

### **Release Form (Cover Page)**

CSCI Class:	<b>Data Warehouse Design</b>
Professor:	<b>Dr J</b>
Term / Year:	<b>Spring 2019</b>
Team Name:	<b>Yellow Team</b>
Student Name (s)	<b>1.Zachary Reese 2. Jonathan Guzman 3.Christopher Iverson 4.Keenan Lewis-Jolly 5. Shawein Smith</b>
Assignment:	<b>Project Assignment</b>
Date of Submission:	<b>4/27/2019</b>

Satisfaction with learning experience with this assignment (check one):

A. Very useful

B. Useful

C. Little or no use

**Post Mortem** –this section is required (type your important comments/lessons learned here):

*Three or more sentences/paragraphs!*

***Release Note:***

We the undersigned (name in italic will serve as signature), take full responsibility for the work performed related to this assignment and for the work records attached, and also identify both as our own original work contribution:

Signature 1: \_\_\_\_\_

Signature 2: \_\_\_\_\_

Signature 3: \_\_\_\_\_

Signature 4: \_\_\_\_\_

Signature 5: \_\_\_\_\_

Signature 6: \_\_\_\_\_

## Table of Contents

<b>Release Form (Cover Page)</b>	<b>0</b>
<b>Executive Summary</b>	<b>4</b>
<b>Project Description</b>	<b>4</b>
<b>Project Organization</b>	<b>5</b>
<b>Project Plan</b>	<b>6</b>
<b>Tasks</b>	<b>6</b>
<b>Project Architecture</b>	<b>8</b>
<b>OLTP Operational Data Model</b>	<b>9</b>
<b>Complete Data Warehouse Design</b>	<b>12</b>
<b><u>Sales</u></b>	
<b>Sales WIP</b>	<b>14</b>
<b>Sales BEAMS</b>	<b>14</b>
<b>Sales DFM</b>	<b>15</b>
<b>Sales Data Vault</b>	<b>16</b>
<b>Sales Data Mart</b>	<b>17</b>
<b>Sales ETL Process</b>	<b>18</b>
<b>Queries - Zac &amp; Jonathan</b>	<b>25</b>
<b><u>Production</u></b>	
<b>Production WIP</b>	<b>30</b>
<b>Production BEAMS</b>	<b>31</b>
<b>Production DFM</b>	<b>32</b>
<b>Production Data Vault</b>	<b>33</b>
<b>Production Data Mart</b>	<b>34</b>

	3
<b>Production ETL Model</b>	<b>35</b>
<b>Queries - Keenan</b>	<b>38</b>
<b><u>Human Resources</u></b>	
<b>Human Resources WIP</b>	<b>43</b>
<b>Human Resources BEAMS</b>	<b>43</b>
<b>Human Resources DFM</b>	<b>44</b>
<b>Human Resources Data Mart</b>	<b>45</b>
<b>Human Resources ETL Process</b>	<b>46</b>
<b>Queries - Chris</b>	<b>50</b>
<b><u>Purchasing</u></b>	
<b>Purchasing WIP</b>	<b>52</b>
<b>Purchasing BEAMS</b>	<b>52</b>
<b>Purchasing DFM</b>	<b>53</b>
<b>Purchasing Data Vault</b>	<b>54</b>
<b>Purchasing Data Mart</b>	<b>55</b>
<b>Purchasing ETL Process</b>	<b>56</b>
<b>Closing Remarks</b>	<b>58</b>
<b>DW Generate Script</b>	<b>58</b>

## Executive Summary

Adventure Works Cycles is a bicycle manufacturing company that operates around the globe with many locations, over 300 employees, and millions of dollars in revenue. The company's resellers are located on three different continents: North America, Europe, and Asia. Adventure Works sells a multitude of products including: Bikes, Accessories, Clothing, Components, Services, and more.

## Project Description

The scope of this project is to create a fully functioning Data Warehouse from the AdventureWorks2014 database. The project is built to Enterprise Data Warehouse standards and consists of the following data marts:

1. Sales: Focuses only on orders sold by a salesperson and tries to answer the following questions, "Which salesperson sold which orders?", "Which territory was the order sold in?", and "Where is a product located at an inventory location?".
2. Production: Focuses on the back end of getting products out the door and answers the following questions, "How much for which product listed?", "What are the current quantities of an inventoried product?", and "Where is a product located at an inventory location?".
3. Human Resources: Focuses on maintaining and updating critical financial and employee records and answers the following, "How much does x employee earn?", "Are all employees paid on time?", and "Which person is best suited for this job?".
4. Purchasing: Focuses on vendors and resellers for AdventureWorks. Answers the questions, "Which vendor makes the most profit?", "Which vendor purchases the most inventory?", and "What is the best way to get our inventory to said vendor?".

## Project Organization

### *Division of Labor*

1. Zachary Reese
  - a. Serves as team leader
  - b. Guide and advise throughout project
  - c. Data Warehouse Modeler
  - d. Data Warehouse implementation
  - e. SSIS instructor
  - f. Sales Data Mart Modeling
  - g. Sales ETL modeling
2. Jonathan Guzman
  - a. DW Architecture development
  - b. Sales data mart implementation
  - c. Sales data mart modeling
  - d. Sales fact table modeling
  - e. Sales fact table implementation
  - f. DW modeling
3. Keenan Lewis-Jolly
  - a. Production fact table implementation
  - b. Production data vault modeling
  - c. Production data mart modeling
  - d. Production ETL modeling
  - e. DW Infrastructure
4. Christopher Iverson
  - a. Human resource fact table modeling
  - b. Human resource data vault modeling
  - c. Human resource data mart modeling
  - d. DW initial implementation
  - e. Human Resources ETL modeling
5. Shawein Smith
  - a. Purchase fact table modeling
  - b. Purchase data vault modeling
  - c. Purchase data mart modeling

- d. Software management
- e. Purchase ETL modeling
- f. Project QA

## Project Plan

Steps	Start date	Completion date
1. Project planning and requirement analysis	1/15/2019	1/30/2019
2. Architecture planning	1/30/2019	2/2/2019
3. Data source analysis aand reconciliation	2/3/2019	2/9/2019
4. User requirements for DW	2/10/2019	2/15/2019
5. Data vault design	2/20/2019	3/2/2019
6. User requirements for data marts	3/4/2019	3/5/2019
7. DFM design for data marts	3/8/2019	3/18/2019
8. ETL	3/15/2019	4/12/2019
9. Implementation of data marts	4/3/2019	4/15/2019
10. Testing and analysis	4/15/2019	4/25/2019
11. Project report	4/19/2019	4/27/2019

## Tasks

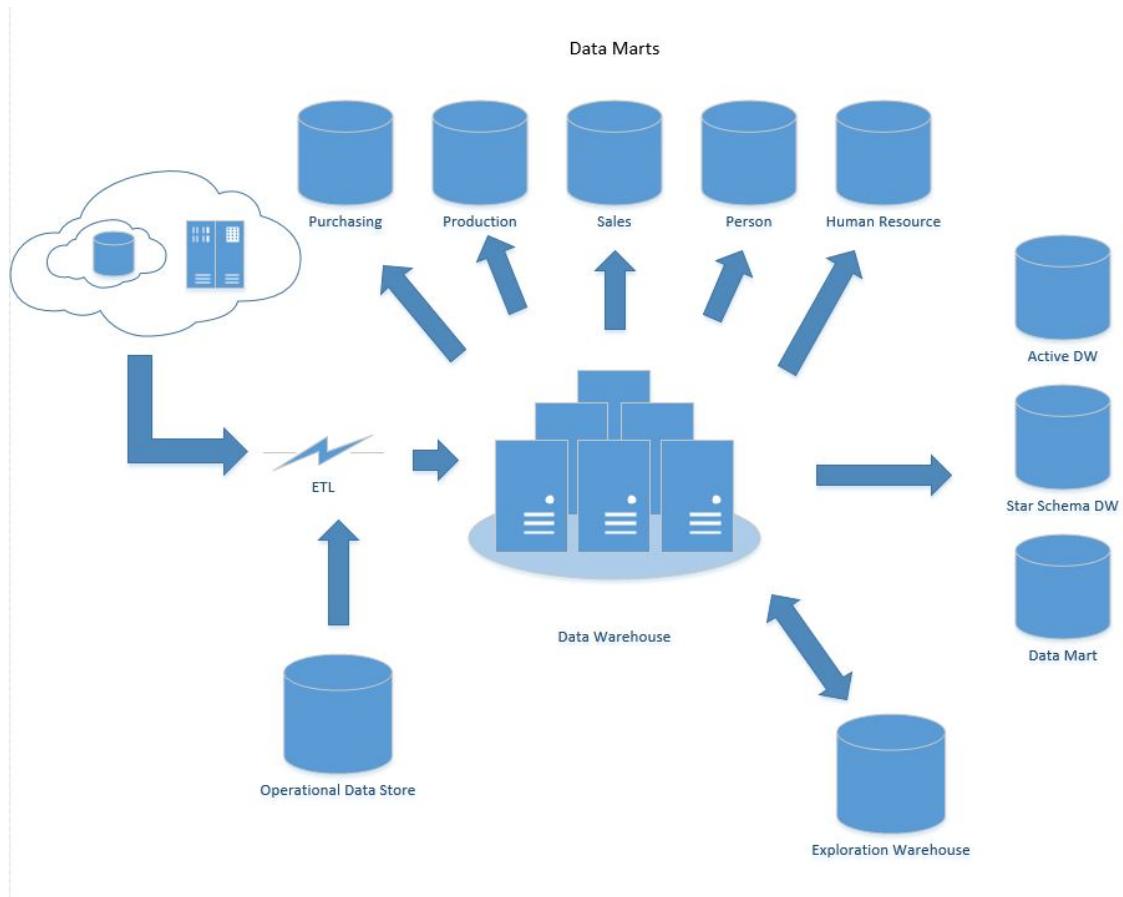
The following tasks have been identified

