

DATA WAREHOUSING AND BUSINESS INTELLIGENCE – MIDTERM PROJECT (SPRING 2018)

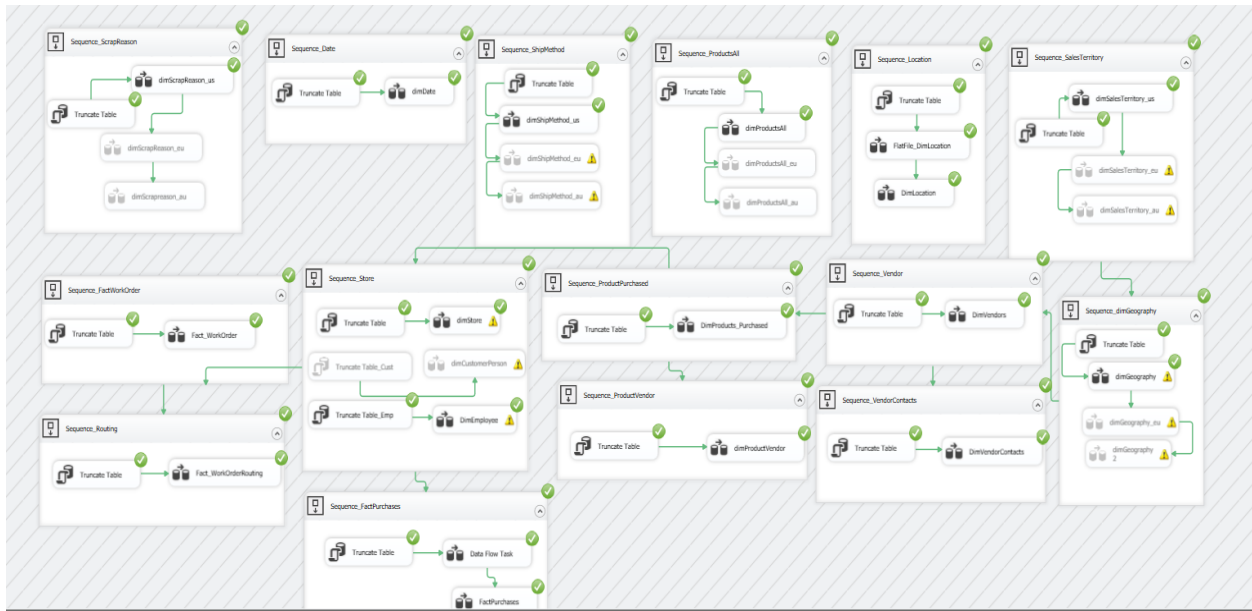
TEAM 4

Schema – adventureworks2016_us

List of Tables and counts:

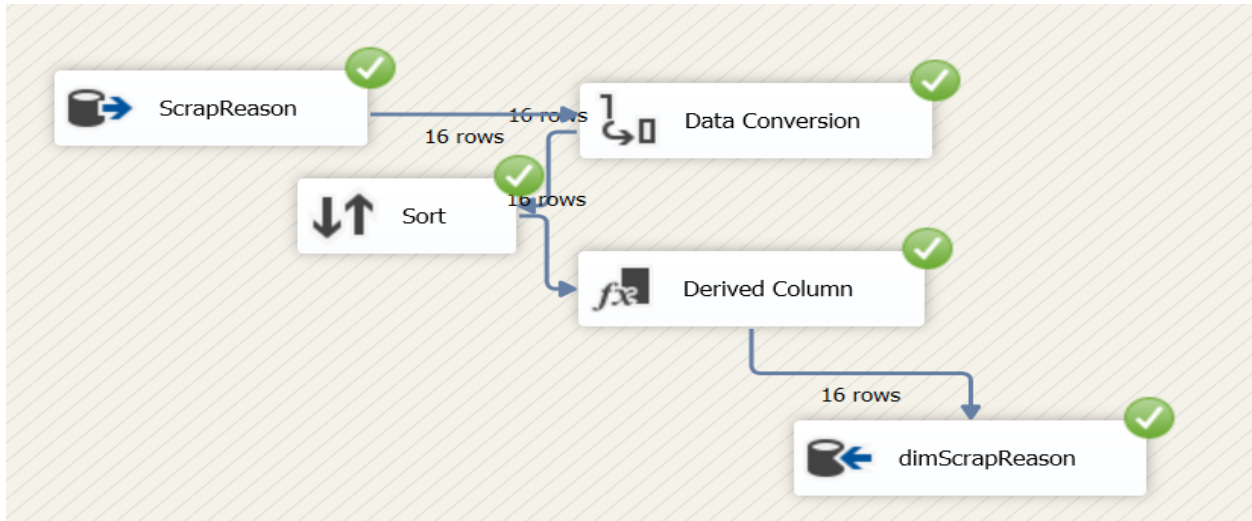
DimDate	3652 rows
DimEmployee	284 rows
DimGeography	402 rows
DimLocation	14 rows
DimProducts_Purchased	265 rows
DimProductsAll	504 rows
DimProductVendor	460 rows
DimSalesTerritory	5 rows
DimScrapReason	16 rows
DimShipMethod	5 rows
DimStore	427 rows
DimVendorContacts	156 rows
DimVendors	104 rows
Fact_WorkOrder	71856 rows
Fact_WorkOrder_Rejects	735 rows
Fact_WorkOrderRouting	66748 rows
Fact_WorkOrderRouting_Rejects	383 rows
FactPurchases	9578 rows
FactPurchases_Rejects	152 rows

SSIS Control Flow:

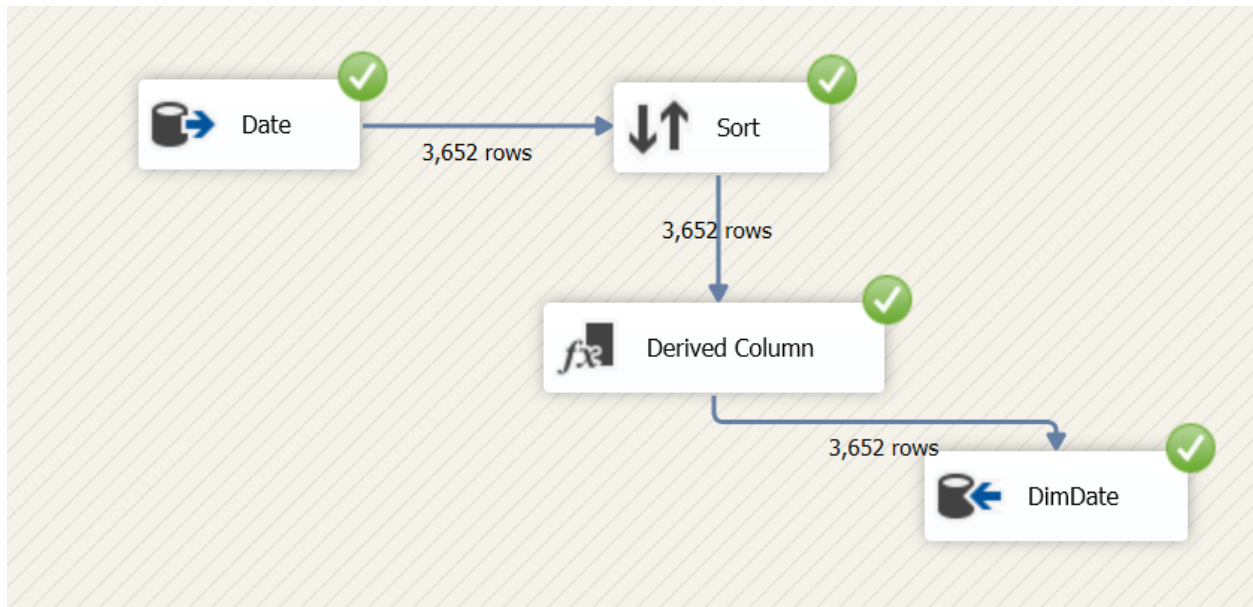


SSIS Jobs:

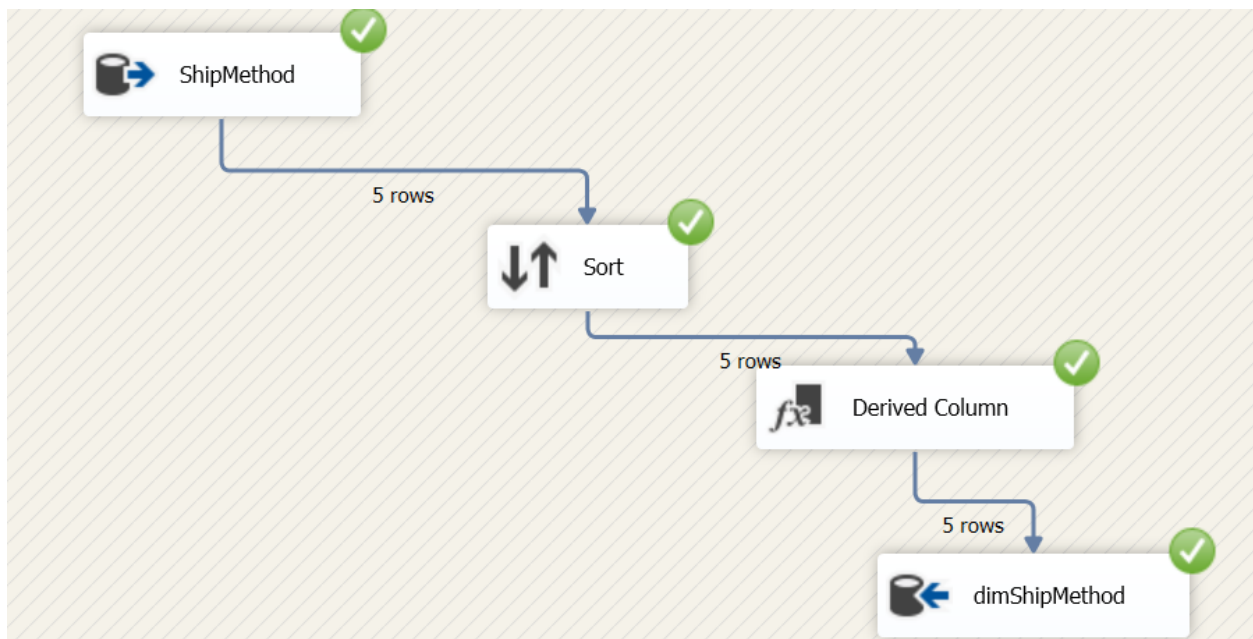
DimScrapReasons



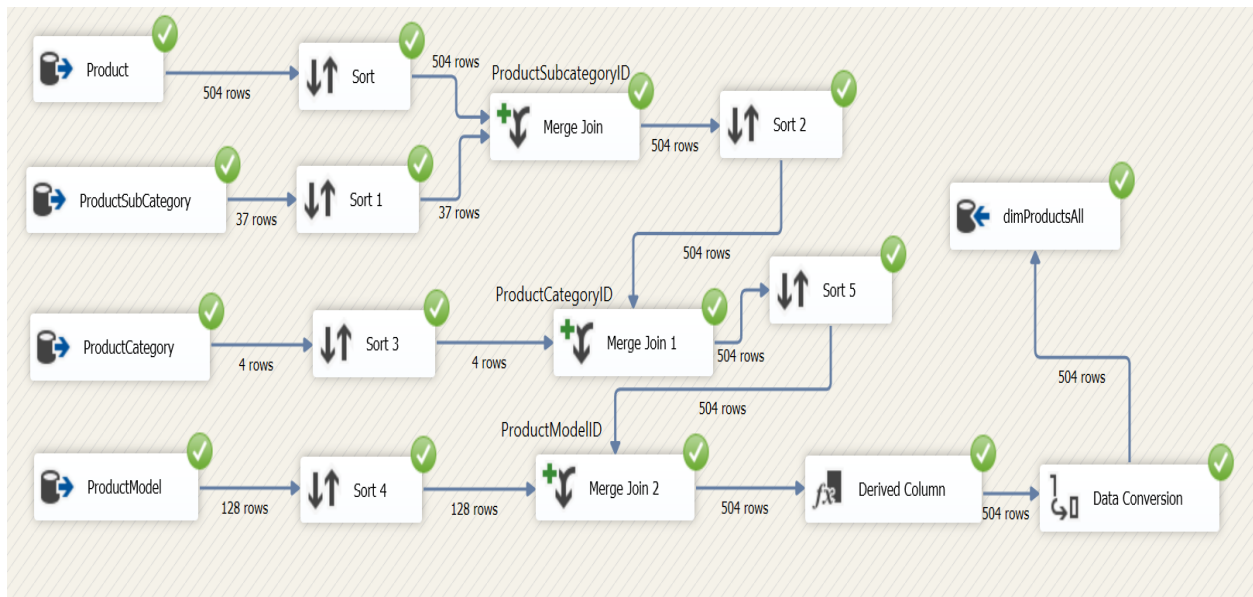
DimDate:



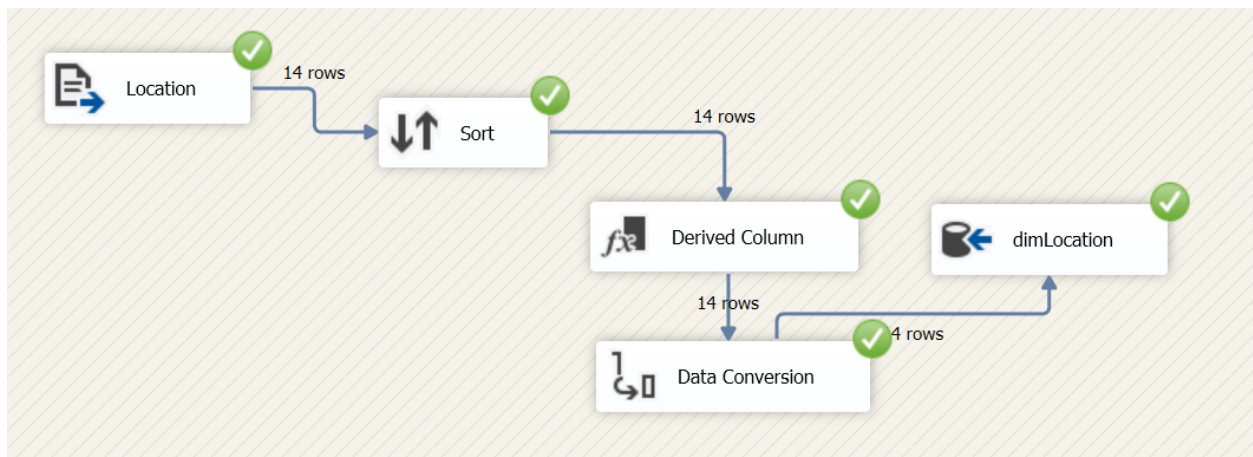
DimShipMethod:



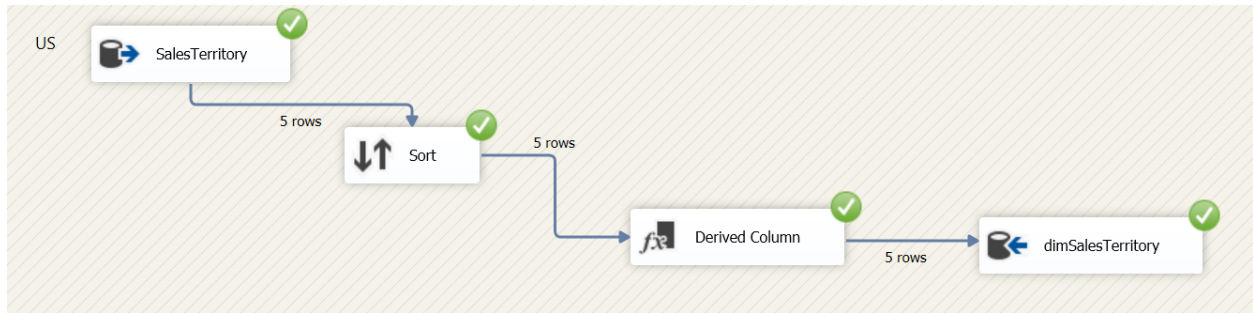
DimProductsAll:



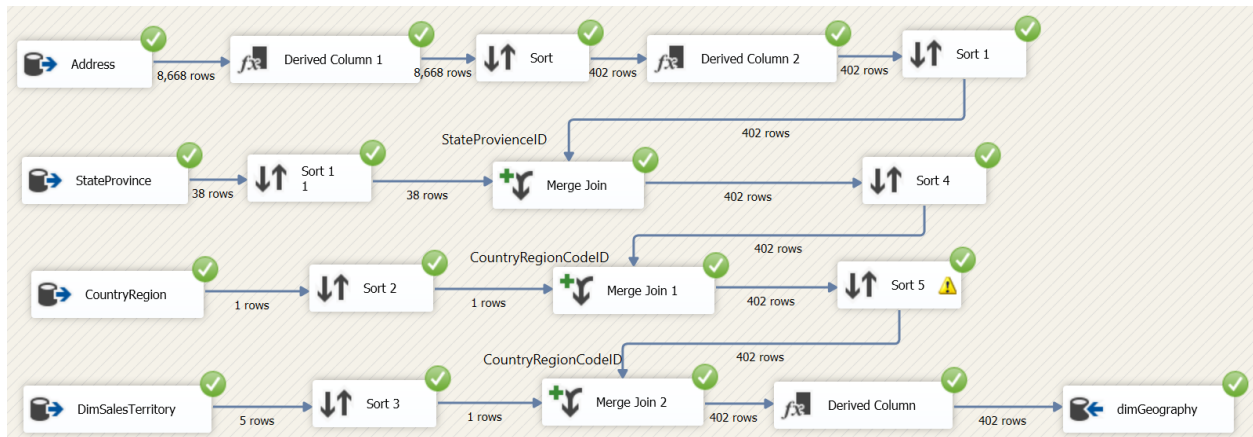
DimLocation:



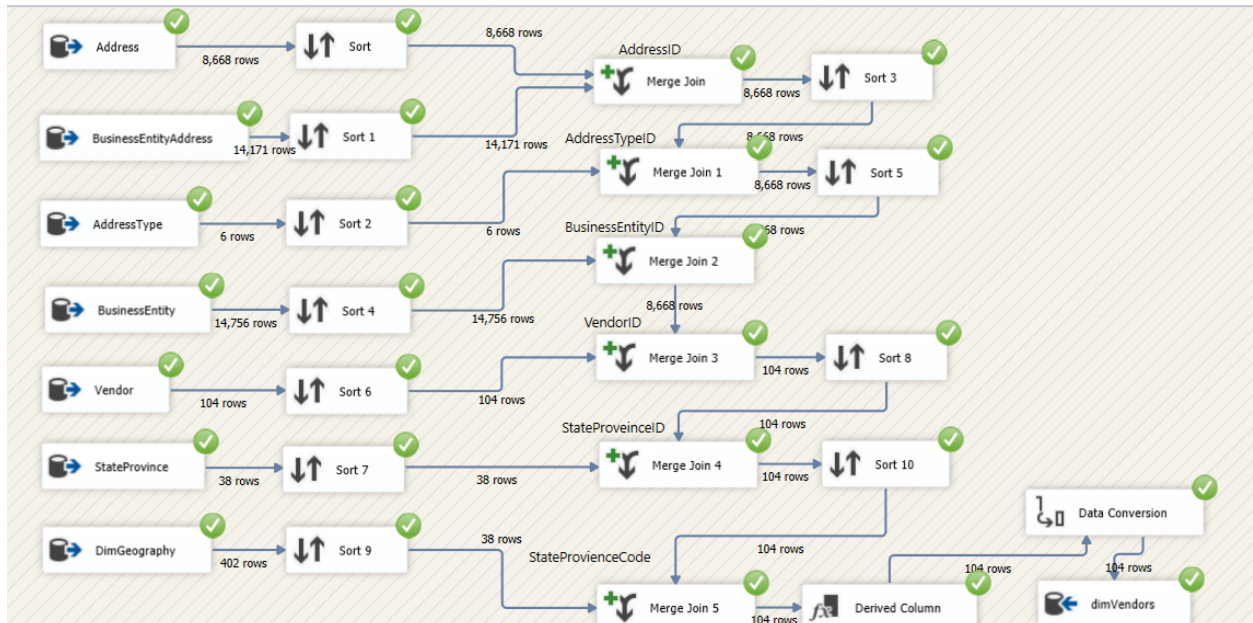
DimSalesTerritory:



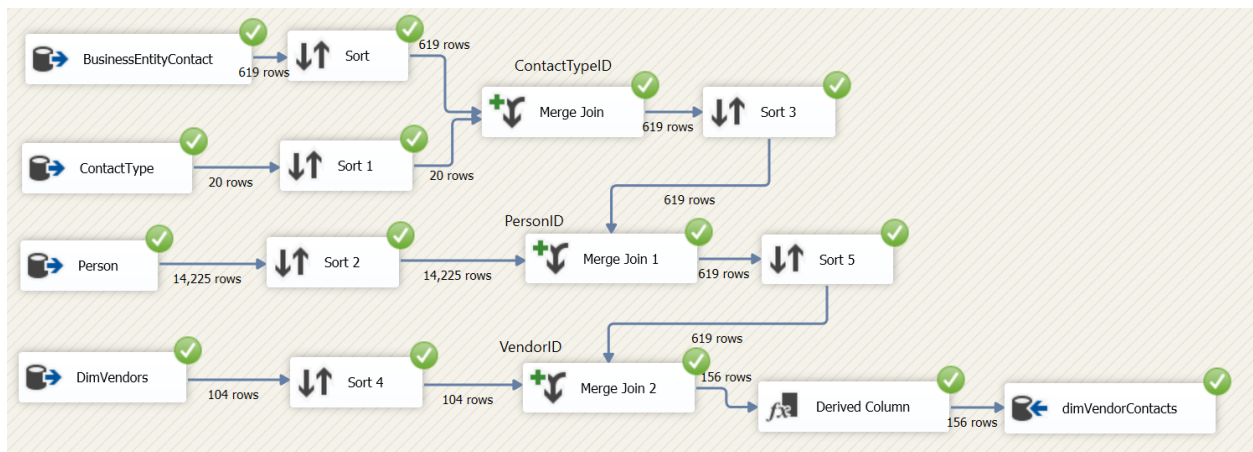
DimGeography:



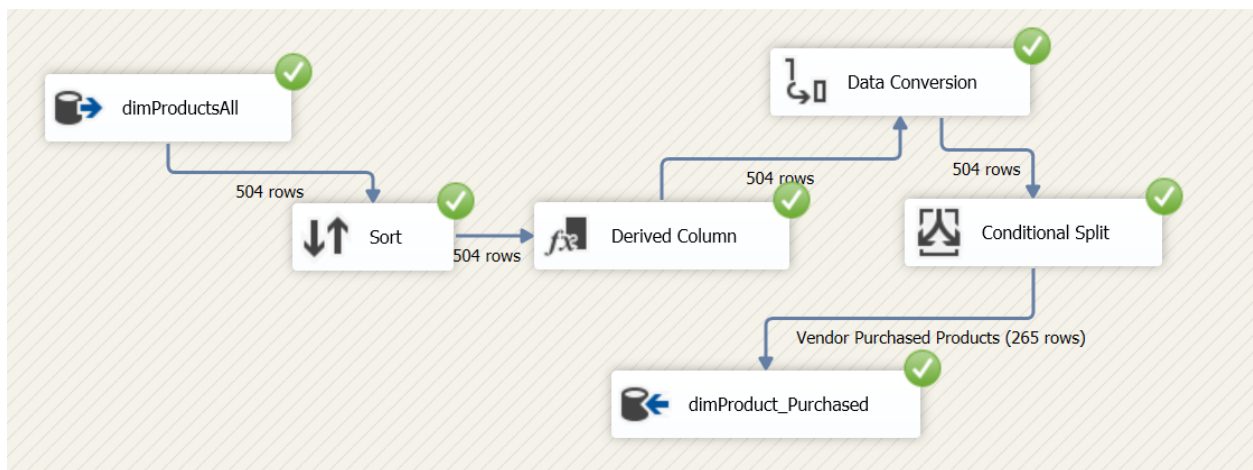
DimVendors:



DimVendorContacts:



DimProductPurchased:




```

graph LR
    PV[ProductVendor  
460 rows] --> S1[Sort]
    S1 -- 460 rows --> MJ1[Merge Join]
    DPP[dimProductsPurchased  
265 rows] --> S2[Sort 1]
    S2 -- 265 rows --> MJ1
    MJ1 -- 460 rows --> S3[Sort 3]
    DV[dimVendor  
104 rows] --> S4[Sort 2]
    S4 -- 104 rows --> MJ2[Merge Join 1]
    MJ1 -- 460 rows --> MJ2
    MJ2 -- 460 rows --> DC[Derived Column]
    DC -- 460 rows --> DCON[Data Conversion]
    DCON -- 460 rows --> DPV[dimProductVendor]
  
```

The diagram illustrates a complex ETL process for a data warehouse. It starts with multiple source tables on the left: Employee (284 rows), EmployeeDepHistory (296 rows), Department (16 rows), EmployeePayHistory (316 rows), Person (14,325 rows), Email Address (14,325 rows), and PersonPhone (14,325 rows). These sources feed into a series of Sort and Merge Join operations. For example, Employee and EmployeeDepHistory are joined and sorted, then merged with Department. The process continues with EmployeePayHistory, Person, Email Address, and PersonPhone, all of which are joined and sorted before being merged into a single stream. This stream then feeds into a second set of operations involving BusinessEntityID and BusinessEntityAddress, which are also joined and sorted. The final output is a Data Conversion step leading to dimEmployee (284 rows). The diagram includes row counts at each stage and green checkmarks indicating successful execution.

