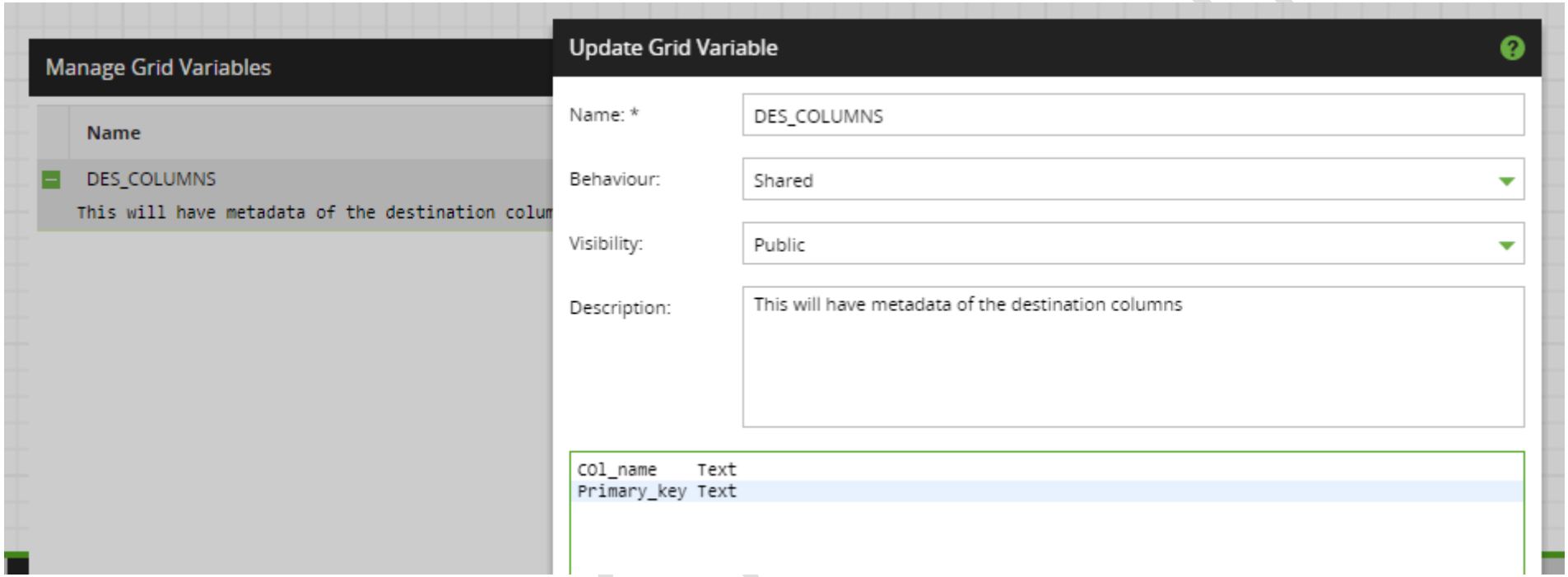
text mode:

JV_DB Text Shared Public RESTAURANTSALES

JV_SCHEMA Text Shared Public RESTAURANT_DB

JV_TABLE_NAME Text Shared Public ORDER_DETAILS

Grid variables:



COl_name	Text
Primary_key	Text

Values:

ORDER_DETAILS_ID	Yes
ORDER_ID	No
ORDER_DATE	No
ORDER_TIME	No
ITEM_ID	No

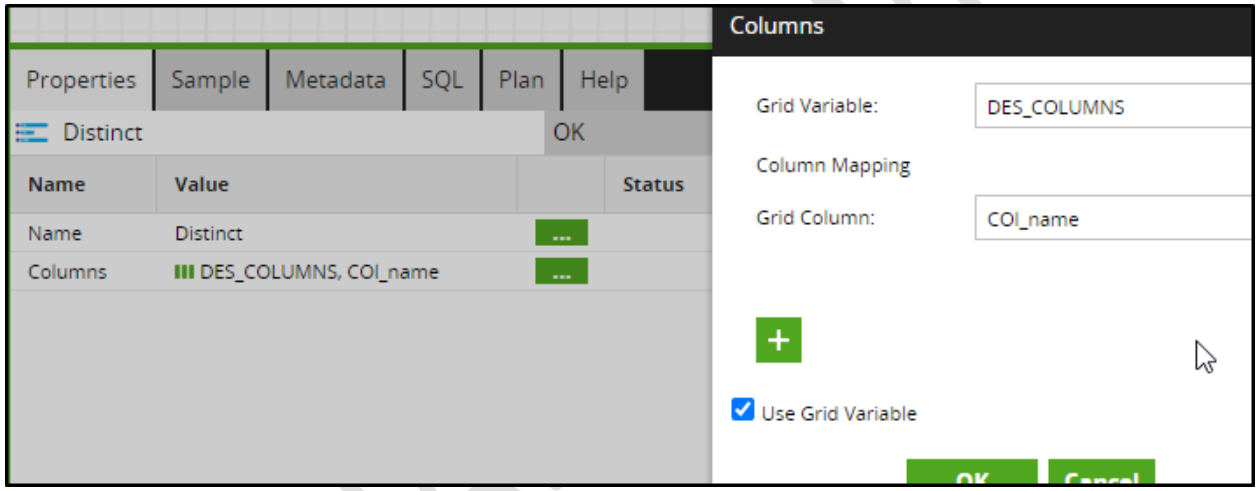
Table input:

Table Input

Name	Value
Name	database_src_tables
Database	${JV_DB}
Schema	${JV_SCHEMA}
Target Table	${JV_TABLE_NAME}
Column N...	DES_COLUMNS, COl_name
Offset	

3.6.1. *Cleaning [DISTINCT + FILTER NULLS]*

Distinct



Filter

Filter

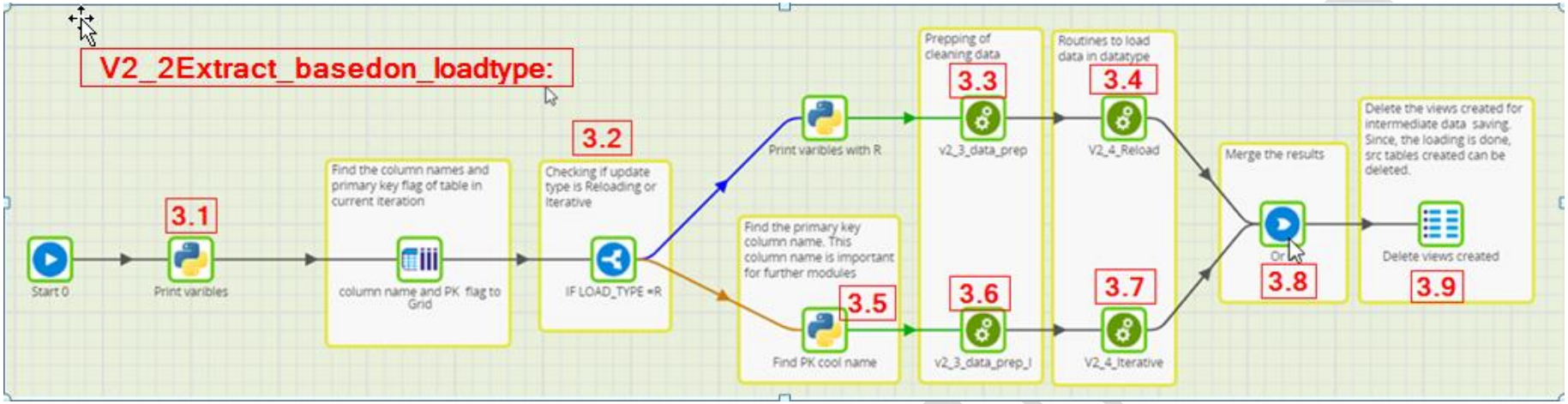
Name	Value
Name	Filter
Filter Con...	ITEM_ID, Not, Ilike, NULL
Combine ...	AND