1. Requirement Analysis
  - a. Form scope of project
  - b. Form requirements for the data warehouse and data marts
  - c. Form guidelines and standards for the warehouse
  - d. Develop division of labor
  - e. Select source system, data, and data profiling
  - f. Test source system data for quality
2. Architecture planning
  - a. Diagram and map overall architecture
  - b. Develop and perceive beginning sizing constraints
  - c. Initial ETL development
3. Extract and Load Data from Source System
  - a. Extract source data
  - b. Develop SQL Server configurations for team
  - c. Load extracted source data into SQL Server
4. Design and Build Data Warehouse Structures
  - a. Development of EDW database built on data vaults
  - b. Development of scripts to generate summary tables

- c. Construct data marts using DFM
- 5. Implementation
  - a. Final implementation of DW
  - b. ETL models
  - c. Develop staging database
- 6. Testing/QA
  - a. Perform data quality tests
  - b. Perform integration tests
  - c. Perform system and environmental tests
  - d. Perform volume and load tests

## Project Architecture

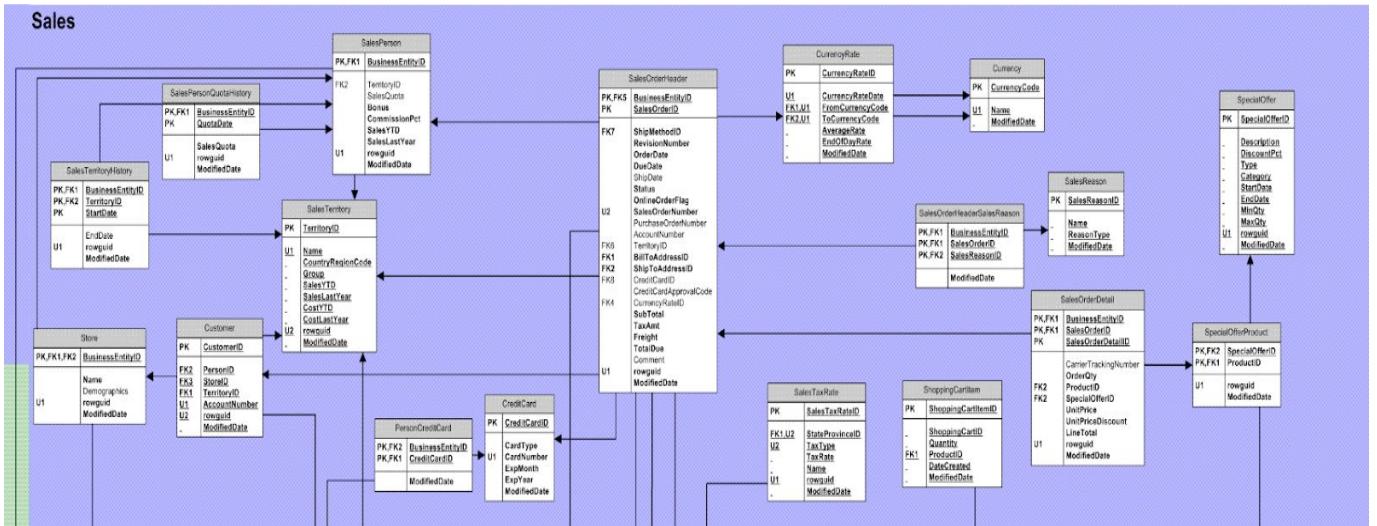
The proposed architecture diagram is below. The source of the data is AdventureWorks2014 database. Data will be extracted from this database, and transformed using ETL tools such as SSIS with Visual Studio. Then the data will be loaded into the respected data marts for analytics.



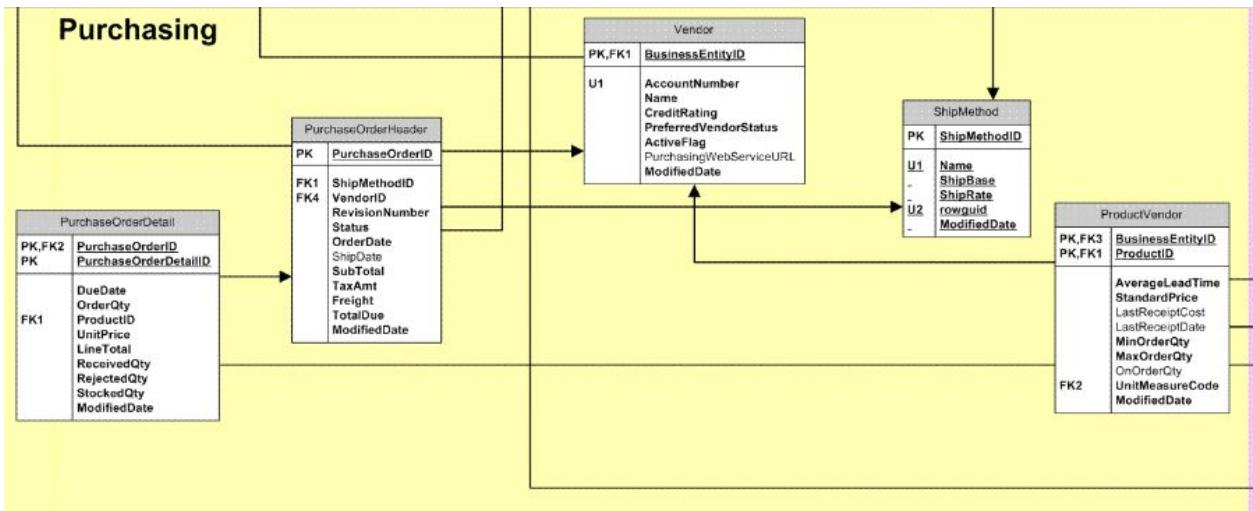
## OLTP Operational Data Model

Below is the Entity Relationship model of our current operational database. This will be used to populate our data marts.

