Contents

[Overview 1](#_Toc175852583)

[Architecture 1](#_Toc175852584)

[Services/Tasks Used 2](#_Toc175852585)

[Source 2](#_Toc175852586)

[Transformation 2](#_Toc175852587)

[Target 4](#_Toc175852588)

[Dimensions 4](#_Toc175852589)

[Facts 4](#_Toc175852590)

[Folder Structure 5](#_Toc175852591)

[1. Executor 5](#_Toc175852592)

[2. load\_gold\_dimension 5](#_Toc175852593)

[3. load\_gold\_fact 6](#_Toc175852594)

[4. loadstaging 6](#_Toc175852595)

[Pipeline Execution 7](#_Toc175852596)

[1. Level\_0 7](#_Toc175852597)

[2. Level\_2 8](#_Toc175852598)

[3. Level\_3 8](#_Toc175852599)

# Overview

## Architecture

A diagram of a cylinder

Description automatically generated

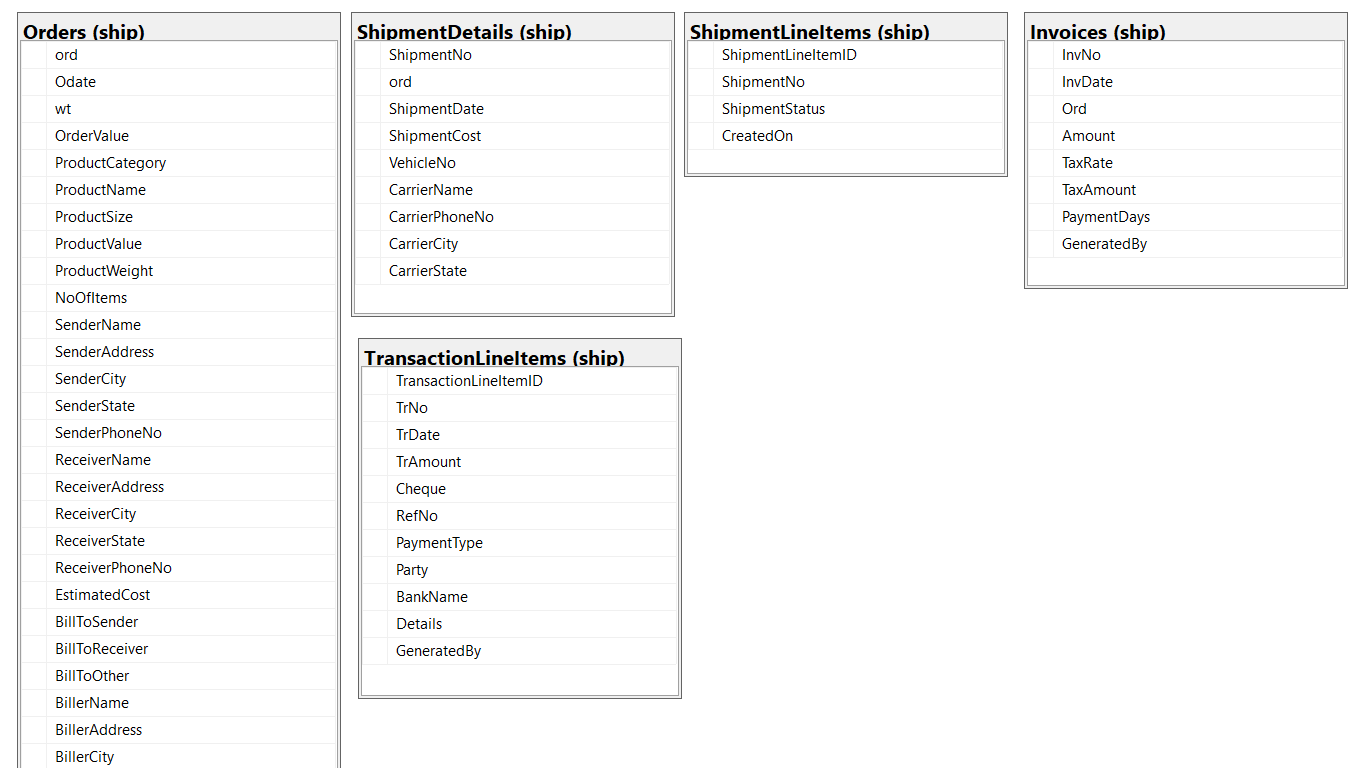
1. ShipmentTracking is source database hold business information.
2. Load Staging pipelines in Azure Data Factory loads data incrementally in ST\_DW database under staging schema. Here raw data is transformed, columns renamed. All data transformation is done using Stored Procedures and Data Flow.
3. Load dimensions and facts pipelines in Azure data factory converts data in dimensions and facts format and load into ST\_DW database under dw schema.

