

## Document Contents:

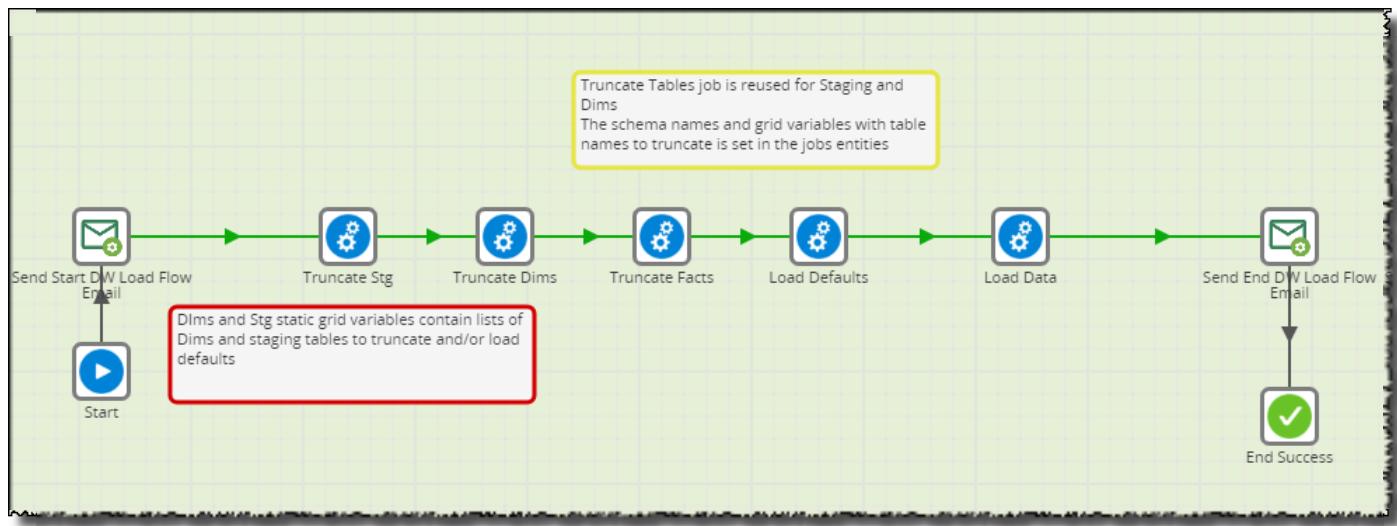
- KD
- Job: Main
  - Send Start DW Load Flow Email
  - Truncate Stg
  - Truncate Dims
  - Truncate Facts
  - Load Defaults
  - Load Data
  - Send End DW Load Flow Email
  - End Success
- Job: Truncate Tables
  - Truncate Tables?
  - Send Start Truncating tables
  - Truncate Tables 0
- Job: Load Defaults
  - Load Defaults?
  - Send Start Load Defaults
  - Grid Iterator 0
  - SQL Load Defaults
- Job: Load
  - Start Load?
  - Start Load
  - Stg\_Load\_Main
  - Dim\_Load\_Main
  - Load Policy Facts
  - End Load
  - Send End DW Load Email
  - Load Compete
  - Nothing To Load
- Job: Start Load
  - Set prev\_loaddate
  - Set is\_SPINN\_EOD\_Complete
  - If is\_SPINN\_EOD\_Complete=1 (new data complete) or is\_SPINN\_EOD\_Complete=-1 (ignore)
  - Set loaddate
  - Set sql\_bookDate
  - Set sql\_currentDate
  - SQL Insert into DW Load Log
  - Send Start DW Load Email
  - Send DW Load Did Not Start email
- Job: Stg\_Load\_Main

- Load Staging?
- Send Start Staging Load Email
- sp\_stg\_producer
- sp\_stg\_policy\_scope
- sp\_stg\_policytransactionextension
- Stg\_Load\_Policy\_Related
- sp\_stg\_policytransaction
- Months To Summarize
- Grid Iterator 0
- sp\_stg\_fpc
- Job: Dim\_Load\_Main
  - Load Dim?
  - Send Start Dimensions Load Email
  - Dim\_Load\_Policy\_Related
  - Complete\_Dim\_Policy\_Realted\_Load
  - Dim\_Load\_Other
  - Dim\_Load\_Coverage\_Related
- Job: Load Policy Facts
  - Load Policy Facts?
  - Load Fact Policy Transactions
  - Load Fact Policy Summaries
- Job: End Load
  - Did Load Start?
  - SQL Update DW Load Log
- Job: Stg\_Load\_Policy\_Related
  - sp\_stg\_product
  - sp\_stg\_insured
  - sp\_stg\_driver
  - sp\_stg\_risk\_coverage
  - sp\_stg\_vehicle
  - sp\_stg\_building
  - sp\_stg\_policy
- Job: Dim\_Load\_Policy\_Related
  - DIM\_APPLICATION
  - DIM\_DRIVER
  - DIM\_INSURED
  - DIM\_PRODUCT
  - DIM\_COVEREDRISK
  - DIM\_VEHICLE
  - DIM\_BUILDING
  - DIM\_RISK\_COVERAGE
- Job: Dim\_Load\_Other
  - DIM\_POLICYTRANSACTIONEXTENSION
  - DIM\_CUSTOMER
  - sp\_dim\_producer
  - DIM\_COMPANY

- DIM\_ADDRESS
- Job: Dim\_Load\_Coverage\_Related
  - DIM\_LIMIT
  - DIM\_CLASSIFICATION
  - DIM\_DEDUCTIBLE
  - DIM\_COVERAGE
- Job: Load Fact Policy Transactions
  - Load Policy Transactions?
  - Send Start Fact Policy Transaction Load Email
  - sp\_fact\_policytransaction
- Job: Load Fact Policy Summaries
  - Load Policy Summaries?
  - Send Start Fact Policy Summaries Load Email
  - sp\_fact\_policycoverage
- Job: DIM\_APPLICATION
  - stg\_policy
  - dim\_application
- Job: DIM\_DRIVER
  - stg\_driver
  - dim\_driver
- Job: DIM\_INSURED
  - stg\_insured
  - dim\_insured
- Job: DIM\_PRODUCT
  - stg\_product
  - dim\_product
- Job: DIM\_COVEREDRISK
  - stg\_coveredrisk
  - dim\_coveredrisk
- Job: DIM\_VEHICLE
  - stg\_vehicle
  - dim\_vehicle
- Job: DIM\_BUILDING
  - stg\_building
  - dim\_building
- Job: DIM\_RISK\_COVERAGE
  - stg\_risk\_coverage
  - dim\_risk\_coverage
- Job: DIM\_POLICYTRANSACTIONEXTENSION
  - stg\_policytransactionextension
  - dim\_policytransactionextension
- Job: DIM\_CUSTOMER
  - stg\_customer
  - dim\_customer
- Job: DIM\_COMPANY
  - stg\_policytransaction

- dim\_company
- Job: DIM\_ADDRESS
  - stg\_vehicle\_building
  - dim\_address
- Job: DIM\_LIMIT
  - stg\_coverage
  - dim\_limit
- Job: DIM\_CLASSIFICATION
  - stg\_coverage
  - dim\_classification
- Job: DIM\_DEDUCTIBLE
  - stg\_coverage
  - dim\_deductible
- Job: DIM\_COVERAGE
  - stg\_coverage
  - dim\_coverage

## Job: Main



Entry point into DW Load jobs

Grid variables:

Name	Visibility	Behaviour	Column Names	Column Types	Description
Dims	Public	Shared	[table_name]	[Text]	
Facts	Public	Shared	[table_name]	[Text]	
Stg	Public	Shared	[table_name]	[Text]	List of tables from staging environment to clean up

Truncate Tables job is reused for Staging and Dims

The schema names and grid variables with table names to truncate is set in the jobs entities

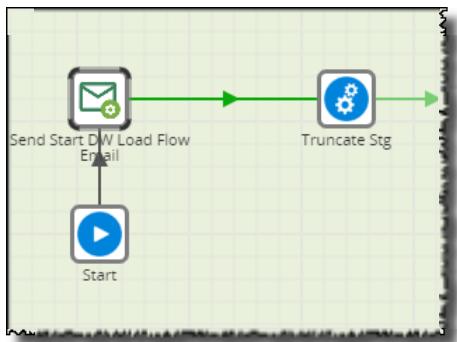
DIMs and STG grid variables is THIS (Main) job STATIC grid variables  
(Can be replaced to dynamic grid variables to populate from a query)

DLms and Stg static grid variables contain lists of Dims and staging tables to truncate and/or load defaults

They can be dynamically populated

## Send Start DW Load Flow Email

Send Email

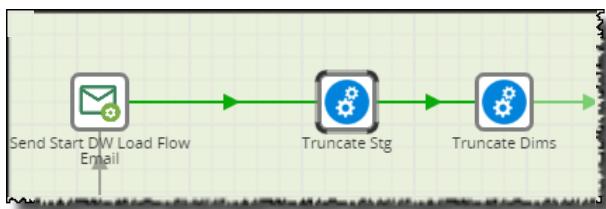


Parameter	Value
To Recipients	\${EMAIL_DESTINATION_ADDRESS}
Cc Recipients	
Subject	\${project_name} Flow Started
Message	Project Name: \${project_name}  Environment: -- Name: \${environment_name} -- Default schema: \${environment_default_schema} DW schema: \${schema_DW} STG schema: \${schema_STG}  Steps: -- Truncate Stg: \${TruncateStg} Truncate DW: \${TruncateDW} -- Load Defaults: \${LoadDefaults} --

	Load Staging: \${LoadStaging} Load Dimensions: \${LoadDimensions} Load Policy Transactions: \${LoadPolicyTransactions} Load Policy Summaries: \${LoadPolicySummaries}
	--
Sender Address	
	\${EMAIL_SENDER_ADDRESS}
Reply Address	
SMTP Username	
	\${EMAIL_AUTHENTICATION_USER}
SMTP Password	
	*****
SMTP Hostname	
	\${EMAIL_SMTP_SERVER}
SMTP Port	
	\${EMAIL_SMTP_PORT}
Enable SSL/TLS	
	Yes
Enable StartTLS	
	Yes

## Truncate Stg

Run Orchestration

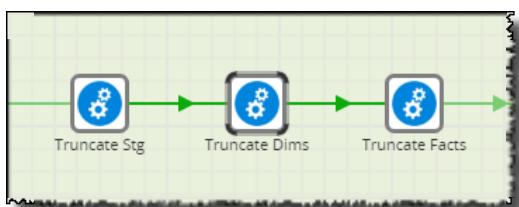


[Jump to job](#)

Parameter	Value
Orchestration Job	Truncate Tables
Set Scalar Variables	DoTruncate, \${TruncateStg}, schema_name, \${schema_STG}, Comment, Staging
Set Grid Variables	Tables, grid

## Truncate Dims

Run Orchestration

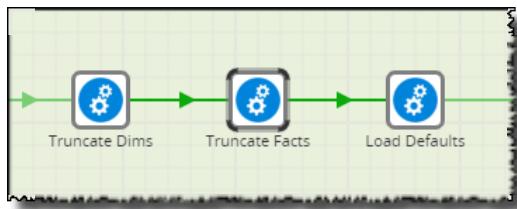


[Jump to job](#)

Parameter	Value
Orchestration Job	Truncate Tables
Set Scalar Variables	DoTruncate, \${TruncateDW}, schema_name, \${schema_DW}, Comment, Dimensions
Set Grid Variables	Tables, grid

## Truncate Facts

Run Orchestration

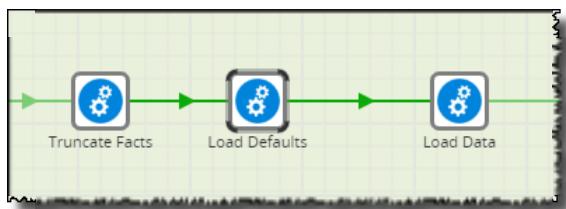


[Jump to job](#)

Parameter	Value
Orchestration Job	Truncate Tables
Set Scalar Variables	DoTruncate, \${TruncateDW}, schema_name, \${schema_DW}, Comment, Facts
Set Grid Variables	Tables, grid

## Load Defaults

Run Orchestration

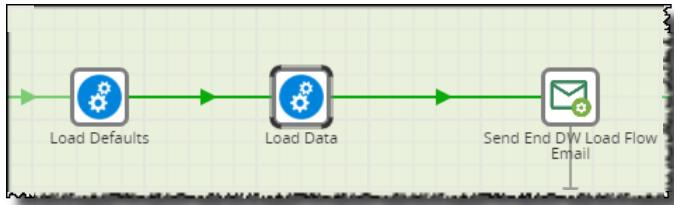


[Jump to job](#)

Parameter	Value
Orchestration Job	Load Defaults
Set Scalar Variables	
Set Grid Variables	Tables, grid

## Load Data

Run Orchestration

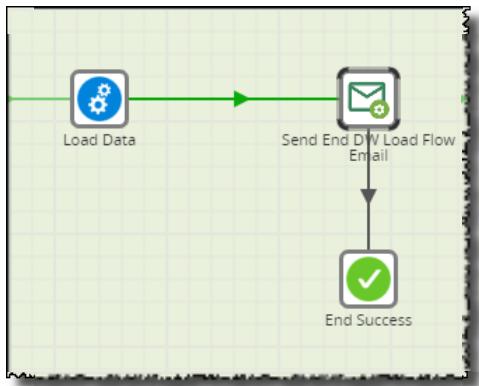


[Jump to job](#)

Parameter	Value
Orchestration Job	Load
Set Scalar Variables	
Set Grid Variables	

## Send End DW Load Flow Email

Send Email



Parameter	Value
To Recipients	#{EMAIL_DESTINATION_ADDRESS}
Cc Recipients	
Subject	#{project_name} Flow Complete
Message	Project Name: #{project_name}
Environment:	-- Name: #{environment_name} -- Default schema: #{environment_default_schema} DW schema: #{schema_DW} STG schema: #{schema_STG}
Steps:	

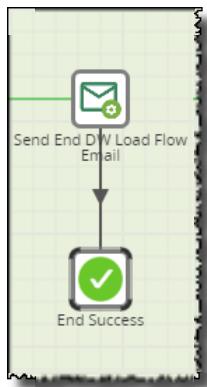
```

-- Truncate Stg: ${TruncateStg}
-- Truncate DW: ${TruncateDW}
-- Load Defaults: ${LoadDefaults}
-- Load Staging: ${LoadStaging}
Load Dimensions: ${LoadDimensions}
Load Policy Transactions: ${LoadPolicyTransactions}
Load Policy Summaries: ${LoadPolicySummaries}
--
```

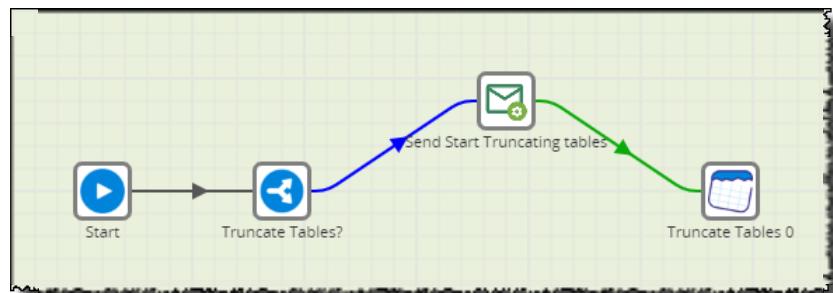
Sender Address	`\${EMAIL_SENDER_ADDRESS}`
Reply Address	
SMTP Username	`\${EMAIL_AUTHENTICATION_USER}`
SMTP Password	*****
SMTP Hostname	`\${EMAIL_SMTP_SERVER}`
SMTP Port	`\${EMAIL_SMTP_PORT}`
Enable SSL/TLS	Yes
Enable StartTLS	Yes

## End Success

End Success



## Job: Truncate Tables



Variables:

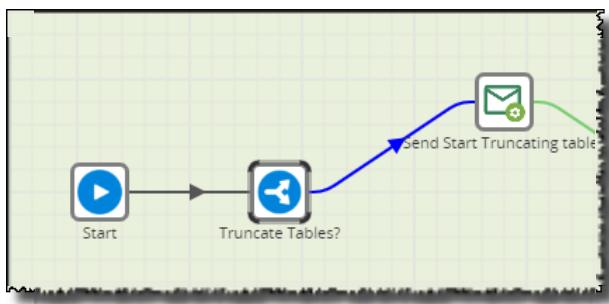
Name	Visibility	Behaviour	Type	Default Value	Description
Comment	Public	Shared	Text	What to truncate	
DoTruncate	Public	Shared	Text	N	
schema_name	Public	Shared	Text	kdlab	

Grid variables:

Name	Visibility	Behaviour	Column Names	Column Types	Description
Tables	Public	Copied	[table_name]	[Text]	List of tables to truncate

## Truncate Tables?

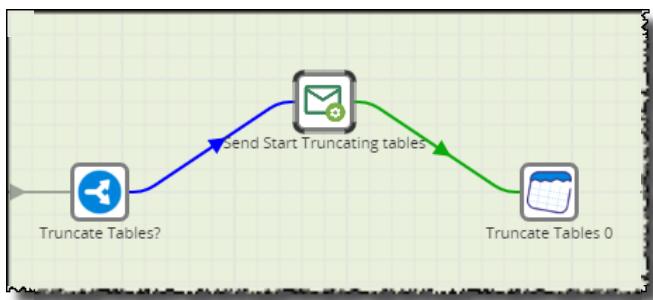
If



Parameter	Value
Mode	Simple
Condition	DoTruncate, Is, Equal to, Y
Combine Conditions	And