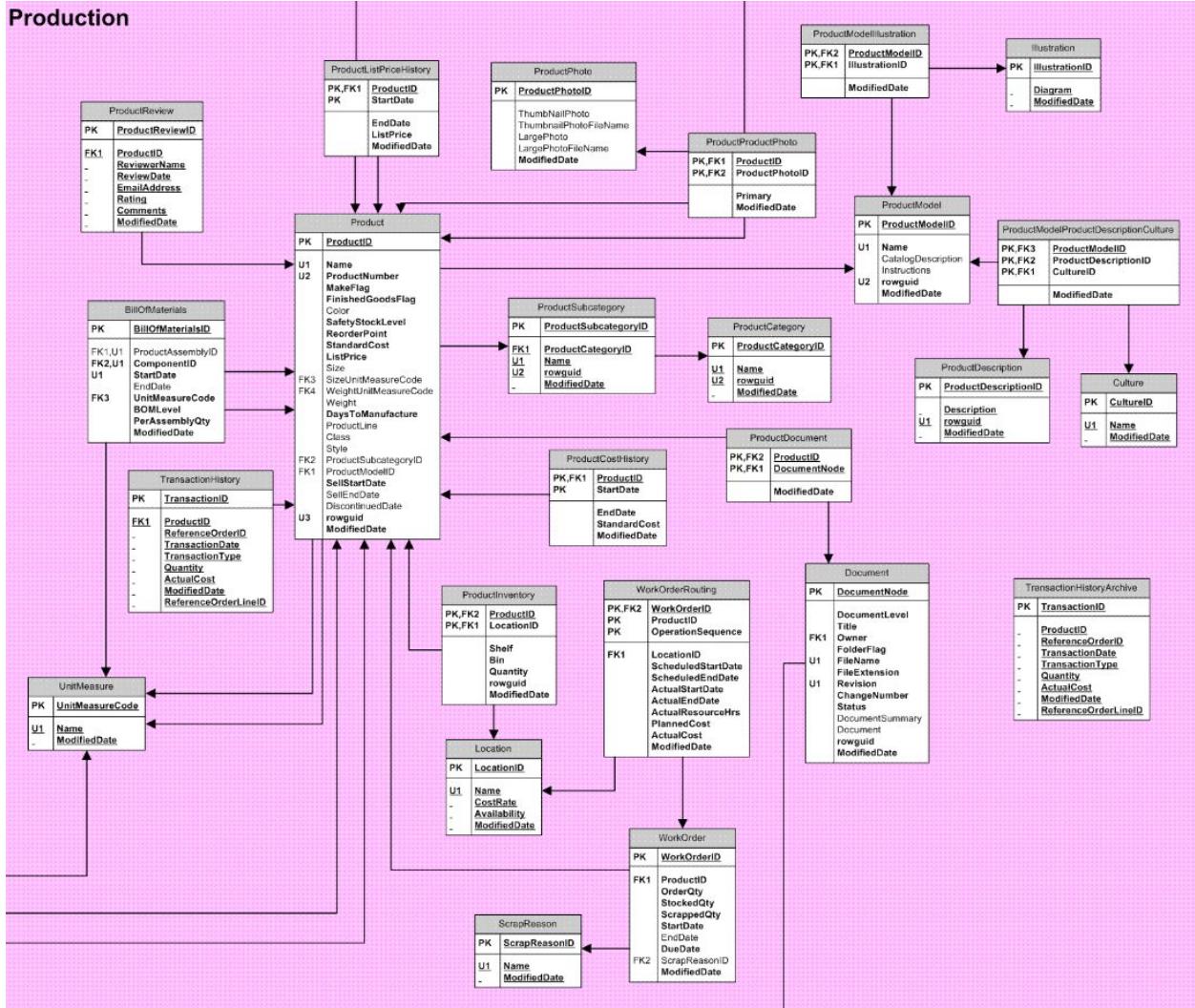
### Sales



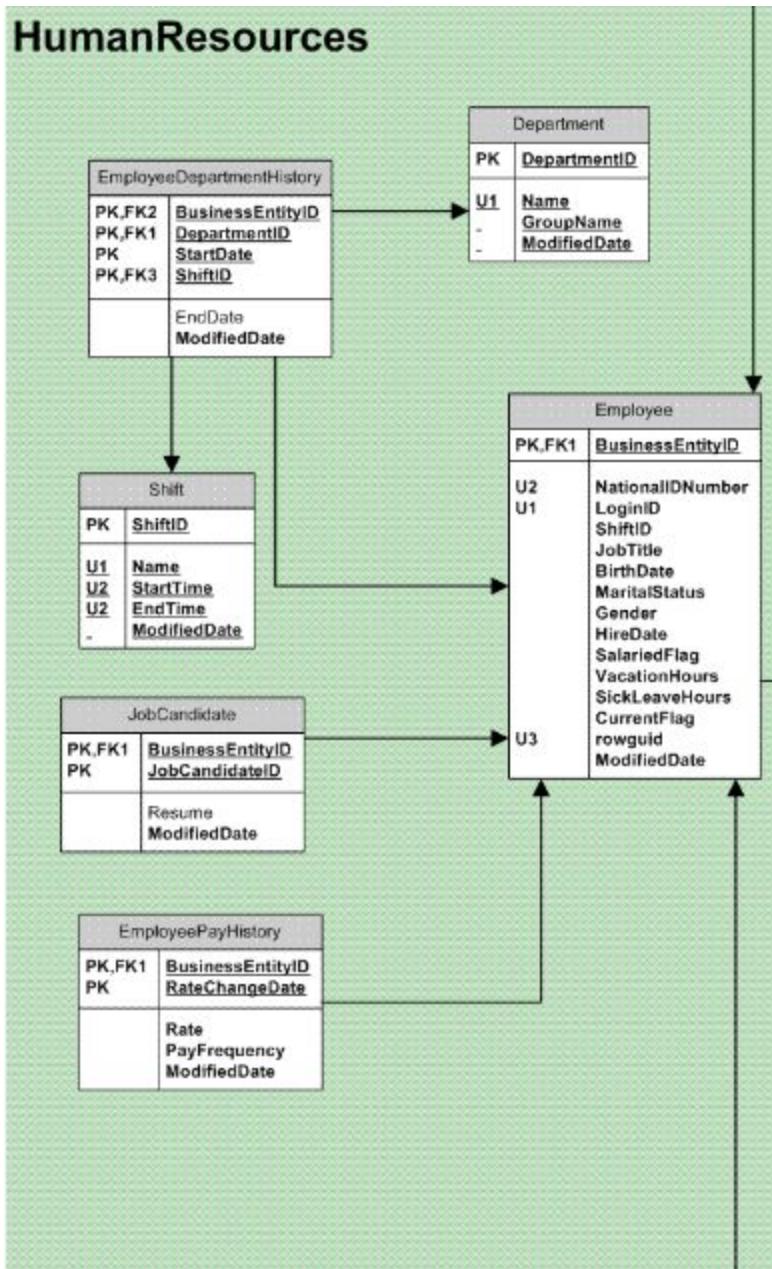
### Purchasing



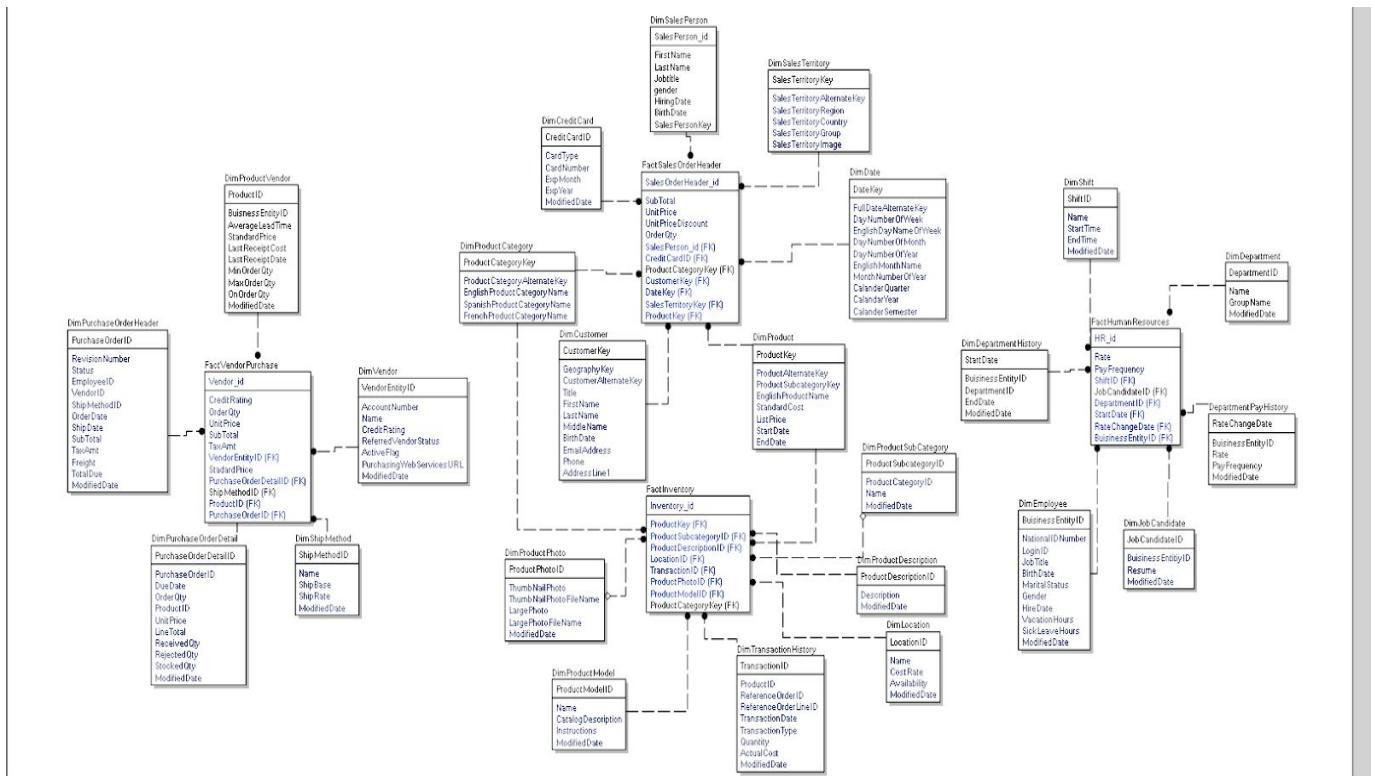
## Production



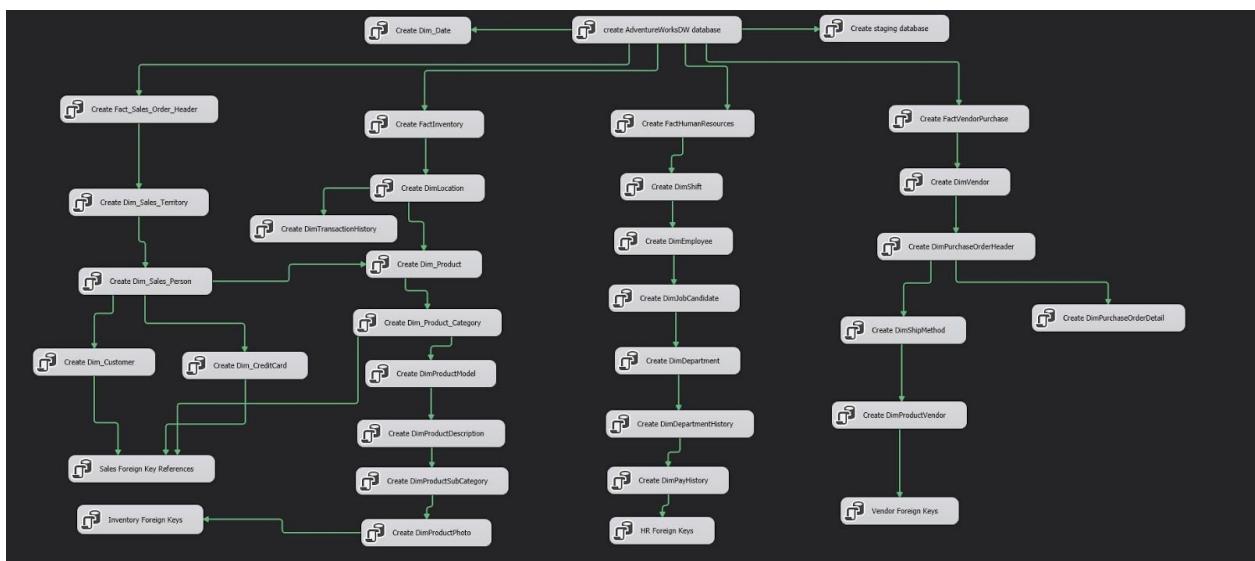
## Human Resources

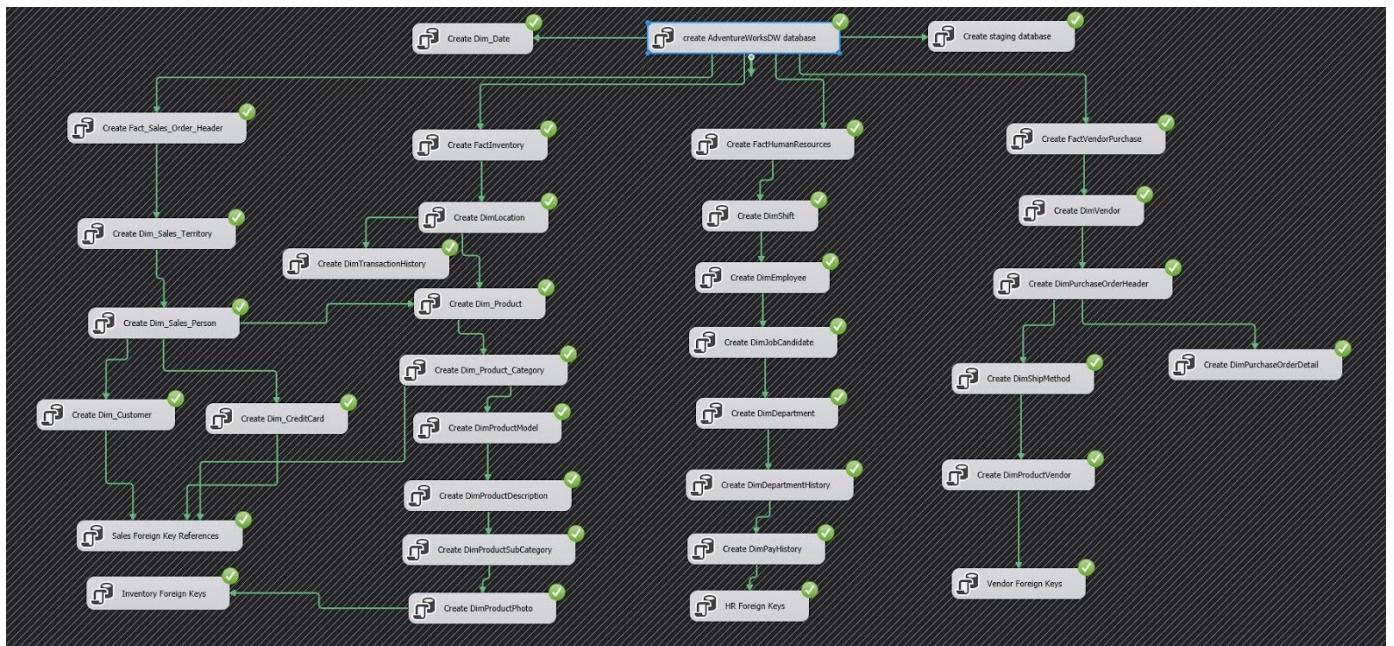


## Complete Data Warehouse Design



SSIS





## Sales - Zac & Jonathan

### WIP Model

Who- Vendor/SalesPerson	Who-Customer	What- Product	When- Time/Date	Where- Location/Region	Which- Category	How - PaymentMethod
D1- Salesperson	D2 - Customer	D3-Product	D4 - Time	D5 - Location	D6 - Product Category	D7 - CreditCard
FirstName	FirstName	ProductID	Date	TerritoryID	ProductCategoryID	CreditCardID
LastName	Last Name	ProductName	Month	CountryRegionCode	Name	CardType
JobTitle	Title	ProductNumber	Year	TerritoryName	SubCategoryName	CardNumber
Gender	EmailPromotion	ListPrice	Day	locationkey	CategoryName	ExpMonth
HiringDate	PersonType	productkey	datekey		productcategorykey	ExpYear
BirthDate	Demographics					ModifiedDate
<b>salespersonkey</b>	<b>CustomerID</b>					BillingAddress
						<b>CreditCardApprovalCode</b>
<b>Observations/Measures</b> TotalDue-\$, SubTotal-\$, UnitPrice-\$, UnitPriceDiscount %, OrderQty #						
Expect to keep history for 8 years						
Preliminary Workload						