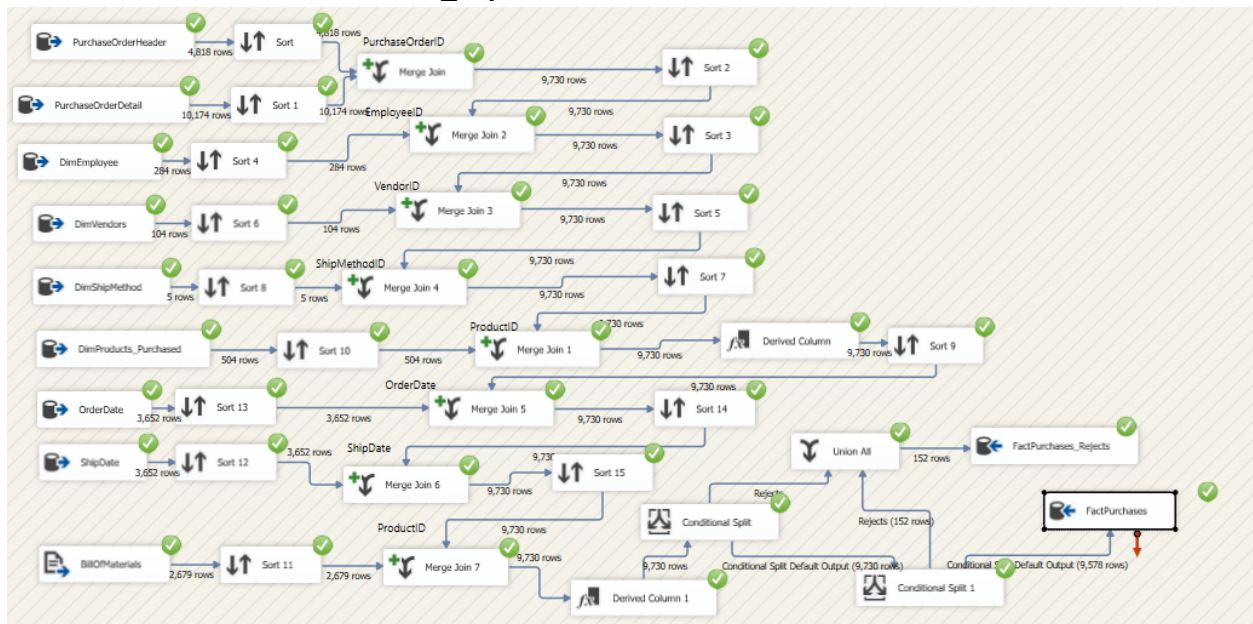
The diagram illustrates a complex data transformation process in Power BI. It starts with six source tables: WorkOrder (72,591 rows), dimProductsAll (504 rows), dimScrapReason (16 rows), StartDate (3,652 rows), EndDate (3,652 rows), and DueDate (3,652 rows). The data flows through several intermediate steps:

- WorkOrder** is sorted and merged with **dimProductsAll** (Sort 1) and **dimScrapReason** (Sort 3) into **Merge Join 1** (72,591 rows).
- Merge Join 1** output is converted to a data type and sorted (Sort 2) to produce **Derived Column 1** (72,591 rows).
- Derived Column 1** is sorted (Sort 4) and merged with **StartDate** (Sort 5) into **Merge Join 2** (72,591 rows).
- Merge Join 2** output is sorted (Sort 6) and merged with **EndDate** (Sort 5) into **Merge Join 3** (72,591 rows).
- Merge Join 3** output is sorted (Sort 8) and merged with **DueDate** (Sort 5) into **Merge Join 4** (72,591 rows).
- Merge Join 4** output is converted to a data type and merged with **Derived Column 1** into **Derived Column 2** (72,591 rows).
- Derived Column 2** is split conditionally (Conditional Split 1) into **NULL_StockQty (5 rows)** and **NOTNULL_StockQty (72,586 rows)**.
- NOTNULL_StockQty** is split conditionally (Conditional Split 2) into **Rejects (729 rows)** and **Approved (71,856 rows)**.
- Rejects** are merged with **NULL_StockQty** into **Union All** (735 rows).
- Union All** is merged with **Fact_WorkOrder_Rejects** (735 rows) into the final **Fact_WorkOrder** table (72,586 rows).

The diagram illustrates the data flow for a Power BI report, showing the transformation of data from source tables into a final fact table. The flow is as follows:

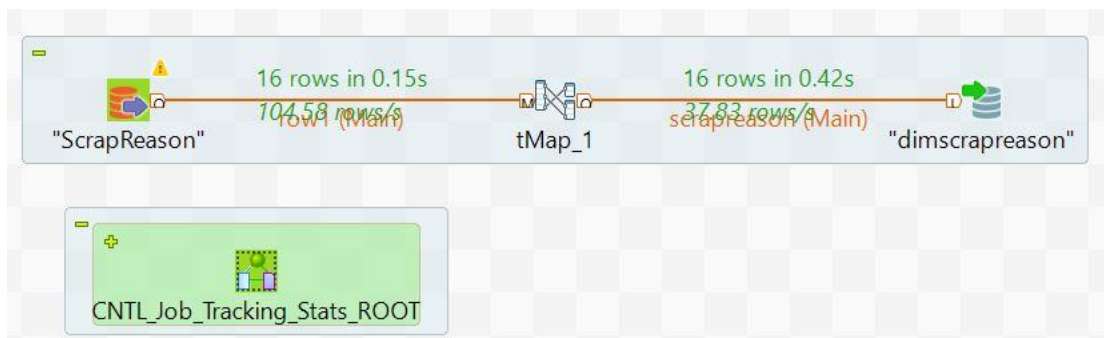
- Source Tables:**
 - WorkOrder (72,591 rows)
 - WorkOrderRouting (67,131 rows)
 - dimProductsAll (504 rows)
 - dimLocation (14 rows)
 - ScheduledStartDate (3,652 rows)
 - ScheduledEndDate (3,652 rows)
 - ActualStartDate (3,652 rows)
 - ActualEndDate (3,652 rows)
- Transformations:**
 - Sort Operations:** Data is sorted by ProductID (Sort 1, Sort 9, Sort 3, Sort 5, Sort 5 1, Sort 5 1 1, Sort 5 1 2) and LocationID (Sort 2, Sort 10, Sort 4, Sort 6, Sort 7, Sort 8).
 - Merge Joins:** Data is merged based on ProductID (Merge Join, Merge Join 6, Merge Join 1, Merge Join 2, Merge Join 3, Merge Join 4, Merge Join 5) and LocationID (Merge Join 1).
 - Data Conversion:** Data is converted from a table to a column (Data Conversion).
 - Derived Columns:** New columns are derived from existing data (Derived Column, Derived Column 1).
 - Conditional Split:** Data is split based on a condition (Conditional Split).
- Final Output:** The final fact table is Fact_WorkOrderRouting_Rejects, which contains 383 rows of rejected data.

