Effectivity Satellites

Book Sections Reference: 5.3.4: "Its purpose is to track when the **link** is active according to the business and provides begin and end dates for this purposes"

See also: "The Data Vault Guru: A pragmatic guide on building a data vault" Chapter 6.3 Effectivity Satellite (Page 301)

As part of the enablement pack WhereScape ships the functionality to use the "Applied Date" to determine the effectivity of the link. You can choose whether you want a default system generated date to populate "Applied Date" or whether you want to choose no more than 1 date from the source.

In cases where multiple dates or fields from the source describes the temporality of the data we provide an option for a different approach, and not to resolve it during load of Raw Vault, but rather further downstream. See details under our Multi-Temporal sources enablement pack.

WhereScape 3D

Dependent objects

Sample Repo

- Look at the examples created in 3D repo **wsDVSamples**

Model Conversion Rules

- 1 of 3 generation of Data Vault rule **(ws3d_grv - Create effectivity satellites)**
- 1 of 3 generation of Data Vault rule **(ws3d_grv - Prepare effectivity satellites)**
- 1 of 3 generation of Data Vault rule **(ws3d_grv - Housekeeping effectivity satellites)**
- 1 additional generation of load and stage **(ws3d_rvls - Create effectivity satellites)**

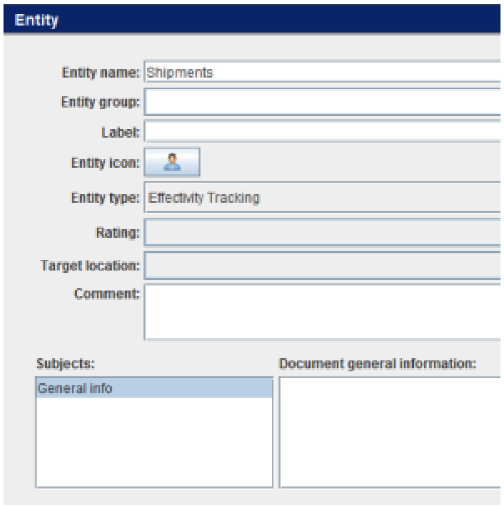
Copy from wsDVSamples or import **EffectivityTracking_MCR.xml**

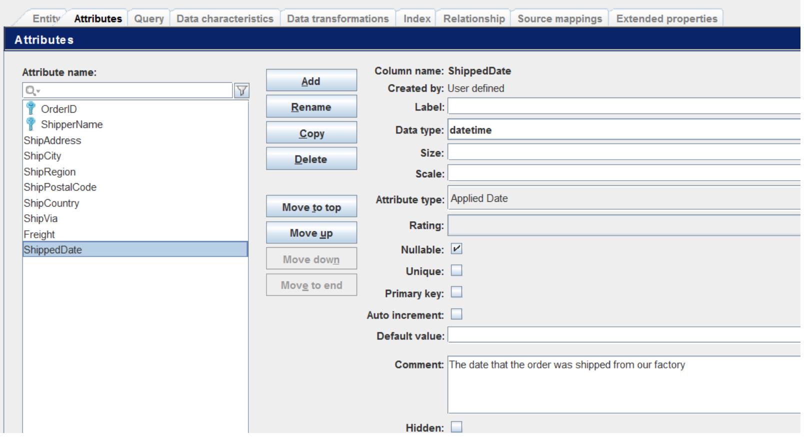
Template

No additional templates are required

Steps to implement

If a particular stage is not mentioned, then you can assume it would follow the standard process of generating a Data Vault.

#	Category	Description
1	Data Vault Design	For the purpose of this document, the Shipments-table has been chosen to create an Effectivity Satellite upon. In this scenario the OrderID shall be the DrivingKey (dss_driving_key) and the ShippedDate has been declared as Applied Date (dss_applied_date) - which is purely for the example purposes. The entity containing the link business keys has been set to the entity-type "Effectivity Tracking"  Entity Entity name: Shipments Entity group: Label: Entity icon: Entity type: Effectivity Tracking Rating: Target location: Comment: Subjects: General info Document general information:

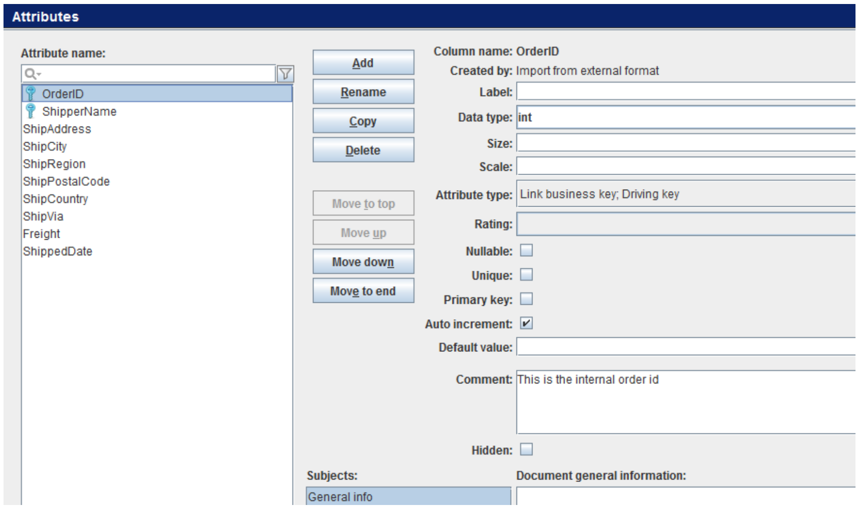


Attributes

Attribute name:

- OrderID
- ShipperName
- ShipAddress
- ShipCity
- ShipRegion
- ShipPostalCode
- ShipCountry
- ShipVia
- Freight
- ShippedDate

Column name: ShippedDate
Created by: User defined
Label:
Data type: datetime
Size:
Scale:
Attribute type: Applied Date
Rating:
Nullable: ☑
Unique: □
Primary key: □
Auto increment: □
Default value:
Comment: The date that the order was shipped from our factory
Hidden: □



Attributes

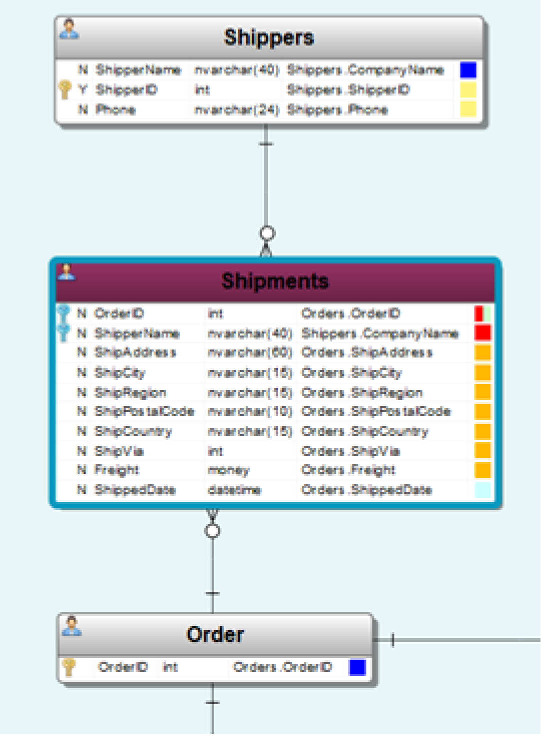
Attribute name:

- OrderID
- ShipperName
- ShipAddress
- ShipCity
- ShipRegion
- ShipPostalCode
- ShipCountry
- ShipVia
- Freight
- ShippedDate

Column name: OrderID
Created by: Import from external format
Label:
Data type: int
Size:
Scale:
Attribute type: Link business key; Driving key
Rating:
Nullable: □
Unique: □
Primary key: □
Auto increment: ☑
Default value:
Comment: This is the internal order id
Hidden: □

Subjects: General info

Document general information:



Shippers

N	ShipperName	nvarchar(40)	Shippers.CompanyName
Y	ShipperID	int	Shippers.ShipperID
N	Phone	nvarchar(24)	Shippers.Phone

Shipments

N	OrderID	int	Orders.OrderID
N	ShipperName	nvarchar(40)	Shippers.CompanyName
N	ShipAddress	nvarchar(60)	Orders.ShipAddress
N	ShipCity	nvarchar(15)	Orders.ShipCity
N	ShipRegion	nvarchar(15)	Orders.ShipRegion
N	ShipPostalCode	nvarchar(10)	Orders.ShipPostalCode
N	ShipCountry	nvarchar(15)	Orders.ShipCountry
N	ShipVia	int	Orders.ShipVia
N	Freight	money	Orders.Freight
N	ShippedDate	datetime	Orders.ShippedDate