Question/Query	What are the most popular items on a monthly/quarterly/yearly basis					
Question/Query	Which territory has the highest amount of orders and which territory brings in the most amount of profit					
Question/Query	What are the most popular items per region					
Question/Query	What are the most frequent and most spending customers per region					
Question/Query	What is the highest selling product in the companies catalog					
Question/Query	What is the most popular credit card type					

## BEAMS

### *SalesOrderHeader Dimension*

SalesOrderHeader								
Customer Name	CustomerID	Purchase Order Number	Order Date	Territory Name	Country Region Code	Sales Person	Sub-Total	Sales Order ID
Who	Who	What	When	Where	Where	Who		What
Zac Reese	29825	PO522145787	2011-05-31 00:00:00.000	Southeast	US	Jim Ross	20565.6206	4365
Jonathan Guzman	29672	PO18850127500	2011-05-31 0:00:00	Southeast	US	Billy Bob	1294.2529	4366
Dr Jovanovic	29734	PO18473189620	2011-05-31 0:00:00	Canada	CA	Kieth Don	32726.4786	4366
Isaac Newton	29994	PO16444174044	2011-05-31 0:00:00	Canada	CA	Timmy Turner	28832.5209	4366
Betsy Ross	29565	PO18009186470	2011-05-31 0:00:00	Southwest	US	Jimmv Neutron	419.4589	4366
Bob Smith	29898	PO16617121983	2011-05-31 0:00:00	Northwest	US	Patrick Star	24432.6088	4366