3.6.2. Save it in a view [CREATE VIEW]

View

Create View

Name	Value
Name	ORDER_DETAILS_VIEW
Database	RESTAURANTSALES
Schema	RESTAURANT_DB_DES
View Name	${JV_TABLE_NAME}_VIEW
Secure View	No
View Type	Standard



3.7. For I: Load the data from the VIEW into the destination database [RUN transformation component]

Create a Transformation job – V2_4_Iterative

Copy the Grid variables and Job variables

Manage Job Variables

Name	Type	Behaviour	Visibility	Value
JV_DB	Text	Shared	Public	RESTAURANTSALES
JV_PRIMARYKEY	Text	Shared	Public	ORDER_DETAILS_ID
JV_SCHEMA	Text	Shared	Public	RESTAURANT_DB
JV_TABLE_NAME	Text	Shared	Public	ORDER_DETAILS

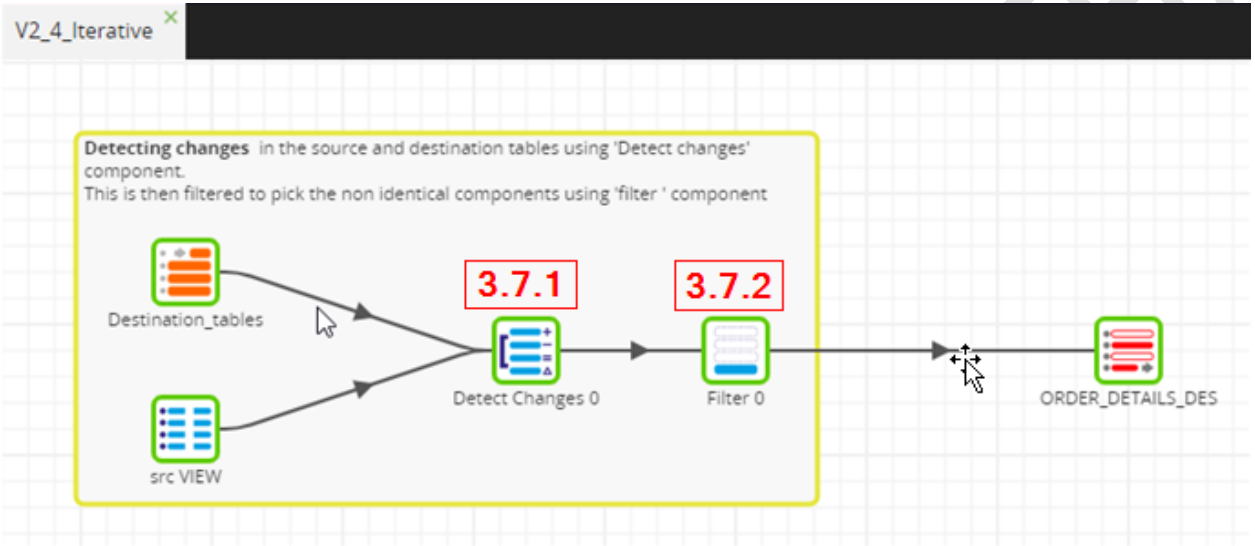
Manage Grid Variables

Name	Behaviour	Visibility	C
DES_COLUMNS	Shared	Public	

Run transformation component:

Run Transformation OK

Name	Value
Name	V2_4_Iterative
Transfor...	V2_4_Iterative
Set Scala...	JV_DB, ${JV_DB}, JV_SCHEMA, ${JV_SCHEMA}, JV_T...
Set Grid ...	DES_COLUMNS, grid



3.7.1. Detect changes [Detect changes]

Table input:

Table Input

Name	Value
Name	Destination_tables
Database	${JV_DB}
Schema	${JV_SCHEMA}_DES
Target T...	${JV_TABLE_NAME}_DES
Column ...	DES_COLUMNS, COl_name
Offset	

SQL query

SELECT * FROM RESTAURANTSALES.RESTAURANT_DB_DES.${JV_TABLE_NAME}_VIEW

Detect changes

Don't worry about the errors at this point. It's because these don't have columns its searching for. Once the iteration start, the right columns will come in.