## Send Start Truncating tables

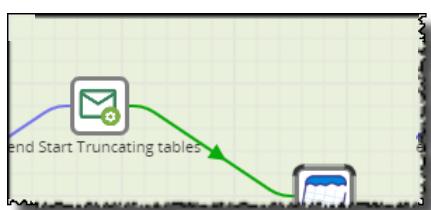
Send Email



Parameter	Value
To Recipients	\${EMAIL_DESTINATION_ADDRESS}
Cc Recipients	
Subject	\${project_name} Truncating
Message	Project Name: \${project_name}
	Environment: -- Name: \${environment_name} -- Default schema: \${environment_default_schema} DW schema: \${schema_DW} STG schema: \${schema_STG}
	Steps: -- Truncate Stg: \${TruncateStg} Truncate DW: \${TruncateDW} -- Load Defaults: \${LoadDefaults} -- Load Staging: \${LoadStaging} Load Dimensions: \${LoadDimensions} Load Policy Transactions: \${LoadPolicyTransactions} Load Policy Summaries: \${LoadPolicySummaries} -- Truncating: \${Comment}
Sender Address	\${EMAIL_SENDER_ADDRESS}
Reply Address	
SMTP Username	\${EMAIL_AUTHENTICATION_USER}
SMTP Password	*****
SMTP Hostname	\${EMAIL_SMTP_SERVER}
SMTP Port	\${EMAIL_SMTP_PORT}
Enable SSL/TLS	Yes
Enable StartTLS	Yes

## Truncate Tables 0

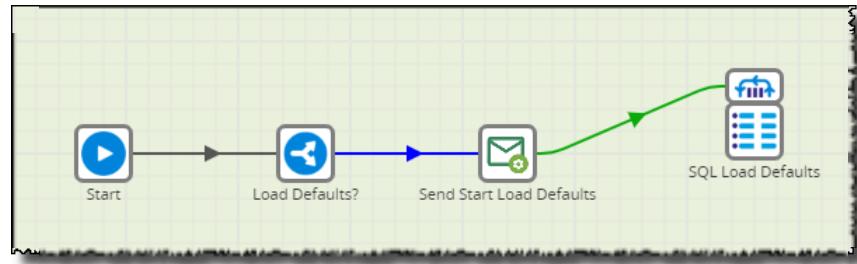
Truncate Table





Parameter	Value
Schema	\${schema_name}
Tables to Truncate	Tables, table_name

## Job: Load Defaults



Variables:

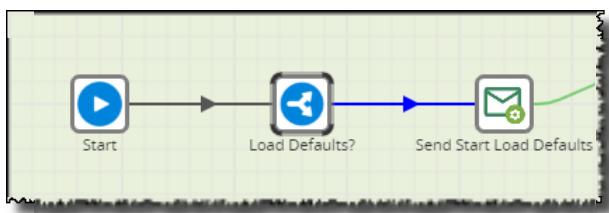
Name	Visibility	Behaviour	Type	Default Value	Description
table_name_to_load_defaults	Public	Shared	Text		

Grid variables:

Name	Visibility	Behaviour	Column Names	Column Types	Description
Tables	Public	Copied	[table_name]	[Text]	List of tables to load default values

## Load Defaults?

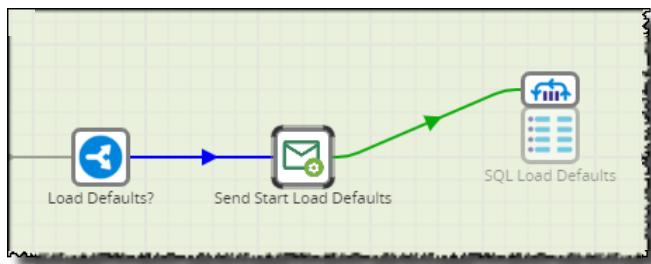
If



Parameter	Value
Mode	Simple
Condition	LoadDefaults, Is, Equal to, Y
Combine Conditions	And

## Send Start Load Defaults

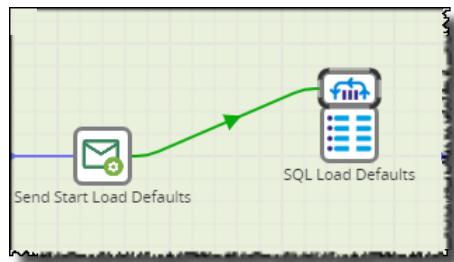
Send Email



Parameter	Value
To Recipients	<code> \${EMAIL_DESTINATION_ADDRESS}</code>
Cc Recipients	
Subject	<code> \${project_name} Load Defaults</code>
Message	<p>Project Name: <code> \${project_name}</code></p> <p>Environment:</p> <p>--</p> <p>Name: <code> \${environment_name}</code></p> <p>--</p> <p>Default schema: <code> \${environment_default_schema}</code></p> <p>DW schema: <code> \${schema_DW}</code></p> <p>STG schema: <code> \${schema_STG}</code></p> <p>Steps:</p> <p>--</p> <p>Truncate Stg: <code> \${TruncateStg}</code></p> <p>Truncate DW: <code> \${TruncateDW}</code></p> <p>--</p> <p>Load Defaults: <code> \${LoadDefaults}</code></p> <p>--</p> <p>Load Staging: <code> \${LoadStaging}</code></p> <p>Load Dimensions: <code> \${LoadDimensions}</code></p> <p>Load Policy Transactions: <code> \${LoadPolicyTransactions}</code></p> <p>Load Policy Summaries: <code> \${LoadPolicySummaries}</code></p> <p>--</p>
Sender Address	<code> \${EMAIL_SENDER_ADDRESS}</code>
Reply Address	
SMTP Username	<code> \${EMAIL_AUTHENTICATION_USER}</code>
SMTP Password	*****
SMTP Hostname	<code> \${EMAIL_SMTP_SERVER}</code>
SMTP Port	<code> \${EMAIL_SMTP_PORT}</code>
Enable SSL/TLS	Yes
Enable StartTLS	Yes

## Grid Iterator 0

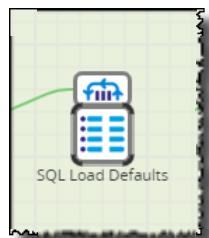
Grid Iterator



Parameter	Value
Grid Variable	Tables
Grid Variable Mapping	table_name, table_name_to_load_defaults
Break on Failure	No
Concurrency	Sequential
Record Values In Task History	Yes
Stop on Condition	No

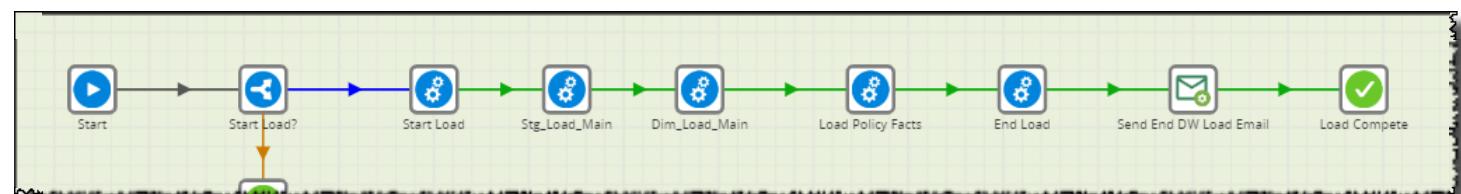
## SQL Load Defaults

SQL Script



Parameter	Value
SQL Script	call kdlab.sp_load_defaults('\${schema_DW}', '\${table_name_to_load_defaults}')

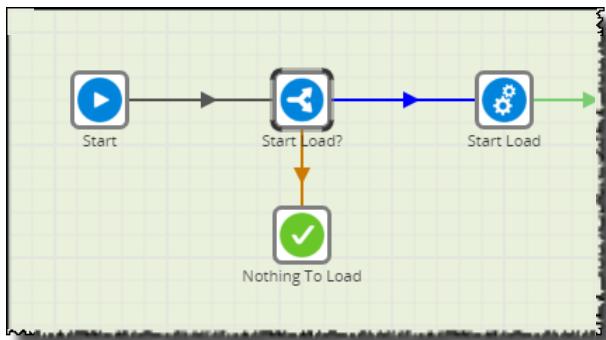
## Job: Load





## Start Load?

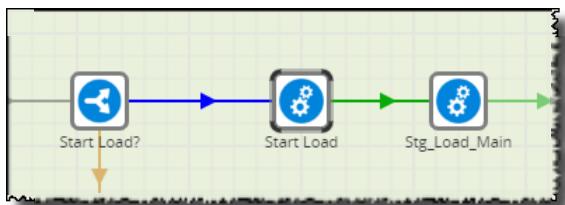
If



Parameter	Value
Mode	Simple
Condition	LoadStaging, Is, Equal to, Y, LoadDimensions, Is, Equal to, Y, LoadPolicyTransactions, Is, Equal to, Y, LoadPolicySummaries, Is, Equal to, Y
Combine Conditions	Or

## Start Load

Run Orchestration

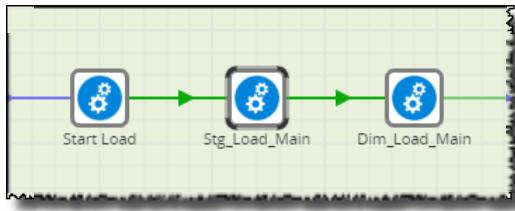


[Jump to job](#)

Parameter	Value
Orchestration Job	Start Load
Set Scalar Variables	
Set Grid Variables	

## Stg\_Load\_Main

Run Orchestration

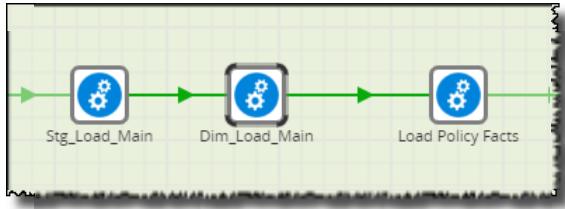


[Jump to job](#)

Parameter	Value
Orchestration Job	Stg_Load_Main
Set Scalar Variables	
Set Grid Variables	

## Dim\_Load\_Main

Run Orchestration

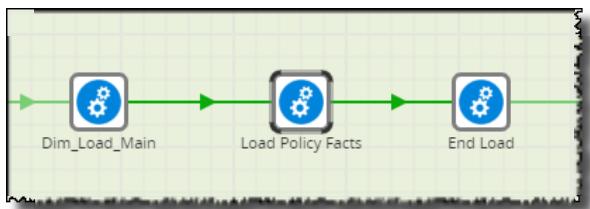


[Jump to job](#)

Parameter	Value
Orchestration Job	Dim_Load_Main
Set Scalar Variables	
Set Grid Variables	

## Load Policy Facts

Run Orchestration



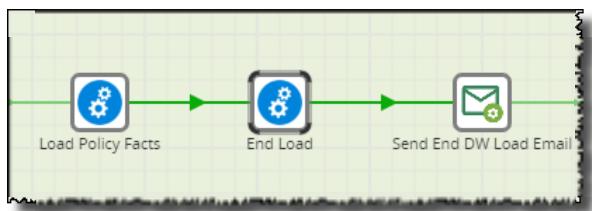
[Jump to job](#)

Parameter	Value
Orchestration Job	Load Policy Facts
Set Scalar Variables	
Set Grid Variables	

---

## End Load

Run Orchestration



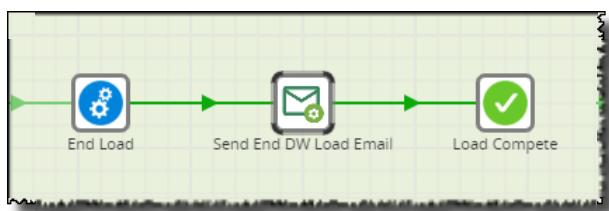
[Jump to job](#)

Parameter	Value
Orchestration Job	End Load
Set Scalar Variables	
Set Grid Variables	

---

## Send End DW Load Email

Send Email



Parameter	Value
To Recipients	\${EMAIL_DESTINATION_ADDRESS}
Cc Recipients	
Subject	\${project_name} Load Complete
Message	Loaddate is \${loaddate} SPINN EOD: \${is_SPINN_EOD_Complete} Project Name: \${project_name}

```

Environment:
--
Name: ${environment_name}
--
Default schema: ${environment_default_schema}
DW schema: ${schema_DW}
STG schema: ${schema_STG}

Steps:
--
Truncate Stg: ${TruncateStg}
Truncate DW: ${TruncateDW}
--
Load Defaults: ${LoadDefaults}
--
Load Staging: ${LoadStaging}
Load Dimensions: ${LoadDimensions}
Load Policy Transactions: ${LoadPolicyTransactions}
Load Policy Summaries: ${LoadPolicySummaries}
--
--
```

Sender Address	\${EMAIL_SENDER_ADDRESS}
----------------	--------------------------

Reply Address	
---------------	--

SMTP Username	\${EMAIL_AUTHENTICATION_USER}
---------------	-------------------------------

SMTP Password	*****
---------------	-------

SMTP Hostname	\${EMAIL_SMTP_SERVER}
---------------	-----------------------

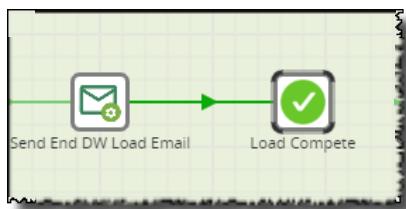
SMTP Port	\${EMAIL_SMTP_PORT}
-----------	---------------------

Enable SSL/TLS	Yes
----------------	-----

Enable StartTLS	Yes
-----------------	-----

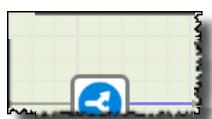
## Load Compete

End Success



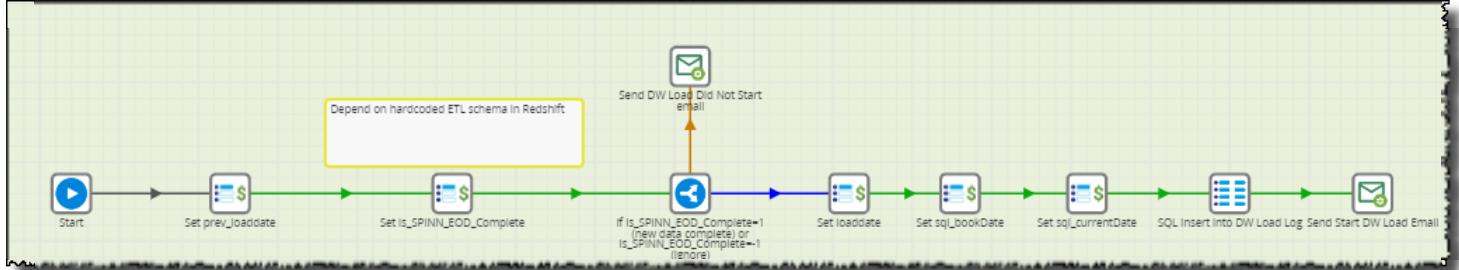
## Nothing To Load

End Success





## Job: Start Load

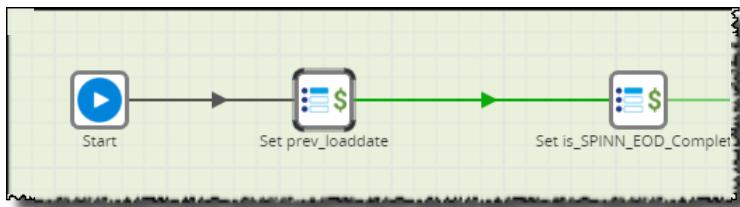


Set Up variables and Go/No Go Load based on SPInn logs completeness

Depend on hardcoded ETL schema in Redshift

## Set prev\_loaddate

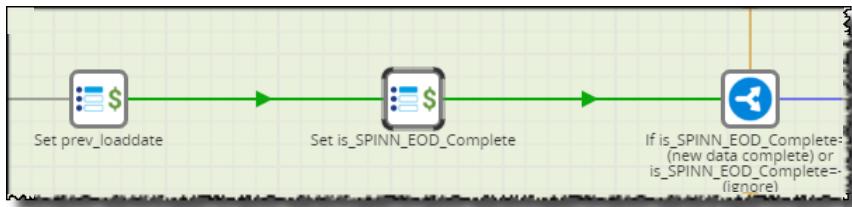
Query Result To Scalar



Parameter	Value
Basic / Advanced	Advanced
SQL Query	<pre>/* Note: this query will be run as a subquery for the purposes of sampling the data. As such it will be restricted by any subquery limitations specified by the current platform. */ select max(prev_loaddate) prev_loaddate from ( select max(bookdate) prev_loaddate from \${schema_STG}.etl_log union all select to_date('1900-01-01','yyyy-mm-dd') prev_loaddate ) data</pre>
Scalar Variable Mapping	prev_loaddate, prev_loaddate

## Set is\_SPINN\_EOD\_Complete

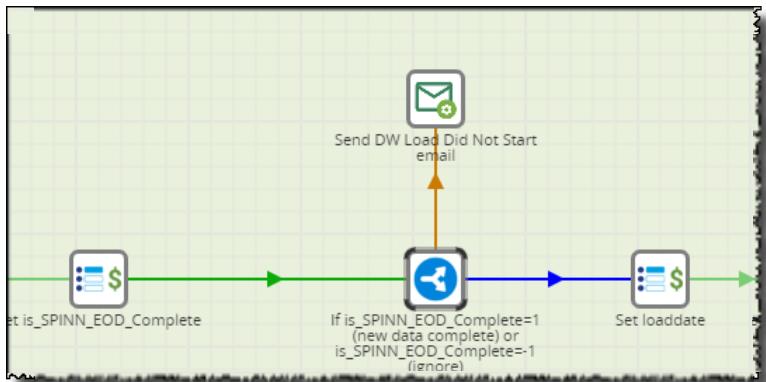
Query Result To Scalar



Parameter	Value
Basic / Advanced	Advanced
SQL Query	<pre>/* Note: this query will be run as a subquery for the purposes of sampling the data. As such it will be restricted by any subquery limitations specified by the current platform. */ with data as ( select distinct 1 JobDone from etl.LogsStatus where Updatetimestamp&gt;coalesce(cast('\${prev_loaddate}' as datetime),'1900-01-01') and JobStatusDaily='Completed' and JobStatusIncrement='Completed' union all select 0 JobDone ) select max(JobDone) JobDone from data</pre>
Scalar Variable Mapping	is_SPINN_EOD_Complete, jobdone

If is\_SPINN\_EOD\_Complete=1 (new data complete) or is\_SPINN\_EOD\_Complete=-1 (ignore)

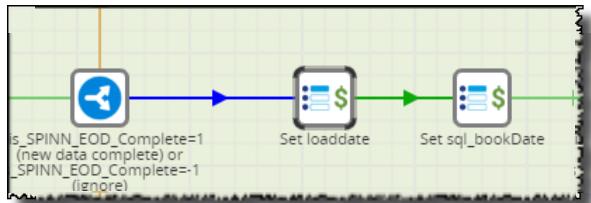
If



Parameter	Value
Mode	Simple
Condition	is_SPINN_EOD_Complete, Is, Equal to, 1, is_SPINN_EOD_Complete, Is, Equal to, -1
Combine Conditions	Or