### *SalesTerritory Dimension*

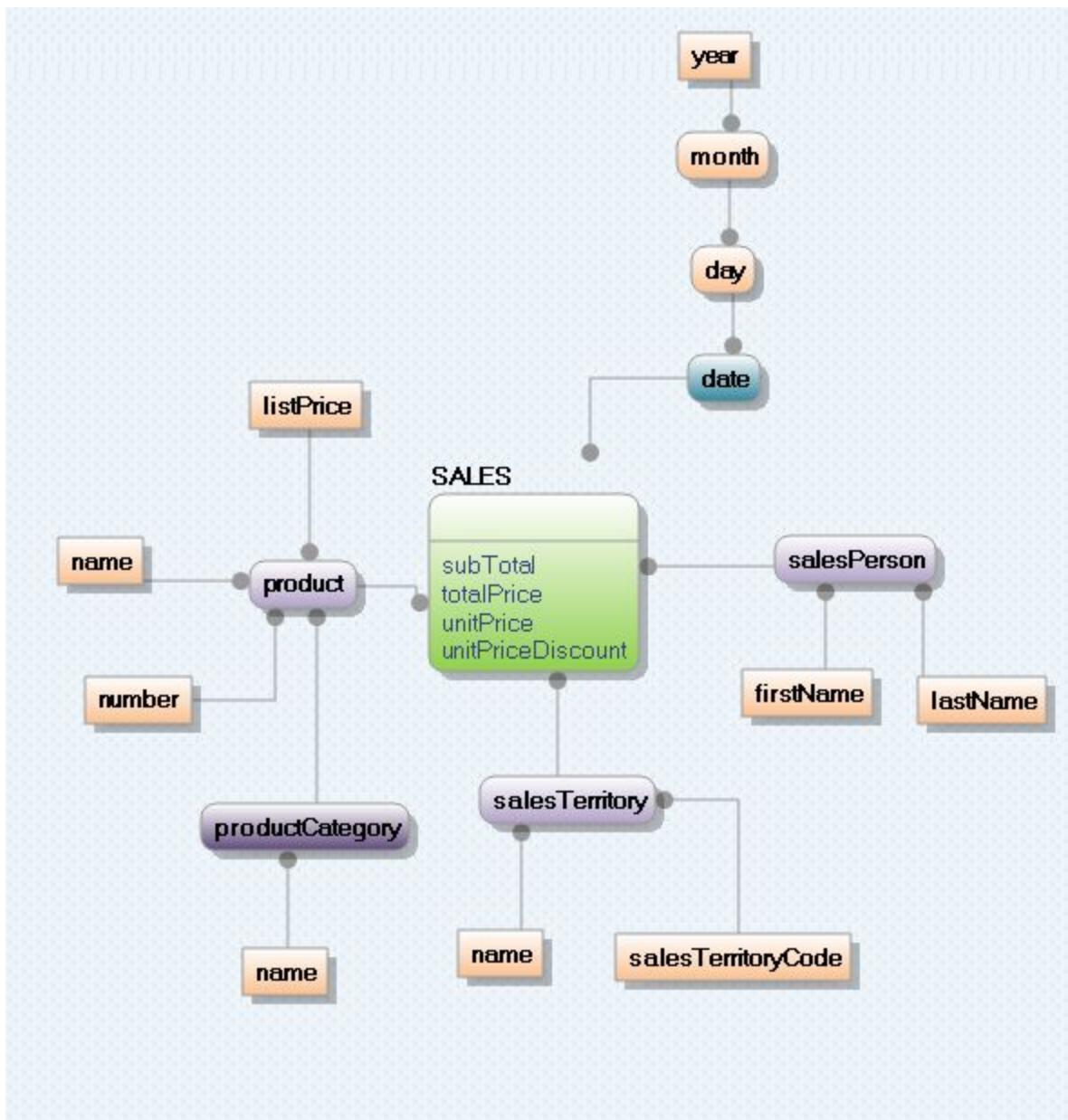
SalesTerritory								
Business Entity ID	First Name	Last Name	Job Title	Store ID	TerritoryID	Territory Name	Country Region Name	
Who	Who	Who	What	Where	Where	Where	Where	Where
274	Stephen	Jiang	North American Sales Manager	7482342	NULL	NULL	United States	
275	Michael	Blythe	Sales Representative	3242114	2	Northeast	United States	
276	Linda	Mitchell	Sales Representative	4249495	4	Southwest	United States	
277	Jillian	Carson	Sales Representative	6934505	3	Central	United States	

## Data Warehouse Design

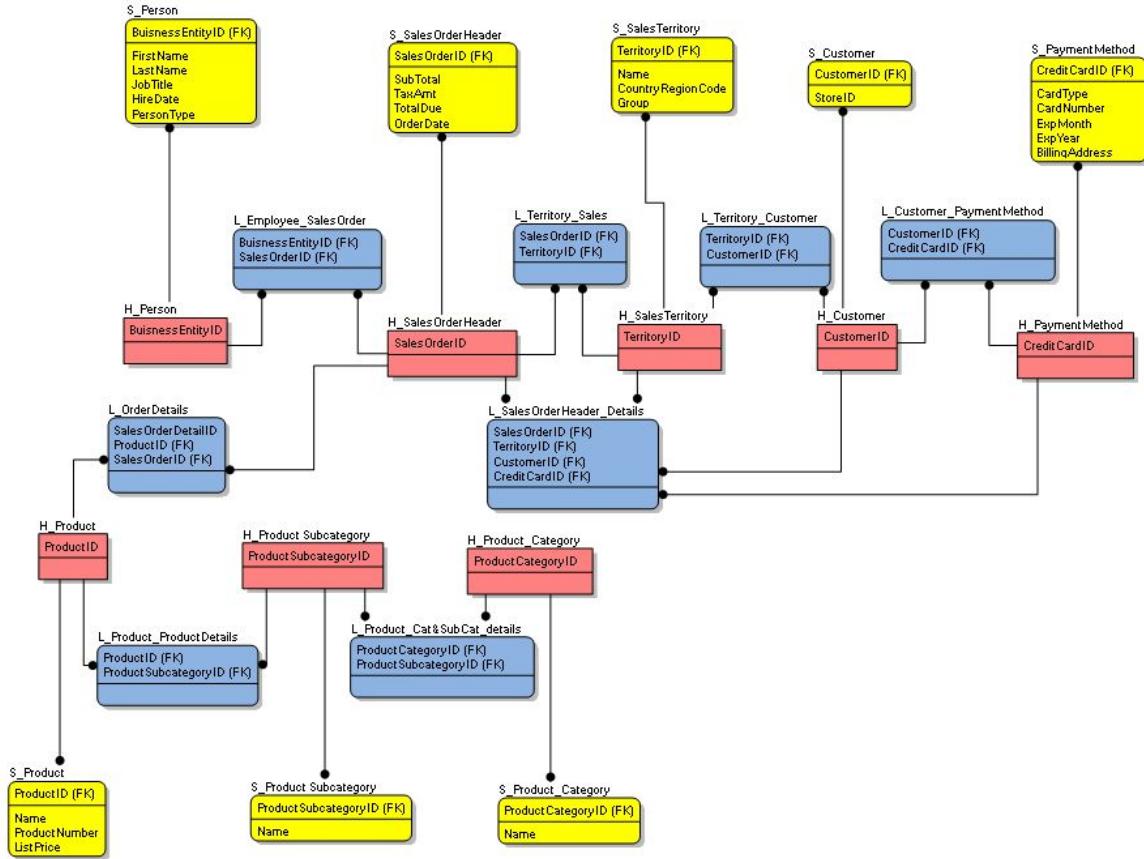
Conceptual Models

Sales - Zac & Jonathan

*Dimensional Fact Model*



## Data Vault - Sales



## Logical Model

### Sales Data Mart

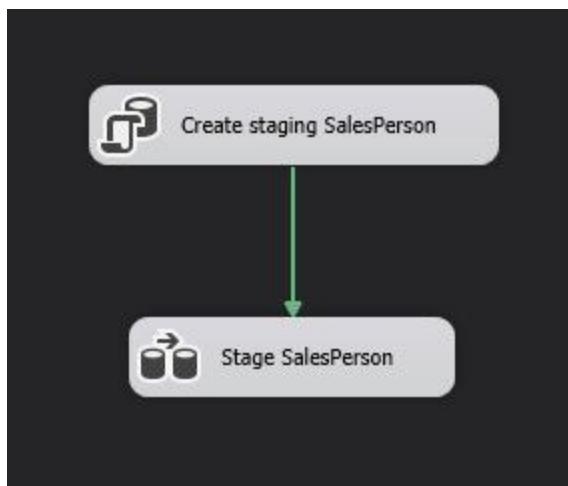


## ETL Process - Sales

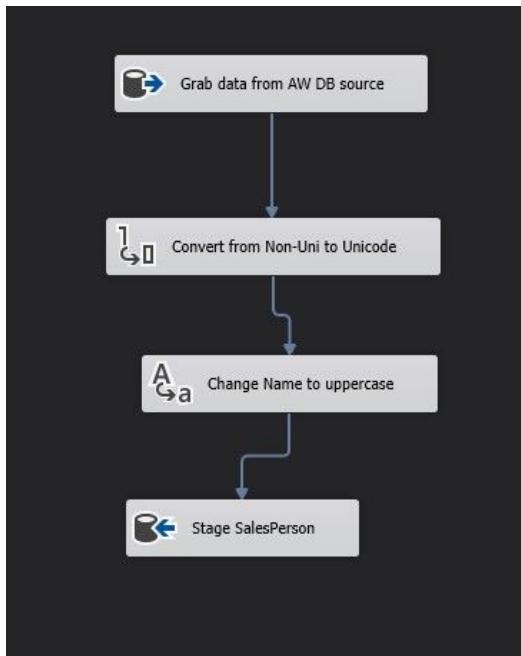
Zac: As group lead, I felt that SSIS was one of the most important areas of focus in the entire DW spectrum. As a group, we have created models to showcase the strength of SSIS in Visual Studio to aid in the ETL process.