Order

OrderID	int	Orders.OrderID

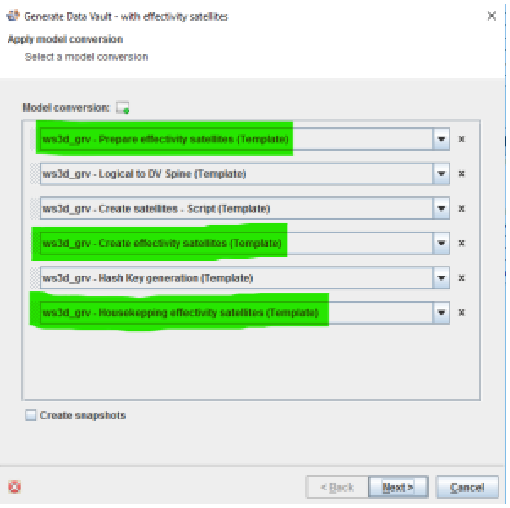
Things to note:

- Attribute types that are being used are:
  - Driving Key (light green)
  - Applied Date (light blue)
  - Entity-Type "Effectivity Tracking" (burgundy)
- The single source of each attribute is shown in the diagram
- Using the dss_applied_date is optional - if not used, the dss_create_time will be used as source

2

Data Vault Model

Generate from the Data Vault Design Model with the standard-process enhanced with the following Rules:



Generate Data Vault - with effectivity satellites

Apply model conversion

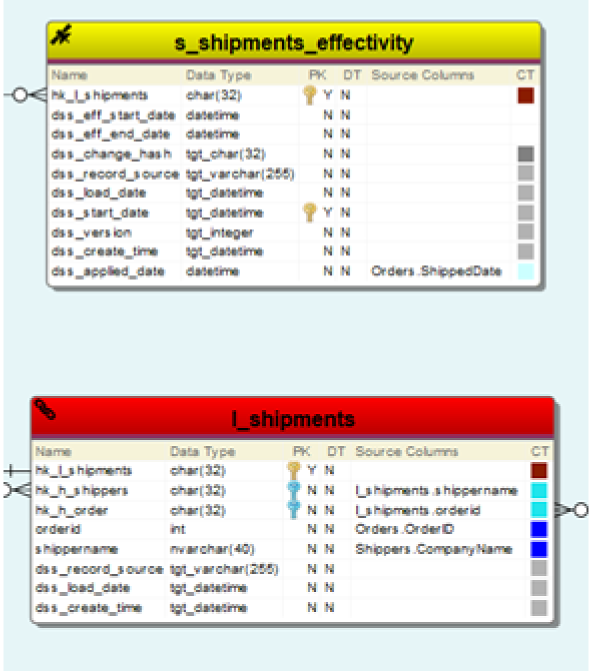
Select a model conversion

Model conversion:

1. ws3d_grv - Prepare effectivity satellites (Template)
2. ws3d_grv - Logical to DV Spine (Template)
3. ws3d_grv - Create satellites - Script (Template)
4. ws3d_grv - Create effectivity satellites (Template)
5. ws3d_grv - Hash Key generation (Template)
6. ws3d_grv - Housekeeping effectivity satellites (Template)

□ Create snapshots

The Data Vault Model should look like this



s_shipments_effectivity

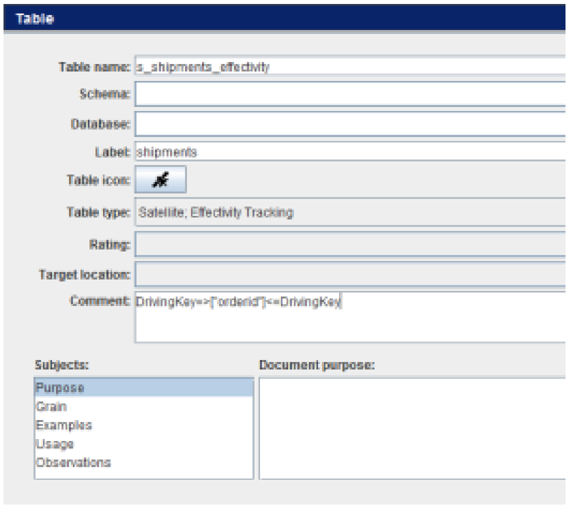
Name	Data Type	PK	DT	Source Columns	CT
hk_l_shipments	char(32)	Y	N		🟥
dss_eff_start_date	datetime	N	N		
dss_eff_end_date	datetime	N	N		
dss_change_hash	tgt_char(32)	N	N		▨
dss_record_source	tgt_varchar(255)	N	N		▨
dss_load_date	tgt_datetime	N	N		▨
dss_start_date	tgt_datetime	Y	N		▨
dss_version	tgt_integer	N	N		▨
dss_create_time	tgt_datetime	N	N		▨
dss_applied_date	datetime	N	N	Orders.ShippedDate	🟦