Set loaddate

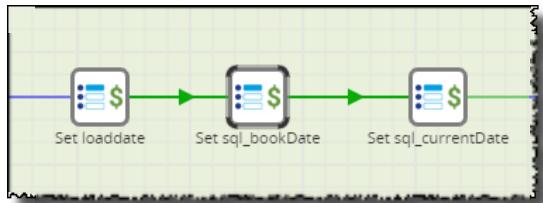
## Query Result To Scalar



Parameter	Value
Basic / Advanced	Advanced
SQL Query	<pre> /* Note: this query will be run as a subquery for the purposes of sampling the data. As such it will be restricted by any subquery limitations specified by the current platform. */ /*It's important to convert into PST because SPInn log time is in PST and use US/Pacific to account for Daylight Saving Time*/ select case when '\${loaddate}'=to_date('1900-01-01','yyyy-mm-dd') then convert_timezone('UTC','US/Pacific', GetDate()) else '\${loaddate}' end LoadDate </pre>
Scalar Variable Mapping	loaddate, loaddate

## Set sql\_bookDate

### Query Result To Scalar



Parameter	Value
Basic / Advanced	Advanced
SQL Query	<pre> /* Note: this query will be run as a subquery for the purposes of sampling the data. As such it will be restricted by any subquery limitations specified by the current platform. */ /*It's important to convert into PST because SPInn log time is in PST and use US/Pacific to account for Daylight Saving Time*/  /* *-----* /*BookDt &gt; sql_bookDate and BookDt &lt;= sql_currentDate */ /* *-----*  select max(sql_bookDate) sql_bookDate from ( select case when '\${sql_bookDate}'=to_date('1900-01-01','yyyy-mm-dd') then --if not set in variables then the latest loaded from the log bookdate </pre>

```

else
'${sql_bookDate}'
end sql_bookDate
from ${schema_STG}.etl_log
union all
select '${sql_bookDate}' sql_bookDate
) data

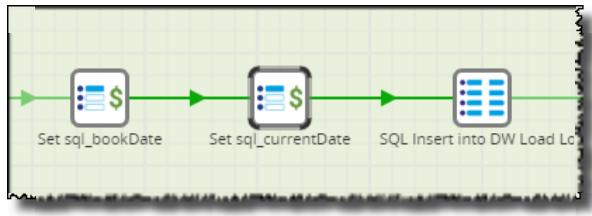
```

Scalar Variable Mapping

sql\_bookDate, sql\_bookDate

## Set sql\_currentDate

Query Result To Scalar



### Parameter

### Value

Basic / Advanced

Advanced

### SQL Query

```

/* Note: this query will be run as a subquery for the purposes of sampling the data.
As such it will be restricted by any subquery limitations specified by the current platform. */
/*It's important to convert into PST because SPIinn log time is in PST and use US/Pacific to account for
Daylight Saving Time*/
/*
*-----*/
/*BookDt > sql_bookDate and BookDt <= sql_currentDate */
/*
*-----*/
select
case
when '${sql_currentDate}'=to_date('1900-01-01','yyyy-mm-dd') then /*if sql_currentDate is not set then
Yesterday*/
dateadd(day,-1,convert_timezone('UTC','US/Pacific', GetDate()) )
else
'${sql_currentDate}'
end sql_currentDate

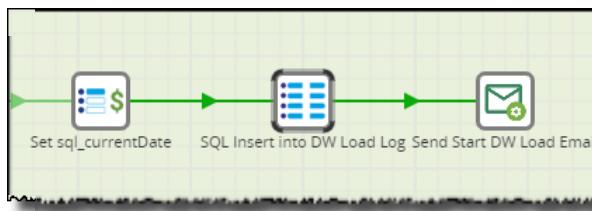
```

Scalar Variable Mapping

sql\_currentDate, sql\_currentDate

## SQL Insert into DW Load Log

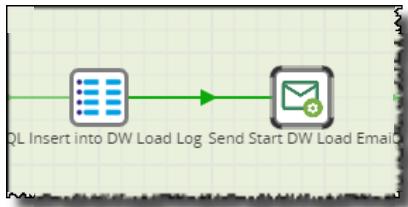
SQL Script



Parameter	Value
SQL Script	<pre>/* *-----* /*BookDt &gt; sql_bookDate and BookDt &lt;= sql_currentDate */ /* According to this condition in every day load */ /* YESTERDAY bookdate is in sql_currentDate */ *-----*/</pre> <p>insert into \${schema_STG}.etl_log (loaddate, bookdate,loadstarttime,loadcomments) values ('\${loaddate}', '\${sql_currentDate}', GetDate(), '\${sql_bookDate}');</p>

## Send Start DW Load Email

Send Email

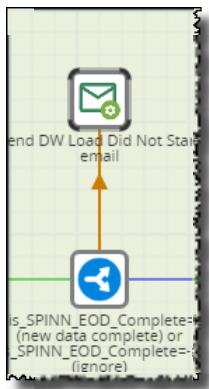


Parameter	Value
To Recipients	\${EMAIL_DESTINATION_ADDRESS}
Cc Recipients	
Subject	\${project_name} Data Load Started
Message	<p>Loaddate is \${loaddate}          SPINN EOD: \${is_SPINN_EOD_Complete}</p> <p>Project Name: \${project_name}</p> <p>Environment:</p> <p>--</p> <p>Name: \${environment_name}</p> <p>--</p> <p>Default schema: \${environment_default_schema}</p> <p>DW schema: \${schema_DW}</p> <p>STG schema: \${schema_STG}</p> <p>Steps:</p> <p>--</p> <p>Truncate Stg: \${TruncateStg}</p> <p>Truncate DW: \${TruncateDW}</p> <p>--</p> <p>Load Defaults: \${LoadDefaults}</p> <p>--</p> <p>Load Staging: \${LoadStaging}</p> <p>Load Dimensions: \${LoadDimensions}</p> <p>Load Policy Transactions: \${LoadPolicyTransactions}</p> <p>Load Policy Summaries: \${LoadPolicySummaries}</p> <p>--</p>
Sender Address	\${EMAIL_SENDER_ADDRESS}
Reply Address	

SMTP Username	\${EMAIL_AUTHENTICATION_USER}
SMTP Password	*****
SMTP Hostname	\${EMAIL_SMTP_SERVER}
SMTP Port	\${EMAIL_SMTP_PORT}
Enable SSL/TLS	Yes
Enable StartTLS	Yes

## Send DW Load Did Not Start email

Send Email



Parameter	Value
To Recipients	\${EMAIL_DESTINATION_ADDRESS}
Cc Recipients	
Subject	\${project_name} Data Load did NOT Start
Message	<p>SPInn EOD not complete or complete with errors</p> <p>SPINN EOD: \${is_SPINN_EOD_Complete}</p> <p>Project Name: \${project_name}</p> <p>Environment:</p> <p>--</p> <p>Name: \${environment_name}</p> <p>--</p> <p>Default schema: \${environment_default_schema}</p> <p>DW schema: \${schema_DW}</p> <p>STG schema: \${schema_STG}</p> <p>Steps:</p> <p>--</p> <p>Truncate Stg: \${TruncateStg}</p> <p>Truncate Dims: \${TruncateDims}</p> <p>--</p> <p>Load Defaults: \${LoadDefaults}</p> <p>--</p>

```

Load Staging: ${LoadStaging}
Load Dimensions: ${LoadDimensions}
Load Policy Transactions: ${LoadPolicyTransactions}
Load Policy Summaries: ${LoadPolicySummaries}
--
```

Sender Address	\${EMAIL_SENDER_ADDRESS}
----------------	--------------------------

Reply Address	
---------------	--

SMTP Username	\${EMAIL_AUTHENTICATION_USER}
---------------	-------------------------------

SMTP Password	*****
---------------	-------

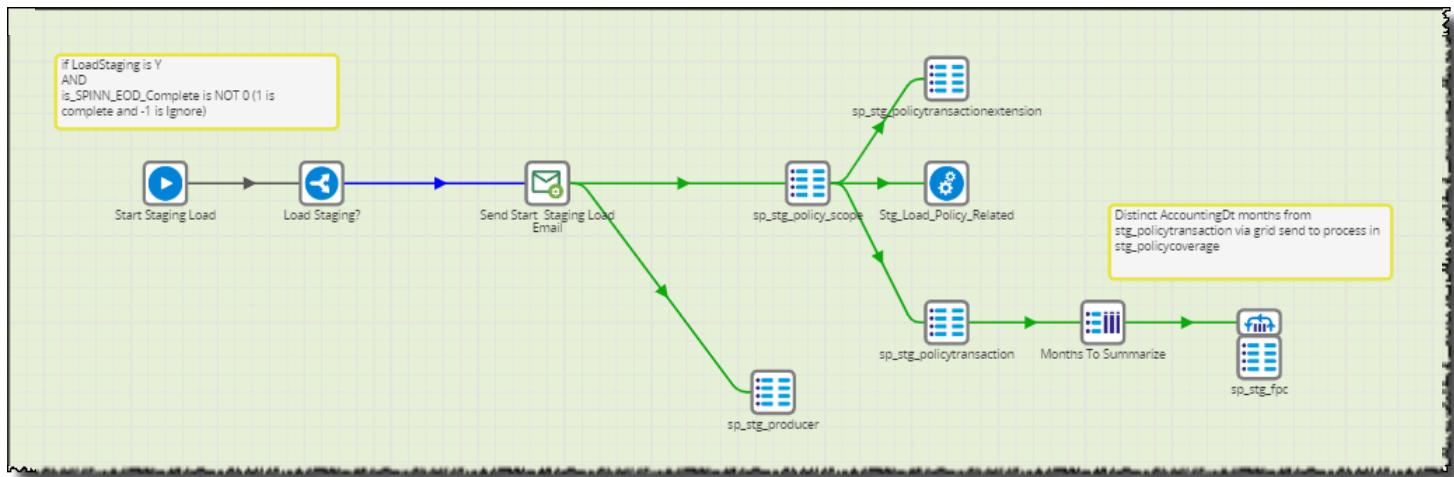
SMTP Hostname	\${EMAIL_SMTP_SERVER}
---------------	-----------------------

SMTP Port	\${EMAIL_SMTP_PORT}
-----------	---------------------

Enable SSL/TLS	Yes
----------------	-----

Enable StartTLS	Yes
-----------------	-----

## Job: Stg\_Load\_Main



Variables:

Name	Visibility	Behaviour	Type	Default Value	Description
pmonth_id	Private	Copied	Numeric		

Grid variables:

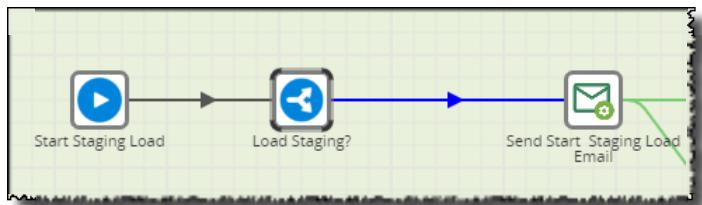
Name	Visibility	Behaviour	Column Names	Column Types	Description
Months	Private	Copied	[month_id]	[Numeric]	Months in the current stg_policytransaction to populate stg and fact_policycoverage

```
if LoadStaging is Y  
AND  
is_SPINN_EOD_Complete is NOT 0 (1 is complete and -1 is Ignore)
```

Distinct AccountingDt months from stg\_policytransaction via grid send to process in stg\_policycoverage

## Load Staging?

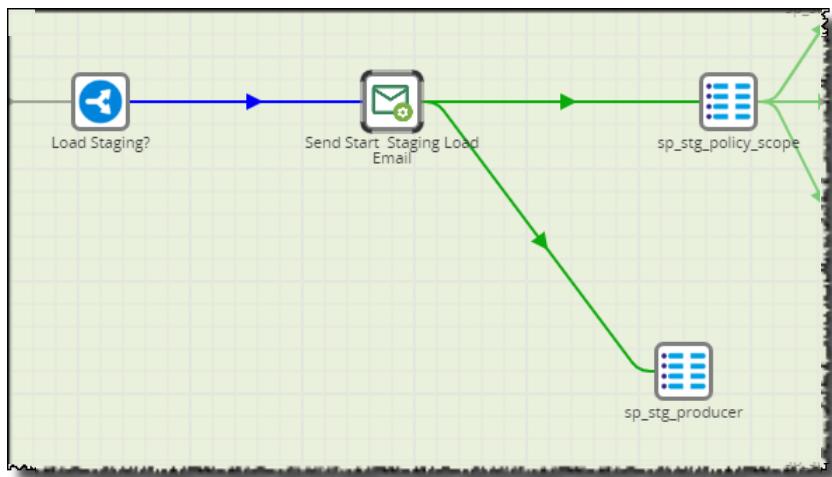
If



Parameter	Value
Mode	Simple
Condition	LoadStaging, Is, Equal to, Y, is_SPINN_EOD_Complete, Not, Equal to, 0
Combine Conditions	And

## Send Start Staging Load Email

Send Email

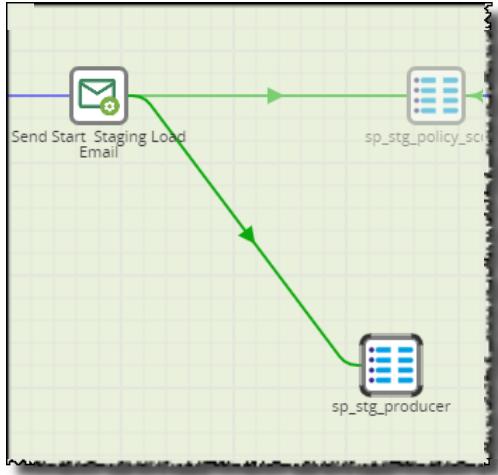


Parameter	Value
-----------	-------

To Recipients	<code> \${EMAIL_DESTINATION_ADDRESS}</code>
Cc Recipients	
Subject	<code> \${project_name} Load of Staging Started</code>
Message	Load of Staging Started
Sender Address	<code> \${EMAIL_SENDER_ADDRESS}</code>
Reply Address	
SMTP Username	<code> \${EMAIL_AUTHENTICATION_USER}</code>
SMTP Password	*****
SMTP Hostname	<code> \${EMAIL_SMTP_SERVER}</code>
SMTP Port	<code> \${EMAIL_SMTP_PORT}</code>
Enable SSL/TLS	Yes
Enable StartTLS	Yes

## sp\_stg\_producer

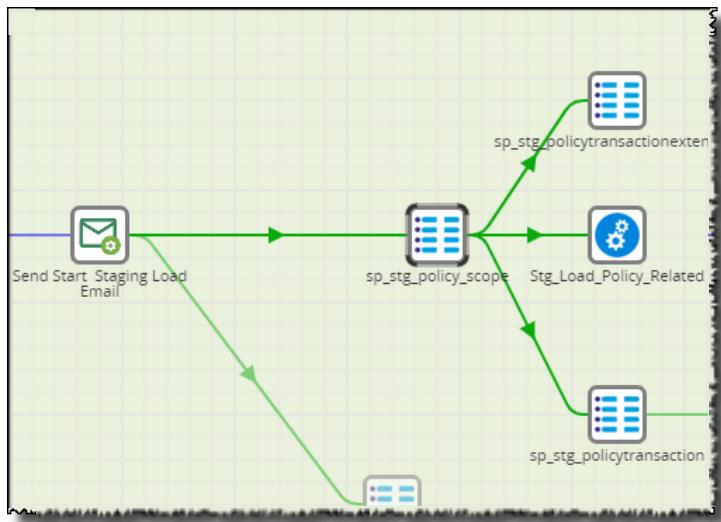
SQL Script



Parameter	Value
SQL Script	<code>call \${schema_STG}.sp_stg_producer( to_date('\${loaddate}', 'yyyy-mm-dd'))</code>

## sp\_stg\_policy\_scope

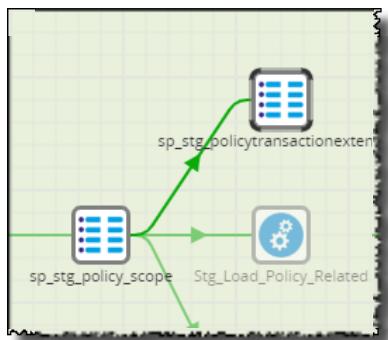
SQL Script



Parameter	Value
SQL Script	call \${schema_STG}.sp_stg_policy_scope(to_date('\${sql_bookDate}', 'yyyy-mm-dd'), to_date('\${sql_currentDate}', 'yyyy-mm-dd'), to_date('\${loaddate}', 'yyyy-mm-dd'))

## sp\_stg\_policytransactionextension

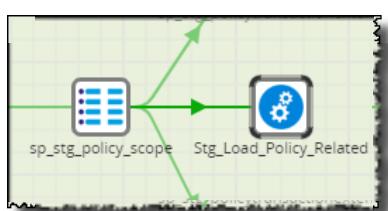
SQL Script



Parameter	Value
SQL Script	call \${schema_STG}.sp_stg_policytransactionextension(to_date('\${sql_bookDate}', 'yyyy-mm-dd'), to_date('\${sql_currentDate}', 'yyyy-mm-dd'), to_date('\${loaddate}', 'yyyy-mm-dd'))

## Stg\_Load\_Policy\_Related

Run Orchestration

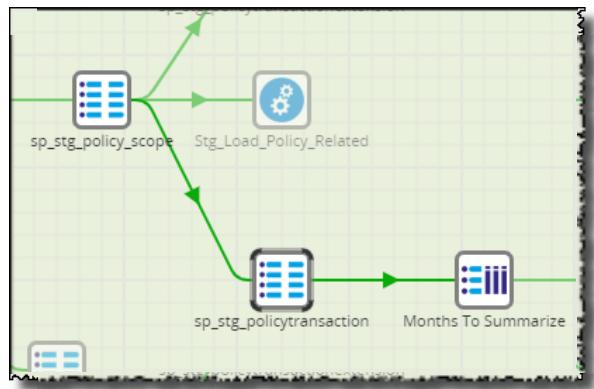


[Jump to job](#)