FactPurchases and FactPurchases_Rejects:

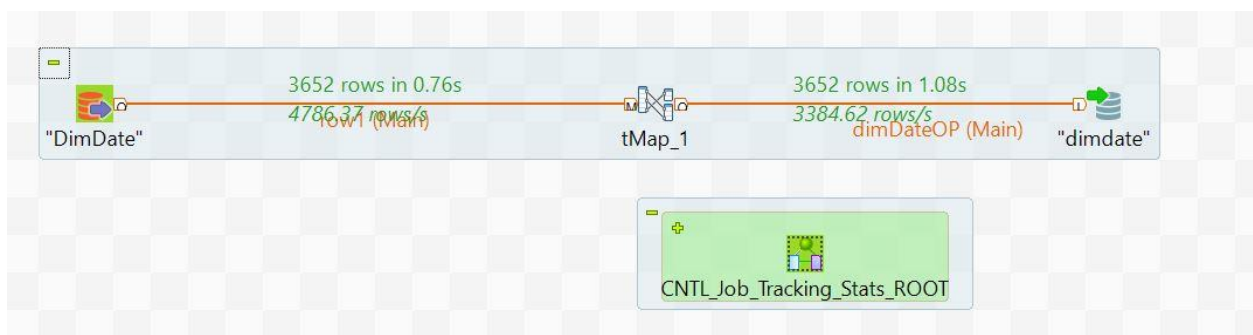


Talend Jobs:

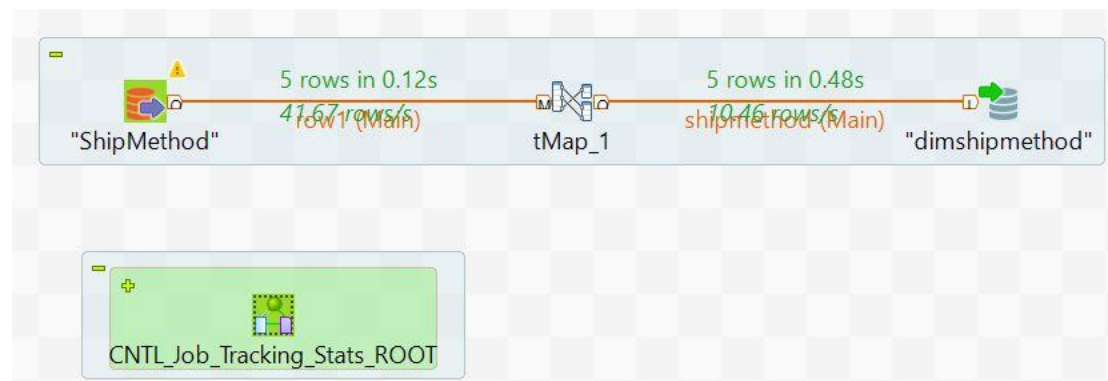
DimScrapReason:



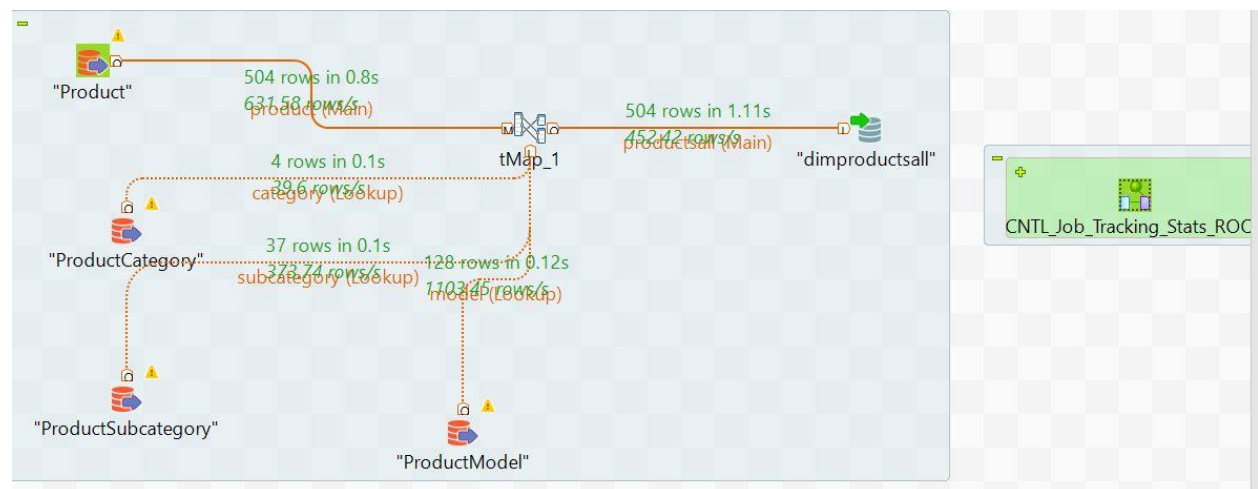
DimDate:



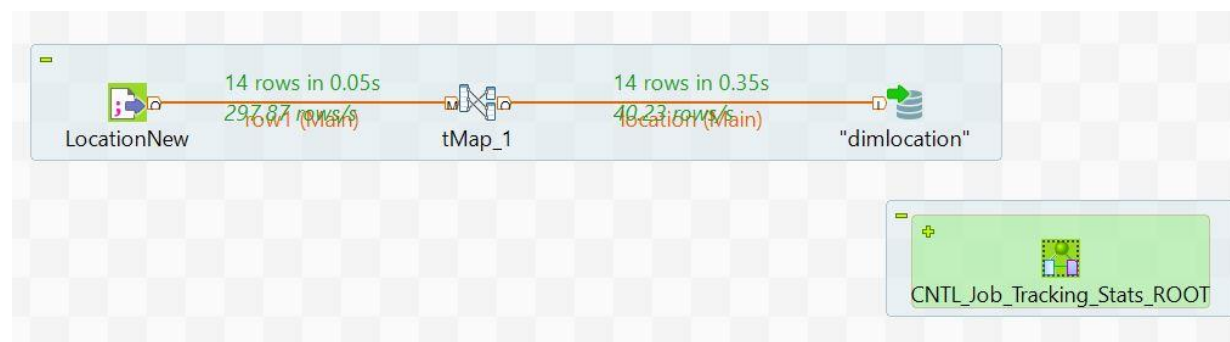
DimShipMethod:



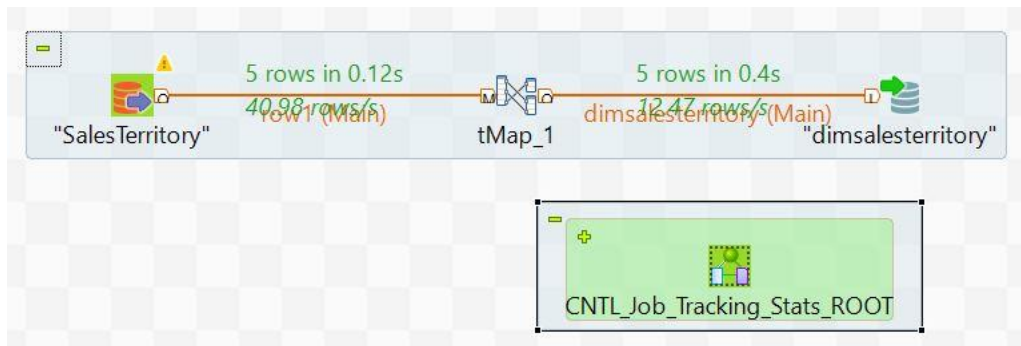
DimProductsAll:



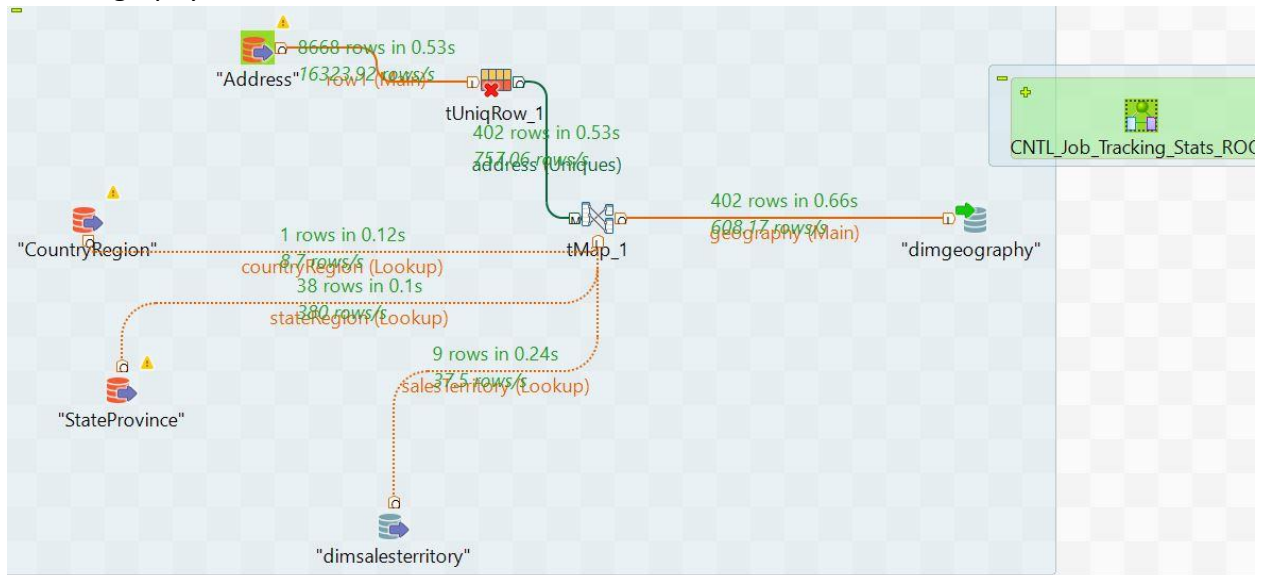
DimLocation:



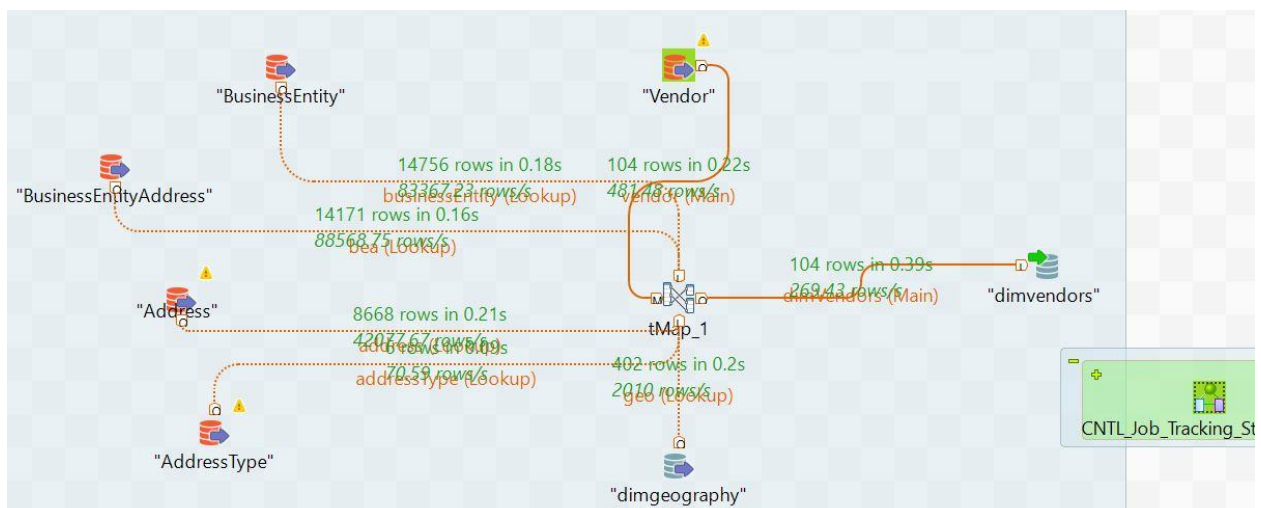
DimSalesTerritory:



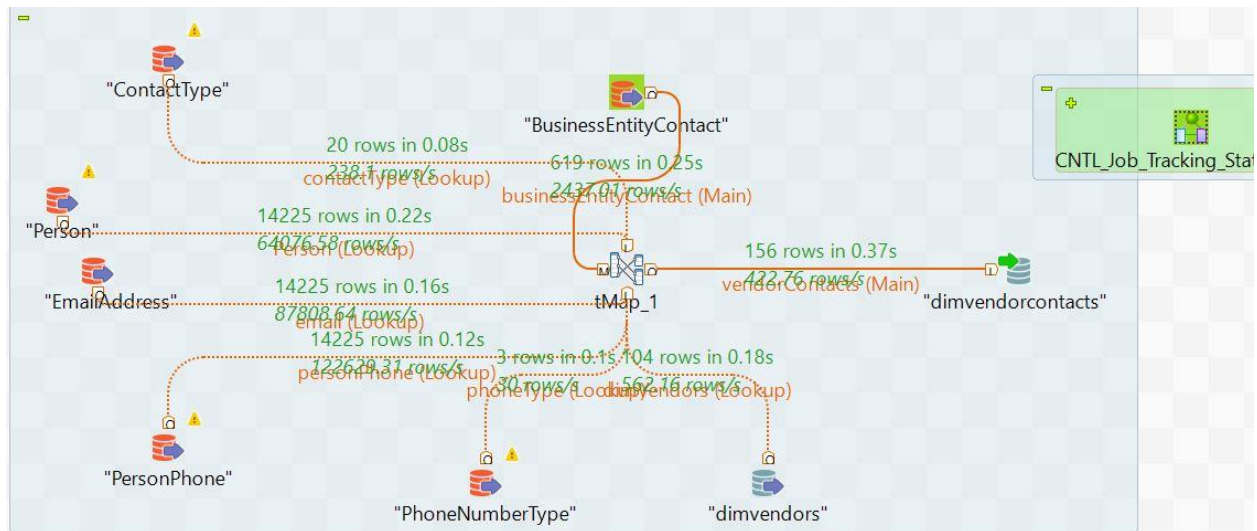
DimGeography:



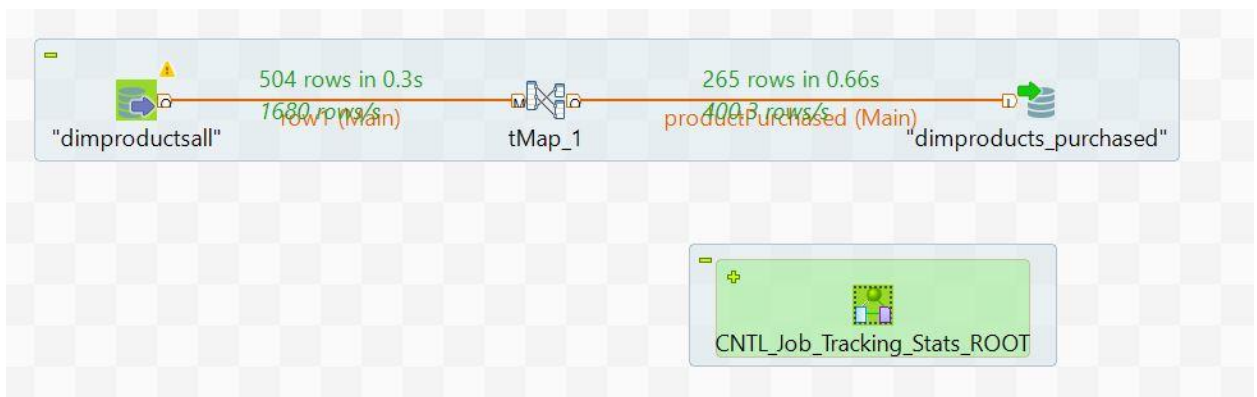
DimVendor:



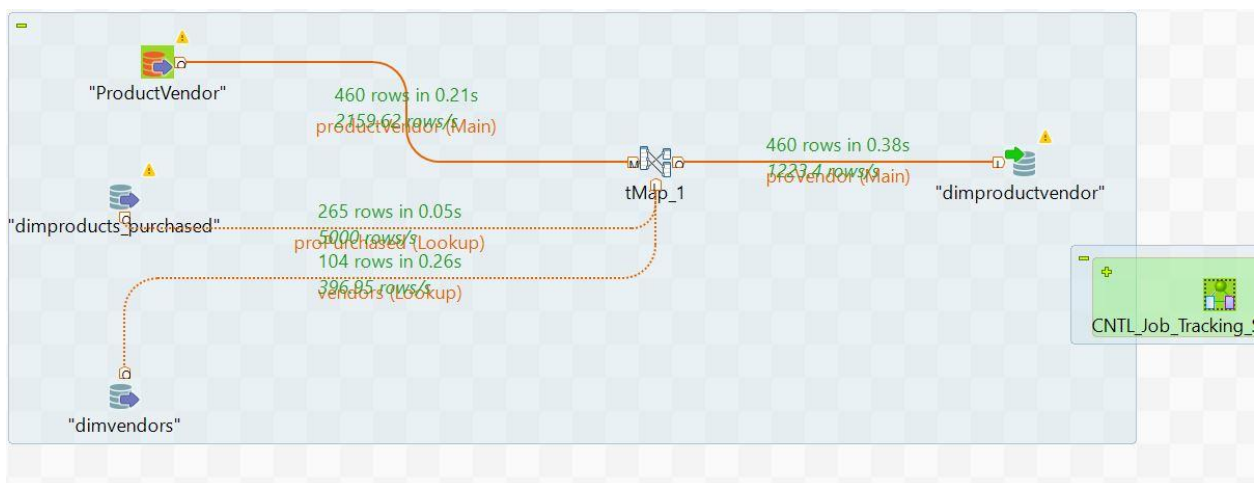
DimVendorContacts:



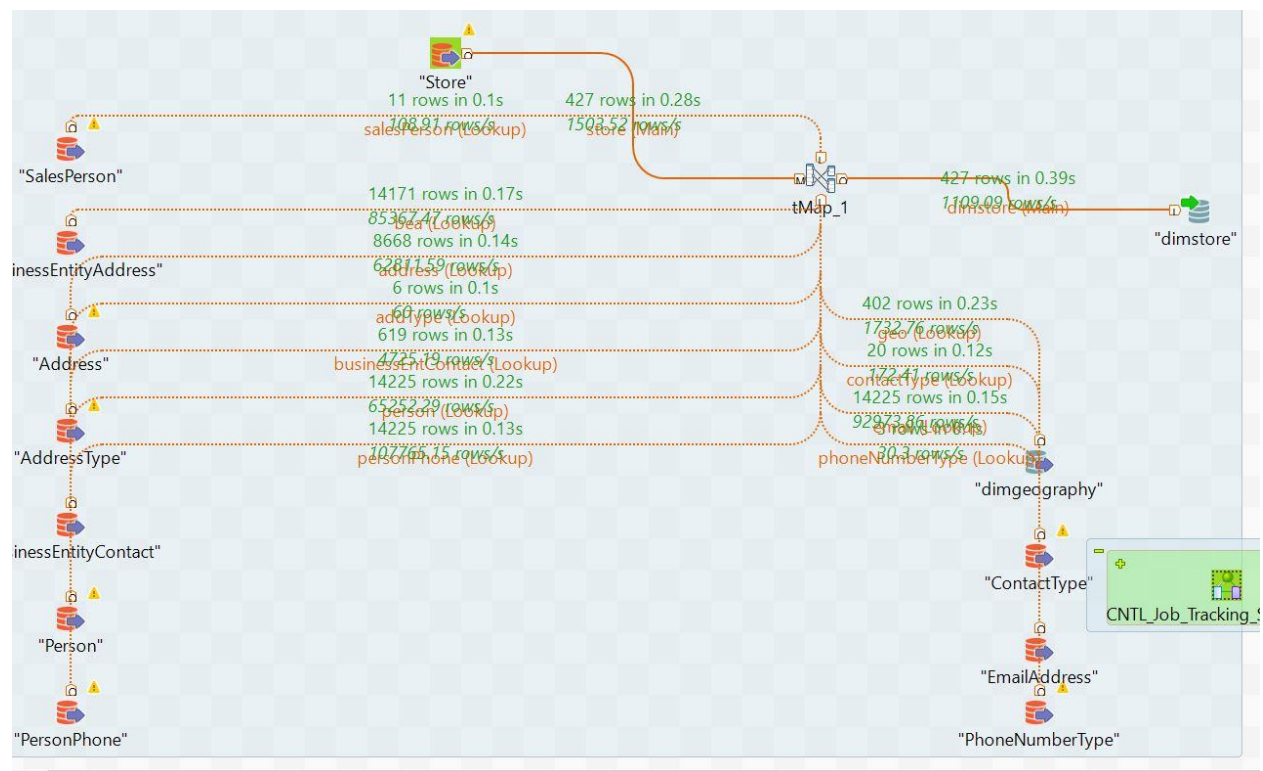
DimProduct_Purchased:



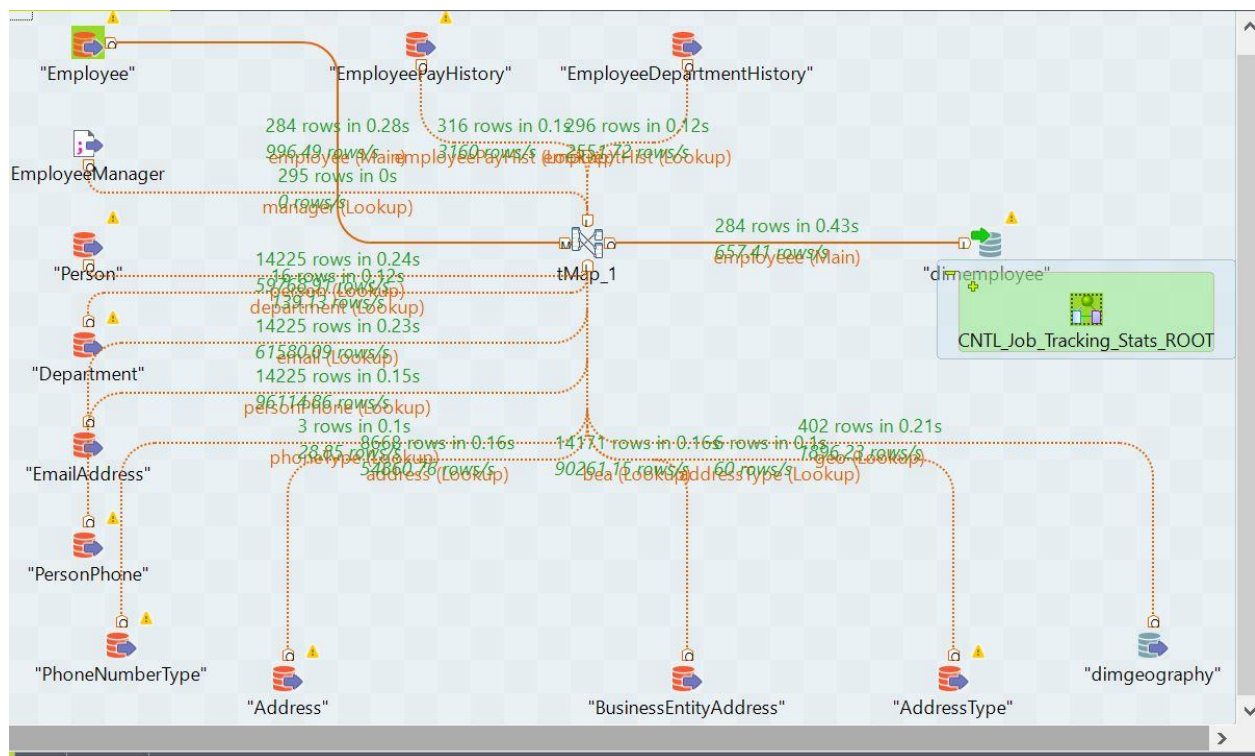
DimProductVendor:



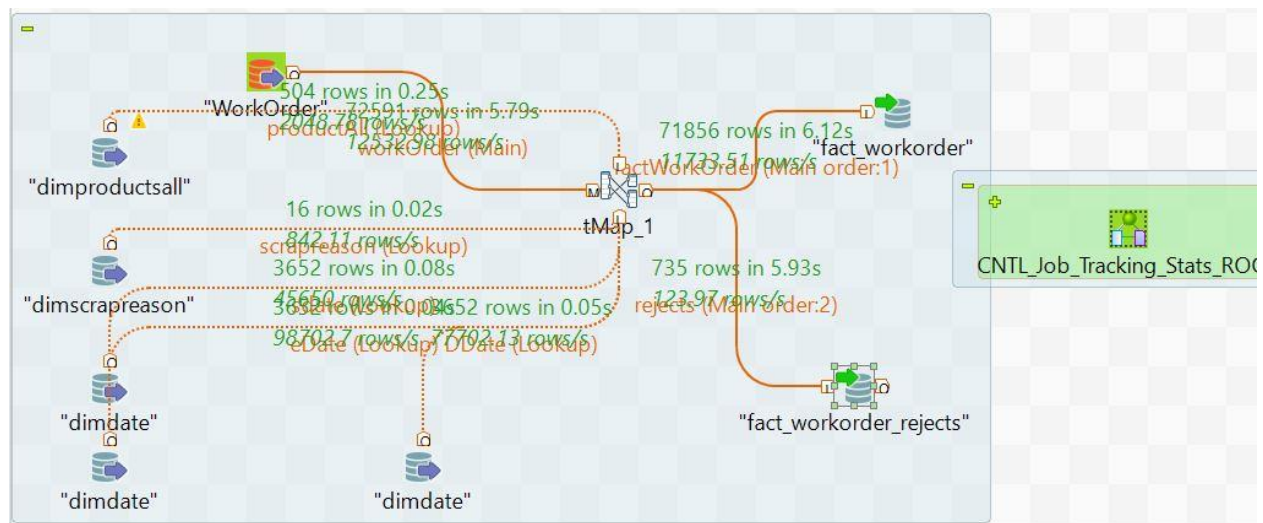
DimStore:



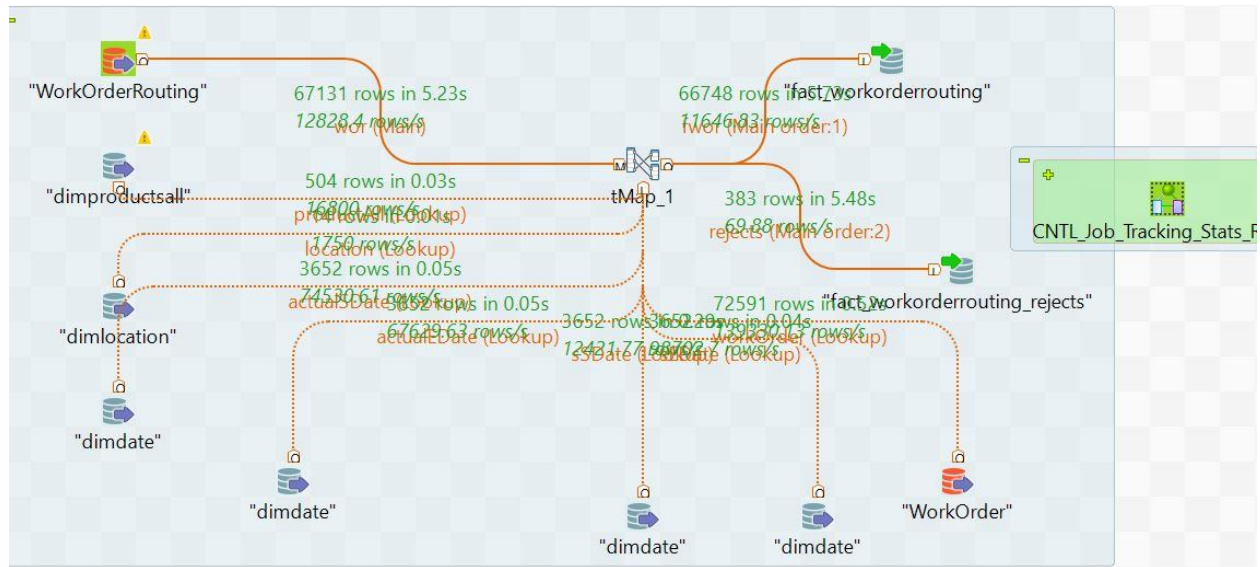
DimEmployee:



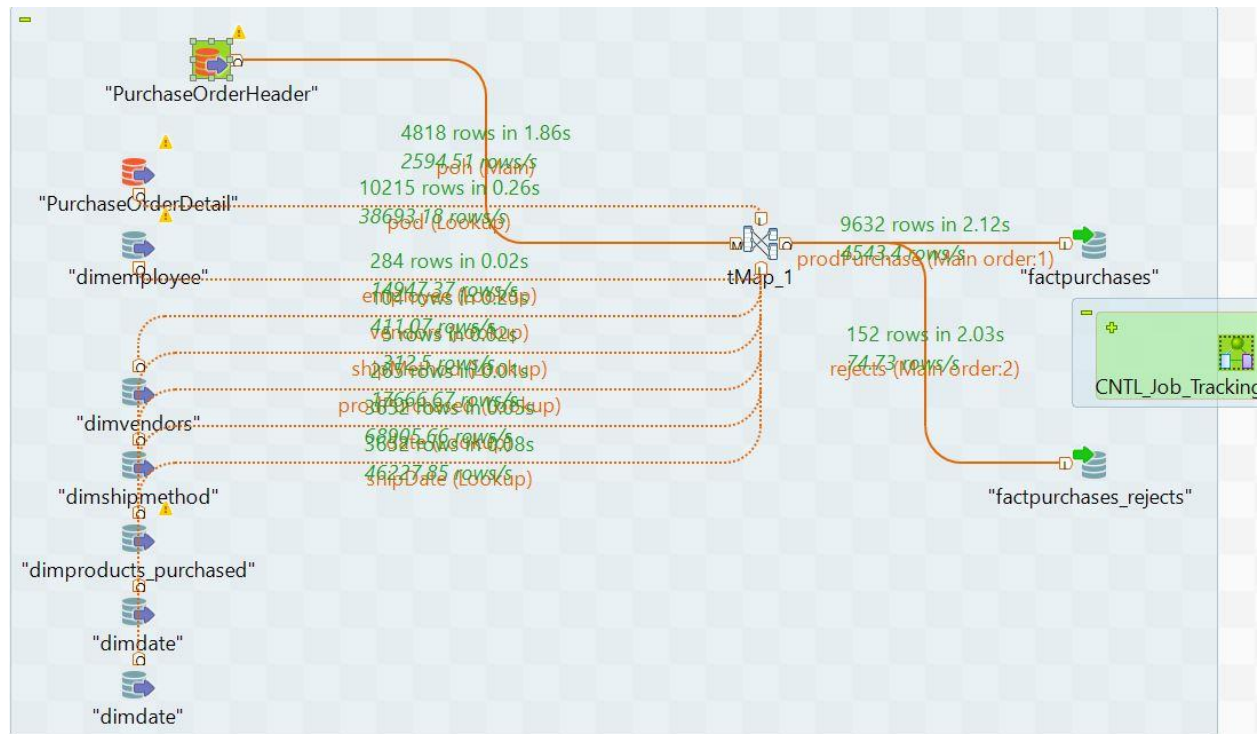
Fact_WorkOrder and Fact_WorkOrder_Rejects:



Fact_WorkOrderRouting and Fact_WorkOrderRouting_Rejects:



FactPurchases and FactPurchases_Rejects:



SQL Queries:

1. Ranked order of Vendors by purchase amount \$ -

```
select rank() over (order by fp.subtotal desc) as Rank_Vendor,  
v.vendorSK, fp.subtotal  
from dbo.DimVendors as v inner join dbo.FactPurchases as fp  
on v.VendorSK = fp.VendorSK  
group by v.vendorSK, fp.subtotal;
```

Rows: 469

2.

--a) Ranked order of products by amount \$ By category

```
select ProductCategoryName,ListPrice,  
rank() over(partition by ProductCategoryName order by listprice desc) as Amount_Rank  
from DimProductsAll  
order by 3,1 desc;
```

Rows: 111

--b) Ranked order of products by amount \$ By Subcategory

```
select ProductSubcategoryName,ListPrice,  
rank() over(order by listprice desc) as Amount_Rank  
from DimProductsAll  
order by 3,1 desc;
```

Rows: 129

--c) Ranked order of products by amount \$ By ProductModel - 20

```
select top 20 ModelName,ListPrice,  
rank() over(order by listprice desc) as Amount_Rank  
from DimProductsAll  
order by 3,1 desc;
```

Rows: 20

--d) Ranked order of products by amount \$ By ProductName - 20

```
select top 20 ProductName,ListPrice,  
  
rank() over(order by listprice desc) as Amount_Rank  
from DimProductsAll  
order by 3,1 desc;
```

Rows: 20

3. Ranked list of employees purchasing products amount \$

```
select distinct concat(e.FirstName,' ',e.LastName) as Name, sum(fp.subtotal) as  
Sum_Amount, rank() over(order by sum(fp.subtotal) asc) as Amount_Rank  
from DimEmployee as e  
join FactPurchases as fp
```

```
on e.EmployeeSK = fp.EmployeeSK
group by FirstName, LastName
order by Sum_Amount;
```

Rows: 12

4. List of employees who purchased products with phone, email & address

```
select distinct concat(e.FirstName, ' ', e.MiddleName, ' ', e.LastName) as
EmployeeName,
e.phone, e.emailaddress,
e.addressline1, ISNULL(e.addressline2, '-')
from dbo.DimEmployee as e
join dbo.FactPurchases as fp
on e.EmployeeSK = fp.EmployeeSK
```

Rows: 12

5. List of purchasing contacts with vendor name, phone, email & address

```
select distinct concat(vc.vendorSK, ' ', vc.firstname, ' ', vc.lastname) as FullName,
v.vendorname, vc.phonenumber,
vc.emailaddress, v.addressline1, ISNULL(v.addressline2, '-') as addressline2
from DimVendorContacts as vc
inner join DimVendors as v
on v.VendorSK = vc.VendorSK
inner join FactPurchases fp
on fp.VendorID=v.BusinessEntityID;
```

Rows: 129