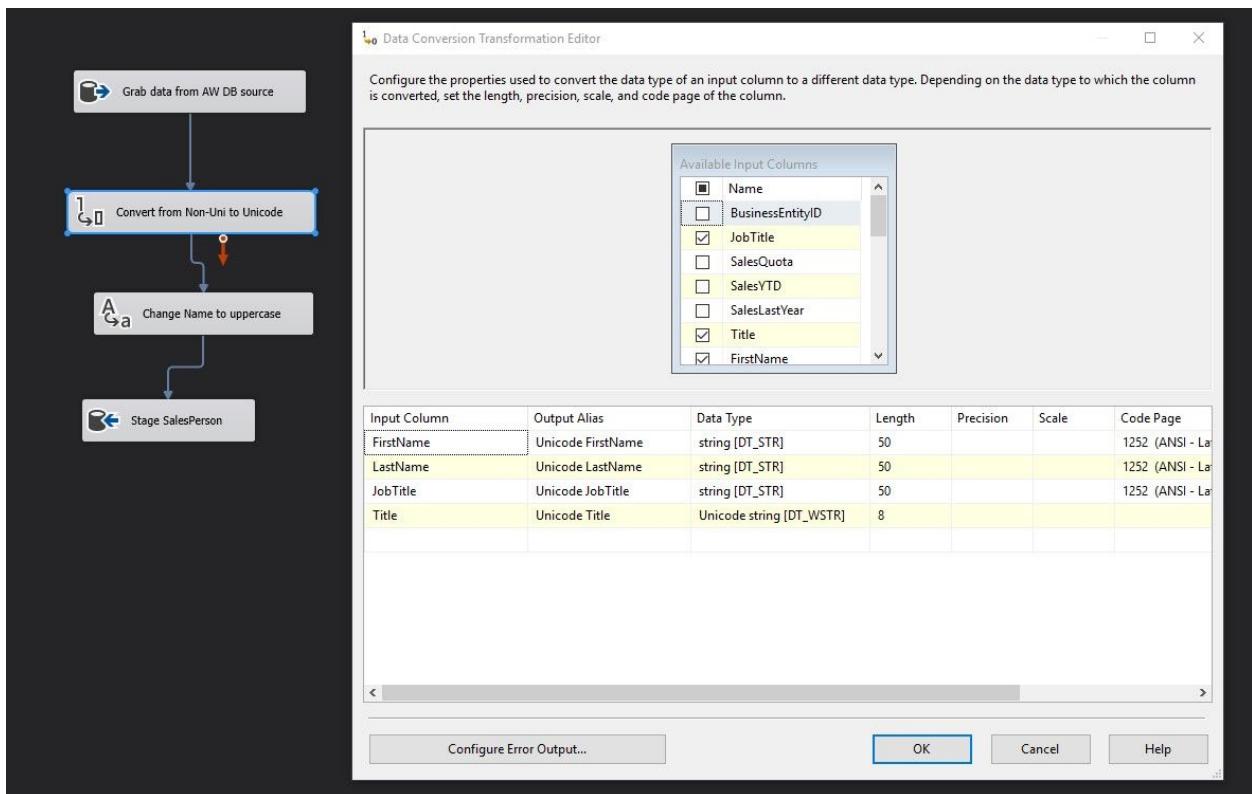
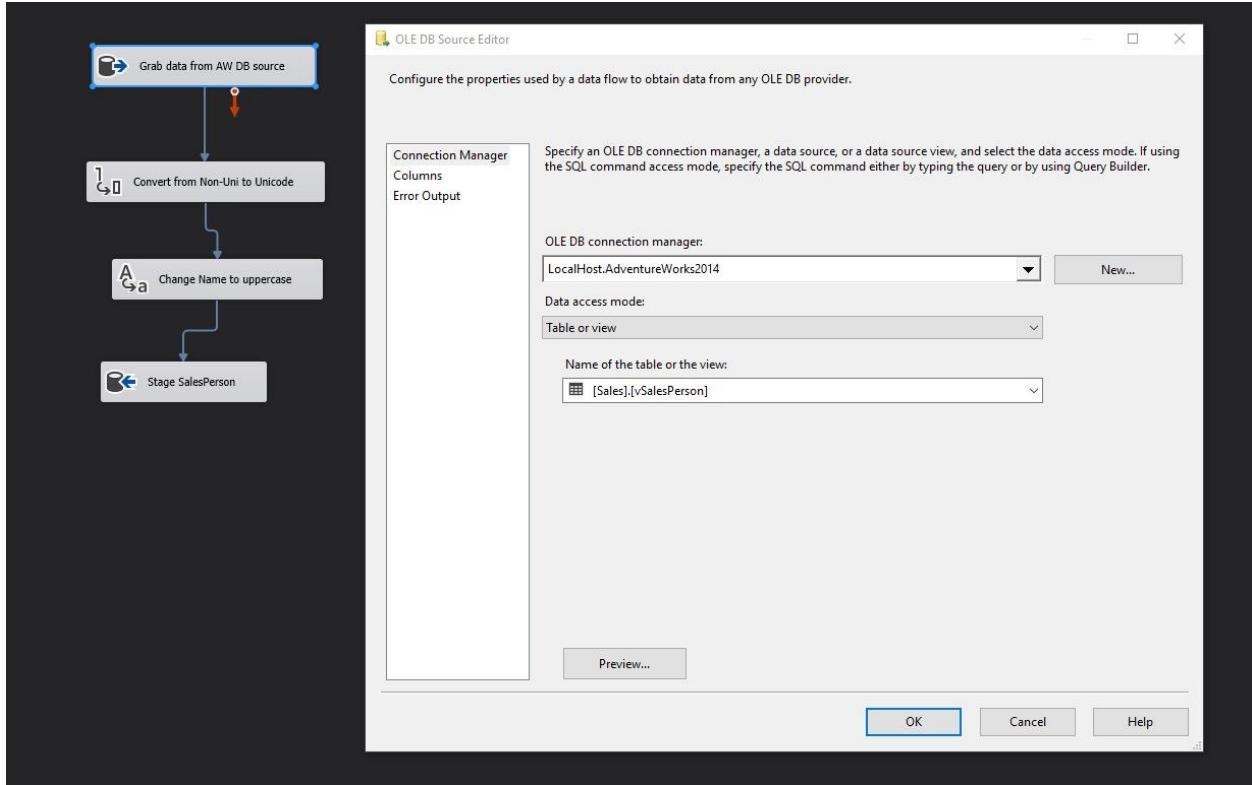
1. Show the process of using SSIS to extract data from a foreign source, in this case the data comes from AdventureWorks database, then we transform the data as it is going from non-unicode to unicode, and finally load the data into a separate source, in this case our staging DW.

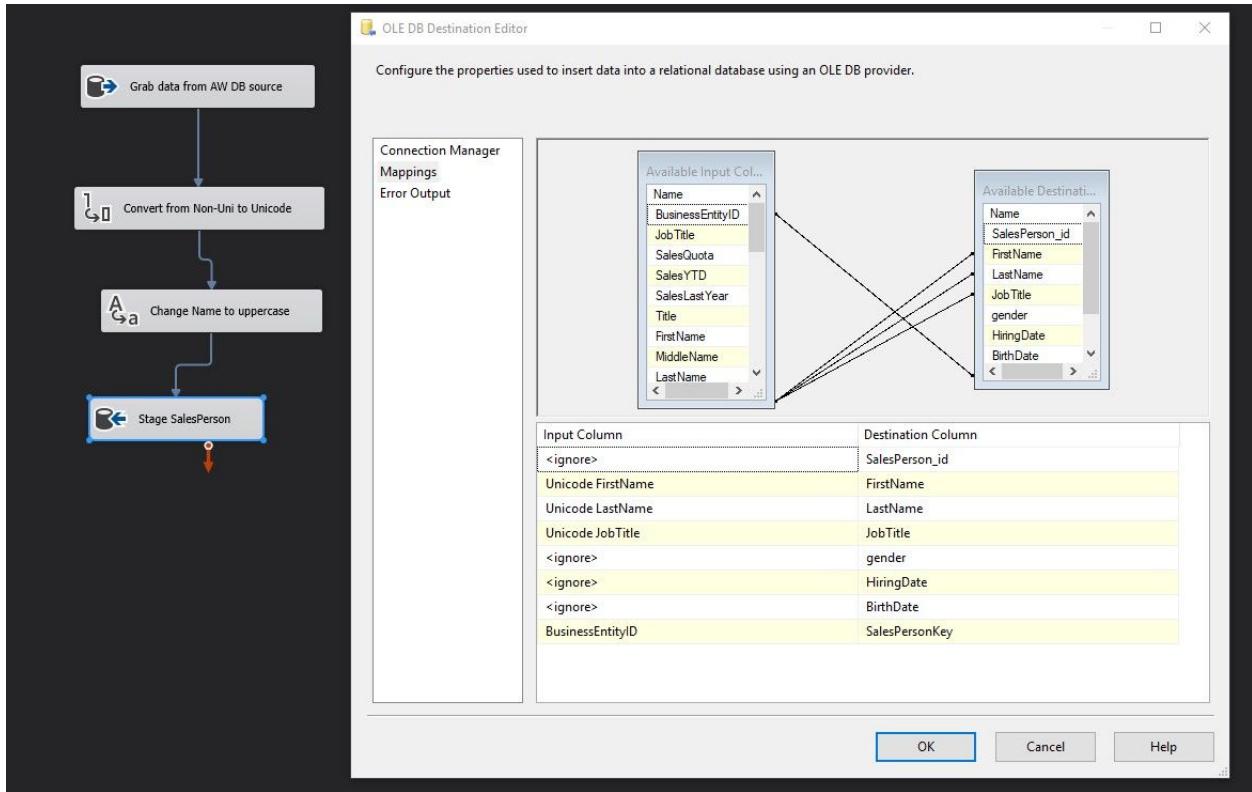
### *Control Flow*



### *Data Flow*







### Final Result

SQLQuery8.sql - Z...gDW (ZAC\Zac (64)) \* → X SQLQuery7.sql - Z...sDW (ZAC\Zac (51)) \* scriptdata.sql - Z...ksDW (ZAC\Zac (62))

```
SELECT * FROM StagingSalesPerson
```