Parameter	Value
Orchestration Job	Stg_Load_Policy_Related
Set Scalar Variables	
Set Grid Variables	

## sp\_stg\_policytransaction

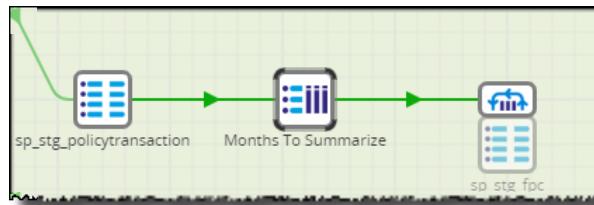
SQL Script



Parameter	Value
SQL Script	call \${schema_STG}.sp_stg_policytransaction(to_date('\${sql_bookDate}', 'yyyy-mm-dd'), to_date('\${sql_currentDate}', 'yyyy-mm-dd'), to_date('\${loaddate}', 'yyyy-mm-dd'))

## Months To Summarize

Query Result To Grid



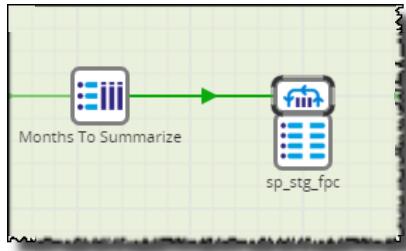
Parameter	Value
Basic / Advanced	Advanced
SQL Query	select distinct cast(to_char(accountingdt,'yyyymm') as int) month_id from kdlab.stg_policytransaction where cast(to_char(bookdt,'yyyymm') as int)>=cast(to_char(accountingdt,'yyyymm') as int) /*no future months, limit to the current and previous months*/
Grid Variable	Months

Grid Variable Mapping

month\_id

## Grid Iterator 0

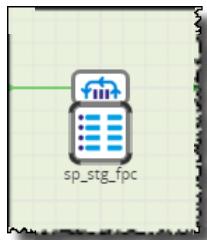
Grid Iterator



Parameter	Value
Grid Variable	Months
Grid Variable Mapping	month_id, pmonth_id
Break on Failure	No
Concurrency	Sequential
Record Values In Task History	Yes
Stop on Condition	No

## sp\_stg\_fpc

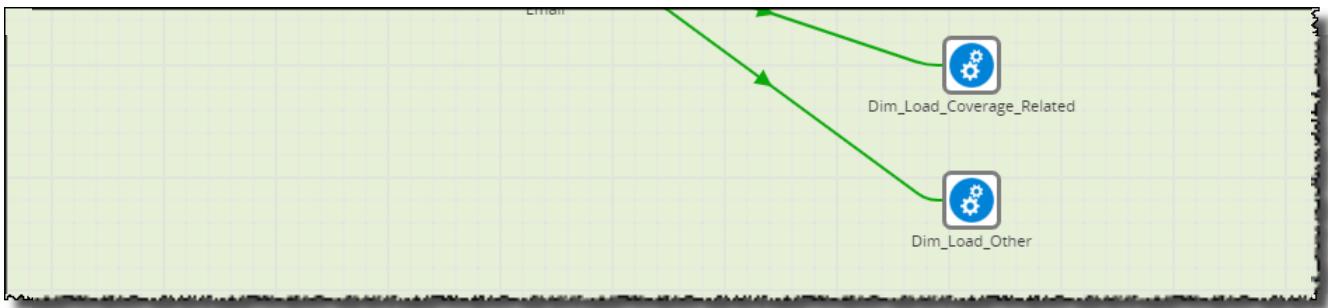
SQL Script



Parameter	Value
SQL Script	call kdlab.sp_stg_fpc(\${pmonth_id}, to_date('\${loaddate}', 'yyyy-mm-dd'))

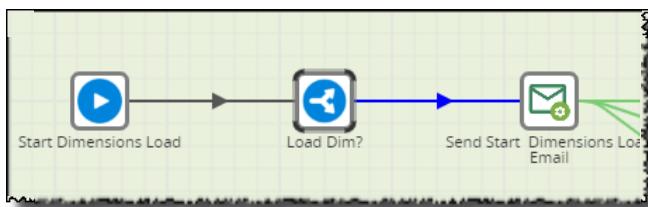
## Job: Dim\_Load\_Main





## Load Dim?

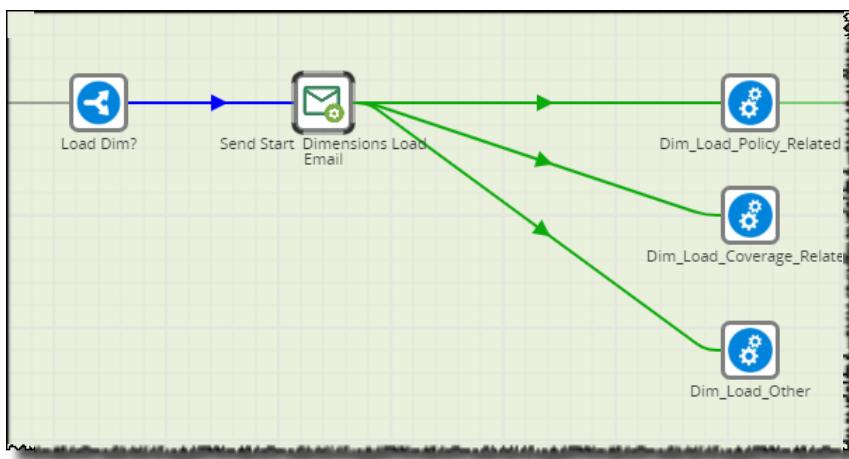
If



Parameter	Value
Mode	Simple
Condition	LoadDimensions, Is, Equal to, Y, is_SPINN_EOD_Complete, Not, Equal to, 0
Combine Conditions	And

## Send Start Dimensions Load Email

Send Email



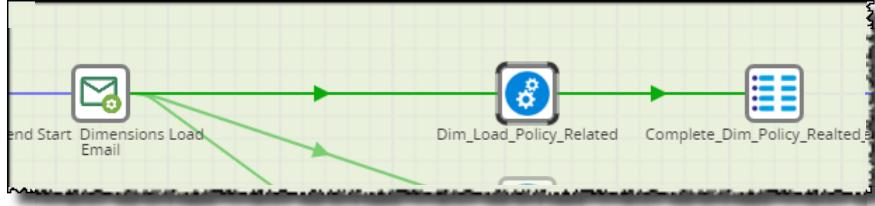
Parameter	Value
To Recipients	\${EMAIL_DESTINATION_ADDRESS}

## Cc Recipients

Subject	`\${project_name} Load of Dimensions Started
Message	Load of Dimensions Started
Sender Address	`\${EMAIL_SENDER_ADDRESS}
Reply Address	
SMTP Username	`\${EMAIL_AUTHENTICATION_USER}
SMTP Password	*****
SMTP Hostname	`\${EMAIL_SMTP_SERVER}
SMTP Port	`\${EMAIL_SMTP_PORT}
Enable SSL/TLS	Yes
Enable StartTLS	Yes

## Dim\_Load\_Policy\_Related

Run Orchestration



[Jump to job](#)

Parameter	Value
Orchestration Job	Dim_Load_Policy_Related
Set Scalar Variables	
Set Grid Variables	

## Complete\_Dim\_Policy\_Related\_Load

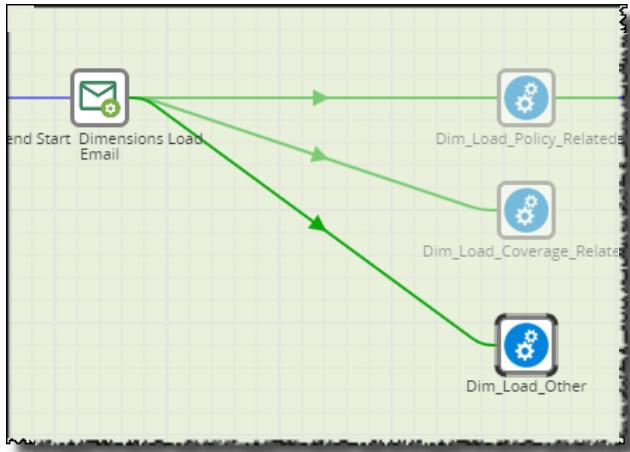
SQL Script



Parameter	Value
SQL Script	call \${schema_DW}.sp_load_complete()

## Dim\_Load\_Other

Run Orchestration

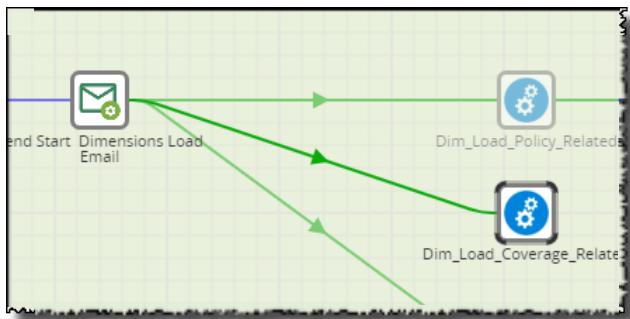


[Jump to job](#)

Parameter	Value
Orchestration Job	Dim_Load_Other
Set Scalar Variables	
Set Grid Variables	

## Dim\_Load\_Coverage\_Related

Run Orchestration



[Jump to job](#)

Parameter	Value
Orchestration Job	Dim_Load_Coverage_Related

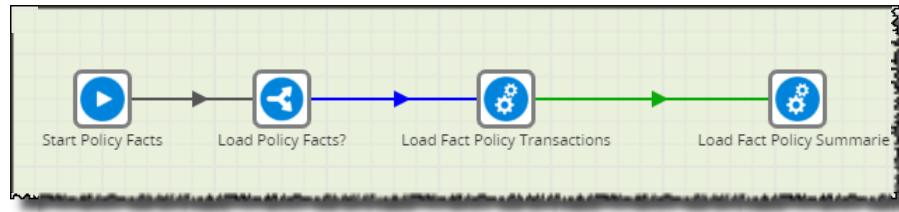
Orchestration Job

Dim\_Load\_Coverage\_Related

Set Scalar Variables

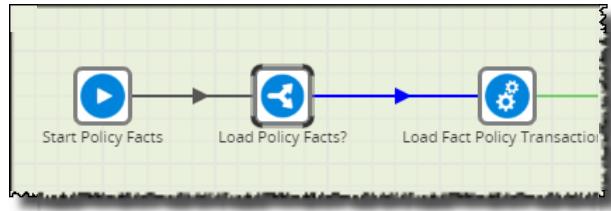
Set Grid Variables

## Job: Load Policy Facts



### Load Policy Facts?

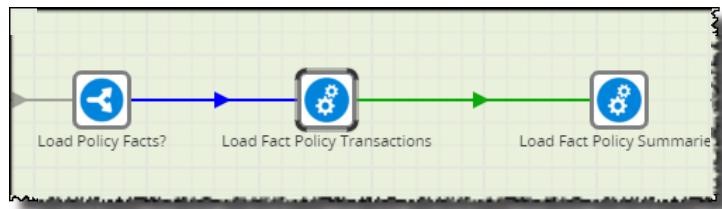
If



Parameter	Value
Mode	Advanced
Condition	((LoadPolicyTransactions == 'Y')   (LoadPolicySummaries == 'Y')) & (is_SPINN_EOD_Complete != 0)

## Load Fact Policy Transactions

Run Orchestration



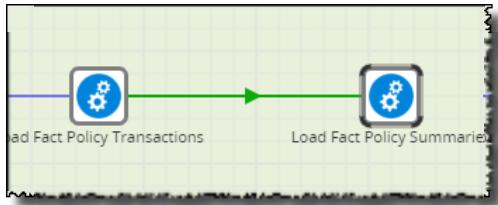
[Jump to job](#)

Parameter	Value
Orchestration Job	Load Fact Policy Transactions
Set Scalar Variables	

Set Grid Variables

## Load Fact Policy Summaries

Run Orchestration

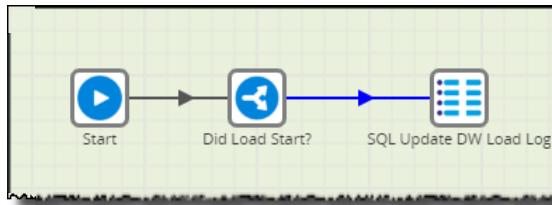


[Jump to job](#)

Parameter	Value
-----------	-------

Orchestration Job	Load Fact Policy Summaries
Set Scalar Variables	
Set Grid Variables	

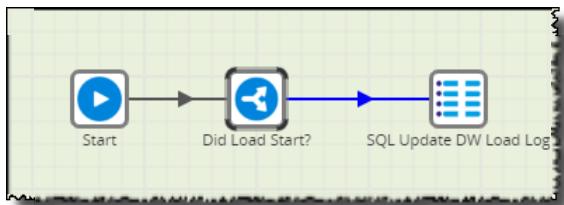
## Job: End Load



Update DW Log

## Did Load Start?

If



Parameter	Value
-----------	-------

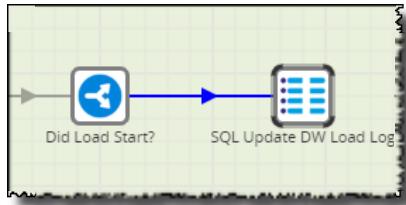
| Mode | Simple |
| Condition | is\_SPINN\_EOD\_Complete, Not, Equal to, 0 |

Combine Conditions

And

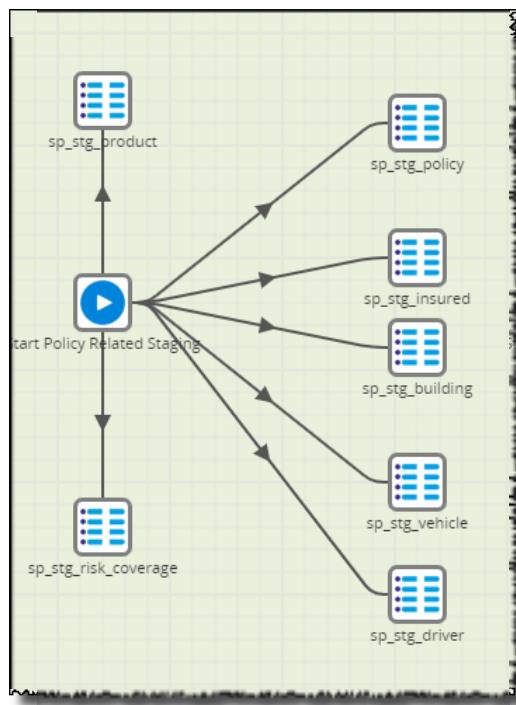
## SQL Update DW Load Log

SQL Script



Parameter	Value
SQL Script	update kdlab.etl_log set loadendtime=convert_timezone('UTC','US/Pacific', GetDate()) where loaddate='\${loaddate}'

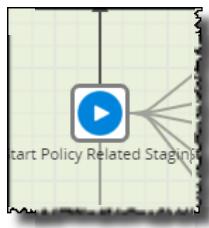
## Job: Stg\_Load\_Policy\_Related



## sp\_stg\_product

SQL Script

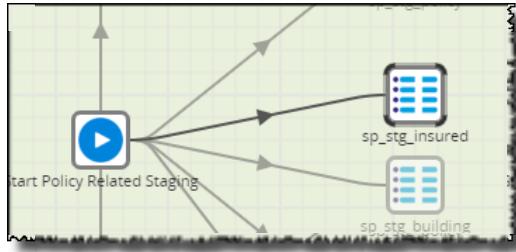




Parameter	Value
SQL Script	call \${schema_STG}.sp_stg_product(to_date('\${sql_bookDate}', 'yyyy-mm-dd'), to_date('\${sql_currentDate}', 'yyyy-mm-dd'), to_date('\${loaddate}', 'yyyy-mm-dd'))

## sp\_stg\_insured

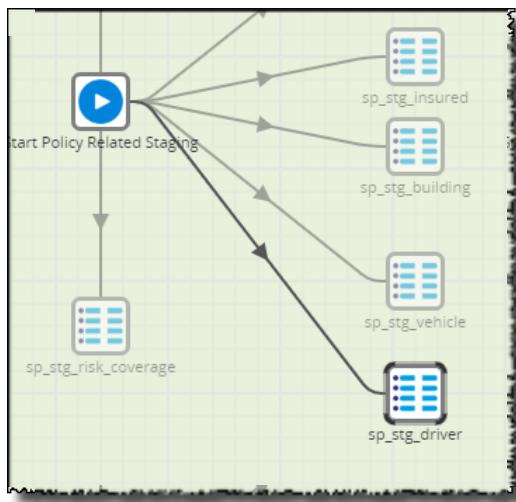
SQL Script



Parameter	Value
SQL Script	call \${schema_STG}.sp_stg_insured(to_date('\${sql_bookDate}', 'yyyy-mm-dd'), to_date('\${sql_currentDate}', 'yyyy-mm-dd'), to_date('\${loaddate}', 'yyyy-mm-dd'))

## sp\_stg\_driver

SQL Script



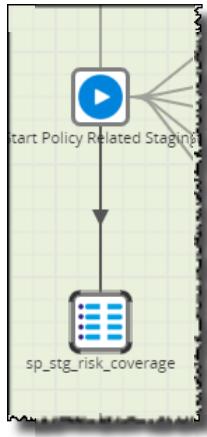
Parameter	Value
SQL Script	call \${schema_STG}.sp_stg_driver(to_date('\${sql_bookDate}', 'yyyy-mm-dd'), to_date('\${sql_currentDate}', 'yyyy-mm-dd'), to_date('\${loaddate}', 'yyyy-mm-dd'))

#### SQL Script

```
call ${schema_STG}.sp_stg_driver(to_date('${sql_bookDate}', 'yyyy-mm-dd'),  
to_date('${sql_currentDate}', 'yyyy-mm-dd'), to_date('${loaddate}', 'yyyy-mm-dd'))
```