## Services/Tasks Used

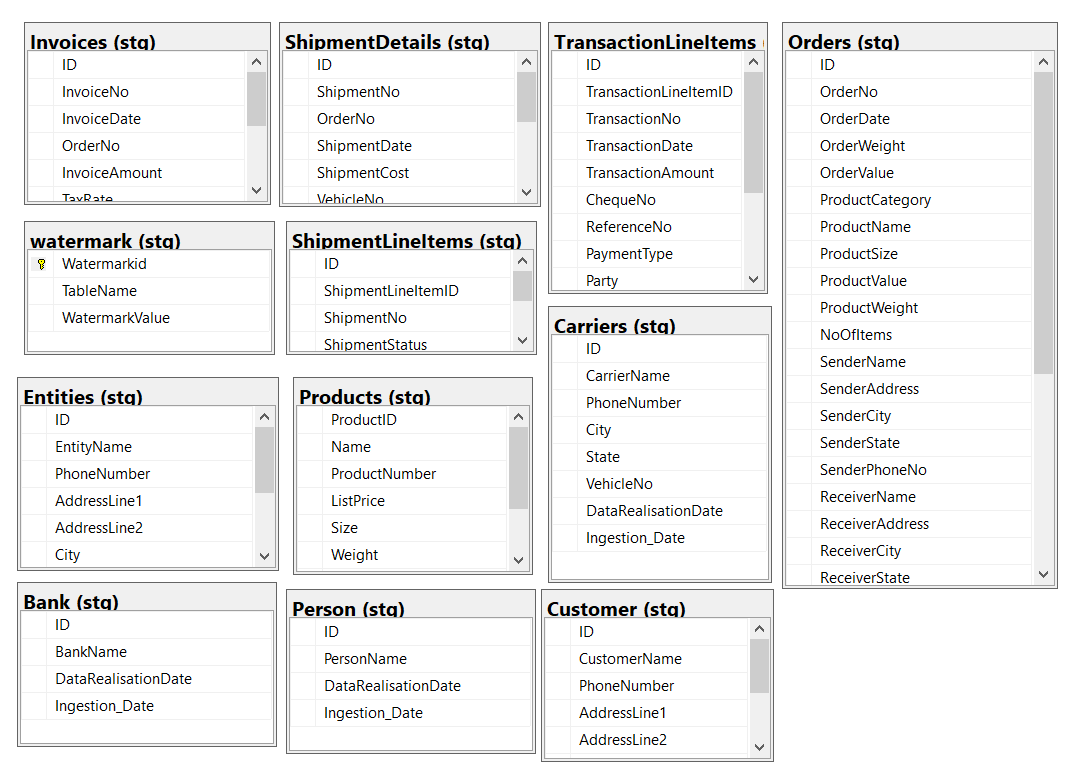
1. Azure SQL Database
2. Azure Key Vault
3. Azure Data Bricks
4. Azure Data Factory

* Lookup
* Set variable
* Stored procedure
* Execute pipeline
* Data flow
* Linked service
* Triggers