100 %

Results Messages

	SalesPerson_id	FirstName	LastName	JobTitle	gender	HiringDate	BirthDate	SalesPersonKey
1	1	Stephen	Jiang	North American Sales Manager	NULL	NULL	NULL	274
2	2	Michael	Blythe	Sales Representative	NULL	NULL	NULL	275
3	3	Linda	Mitchell	Sales Representative	NULL	NULL	NULL	276
4	4	Jillian	Carson	Sales Representative	NULL	NULL	NULL	277
5	5	Garett	Vargas	Sales Representative	NULL	NULL	NULL	278
6	6	Tsvi	Reiter	Sales Representative	NULL	NULL	NULL	279
7	7	Pamela	Ansmann-Wolfe	Sales Representative	NULL	NULL	NULL	280
8	8	Shu	Ito	Sales Representative	NULL	NULL	NULL	281
9	9	José	Saraiva	Sales Representative	NULL	NULL	NULL	282
10	10	David	Campbell	Sales Representative	NULL	NULL	NULL	283
11	11	Tete	Mensa-Annan	Sales Representative	NULL	NULL	NULL	284
12	12	Syed	Abbas	Pacific Sales Manager	NULL	NULL	NULL	285
13	13	Lynn	Tsoflias	Sales Representative	NULL	NULL	NULL	286
14	14	Amy	Alberts	European Sales Manager	NULL	NULL	NULL	287
15	15	Rachel	Valdez	Sales Representative	NULL	NULL	NULL	288
16	16	Jae	Pak	Sales Representative	NULL	NULL	NULL	289
17	17	Ranjit	Varkey Chudukatil	Sales Representative	NULL	NULL	NULL	290

SQLQuery8.sql - Z...gDW (ZAC\Zac (64)) \* → X SQLQuery7.sql - Z...sDW (ZAC\Zac (51)) \* scriptdata.sql - Z...ksDW (ZAC\Zac (62))

```
SELECT * FROM StagingSalesPerson
```