### sp\_stg\_risk\_coverage

#### SQL Script



#### Parameter

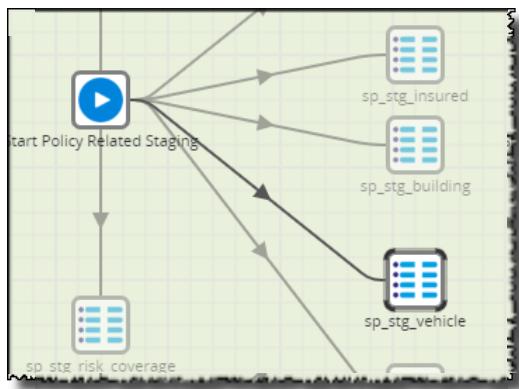
#### Value

#### SQL Script

```
call ${schema_STG}.sp_stg_risk_coverage(to_date('${sql_bookDate}', 'yyyy-mm-dd'),  
to_date('${sql_currentDate}', 'yyyy-mm-dd'), to_date('${loaddate}', 'yyyy-mm-dd'))
```

### sp\_stg\_vehicle

#### SQL Script



#### Parameter

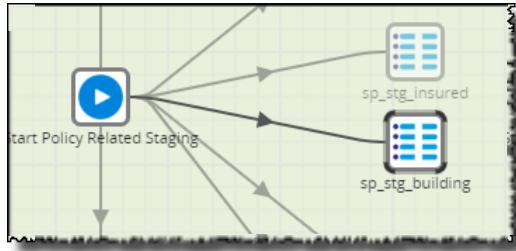
#### Value

#### SQL Script

```
call ${schema_STG}.sp_stg_vehicle(to_date('${sql_bookDate}', 'yyyy-mm-dd'),  
to_date('${sql_currentDate}', 'yyyy-mm-dd'), to_date('${loaddate}', 'yyyy-mm-dd'))
```

### sp\_stg\_building

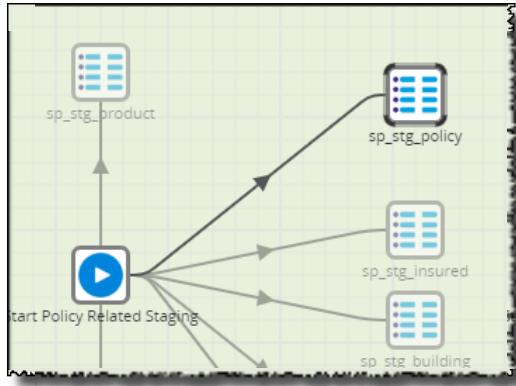
## SQL Script



Parameter	Value
SQL Script	call \${schema_STG}.sp_stg_building(to_date('\${sql_bookDate}', 'yyyy-mm-dd'), to_date('\${sql_currentDate}', 'yyyy-mm-dd'), to_date('\${loaddate}', 'yyyy-mm-dd'))

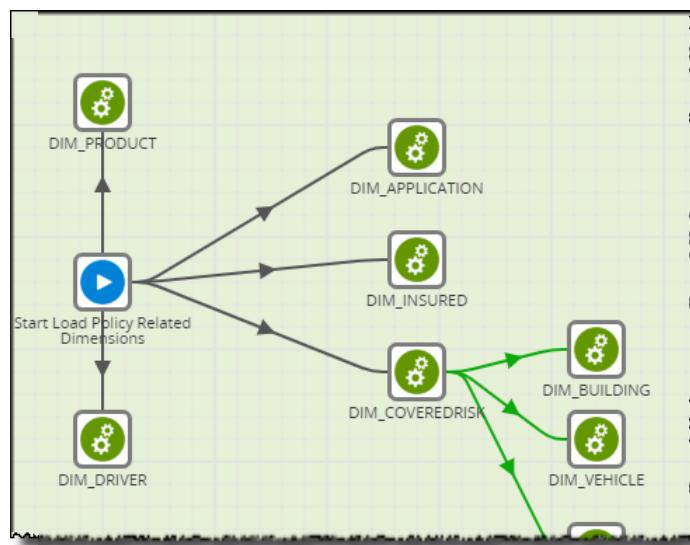
## sp\_stg\_policy

### SQL Script



Parameter	Value
SQL Script	call \${schema_STG}.sp_stg_policy(to_date('\${sql_bookDate}', 'yyyy-mm-dd'), to_date('\${sql_currentDate}', 'yyyy-mm-dd'), to_date('\${loaddate}', 'yyyy-mm-dd'))

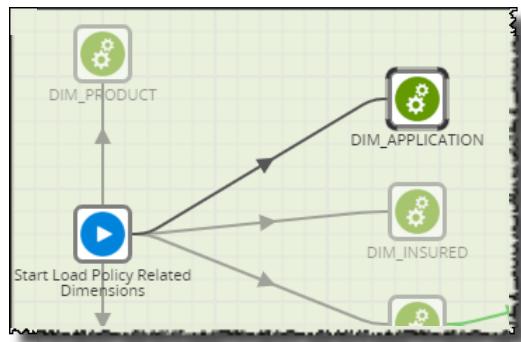
## Job: Dim\_Load\_Policy\_Related





## DIM\_APPLICATION

Run Transformation



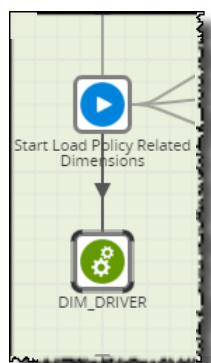
[Jump to job](#)

Parameter	Value
Transformation Job	DIM_APPLICATION
Set Scalar Variables	
Set Grid Variables	

---

## DIM\_DRIVER

Run Transformation



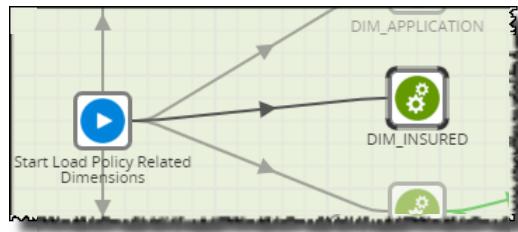
[Jump to job](#)

Parameter	Value
Transformation Job	DIM_DRIVER
Set Scalar Variables	

Set Grid Variables

## DIM\_INSURED

Run Transformation



[Jump to job](#)

Parameter	Value
Transformation Job	DIM_INSURED
Set Scalar Variables	
Set Grid Variables	

## DIM\_PRODUCT

Run Transformation

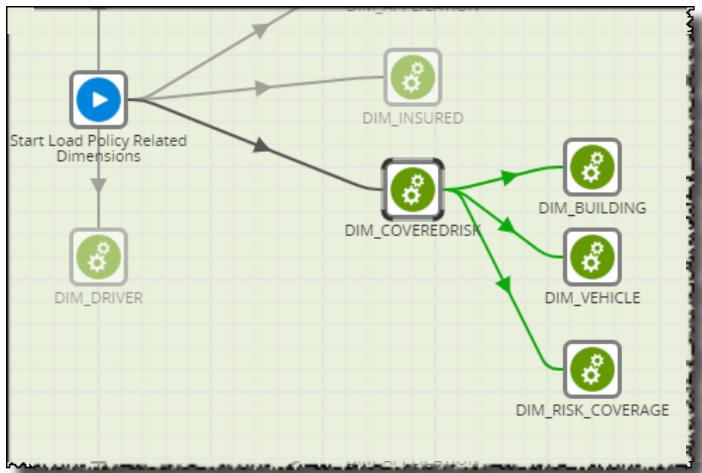


[Jump to job](#)

Parameter	Value
Transformation Job	DIM_PRODUCT
Set Scalar Variables	
Set Grid Variables	

## DIM\_COVEREDRISK

Run Transformation

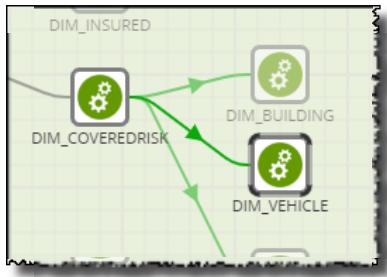


[Jump to job](#)

Parameter	Value
Transformation Job	DIM_COVEREDRISK
Set Scalar Variables	
Set Grid Variables	

## DIM\_VEHICLE

Run Transformation

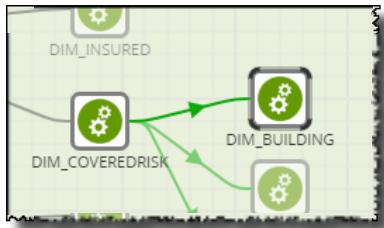


[Jump to job](#)

Parameter	Value
Transformation Job	DIM_VEHICLE
Set Scalar Variables	
Set Grid Variables	

## DIM\_BUILDING

Run Transformation

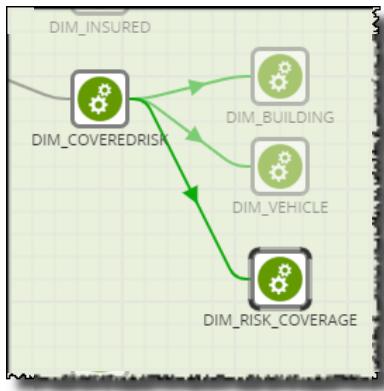


[Jump to job](#)

Parameter	Value
Transformation Job	DIM_BUILDING
Set Scalar Variables	
Set Grid Variables	

## DIM\_RISK\_COVERAGE

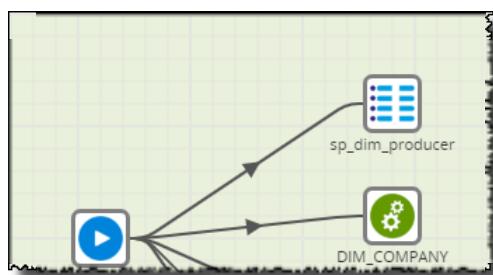
Run Transformation

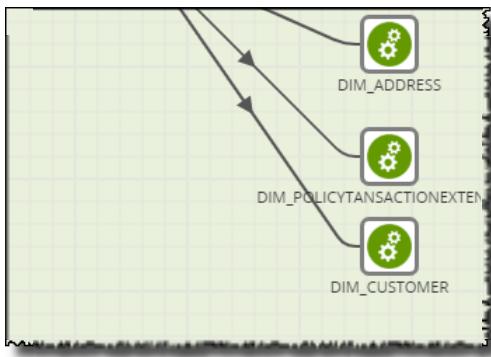


[Jump to job](#)

Parameter	Value
Transformation Job	DIM_RISK_COVERAGE
Set Scalar Variables	
Set Grid Variables	

## Job: Dim\_Load\_Other



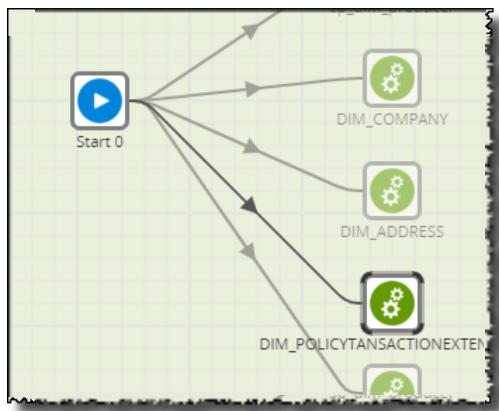


not directly related to a policy or coverage

---

## DIM\_POLICYACTIONEXTENSION

Run Transformation



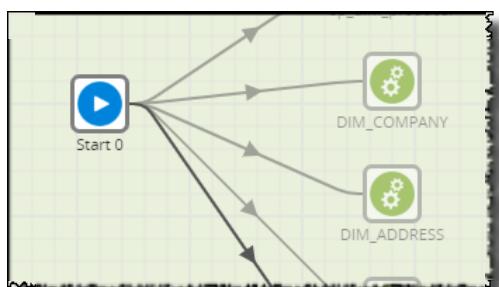
[Jump to job](#)

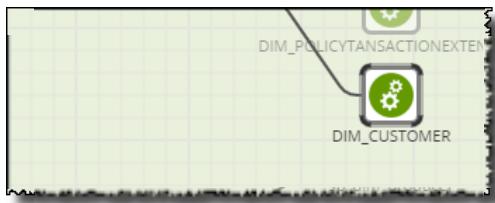
Parameter	Value
Transformation Job	DIM_POLICYACTIONEXTENSION
Set Scalar Variables	
Set Grid Variables	

---

## DIM\_CUSTOMER

Run Transformation





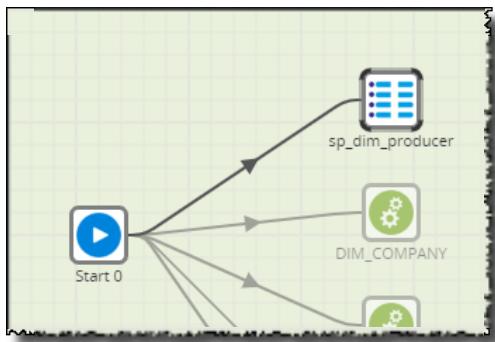
[Jump to job](#)

Parameter	Value
Transformation Job	DIM_CUSTOMER
Set Scalar Variables	
Set Grid Variables	

---

## sp\_dim\_producer

SQL Script

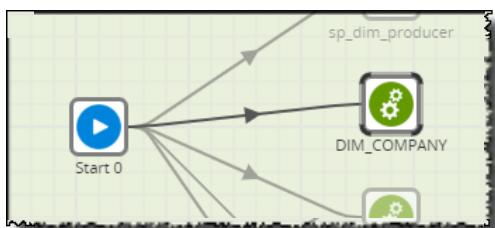


Parameter	Value
SQL Script	call \${schema_DW}.sp_dim_producer()

---

## DIM\_COMPANY

Run Transformation



[Jump to job](#)

Parameter	Value

Transformation Job

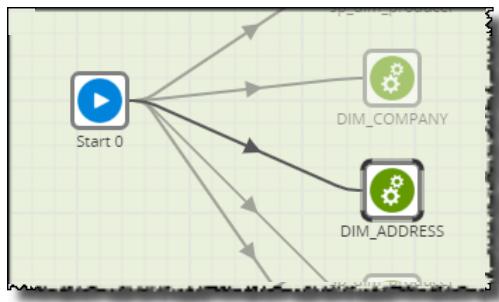
DIM\_COMPANY

Set Scalar Variables

Set Grid Variables

## DIM\_ADDRESS

Run Transformation



[Jump to job](#)

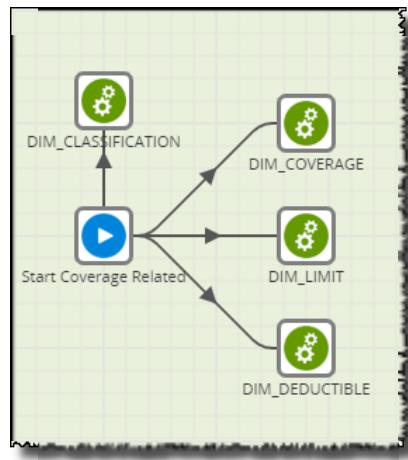
Parameter	Value
-----------	-------

Transformation Job DIM\_ADDRESS

Set Scalar Variables

Set Grid Variables

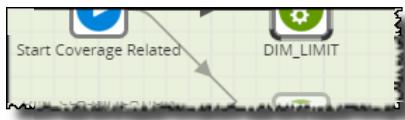
## Job: Dim\_Load\_Coverage\_Related



## DIM\_LIMIT

Run Transformation





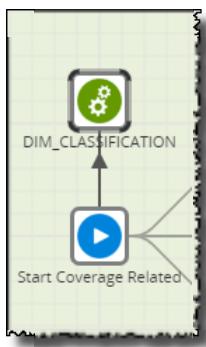
[Jump to job](#)

Parameter	Value
Transformation Job	DIM_LIMIT
Set Scalar Variables	
Set Grid Variables	

---

## DIM\_CLASSIFICATION

Run Transformation



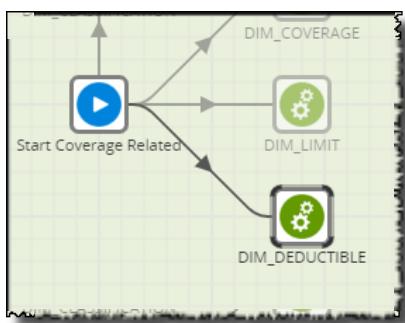
[Jump to job](#)

Parameter	Value
Transformation Job	DIM_CLASSIFICATION
Set Scalar Variables	
Set Grid Variables	

---

## DIM\_DEDUCTIBLE

Run Transformation



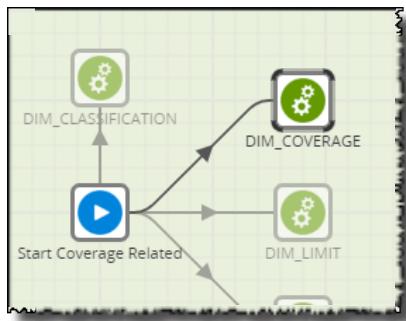
[Jump to job](#)

Parameter	Value
Transformation Job	DIM_DEDUCTIBLE
Set Scalar Variables	
Set Grid Variables	

---

## DIM\_COVERAGE

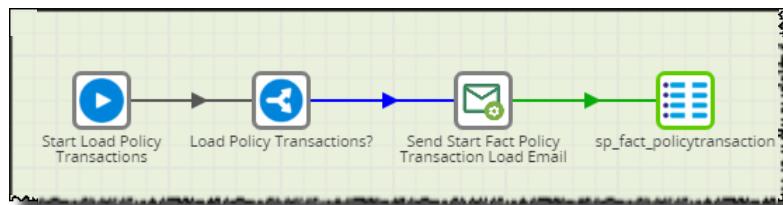
Run Transformation



[Jump to job](#)

Parameter	Value
Transformation Job	DIM_COVERAGE
Set Scalar Variables	
Set Grid Variables	

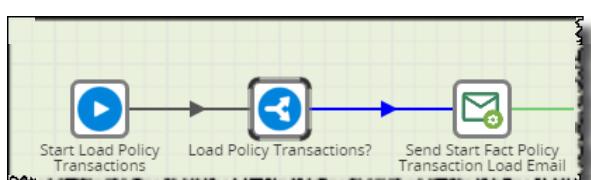
## Job: Load Fact Policy Transactions



---

## Load Policy Transactions?

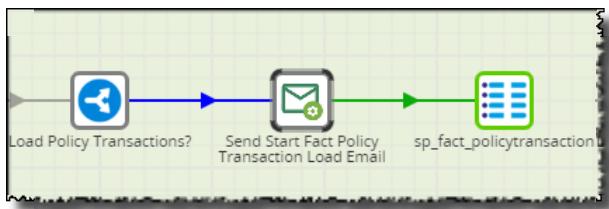
If



Parameter	Value
Mode	Simple
Condition	LoadPolicyTransactions, Is, Equal to, Y, is_SPINN_EOD_Complete, Not, Equal to, 0
Combine Conditions	And