Detect Changes — Input components are not in a valid state.

Name	Value		Status
Name	Detect Changes 0	...	OK
Master T...	Destination_tables	...	OK
Match Ke...	DES_COLUMNS, COl_name	...	No columns that appear in both inputs.
Compare...	DES_COLUMNS, COl_name	...	No columns that appear in both inputs.
Output C...	master_${JV_PRIMARYKEY}, compare_${JV_...	...	OK
Indicator...	Indicator	...	OK

Output Column Mapping

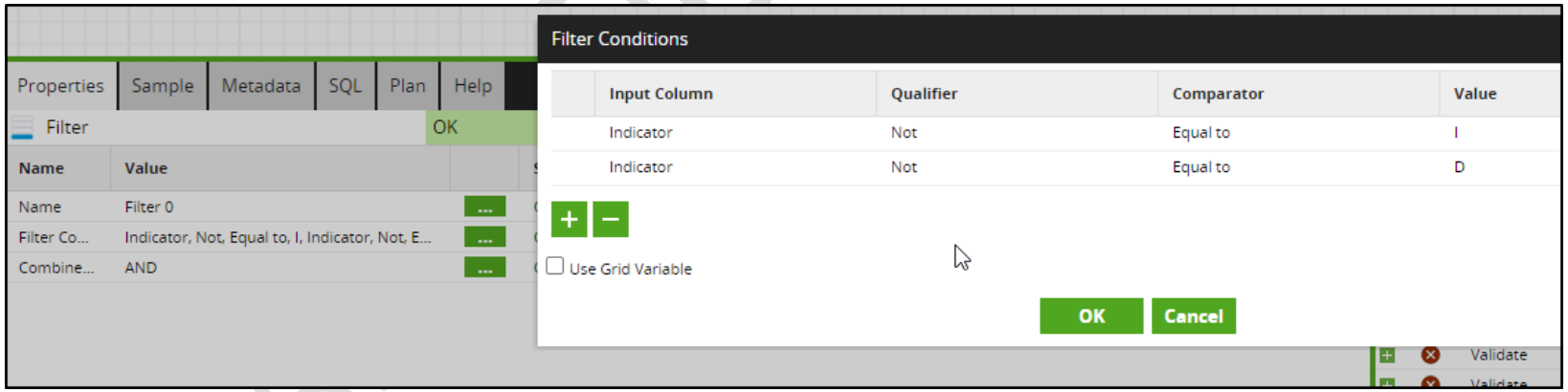
Input Column	Output Column
master_${JV_PRIMARYKEY}	compare_${JV_PRIMARYKEY}

Test mode:

master_${JV_PRIMARYKEY} compare_${JV_PRIMARYKEY}

3.7.2. Filter rows that need to be Upserted [filter]

Filter



Text mode:

Indicator	Not	Equal to	I
Indicator	Not	Equal to	D

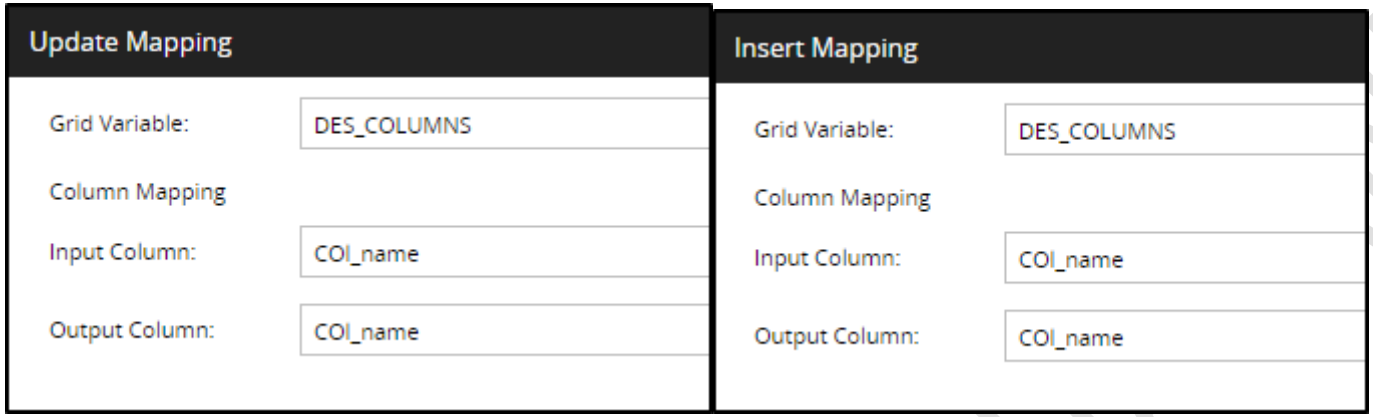
Upsert

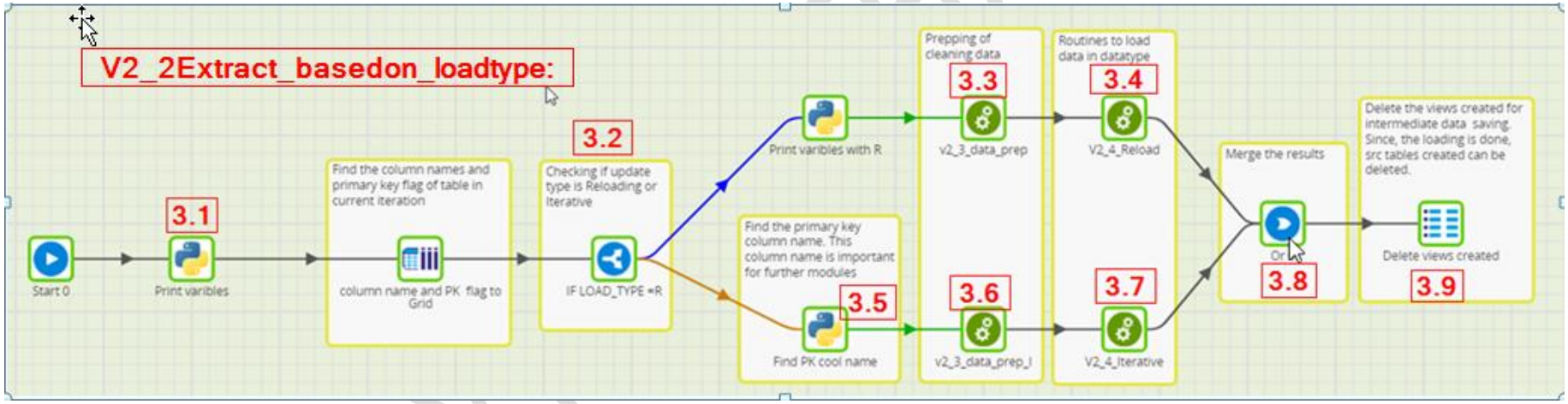
Table Update — OK

Name	Value
Name	ORDER_DETAILS_DES
Warehou...	COMPUTE_WH
Database	${JV_DB}
Schema	${JV_SCHEMA}_DES
Target Ta...	${JV_TABLE_NAME}_DES
Target Ali...	target
Source Al...	input
Join Expr...	"target".${JV_PRIMARYKEY}="input".${JV_P...
When M...	"target".${JV_PRIMARYKEY}="input".${JV_P...
Update ...	DES_COLUMNS, COl_name, COl_name
Include ...	Yes
Insert M...	DES_COLUMNS, COl_name, COl_name