l_shipments

Name	Data Type	PK	DT	Source Columns	CT
hk_l_shipments	char(32)	Y	N		🟥
hk_h_shippers	char(32)	N	N	l_shipments.shippername	🟦
hk_h_order	char(32)	N	N	l_shipments.orderid	🟦
orderid	int	N	N	Orders.OrderID	🟦
shippername	nvarchar(40)	N	N	Shippers.CompanyName	🟦
dss_record_source	tgt_varchar(255)	N	N		▨
dss_load_date	tgt_datetime	N	N		▨
dss_create_time	tgt_datetime	N	N		▨

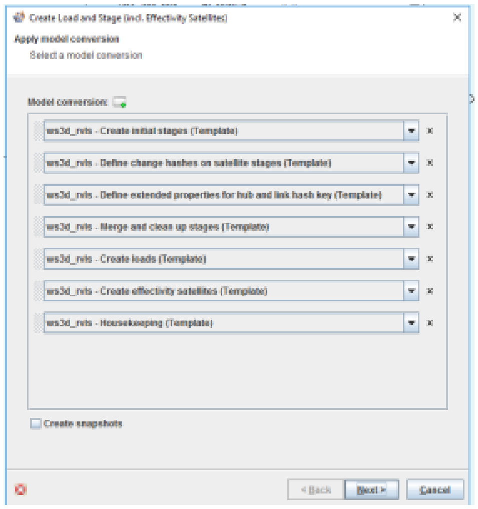
Things to note:

- All of the extra DV control columns have been added to the entities as per DV 2.0 rules
- Hub and Change hash keys have been added where needed.
- The setting of the attribute "Driving Key" has been copied into the comment of the newly created satellite.

		 Table Table name: s_shipments_effectivity Schema: Database: Label: shipments Table icon: Table type: Satellite; Effectivity Tracking Rating: Target location: Comment: DrivingKey=>["orderid"]<=DrivingKey Subjects: Purpose Grain Examples Usage Observations Document purpose:

<table>
<tr>
<td>3</td>
<td><b>Load and Staging Model</b></td>
<td>

Generate from the Data Vault Design Model with the standard-process enhanced with the following Rules:



Create Load and Stage (incl. Effectivity Satellites)

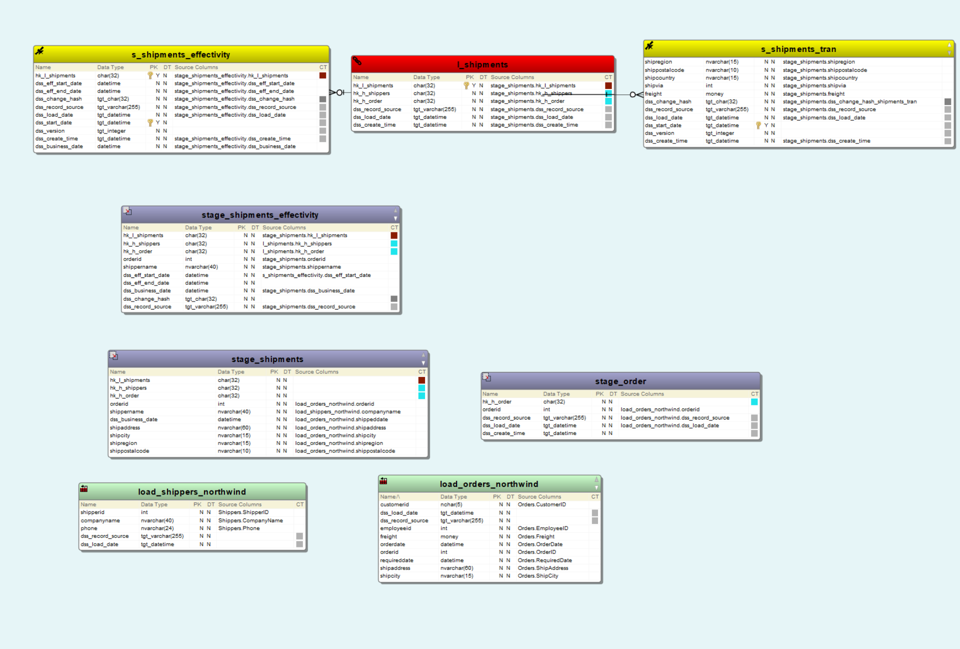
Apply model conversion

Select a model conversion

Model conversion:

- ws3d_rvis - Create initial stages (Template)
- ws3d_rvis - Define change hashes on satellite stages (Template)
- ws3d_rvis - Define extended properties for hub and link hash key (Template)
- ws3d_rvis - Merge and clean up stages (Template)
- ws3d_rvis - Create loads (Template)
- ws3d_rvis - Create effectivity satellites (Template)
- ws3d_rvis - Housekeeping (Template)

☐ Create snapshots



s_shipments_effectivity

Name	Data Type	PK	DT	Source Columns	CT
hk_l_shipments	char(32)	Y	N	stage_shipments_effectivity.hk_l_shipments	🟥
dss_eff_start_date	datetime	N	N	stage_shipments_effectivity.dss_eff_start_date	
dss_eff_end_date	datetime	N	N	stage_shipments_effectivity.dss_eff_end_date	
dss_change_hash	tgt_char(32)	N	N	stage_shipments_effectivity.dss_change_hash	⬛
dss_record_source	tgt_varchar(255)	N	N	stage_shipments_effectivity.dss_record_source	▨
dss_load_date	tgt_datetime	N	N	stage_shipments_effectivity.dss_load_date	▨
dss_start_date	tgt_datetime	Y	N		▨
dss_version	tgt_integer	N	N		▨
dss_create_time	tgt_datetime	N	N	stage_shipments_effectivity.dss_create_time	▨
dss_business_date	datetime	N	N	stage_shipments_effectivity.dss_business_date	

l_shipments

Name	Data Type	PK	DT	Source Columns	CT
hk_l_shipments	char(32)	Y	N	stage_shipments.hk_l_shipments	🟥
hk_h_shippers	char(32)	N	N	stage_shipments.hk_h_shippers	🟦
hk_h_order	char(32)	N	N	stage_shipments.hk_h_order	🟦
dss_record_source	tgt_varchar(255)	N	N	stage_shipments.dss_record_source	▨
dss_load_date	tgt_datetime	N	N	stage_shipments.dss_load_date	▨
dss_create_time	tgt_datetime	N	N	stage_shipments.dss_create_time	▨

s_shipments_tran

shipregion	nvarchar(15)	N	N	stage_shipments.shipregion	
shippostalcode	nvarchar(10)	N	N	stage_shipments.shippostalcode	
shipcountry	nvarchar(15)	N	N	stage_shipments.shipcountry	
shipvia	int	N	N	stage_shipments.shipvia	
freight	money	N	N	stage_shipments.freight	
dss_change_hash	tgt_char(32)	N	N	stage_shipments.dss_change_hash_shipments_tran	⬛
dss_record_source	tgt_varchar(255)	N	N	stage_shipments.dss_record_source	▨
dss_load_date	tgt_datetime	N	N	stage_shipments.dss_load_date	▨
dss_start_date	tgt_datetime	Y	N		▨
dss_version	tgt_integer	N	N		▨
dss_create_time	tgt_datetime	N	N	stage_shipments.dss_create_time	▨

stage_shipments_effectivity

Name	Data Type	PK	DT	Source Columns	CT
hk_l_shipments	char(32)	N	N	stage_shipments.hk_l_shipments	🟥
hk_h_shippers	char(32)	N	N	l_shipments.hk_h_shippers	🟦
hk_h_order	char(32)	N	N	l_shipments.hk_h_order	🟦
orderid	int	N	N	stage_shipments.orderid	
shippername	nvarchar(40)	N	N	stage_shipments.shippername	
dss_eff_start_date	datetime	N	N	s_shipments_effectivity.dss_eff_start_date	
dss_eff_end_date	datetime	N	N		
dss_business_date	datetime	N	N	stage_shipments.dss_business_date	
dss_change_hash	tgt_char(32)	N	N		⬛
dss_record_source	tgt_varchar(255)	N	N	stage_shipments.dss_record_source	▨