## Send Start Fact Policy Transaction Load Email

Send Email

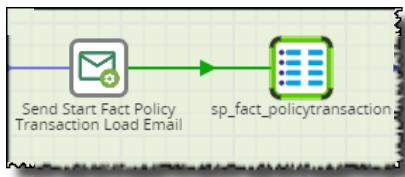


Parameter	Value
To Recipients	\${EMAIL_DESTINATION_ADDRESS}
Cc Recipients	
Subject	\${project_name} Load of Fact Policy Transaction Started
Message	Load of Fact Policy Transaction Started
Sender Address	\${EMAIL_SENDER_ADDRESS}
Reply Address	
SMTP Username	\${EMAIL_AUTHENTICATION_USER}
SMTP Password	*****
SMTP Hostname	\${EMAIL_SMTP_SERVER}
SMTP Port	\${EMAIL_SMTP_PORT}
Enable SSL/TLS	Yes
Enable StartTLS	Yes

## sp\_fact\_policytransaction

SQL Script

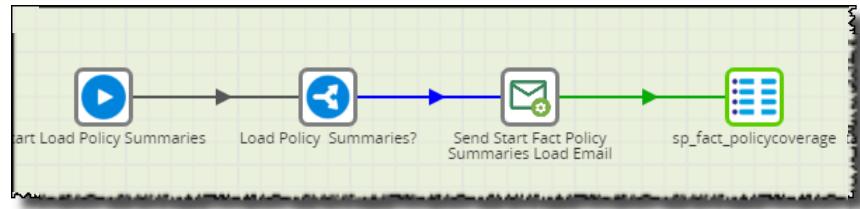




Parameter	Value
SQL Script	call \${schema_STG}.sp_fact_policytransaction()

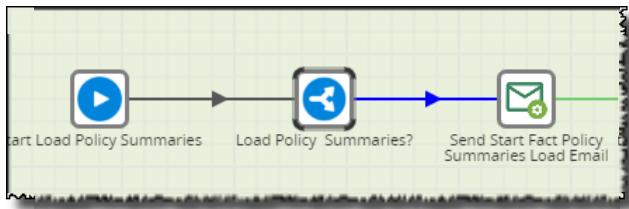
```
call kdlab.sp_fact_policytransaction()
```

## Job: Load Fact Policy Summaries



## Load Policy Summaries?

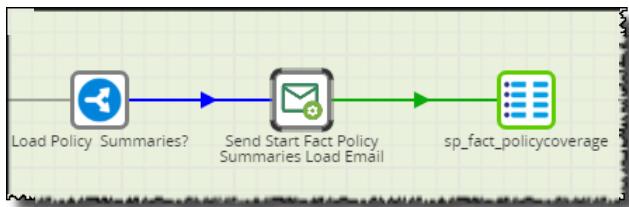
If



Parameter	Value
Mode	Simple
Condition	LoadPolicySummaries, Is, Equal to, Y, is_SPINN_EOD_Complete, Not, Equal to, 0
Combine Conditions	And

## Send Start Fact Policy Summaries Load Email

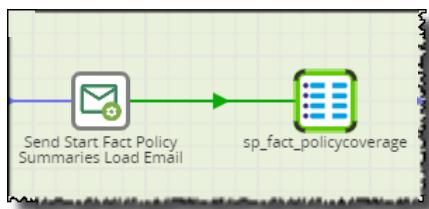
Send Email



Parameter	Value
To Recipients	\${EMAIL_DESTINATION_ADDRESS}
Cc Recipients	
Subject	\${project_name} Load of Fact Policy Summaries Started
Message	Load of Fact Policy Summaries Started
Sender Address	\${EMAIL_SENDER_ADDRESS}
Reply Address	
SMTP Username	\${EMAIL_AUTHENTICATION_USER}
SMTP Password	*****
SMTP Hostname	\${EMAIL_SMTP_SERVER}
SMTP Port	\${EMAIL_SMTP_PORT}
Enable SSL/TLS	Yes
Enable StartTLS	Yes

## sp\_fact\_policycoverage

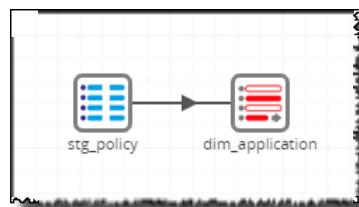
SQL Script



Parameter	Value
SQL Script	call \${schema_STG}.sp_fact_policycoverage()

```
call kdlab.sp_fact_policycoverage()
```

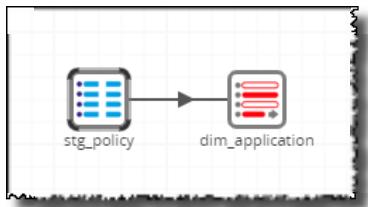
## Job: DIM\_APPLICATION



to load quotes and policies

## stg\_policy

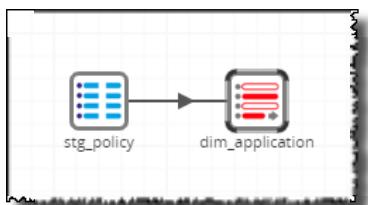
SQL



Parameter	Value
SQL Query	<pre>with data as ( select row_number()over (partition by systemid order by bookdt) rn, * from \${schema_STG}.stg_policy ) select policy_uniqueid as policy_id, 0 as CurrentFlg, * from data where rn=1</pre>

## dim\_application

Table Update



Parameter	Value
Schema	\${schema_DW}
Target Table Name	dim_application
Fix Data Type Mismatches	No
Column Mapping	loaddate, loaddate, policy_id, policy_id, systemid, systemid, currentflg, currentflg, bookdt, bookdt, transactioneffectivedt, transactioneffectivedt, policy_uniqueid, policy_uniqueid, transactioncd, transactioncd, policynumber, policynumber, term, term, effectivedate, effectivedate, expirationdate, expirationdate, carriercd, carriercd, companycd, companycd, termdays, termdays, carriergroupcd, carriergroupcd, statecd, statecd, businesssourcecd, businesssourcecd, previouscarriercd, previouscarriercd, policyformcode, policyformcode, subtypecd, subtypecd, payplaned, payplaned, inceptiondt, inceptiondt, priorpolicynumber, priorpolicynumber, previouspolicynumber, previouspolicynumber, affinitygroupcd, affinitygroupcd, programind, programind, relatedpolicynumber, relatedpolicynumber, twopaydiscountind, twopaydiscountind, quotenumber, quotenumber, renewaltermcd, renewaltermcd, rewritepolicyref, rewritepolicyref, rewritefrompolicyref, rewritefrompolicyref, canceldt, canceldt, reinstatedt, reinstatedt, persistencydiscountdt, persistencydiscountdt, paperlessdelivery, paperlessdelivery, multicarddiscountind, multicarddiscountind, latefee, latefee, nsfee, nsfee, installmentfee, installmentfee, batchquotesourcecd, batchquotesourcecd, waivepolicyfeeind, waivepolicyfeeind, liabilitylimitcpl,

```

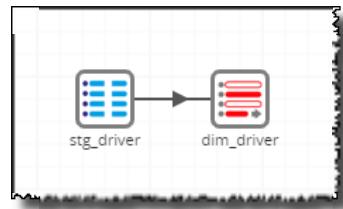
liabilitylimitcpl, liabilityreductionind, liabilityreductionind, liabilitylimitolt, liabilitylimitolt,
personalliabilitylimit, personalliabilitylimit, gloccurrencelimit, gloccurrencelimit, glaggregatelimit,
glaggregatelimit, policy_spinn_status, policy_spinn_status, bilimit, bilimit, pdllimit, pdllimit, umbilimit,
umbilimit, medpaylimit, medpaylimit, multipolicydiscount, multipolicydiscount,
multipolicyautodiscount, multipolicyautodiscount, multipolicyautonumber, multipolicyautonumber,
multipolicyhomediscount, multipolicyhomediscount, homerelatedpolicynumber,
homerelatedpolicynumber, multipolicyumbrelladiscount, multipolicyumbrelladiscount,
umbrellarelatedpolicynumber, umbrellarelatedpolicynumber, cseemployeediscountind,
cseemployeediscountind, fullpaydiscountind, fullpaydiscountind, primarypolicynumber,
primarypolicynumber, landlordind, landlordind, personalinjuryind, personalinjuryind,
vehiclelistconfirmedind, vehiclelistconfirmedind, altsubtypecd, altsubtypecd, firstpayment,
firstpayment, lastpayment, lastpayment, balanceamt, balanceamt, paidamt, paidamt,
product_uniqueid, product_uniqueid, company_uniqueid, company_uniqueid, producer_uniqueid,
producer_uniqueid, firstinsured_uniqueid, firstinsured_uniqueid, accountref, accountref,
customer_uniqueid, customer_uniqueid, mgafeeplancd, mgafeeplancd, mgafeepct, mgafeepct,
tpafeeplancd, tpafeeplancd, tpafeeplancd, applicationnumber, applicationnumber,
application_updatetimestamp, application_updatetimestamp, quoteinfo_updatedt,
quoteinfo_updatedt, quoteinfo_adduser_uniqueid, quoteinfo_adduser_uniqueid,
original_policy_uniqueid, original_policy_uniqueid, application_type, application_type,
quoteinfo_type, quoteinfo_type, application_status, application_status, quoteinfo_status,
quoteinfo_status, quoteinfo_closereasoncd, quoteinfo_closereasoncd, quoteinfo_closesubreasoncd,
quoteinfo_closesubreasoncd, quoteinfo_closecomment, quoteinfo_closecomment,
writtenpremiumamt, writtenpremiumamt, fulltermamt, fulltermamt, commissionamt, commissionamt

```

Unique Keys systemid

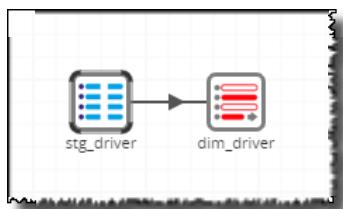
Update Strategy Update/Insert

## Job: DIM\_DRIVER



## stg\_driver

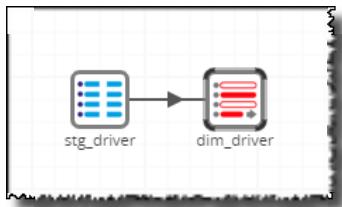
SQL



Parameter	Value
SQL Query	<pre> with data as ( select row_number()over (partition by driver_uniqueid order by bookdt) rn, * from \${schema_STG}.stg_driver ) select policy_uniqueid as policy_id, 0 currentFlg, * from data where rn=1 </pre>

## dim\_driver

Table Update



Parameter	Value
Schema	\${schema_DW}
Target Table Name	dim_driver
Fix Data Type Mismatches	No
Column Mapping	loaddate, loaddate, policy_id, policy_id, systemid, systemid, currentflg, currentflg, bookdt, bookdt, transactioneffectivedt, transactioneffectivedt, policy_uniqueid, policy_uniqueid, drivernumber, drivernumber, driver_uniqueid, driver_uniqueid, spinndriver_id, spinndriver_id, status, status, firstname, firstname, lastname, lastname, licensenumber, licensenumber, licensedt, licensedt, driverinfood, driverinfood, drivertypecd, driverstatuscd, driverstatuscd, licensedstateprovcd, licensedstateprovcd, relationshiptoinsuredcd, relationshiptoinsuredcd, scholasticdiscountind, scholasticdiscountind, mvrrequestind, mvrrequestind, mvrstatus, mvrstatus, mvrstatusdt, mvrstatusdt, maturedriverind, maturedriverind, drivertrainingind, drivertrainingind, gooddriverind, gooddriverind, accidentpreventioncoursecompletiondt, accidentpreventioncoursecompletiondt, accidentpreventioncourseind, accidentpreventioncourseind, scholasticcertificationdt, scholasticcertificationdt, activemilitaryind, activemilitaryind, permanentlicenseind, permanentlicenseind, newtostateind, newtostateind, persontypecd, persontypecd, gendercd, gendercd, birthdt, birthdt, maritalstatuscd, maritalstatuscd, occupationclasscd, occupationclasscd, positiontitle, positiontitle, currentresidencecd, currentresidencecd, civilservantind, civilservantind, retiredind, retiredind, newteenexpirationdt, newteenexpirationdt, attachedvehiclerref, attachedvehiclerref, viol_pointschargedterm, viol_pointschargedterm, acci_pointschargedterm, acci_pointschargedterm, susp_pointschargedterm, susp_pointschargedterm, other_pointschargedterm, other_pointschargedterm, gooddriverpoints_chargedterm, gooddriverpoints_chargedterm, sr22feeind, sr22feeind, maturecertificationdt, maturecertificationdt, agefirstlicensed, agefirstlicensed
Unique Keys	driver_uniqueid
Update Strategy	Update/Insert

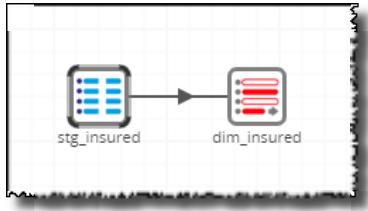
## Job: DIM\_INSURED



to load insured

stg\_insured

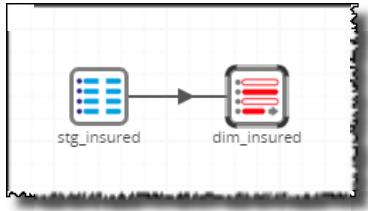
SQL



Parameter	Value
SQL Query	<pre>with data as ( select row_number()over (partition by systemid order by bookdt) rn, * from \${schema_STG}.stg_insured ) select systemid insured_id , policy_uniqueid as policy_id , 0 CurrentFlg * ,from data where rn=1</pre>

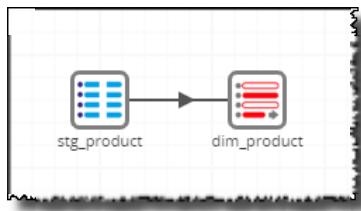
## dim\_insured

Table Update



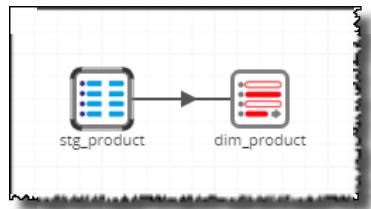
Parameter	Value
Schema	`\${schema_DW}`
Target Table Name	dim_insured
Fix Data Type Mismatches	No
Column Mapping	insured_id, insured_id, loaddate, loaddate, policy_id, policy_id, systemid, systemid, currentflg, currentflg, bookdt, bookdt, transactioneffectivedt, transactioneffectivedt, policy_uniqueid, policy_uniqueid, insured_uniqueid, insured_uniqueid, first_name, first_name, last_name, last_name, commercialname, commercialname, dob, dob, occupation, occupation, gender, gender, maritalstatus, maritalstatus, address1, address1, address2, address2, county, county, city, city, state, state, postalcode, postalcode, country, country, telephone, telephone, mobile, mobile, email, email, jobtitle, jobtitle, insurancescore, insurancescore, overriddeninsurancescore, overriddeninsurancescore, overriddeninsurancescore, overriddeninsurancescore, ratepageeffectivedt, ratepageeffectivedt, insscoretiervalueband, insscoretiervalueband, financialstabilitytier, financialstabilitytier
Unique Keys	insured_id
Update Strategy	Update/Insert

Job: DIM\_PRODUCT



## stg\_product

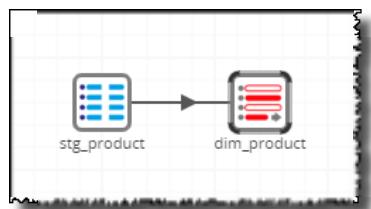
SQL



Parameter	Value
SQL Query	<pre>with data as ( select row_number()over (partition by product_uniqueid order by startdt) rn, * from \${schema_STG}.stg_product ) select * from data where rn=1</pre>

## dim\_product

Table Update



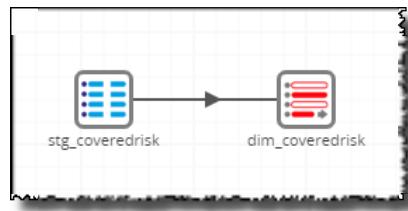
Parameter	Value
Schema	`\${schema_DW}`
Target Table Name	dim_product
Fix Data Type Mismatches	No
Column Mapping	loaddate, loaddate, product_uniqueid, product_uniqueid, carriergroupcd, carriergroupcd, description, description, subtypecd, subtypecd, productversion, productversion, name, name, producttypecd, producttypecd, carriercd, carriercd, isselect, isselect, linecd, linecd, altsubtypecd, altsubtypecd, subtypeshortdesc, subtypeshortdesc, subtypefulldesc, subtypefulldesc, policynumberprefix, policynumberprefix, startdt, startdt, stopdt, stopdt, renewalstartdt, renewalstartdt, renewalstopdt, renewalstopdt, statecd, statecd, contract, contract, lob, lob, propertyform, propertyform,

prerenewaldays, prerenewaldays, autorenewaldays, autorenewaldays, mgafeeplancd,  
mgafeeplancd, tpafeeplancd, tpafeeplancd

Unique Keys product\_uniqueid

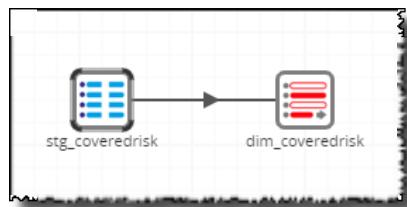
Update Strategy Update/Insert

## Job: DIM\_COVEREDRISK



### stg\_coveredrisk

SQL



#### Parameter

#### Value

SQL Query	<pre>with stg_query as(select Building_Uniqueid coveredrisk_uniqueid ,LOADDATE ,SystemId ,BookDt ,TransactionEffectiveDt ,case when Status='Deleted' then 1 else 0 end deleted_indicator ,BldgNumber Risk_Number ,Risk_UniqueId ,Policy_UniqueId ,isnull(Risk_Type,'~') Risk_Type from \${schema_STG}.STG_BUILDING union all select Vehicle_Uniqueid coveredrisk_uniqueid ,LOADDATE ,SystemId ,BookDt ,TransactionEffectiveDt ,case when Status='Deleted' then 1 else 0 end deleted_indicator ,VehNumber Risk_Number ,Risk_UniqueId ,Policy_UniqueId ,isnull(Risk_Type,'~') Risk_Type from \${schema_STG}.STG_VEHICLE) ,data as ( select row_number()over (partition by coveredrisk_uniqueid order by bookdt) rn, * from stg_query ) select Policy_UniqueId Policy_Id</pre>
-----------	---