JOIN expression: "target".${JV_PRIMARYKEY}="input".${JV_PRIMARYKEY}

WHEN marched: "target".${JV_PRIMARYKEY}="input".${JV_PRIMARYKEY} Update





3.8. Irrespective of R or I , the pipeline needs to be merged [OR]

OR

3.9. Since the cleaning and loading is completed, Unnecessary data/ structure is removed [SQL SCRIPT]

SQL script

SQL Script OK

Name	Value		Sta
Name	Delete views created	...	OK
SQL Script	DROP VIEW ${JV_DB}.${JV_SCHEMA}_DES.$...	...	OK

Script:

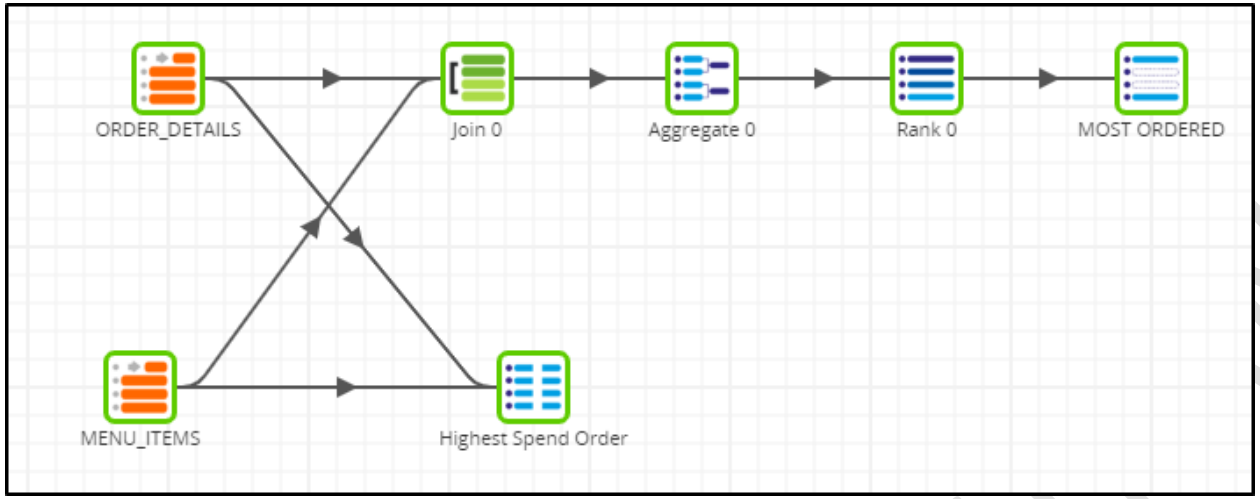
```
DROP VIEW ${JV_DB}.${JV_SCHEMA}_DES.${JV_TABLE_NAME}_VIEW;

-- TRUNCATE TABLE ${JV_DB}.${JV_SCHEMA}.${JV_TABLE_NAME}
```

4. ***You can perform transformations to retrieve value /KPI OR use the data as input to any visualization tool. [RUN transformation component]***

Run Transformation OK

Name	Value		Status
Name	v2_5_transformation	...	OK
Transfor...	v2_5_transformation	...	OK
Set Scala...		...	OK
Set Grid ...		...	OK

Create Transformation Job – V2_5_Tranformation



ORDER DETAILS/Menu Items:

Table Input

Name	Value
Name	ORDER_DETAILS
Database	RESTAURANTSALES
Schema	RESTAURANT_DB_DES
Target Ta...	ORDER_DETAILS_DES
Column N...	ORDER_DETAILS_ID, ORDER_ID...
Offset	