## Source

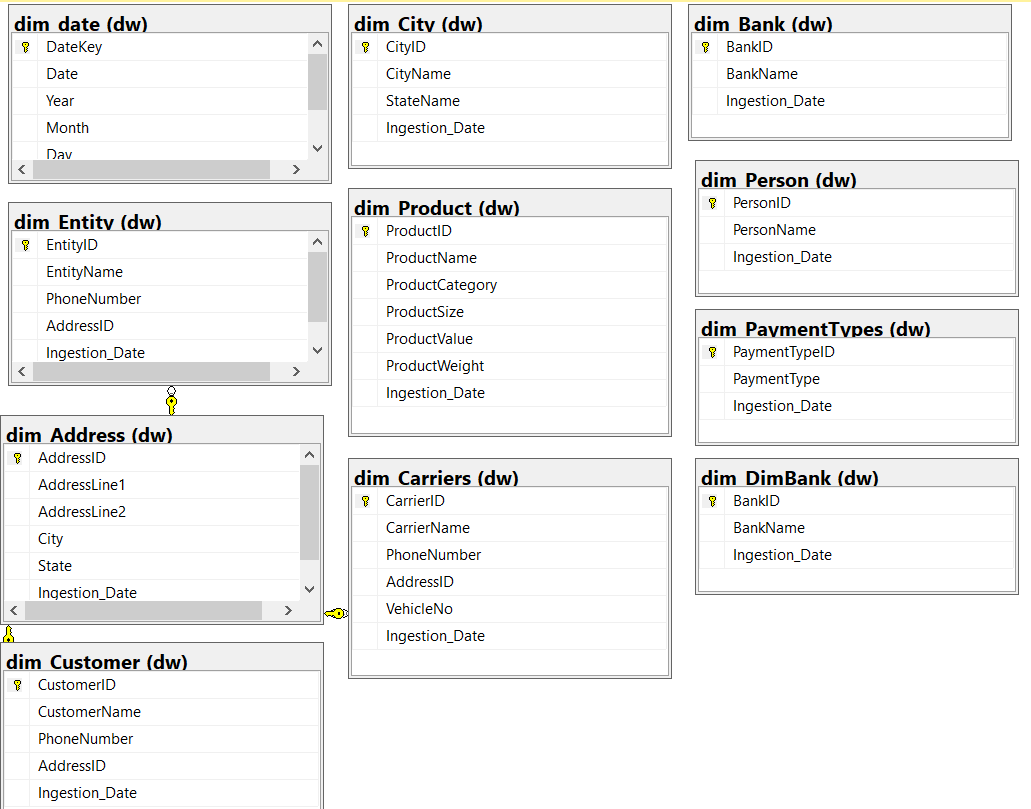


## Transformation

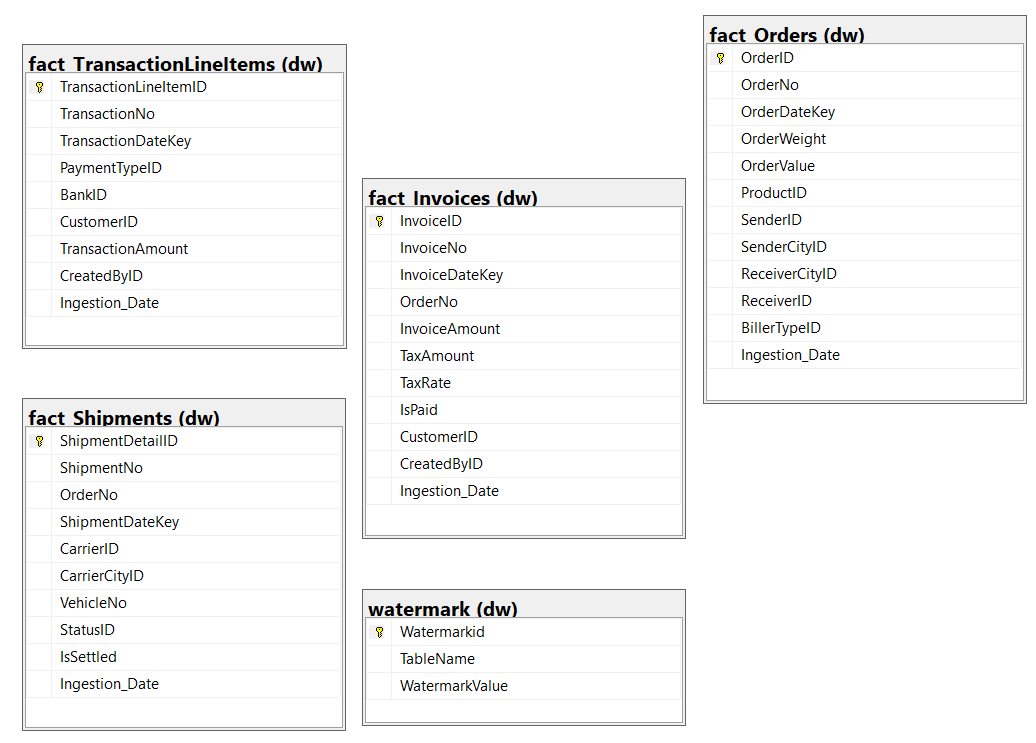


## Target

### Dimensions



### Facts



## Folder Structure

### 1. Executor

A screenshot of a computer

Description automatically generated

### 2. load\_gold\_dimension

A screenshot of a computer

Description automatically generated

### 3. load\_gold\_fact

A screenshot of a computer

Description automatically generated

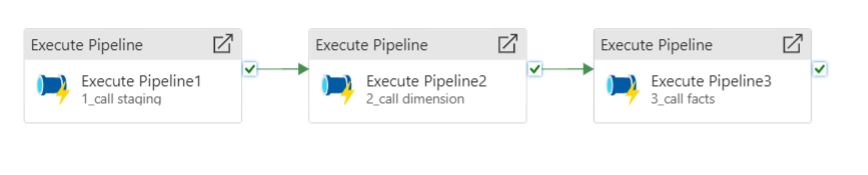
### 4. loadstaging

A screenshot of a computer

Description automatically generated

## Pipeline Execution

### Level\_0



At Level\_0 we have master pipeline or we can say parent pipeline, Purpose of this pipeline is to call staging, populate dimensions and populate facts pipelines.

Pipelines are called in sequence, in case of any failure following pipelines are halted.