stage_shipments

Name	Data Type	PK	DT	Source Columns	CT
hk_l_shipments	char(32)	N	N		🟥
hk_h_shippers	char(32)	N	N		🟦
hk_h_order	char(32)	N	N		🟦
orderid	int	N	N	load_orders_northwind.orderid	
shippername	nvarchar(40)	N	N	load_shippers_northwind.companyname	
dss_business_date	datetime	N	N	load_orders_northwind.shippeddate	
shipaddress	nvarchar(60)	N	N	load_orders_northwind.shipaddress	
shipcity	nvarchar(15)	N	N	load_orders_northwind.shipcity	
shipregion	nvarchar(15)	N	N	load_orders_northwind.shipregion	
shippostalcode	nvarchar(10)	N	N	load_orders_northwind.shippostalcode	

stage_order

Name	Data Type	PK	DT	Source Columns	CT
hk_h_order	char(32)	N	N		🟦
orderid	int	N	N	load_orders_northwind.orderid	
dss_record_source	tgt_varchar(255)	N	N	load_orders_northwind.dss_record_source	▨
dss_load_date	tgt_datetime	N	N	load_orders_northwind.dss_load_date	▨
dss_create_time	tgt_datetime	N	N		▨

load_shippers_northwind

Name	Data Type	PK	DT	Source Columns	CT
shipperid	int	N	N	Shippers.ShipperID	
companyname	nvarchar(40)	N	N	Shippers.CompanyName	
phone	nvarchar(24)	N	N	Shippers.Phone	
dss_record_source	tgt_varchar(255)	N	N		▨
dss_load_date	tgt_datetime	N	N		▨

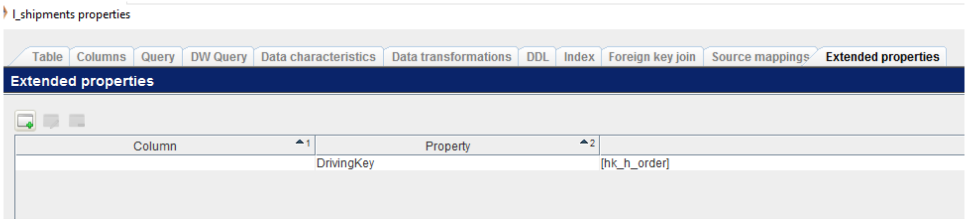
load_orders_northwind

Name/\	Data Type	PK	DT	Source Columns	CT
customerid	nchar(5)	N	N	Orders.CustomerID	
dss_load_date	tgt_datetime	N	N		▨
dss_record_source	tgt_varchar(255)	N	N		▨
employeeid	int	N	N	Orders.EmployeeID	
freight	money	N	N	Orders.Freight	
orderdate	datetime	N	N	Orders.OrderDate	
orderid	int	N	N	Orders.OrderID	
requireddate	datetime	N	N	Orders.RequiredDate	
shipaddress	nvarchar(60)	N	N	Orders.ShipAddress	
shipcity	nvarchar(15)	N	N	Orders.ShipCity	

Things to note:

- There should be an additional stage for every effective satellite which is sourced mainly from the original stage for the link - but also from the link and the effectivity-satellite itself

The setting "Driving Key" from the comment has now moved to the extended Properties:



l_shipments properties

Extended properties

Column	Property	
	DrivingKey	[hk_h_order]

</td>
</tr>
<tr>
<td>4</td>
<td><b>Export to RED</b></td>
<td>Export the RED Export model version to RED using the Basic Process described previously.
The model should deploy to RED without any issues and all Load, Stage and Data Vault objects can be loaded successfully.</td>
</tr>
</table>

WhereScape RED

*Currently only target SQL Server is supported

Dependent objects

Templates

3 new templates need to be used or the logic incorporated into existing templates

- cust_sqlserver_proc_dv_stage (Procedure template for Data Vault Stage)
- cust_sqlserver_proc_dv_stage_effectivity_utility
- cust_sqlserver_utility_dv

Steps to implement

Via application installation

#	Description of change
1	Unzip file app_tem_SQLDV_EFSAT_202210040854
2	Install application DV_SQL_EFSAT from wsl files in the zipped folder

Via copying the content from template files

The 3 templates have also been attached separately to look at.