JOIN:

Properties | Sample | Metadata | SQL | Plan | Help

Join — OK

Name	Value
Name	Join 0
Main Table	ORDER_DETAILS
Main Table Alias	order_details
Joins	MENU_ITEMS, Menu_items, Inner
Join Expressions	"order_details"."ITEM_ID"="Menu_items"."MENU_ITEM_ID", order_details
Output Columns	order_details.ORDER_DETAILS_ID, ORDER_DETAILS_ID, order_details.ORD

JOINS: "order_details"."ITEM_ID"="Menu_items"."MENU_ITEM_ID"

JOIN Expression: "order_details"."ITEM_ID"="Menu_items"."MENU_ITEM_ID"

OUTPUT COLUMS:

order_details.ORDER_DETAILS_ID ORDER_DETAILS_ID
order_details.ORDER_ID ORDER_ID
order_details.ORDER_DATE ORDER_DATE
order_details.ORDER_TIME ORDER_TIME
order_details.ITEM_ID ITEM_ID
Menu_items.MENU_ITEM_ID MENU_ITEM_ID
Menu_items.ITEM_NAME MENU_ITEM_NAME
Menu_items.CATEGORYMENU_CATEGORY
Menu_items.PRICE MENU_PRICE

AGGREGATE:

Properties | Sample | Metadata | SQL | Plan | Help

Aggregate OK

Name	Value
Name	Aggregate 0
Groupings	MENU_ITEM_NAME
Aggregations	MENU_ITEM_NAME, Count
Grouping Type	Group By

RANK:

Properties | Sample | Metadata | SQL | Plan | Help

Aggregate OK

Name	Value
Name	Aggregate 0
Groupings	MENU_ITEM_NAME
Aggregations	MENU_ITEM_NAME, Count
Grouping Type	Group By

FIRST/LAST:

First/Last

Name	Value
Name	MOST ORDERED
Grouping Colu...	MENU_ITEM_NAME
Ordering within...	DenseRank, Asc
First/Last Colu...	DenseRank, First
Ignore Nulls	No

SQL:

First/Last

Name	Value
Name	MOST ORDERED
Grouping Colu...	MENU_ITEM_NAME
Ordering within...	DenseRank, Asc
First/Last Colu...	DenseRank, First
Ignore Nulls	No

```
SELECT OD.order_ID, SUM(PRICE) ORDER_SPEND FROM RESTAURANTSALES.RESTAURANT_DB_DES.ORDER_DETAILS_DES OD
INNER JOIN RESTAURANTSALES.RESTAURANT_DB_DES.MENU_ITEMS_DES MI
ON OD.ITEM_ID=MI.MENU_ITEM_ID
GROUP BY order_ID
ORDER BY 2 DESC LIMIT 1
```

SNS Message: Same as first SNS

Input to Viz tool:

You could use this data you stored in the destination database (in snowflake) as input to any visualization tool.

Here, we will just integrate with power BI and if time permits we can change the data src to new destination database.

GIT Integration:

- **https://docs.matillion.com/data-productivity-cloud/designer/docs/git-overview/**

Scheduling:

- **Project>schedules>add schedules>create new schedules**
- **Uses Cron expressions**
- **https://docs.matillion.com/data-productivity-cloud/designer/docs/schedules/**

References:

1. https://www.matillion.com/blog/matillion-recognized-as-a-challenger-in-the-2023-gartner-magic-quadrant-for-data-integration-tools
2. https://docs.matillion.com/metl/docs/2037630/
3. https://www.matillion.com/blog/building-data-pipelines-always-on-tables-with-matillion-etl
4. https://www.matillion.com/blog/what-is-etl-the-ultimate-guide
5. https://docs.matillion.com/metl/docs/2917841/
6. https://docs.matillion.com/metl/docs/2943425/#contact-support