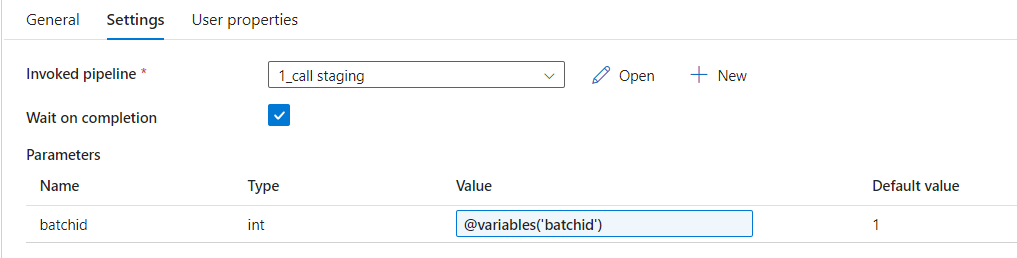
Pipeline accepts a variable batchid

A screenshot of a computer

Description automatically generated

Batchid plays a role to identify when a data is populated.

Each pipeline accepts batchid as a parameter.



execute staging, dimension & facts master pipelines (detailing parameters & variables)

### Level\_2

A screenshot of a computer

Description automatically generated

At Level\_2 we have child pipelines which call further worker pipelines.

### 3. Level\_3

Level\_3 is the worker pipelines used to copy/move, transform and load data from one format to other. Here we are using stored procedures to do the processing as Source and Destination database exist on same server.

#### Load Staging

A diagram of a computer

Description automatically generated

#### Populate Dimensions

A screenshot of a computer

Description automatically generated

#### Populate facts

A computer screen shot of a diagram

Description automatically generated

For worker pipeline I have used above set of activities to perform loading and transformation.

**Get\_pipelinelogid** : It is a lookup activity which executes a procedure to create a new row for each pipeline in pipeline log and return the logid. Which is further used to update the success and failure status of pipeline.

Create procedure [dbo].[get\_pipelinelog](@batchid int,@PipelineName varchar(50))

as

begin

declare @pipelinelogid int

Insert Into pipelinelog(BatchID,PipelineName,ExecutionStatus,CreatedOn)

Values(@batchid,@PipelineName,NULL,getdate())

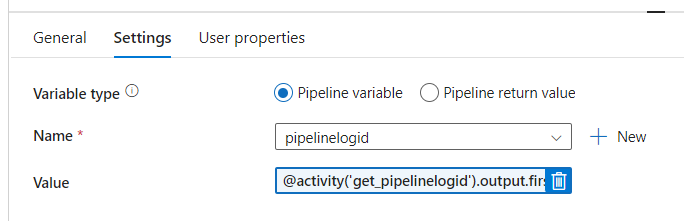
set @pipelinelogid = SCOPE\_IDENTITY()

Select @pipelinelogid as PipelineLogID

End

GO

**Set variable :** In this activity we set output of above activity to a variable.

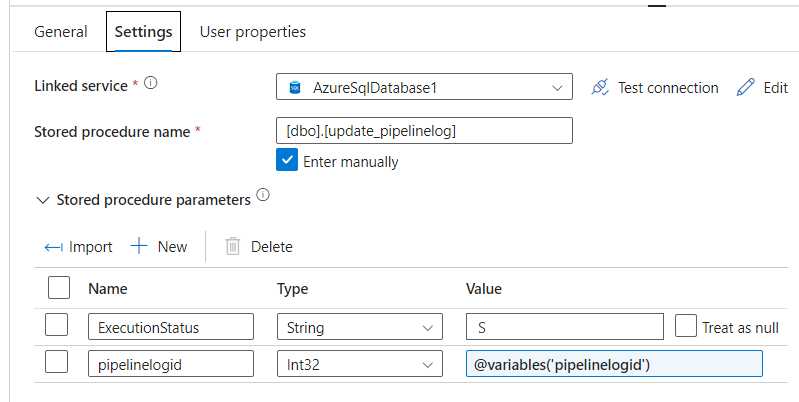


**Load dw fact\_invoices :** Under this activity I am calling stored procedure which perform all the data transformation and loading.

A screenshot of a computer

Description automatically generated

**Update Success Log :** Under this activity I execute a stored procedure which accepts the pipelinelogid and Success Status and update the log.



**Update Watermark Value :** I am using a master table to support incremental load of data. In this table I manage one entry for each table.

A screenshot of a list

Description automatically generated

After successful execution of data load activity, this activity executes and set WatermarkValue to max data date. Thus in next run I am able to support incremental load of data.

**Update Fail Log :** Update fail log activity is executed whenever there is any error. Here I am executing a stored procedure which accepts pipelinelogid and Failure status.

A screenshot of a computer

Description automatically generated