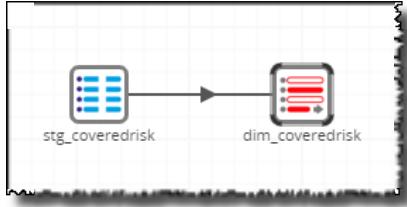
```

,0 CurrentFlg
*
from data
where rn=1

```

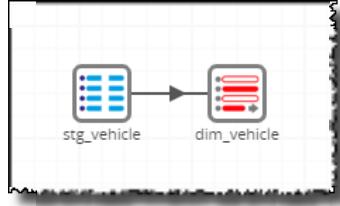
## dim\_coveredrisk

Table Update



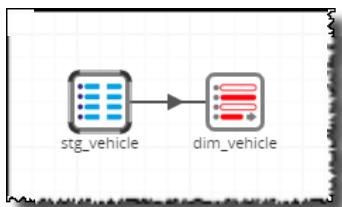
Parameter	Value
Schema	\${schema_DW}
Target Table Name	dim_coveredrisk
Fix Data Type Matches	No
Column Mapping	LOADDATE, LOADDATE, POLICY_ID, POLICY_ID, SystemId, SystemId, CurrentFlg, CurrentFlg, BookDt, BookDt, TransactionEffectiveDt, TransactionEffectiveDt, POLICY_UNIQUEID, POLICY_UNIQUEID, deleted_indicator, deleted_indicator, risk_number, risk_number, risk_type, risk_type, RISK_UNIQUEID, RISK_UNIQUEID, COVEREDRISK_UNIQUEID, COVEREDRISK_UNIQUEID
Unique Keys	coveredrisk_uniqueid
Update Strategy	Update/Insert

## Job: DIM\_VEHICLE



## stg\_vehicle

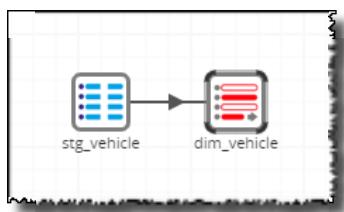
SQL



Parameter	Value
SQL Query	<pre> with data as ( SELECT ROW_NUMBER() OVER(partition by cr.coveredrisk_id order by stg.BookDt) rn, cr.coveredrisk_id as vehicle_id, stg.* FROM STG_VEHICLE stg join \${schema_STG}.DIM_COVEREDRISK cr on stg.Vehicle_Uniqueid=cr.coveredrisk_uniqueid and stg.policy_uniqueid=cr.policy_id and stg.SystemId=cr.SystemId ) select policy_uniqueid as policy_id, 0 CurrentFlg, * from data where rn=1 </pre>

## dim\_vehicle

Table Update



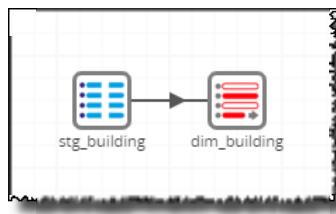
Parameter	Value
Schema	`\${schema_DW}`
Target Table Name	dim_vehicle
Fix Data Type Mismatches	No
Column Mapping	<p>vehicle_id, vehicle_id, loaddate, loaddate, policy_id, policy_id, systemid, systemid, currentflg, currentflg, bookdt, bookdt, transactioneffectivedt, transactioneffectivedt, policy_uniqueid, policy_uniqueid, risk_uniqueid, risk_uniqueid, vehnumber, vehnumber, vehicle_uniqueid, vehicle_uniqueid, spinnvehicle_id, spinnvehicle_id, status, status, stateprovcd, stateprovcd, county, county, postalcode, postalcode, city, city, addr1, addr1, addr2, addr2, garagaddrflg, garagaddrflg, latitude, latitude, longitude, longitude, garagpostalcode, garagpostalcode, garagpostalcodedeflg, garagpostalcodedeflg, manufacturer, manufacturer, model, model, modelyr, modelyr, vehidentificationnumber, vehidentificationnumber, validvinind, validvinind, vehlicensenumber, vehlicensenumber, registrationstateprovcd, registrationstateprovcd, vehbodytypecd, vehbodytypecd, performancecd, performancecd, restraintcd, restraintcd, antibrakingsystemcd, antibrakingsystemcd, antitheftcd, antitheftcd, enginesize, enginesize, enginecylinders, enginecylinders, enginehorsepower, enginehorsepower, enginetype, enginetype, vehusecd, vehusecd, garageteritory, garageteritory, collisionded, collisionded, comprehensiveded, comprehensiveded, statedamt, statedamt, classcd, classcd, ratingvalue, ratingvalue, costnewamt, costnewamt, estimatedannualdistance, estimatedannualdistance, estimatedworkdistance, estimatedworkdistance, leasedvehind, leasedvehind, purchasedt, purchasedt, statedamtind, statedamtind, neworusedind, neworusedind, carpoolind, carpoolind, odometerreading, odometerreading, weekspermonthdriven, weekspermonthdriven, daylightrunninglightsind, daylightrunninglightsind, passiveseatbeltind, passiveseatbeltind, daysperweekdriven, daysperweekdriven, umpdlimit, umpdlimit, towingandalborind, towingandalborind, rentalreimbursementind, rentalreimbursementind, liabilitywaiveind, liabilitywaiveind, ratefeesind, ratefeesind, optionalequipmentvalue, optionalequipmentvalue, customizingequipmentind, customizingequipmentind, customizingequipmentdesc, customizingequipmentdesc, invalidvinacknowledgementind, invalidvinacknowledgementind, ignoreumpdwcdind, ignoreumpdwcdind, recalculateratingsymbolind, recalculateratingsymbolind, programtypecd, programtypecd, cmpratingvalue, cmpratingvalue,</p>

colratingvalue, colratingvalue, liabilityratingvalue, liabilityratingvalue, medpayratingvalue,  
 medpayratingvalue, racmpratingvalue, racmpratingvalue, racolratingvalue, racolratingvalue,  
 rabiratingsymbol, rabiratingsymbol, rapdratingsymbol, rapdratingsymbol, ramedpayratingsymbol,  
 ramedpayratingsymbol, estimatedannualdistanceoverride, estimatedannualdistanceoverride,  
 originalestimatedannualmiles, originalestimatedannualmiles, reportedmileagenonsave,  
 reportedmileagenonsave, mileage, mileage, estimatednoncommutemiles,  
 estimatednoncommutemiles, titlehistoryissue, titlehistoryissue, odometerproblems,  
 odometerproblems, bundle, bundle, loanleasegap, loanleasegap, equivalentreplacementcost,  
 equivalentreplacementcost, originalequipmentmanufacturer, originalequipmentmanufacturer,  
 optionalrideshare, optionalrideshare, medicalpartsaccessibility, medicalpartsaccessibility,  
 odometerreadingprior, odometerreadingprior, reportedmileagenonsavedprior,  
 reportedmileagenonsavedtprior, fullglasscovind, fullglasscovind, boatlengthfeet, boatlengthfeet,  
 motorhorsepower, motorhorsepower, replacementof, replacementof, reportedmileagenonsavedt,  
 reportedmileagenonsavedt, manufacturersymbol, manufacturersymbol, modelsymbol, modelsymbol,  
 bodystylesymbol, bodystylesymbol, symbolcode, symbolcode, verifiedmileageoverride,  
 verifiedmileageoverride

Unique Keys vehicle\_id

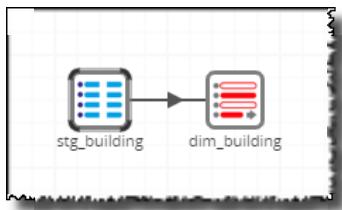
Update Strategy Update/Insert

## Job: DIM\_BUILDING



### stg\_building

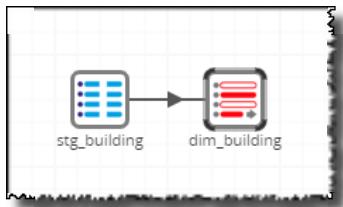
SQL



Parameter	Value
SQL Query	<pre> with data as ( select row_number()over (partition by cr.coveredrisk_id order by stg.bookdt) rn, cr.coveredrisk_id as building_id, stg.* from \${schema_STG}.stg_building stg join \${schema_STG}.DIM_COVEREDRISK cr on stg.Building_UniqueId=cr.coveredrisk_uniqueid and stg.policy_uniqueid=cr.policy_id and stg.SystemId=cr.SystemId ) select policy_uniqueid as policy_id , 0 CurrentFlg , * from data where rn=1 </pre>

## dim\_building

Table Update



Parameter	Value
Schema	\${schema_DW}
Target Table Name	dim_building
Fix Data Type Mismatches	No
Column Mapping	building_id, building_id, loaddate, loaddate, policy_id, policy_id, systemid, systemid, currentflg, currentflg, bookdt, bookdt, transactioneffectivedt, transactioneffectivedt, policy_uniqueid, policy_uniqueid, risk_uniqueid, risk_uniqueid, bldgnumber, bldgnumber, building_uniqueid, building_uniqueid, spinnbuilding_id, spinnbuilding_id, status, status, stateprovcd, stateprovcd, county, county, postalcode, postalcode, city, city, addr1, addr1, addr2, addr2, businesscategory, businesscategory, businessclass, businessclass, constructioncd, constructioncd, roofcd, roofcd, yearbuilt, yearbuilt, sqft, sqft, stories, stories, units, units, occupancycd, occupancycd, protectionclass, protectionclass, territorycd, territorycd, buildinglimit, buildinglimit, contentslimit, contentslimit, valuationmethod, valuationmethod, inflationguardpct, inflationguardpct, ordinanceorlawind, ordinanceorlawind, scheduledpremiummod, scheduledpremiummod, windhailexclusion, windhailexclusion, covalimit, covalimit, covblimit, covblimit, covclimit, covclimit, covdlimit, covdlimit, covelimit, covelimit, covflimit, covflimit, allperilded, allperilded, burglaryalarmtype, burglaryalarmtype, firealarmtype, firealarmtype, covblimitincluded, covblimitincluded, covblimitincrease, covblimitincrease, covblimitincluded, covblimitincluded, covblimitincrease, covblimitincrease, covdlimitincluded, covdlimitincluded, covdlimitincrease, covdlimitincrease, ordinanceorlawpct, ordinanceorlawpct, neighborhoodcrimewatchind, neighborhoodcrimewatchind, employeecreditind, employeecreditind, multipolicyind, multipolicyind, homewarrantycreditind, homewarrantycreditind, yearoccupied, yearoccupied, yearpurchased, yearpurchased, typeofstructure, typeofstructure, feettofirehydrant, feettofirehydrant, numberoffamilies, numberoffamilies, milesfromfirestation, milesfromfirestation, rooms, rooms, roofpitch, roofpitch, firedistrict, firedistrict, sprinklersystem, sprinklersystem, fireextinguisherind, fireextinguisherind, kitchenfireextinguisherind, kitchenfireextinguisherind, deadboltind, deadboltind, gatedcommunityind, gatedcommunityind, centralheatingind, centralheatingind, foundation, foundation, wiringrenovation, wiringrenovation, wiringrenovationcompletetyear, wiringrenovationcompletetyear, plumbingrenovation, plumbingrenovation, heatingrenovation, heatingrenovation, plumbingrenovationcompletetyear, plumbingrenovationcompletetyear, exteriorpaintrenovation, exteriorpaintrenovation, heatingrenovationcompletetyear, heatingrenovationcompletetyear, circuitbreakersind, circuitbreakersind, copperwiringind, copperwiringind, exteriorpaintrenovationcompletetyear, exteriorpaintrenovationcompletetyear, copperpipesind, copperpipesind, earthquakeretrofitind, earthquakeretrofitind, primaryfuelsource, primaryfuelsource, secondaryfuelsource, secondaryfuelsource, usagetype, usagetype, homegardcreditind, homegardcreditind, multipolicynumber, multipolicynumber, localfirealarmind, localfirealarmind, numlosses, numlosses, covalimitincrease, covalimitincrease, covalimitincluded, covalimitincluded, monthsrentedout, monthsrentedout, roofreplacement, roofreplacement, safeguardplusind, safeguardplusind, covelimitincluded, covelimitincluded, roofreplacementcompletetyear, roofreplacementcompletetyear, covelimitincrease, covelimitincrease, owneroccupiedunits, owneroccupiedunits, tenantoccupiedunits, tenantoccupiedunits, replacementcostdwellingind, replacementcostdwellingind, feettopropertyline, feettopropertyline, galvanizedpipeind, galvanizedpipeind, workerscompinservant, workerscompinservant, workerscompoutservant, workerscompoutservant, liabilityterritorycd, liabilityterritorycd, premisesliabilitymedpayind, premisesliabilitymedpayind, relatedprivatestructureexclusion, relatedprivatestructureexclusion, vandalismexclusion, vandalismexclusion, vandalismmind, vandalismmind, roofexclusion, roofexclusion, expandedreplacementcostind, expandedreplacementcostind, replacementvalueind, replacementvalueind, otherpolicynumber1, otherpolicynumber1, otherpolicynumber1, otherpolicynumber2, otherpolicynumber2, otherpolicynumber3, otherpolicynumber3, primarypolicynumber, primarypolicynumber, otherpolicynumbers, otherpolicynumbers, reportedfirehazardsscore, reportedfirehazardsscore, firehazardsscore, firehazardsscore, reportedsteepslopeind, reportedsteepslopeind, steepslopeind, steepslopeind, reportedhomereplacementcost, reportedhomereplacementcost, reportedprotectionclass, reportedprotectionclass, earthquakezone,

earthquakezone, mmiscore, mmiscore, homeinspectiondiscountind, homeinspectiondiscountind, ratingtier, ratingtier, soiltypcd, soiltypcd, reportedfirelineassessment, reportedfirelineassessment, aaisfireprotectionclass, aaisfireprotectionclass, inspectionscore, inspectionscore, annualrents, annualrents, pitchofrooft, pitchofrooft, totallivingsqft, totallivingsqft, parkingsqft, parkingsqft, parkingtype, parkingtype, retrofitcompleted, retrofitcompleted, numpools, numpools, fullyfenced, fullyfenced, divingboard, divingboard, gym, gym, freeweights, freeweights, wirefencing, wirefencing, otherrecreational, otherrecreational, otherrecreationaldesc, otherrecreationaldesc, healthinspection, healthinspection, healthinspectiondt, healthinspectiondt, healthinspectioncited, healthinspectioncited, priordefectrepairs, priordefectrepairs, msbreconstructionestimate, msbreconstructionestimate, biindemnityperiod, biindemnityperiod, equipmentbreakdown, equipmentbreakdown, moneysecurityonpremises, moneysecurityonpremises, moneysecurityoffpremises, moneysecurityoffpremises, waterbackupsump, waterbackupsump, sprinkleredbuildings, sprinkleredbuildings, surveillancecams, surveillancecams, gatedcomplexkeyaccess, gatedcomplexkeyaccess, eqretrofit, eqretrofit, unitsperbuilding, unitsperbuilding, numstories, numstories, constructionquality, constructionquality, burglaryrobbery, burglaryrobbery, nfpaclassification, nfpaclassification, areasofcoverage, areasofcoverage, codetector, codetector, smokedetector, smokedetector, smokedetectorinspectind, smokedetectorinspectind, waterheatersecured, waterheatersecured, boltedorsecured, boltedorsecured, softstorycripple, softstorycripple, seniorhousingpct, seniorhousingpct, designatedseniorhousing, designatedseniorhousing, studenthousingpct, studenthousingpct, designatedstudenthousing, designatedstudenthousing, priorlosses, priorlosses, tenantevictions, tenantevictions, vacancyrateexceed, vacancyrateexceed, seasonalrentals, seasonalrentals, condoinsuingagmt, condoinsuingagmt, gasvalve, gasvalve, owneroccupiedpct, owneroccupiedpct, restaurantname, restaurantname, hoursofoperation, hoursofoperation, restaurantsqft, restaurantsqft, seatingcapacity, seatingcapacity, annualgrosssales, annualgrosssales, seasonalorclosed, seasonalorclosed, barcocktaillounge, barcocktaillounge, liveentertainment, liveentertainment, beerwinegrosssales, beerwinegrosssales, distilledspiritsserved, distilledspiritsserved, kitchendeepfryer, kitchendeepfryer, solidfuelcooking, solidfuelcooking, ansulsystem, ansulsystem, ansulannualinspection, ansulannualinspection, tenantnameslist, tenantnameslist, tenantbusinessstype, tenantbusinessstype, tenantglliability, insuredoccupiedportion, insuredoccupiedportion, valetparking, valetparking, lessorsqft, lessorsqft, buildingrisknumber, buildingrisknumber, multipolicyindumbrella, multipolicyindumbrella, poolind, poolind, studsuprenovation, studsuprenovation, studsuprenovationcompleteyear, studsuprenovationcompleteyear, multipolicynumberumbrella, multipolicynumberumbrella, rctmsbamt, rctmsbamt, rctmsbhomestyle, rctmsbhomestyle, winsoverridenonsmokerdiscount, winsoverridenonsmokerdiscount, winsoverrideseniordiscount, winsoverrideseniordiscount, itv, itv, itvdate, itvdate, msbreporttype, msbreporttype, vandalismdesiredind, vandalismdesiredind, woodshakesiding, woodshakesiding, cseagent, cseagent, propertymanager, propertymanager, rentersinsurance, rentersinsurance, waterdetectiondevice, waterdetectiondevice, autohomeind, autohomeind, earthquakeumbrellaind, earthquakeumbrellaind, landlordind, landlordind, lossassessment, lossassessment, gasshutoffind, gasshutoffind, waterded, waterded, serviceline, serviceline, functionalreplacementcost, functionalreplacementcost, milesofstreet, milesofstreet, hoaexteriorstructure, hoaexteriorstructure, retailportiondevelopment, retailportiondevelopment, lightindustrialtype, lightindustrialtype, lightindustrialdescription, lightindustrialdescription, poolcoveragelimit, poolcoveragelimit, multifamilyresidentialbuildings, multifamilyresidentialbuildings, singlefamilydwellings, singlefamilydwellings, annualpayroll, annualpayroll, annualrevenue, annualrevenue, bedsoccupied, bedsoccupied, emergencylighting, emergencylighting, exitsignsposted, exitsignsposted, fulltimestaff, fulltimestaff, licensedbeds, licensedbeds, numberoffireextinguishers, numberoffireextinguishers, otherfireextinguishers, otherfireextinguishers, oxygentanks, oxygentanks, parttimestaff, parttimestaff, smokingpermitted, smokingpermitted, staffonduty, staffonduty, typeoffireextinguishers, typeoffireextinguishers, covaddr\_secondaryresidence, covaddr\_secondaryresidence, covaddrprem\_secondaryresidence, covaddrprem\_secondaryresidence, covaddrprem\_secondaryresidence, hodeluxe, hodeluxe, latitude, latitude, longitude, longitude, linecd, linecd, wuiclass, wuiclass, censusblock, censusblock, wateriskscore, wateriskscore, landlordlosspreventionservices, landlordlosspreventionservices, enhancedwatercoverage, enhancedwatercoverage, landlordproperty, landlordproperty, liabilityextendedtoothers, liabilityextendedtoothers, lossofuseextendedtime, lossofuseextendedtime, onpremisestheft, onpremisestheft, bedbugmitigation, bedbugmitigation, habitabilityexclusion, habitabilityexclusion, wildfirehazardpotential, wildfirehazardpotential, backupofsewersanddrains, backupofsewersanddrains, vegetationsetbackft, vegetationsetbackft, yarddebriscoveragearea, yarddebriscoveragearea, yarddebriscoveragepercentage, yarddebriscoveragepercentage, capetampoline, capetampoline, capepool, capepool, roofconditionrating, roofconditionrating, trampolineind, trampolineind, plumbingmaterial, plumbingmaterial, centralizedheating, centralizedheating, firedistrictsubscriptioncode, firedistrictsubscriptioncode, roofcondition, roofcondition

Unique Keys	building_id
Update Strategy	Update/Insert

## Job: DIM\_RISK\_COVERAGE



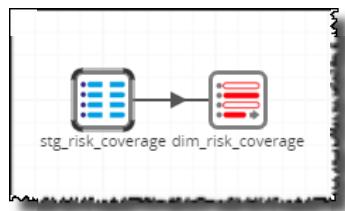


quotes reporting per coverage/limit/deductible

---

## stg\_risk\_coverage

SQL



Parameter	Value
SQL Query	<pre>with data as ( select row_number()over (partition by cr.coveredrisk_id order by stg.bookdt) rn, cr.coveredrisk_id, stg.policy_uniqueid as policy_id, 0 CurrentFlg, stg.* from \${schema_STG}.stg_risk_coverage stg join \${schema_STG}.dim_coveredrisk cr on stg.Risk_Uniqueid=cr.risk_uniqueid and stg.policy_uniqueid=cr.policy_id and stg.SystemId=cr.SystemId ) select * from data where rn=1</pre>

---

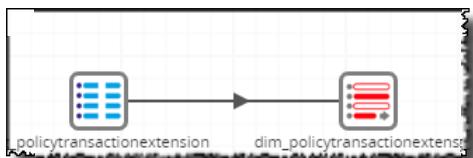
## dim\_risk\_coverage

Table Update



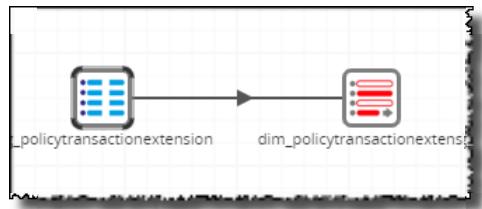
Parameter	Value
Schema	\${schema_DW}
Target Table Name	dim_risk_coverage

Job: DIM\_POLICYTANSACTIONEXTENSION



## stg\_policytransactionextension

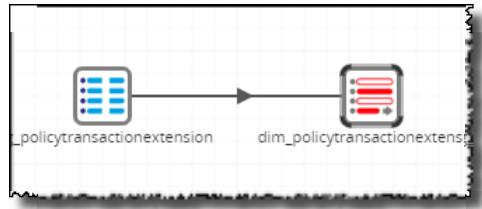
SQL



Parameter	Value
SQL Query	<pre>with data as ( select row_number()over (partition by policytransaction_uniqueid order by transactionnumber) rn, * from \${schema_STG}.stg_policytransactionextension ) select * from data where rn=1</pre>

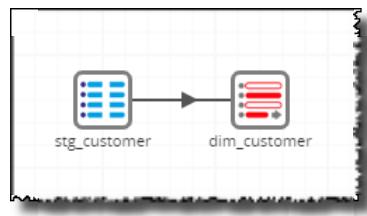
## dim\_policytransactionextension

Table Update



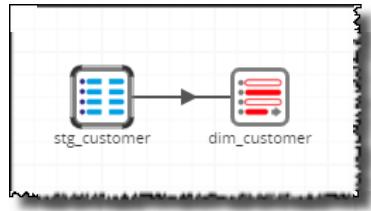
Parameter	Value
Schema	\${schema_DW}
Target Table Name	dim_policytransactionextension
Fix Data Type Mismatches	No
Column Mapping	loaddate, loaddate, policy_uniqueid, policy_id, systemid, systemid, bookdt, bookdt, transactioneffectivedt, transactioneffectivedt, policy_uniqueid, policy_uniqueid, policytransaction_uniqueid, policytransaction_uniqueid, transactionnumber, transactionnumber, transactioncd, transactioncd, transactionlongdescription, transactionlongdescription, transactionshortdescription, transactionshortdescription, canceltypecd, canceltypecd, cancelrequestedbycd, cancelrequestedbycd, cancelreason, cancelreason
Unique Keys	policytransaction_uniqueid
Update Strategy	Update/Insert

## Job: DIM\_CUSTOMER



### stg\_customer

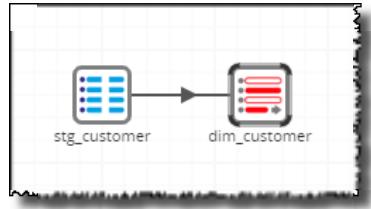
SQL



Parameter	Value
SQL Query	<pre>with data as ( select row_number()over (partition by customer_uniqueid order by changedate) rn, * from \${schema_STG}.stg_customer ) select customer_uniqueid as customer_id, * from data where rn=1</pre>

### dim\_customer

Table Update



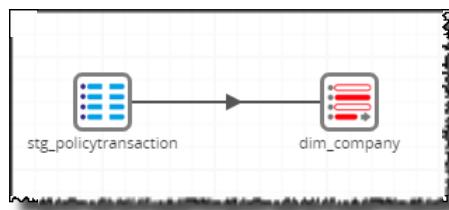
Parameter	Value
Schema	\${schema_DW}
Target Table Name	dim_customer
Fix Data Type Mismatches	No
Column Mapping	customer_id, customer_id, loaddate, loaddate, customer_uniqueid, customer_uniqueid, status, status, entitytypecd, entitytypecd, first_name, first_name, last_name, last_name, commercialname,

	commercialname, dob, dob, gender, gender, maritalstatus, maritalstatus, address1, address1, address2, address2, county, county, city, city, state, state, postalcode, postalcode, phone, phone, mobile, mobile, email, email, preferreddeliverymethod, preferreddeliverymethod, portalinvitationsentdt, portalinvitationsentdt, paymentreminderind, paymentreminderind, paymentreminderind, changedate, changedate
--	--

Unique Keys	customer_uniqueid
-------------	-------------------

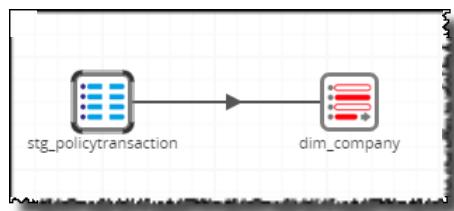
Update Strategy	Update/Insert
-----------------	---------------

## Job: DIM\_COMPANY



### stg\_policytransaction

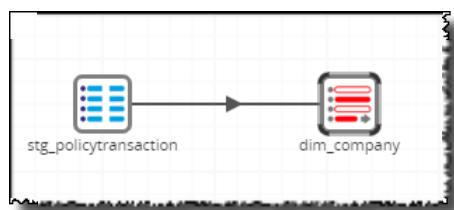
SQL



Parameter	Value
SQL Query	<pre> with data as (select distinct company_uniqueid, split_part(company_uniqueid,'-',1) carriercd , split_part(company_uniqueid,'-',2) companycd from \${schema_STG}.stg_policytransaction ) select to_date('\${loaddate}','yyyy-mm-dd') loaddate, * from data   </pre>

### dim\_company

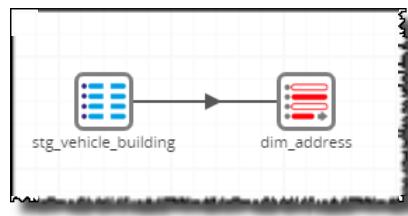
Table Update



Parameter	Value
-----------	-------

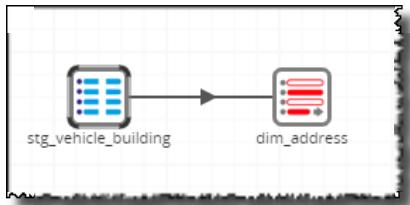
Schema	\${schema_DW}
Target Table Name	dim_company
Fix Data Type Mismatches	No
Column Mapping	company_uniqueid, company_uniqueid, loaddate, loaddate, carriercd, carriercd, companycd, companycd
Unique Keys	company_uniqueid
Update Strategy	Update/Insert

## Job: DIM\_ADDRESS



### stg\_vehicle\_building

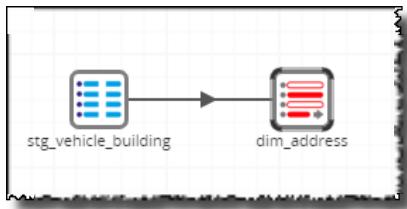
SQL



Parameter	Value
SQL Query	<pre> with data as (select distinct addr1 address1 ,addr2 address2 ,county ,city ,stateprovcd state ,postalcode from \${schema_STG}.stg_building union select distinct addr1 address1 ,addr2 address2 ,county ,city ,stateprovcd state ,postalcode from \${schema_STG}.stg_vehicle ) select to_date('\${loaddate}','yyyy-mm-dd') loaddate, * from data </pre>

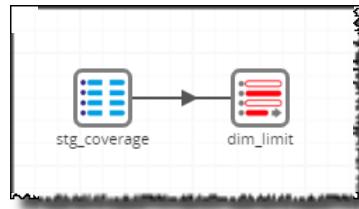
## dim\_address

Table Update



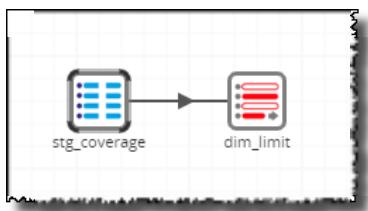
Parameter	Value
Schema	`\${schema_DW}`
Target Table Name	dim_address
Fix Data Type Mismatches	No
Column Mapping	loaddate, loaddate, address1, address1, address2, address2, county, county, city, city, state, state, postalcode, postalcode
Unique Keys	address1, address2, county, city, state, postalcode
Update Strategy	Update/Insert

## Job: DIM\_LIMIT



## stg\_coverage

SQL



Parameter	Value
SQL Query	<pre>select DISTINCT COV_LIMIT1 , COV_LIMIT2 , isnull(cast(case when COV_LIMIT1 !~ ('[^.0-9\.-]') then COV_LIMIT1 else public.removeNonNumeric(COV_LIMIT1) end as float),0) COV_LIMIT1_VALUE , isnull(cast(case when COV_LIMIT2 !~ ('[^.0-9\.-]') then COV_LIMIT2</pre>

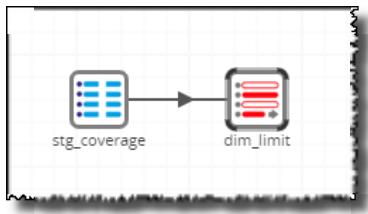
```

else public.removeNotNumeric(COV_LIMIT2)
end as float),0) COV_LIMIT2_VALUE
, LOADDATE
from ${schema_STG}.stg_policytransaction

```

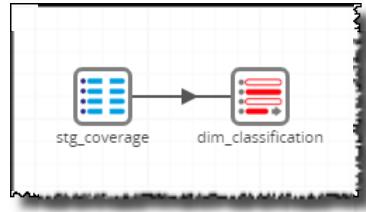
## dim\_limit

Table Update



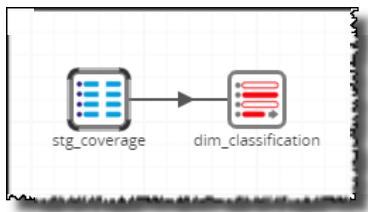
Parameter	Value
Schema	`\${schema_DW}`
Target Table Name	dim_limit
Fix Data Type Mismatches	No
Column Mapping	loaddate, loaddate, cov_limit1, cov_limit1, cov_limit2, cov_limit2, cov_limit1_value, cov_limit1_value, cov_limit2_value, cov_limit2_value
Unique Keys	cov_limit1, cov_limit2
Update Strategy	Update/Insert

## Job: DIM\_CLASSIFICATION



## stg\_coverage

SQL



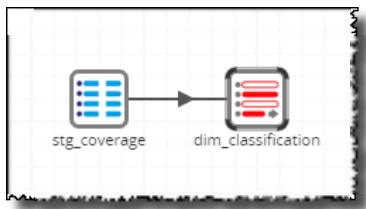
Parameter	Value

## SQL Query

```
select
DISTINCT
ltrim(rtrim(COV_CLASSCODE)) CLASS_CODE
, case when charindex('-',COV_CLASSCODE)-1>0 then ltrim(left(COV_CLASSCODE,charindex('-',COV_CLASSCODE)-1))) else '~' end CLASS_CODENAME
, case when charindex('-',COV_CLASSCODE)-1>0 then
ltrim(rtrim(right(COV_CLASSCODE,len(COV_CLASSCODE) - charindex('-',COV_CLASSCODE)))) else
'~' end CLASS_CODEDESCRIPTION
, LOADDATE
from ${schema_STG}.stg_policytransaction
```

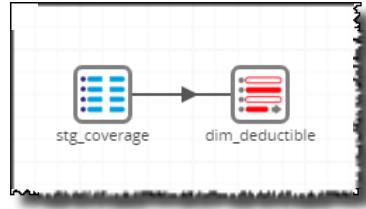
## dim\_classification

### Table Update



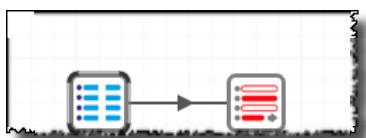
Parameter	Value
Schema	`\${schema_DW}`
Target Table Name	dim_classification
Fix Data Type Mismatches	No
Column Mapping	loaddate, loaddate, class_code, class_code, class_codename, class_codename, class_codedescription, class_codedescription
Unique Keys	class_code, class_codename, class_codedescription
Update Strategy	Update/Insert

## Job: DIM\_DEDUCTIBLE



## stg\_coverage

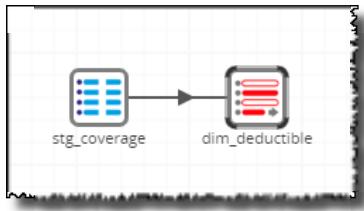
### SQL



Parameter	Value
SQL Query	<pre> select DISTINCT COV_DEDUCTIBLE1, COV_DEDUCTIBLE2, isnull(cast(case when COV_DEDUCTIBLE1 !~ ('[^0-9\.-]') then COV_DEDUCTIBLE1 else public.removeNonNumeric(COV_DEDUCTIBLE1) end as float),0) COV_DEDUCTIBLE1_value , isnull(cast(case when COV_DEDUCTIBLE2 !~ ('[^0-9\.-]') then COV_DEDUCTIBLE2 else public.removeNonNumeric(COV_DEDUCTIBLE2) end as float),0) COV_DEDUCTIBLE2_value , LOADDATE from \${schema_STG}.stg_policytransaction </pre>

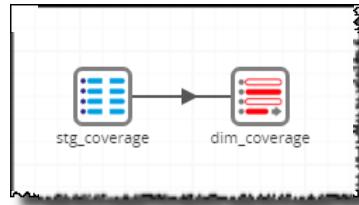
## dim\_deductible

Table Update



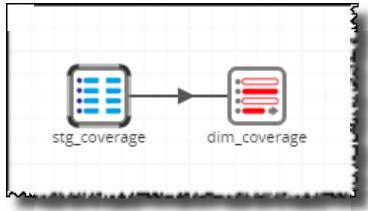
Parameter	Value
Schema	\${schema_DW}
Target Table Name	dim_deductible
Fix Data Type Mismatches	No
Column Mapping	loaddate, loaddate, cov_deductible1, cov_deductible1, cov_deductible2, cov_deductible2, cov_deductible1_value, cov_deductible1_value, cov_deductible2_value, cov_deductible2_value
Unique Keys	cov_deductible1, cov_deductible2
Update Strategy	Update/Insert

## Job: DIM\_COVERAGE



## stg\_coverage

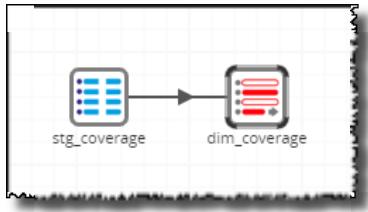
## SQL



Parameter	Value
SQL Query	<pre>select DISTINCT COV_CODE , COV_SUBCODE , COV_ASL , COV_SUBLINE , LOADDATE from \${schema_STG}.stg_policytransaction</pre>

## dim\_coverage

### Table Update



Parameter	Value
Schema	`\${schema_DW}`
Target Table Name	dim_coverage
Fix Data Type Mismatches	No
Column Mapping	loaddate, loaddate, cov_code, cov_code, cov_subcode, cov_subcode, cov_asl, cov_asl, cov_subline, cov_subline
Unique Keys	cov_code, cov_subcode, cov_asl, cov_subline
Update Strategy	Update/Insert