100 %

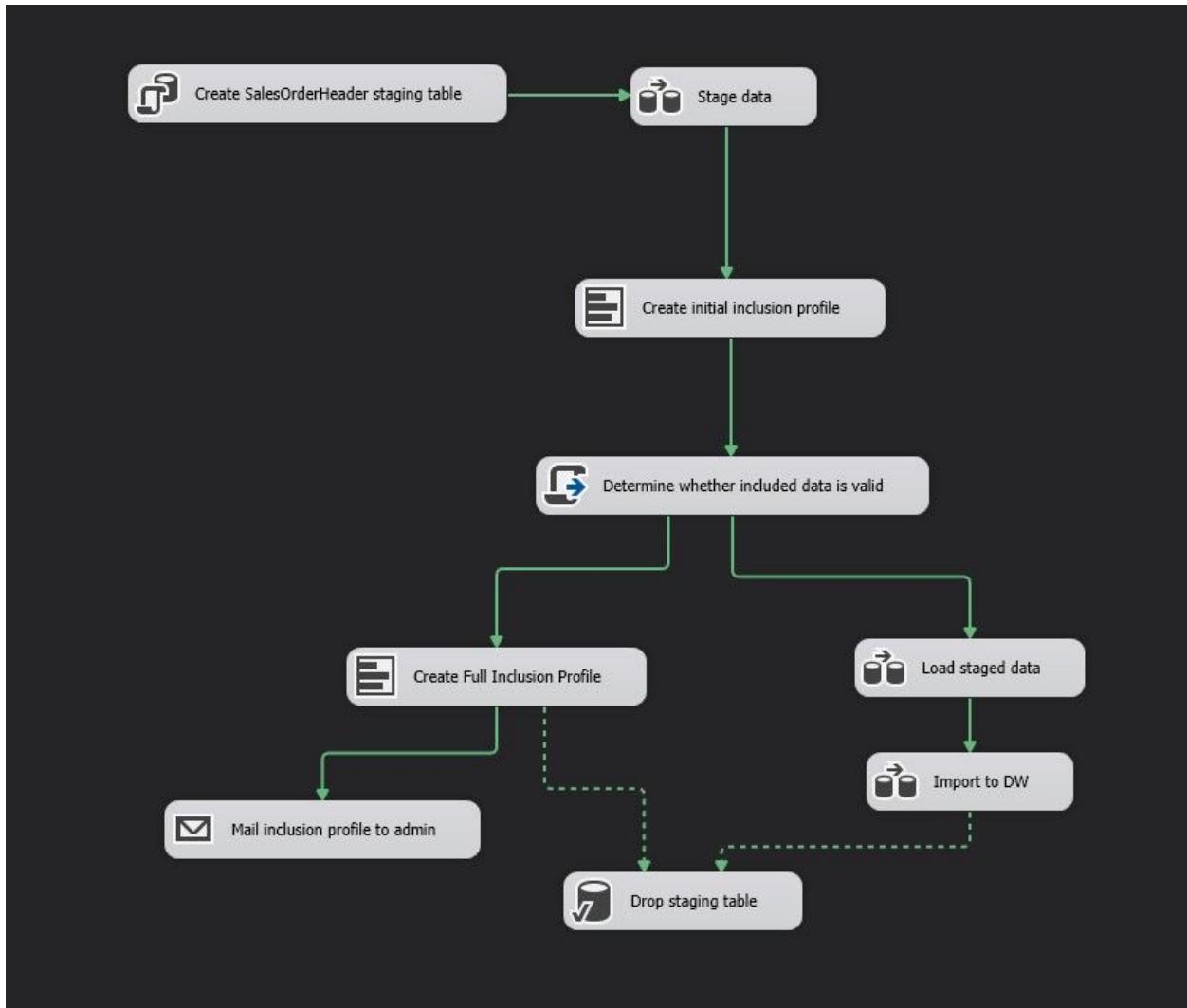
Results Messages

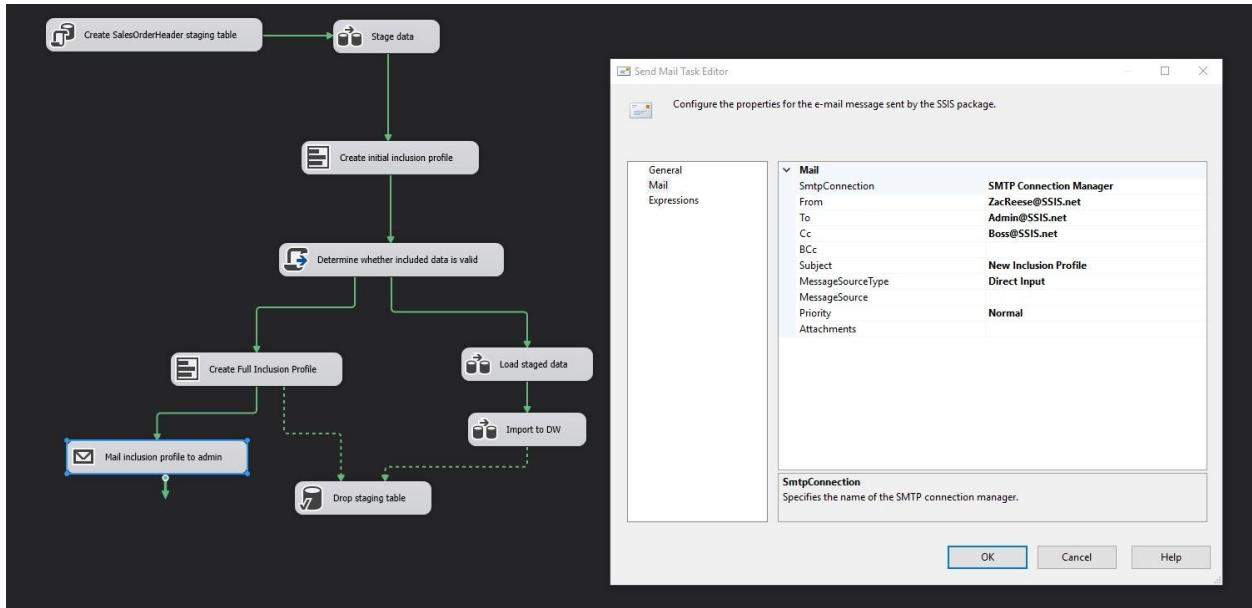
	SalesPerson_id	FirstName	LastName	JobTitle	gender	HiringDate	BirthDate	SalesPersonKey
1	1	STEPHEN	JIANG	North American Sales Manager	NULL	NULL	NULL	274
2	2	MICHAEL	BLYTHE	Sales Representative	NULL	NULL	NULL	275
3	3	LINDA	MITCHELL	Sales Representative	NULL	NULL	NULL	276
4	4	JILLIAN	CARSON	Sales Representative	NULL	NULL	NULL	277
5	5	GARRETT	VARGAS	Sales Representative	NULL	NULL	NULL	278
6	6	TSVI	REITER	Sales Representative	NULL	NULL	NULL	279
7	7	PAMELA	ANSMAN-WOLFE	Sales Representative	NULL	NULL	NULL	280
8	8	SHU	ITO	Sales Representative	NULL	NULL	NULL	281
9	9	JOSÉ	SARAIVA	Sales Representative	NULL	NULL	NULL	282
10	10	DAVID	CAMPBELL	Sales Representative	NULL	NULL	NULL	283
11	11	TETE	MENSA-ANNAN	Sales Representative	NULL	NULL	NULL	284
12	12	SYED	ABBAS	Pacific Sales Manager	NULL	NULL	NULL	285
13	13	LYNN	TSOFLIAS	Sales Representative	NULL	NULL	NULL	286
14	14	AMY	ALBERTS	European Sales Manager	NULL	NULL	NULL	287
15	15	RACHEL	VALDEZ	Sales Representative	NULL	NULL	NULL	288
16	16	JAE	PAK	Sales Representative	NULL	NULL	NULL	289
17	17	RANJIT	VARKEY CHUDUKATIL	Sales Representative	NULL	NULL	NULL	290

## ETL Process - Sales

2. Showing the strength of SSIS control flow. In this diagram we are able to extract our data, load it into our staging data warehouse, and verify the contents of the data. We can create an inclusion profile about the data, what its contents are, the validity of the data, its usefulness, etc. From there we can choose to load the transformed data to the production data warehouse or further analyse the data, followed by emailing the report.

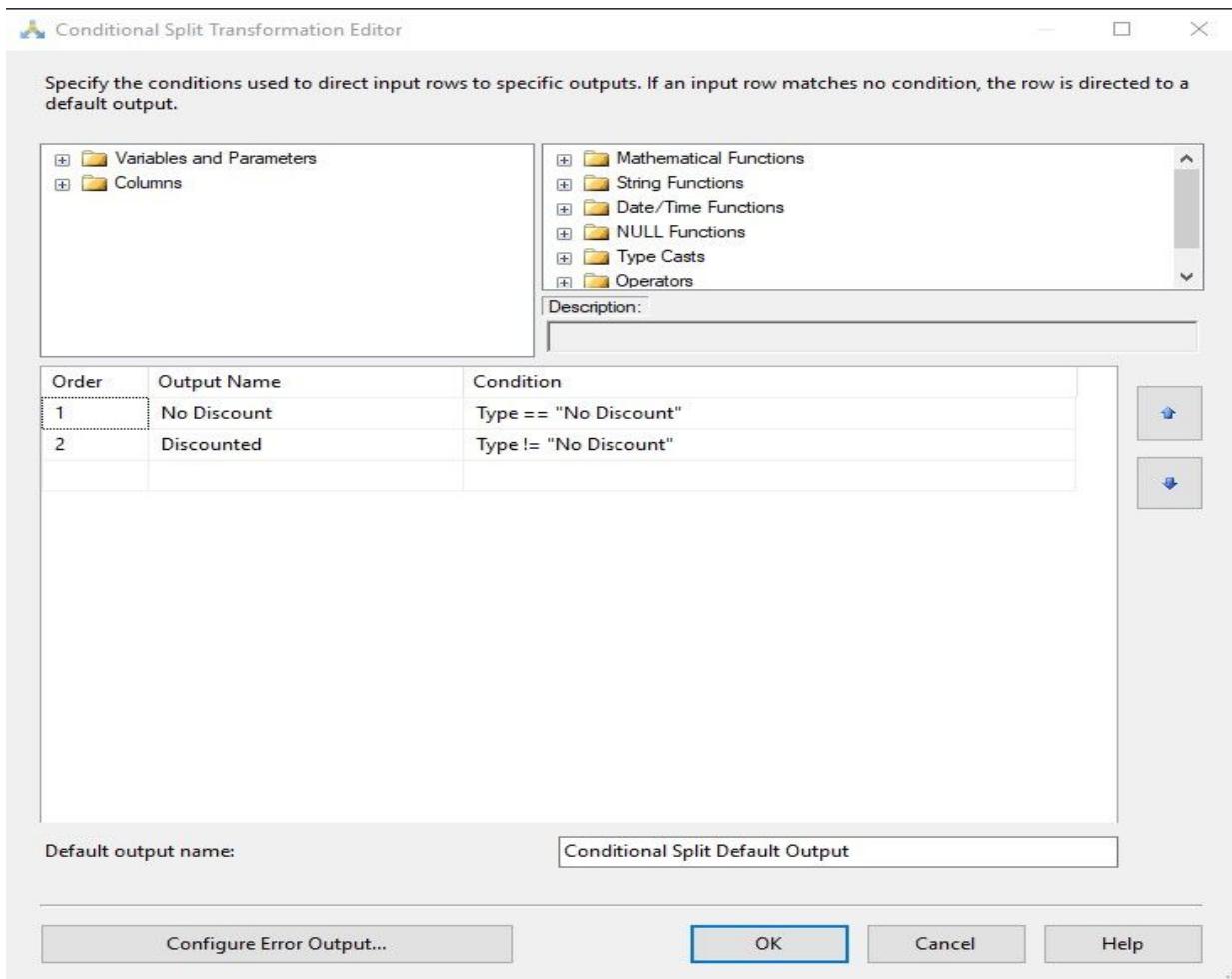
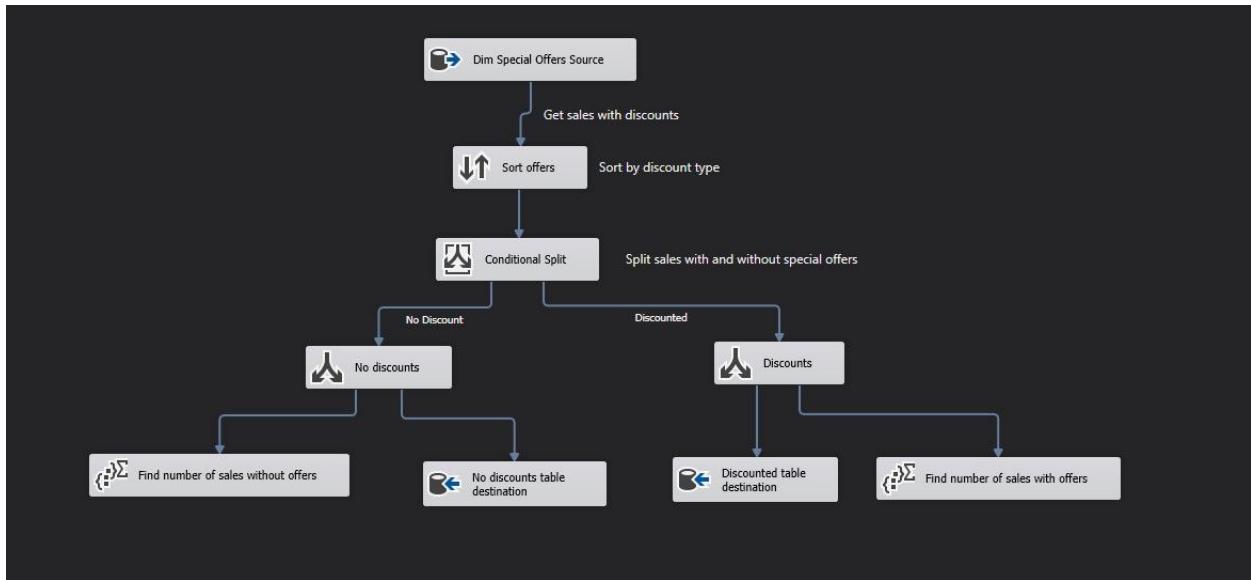
### *Control Flow*





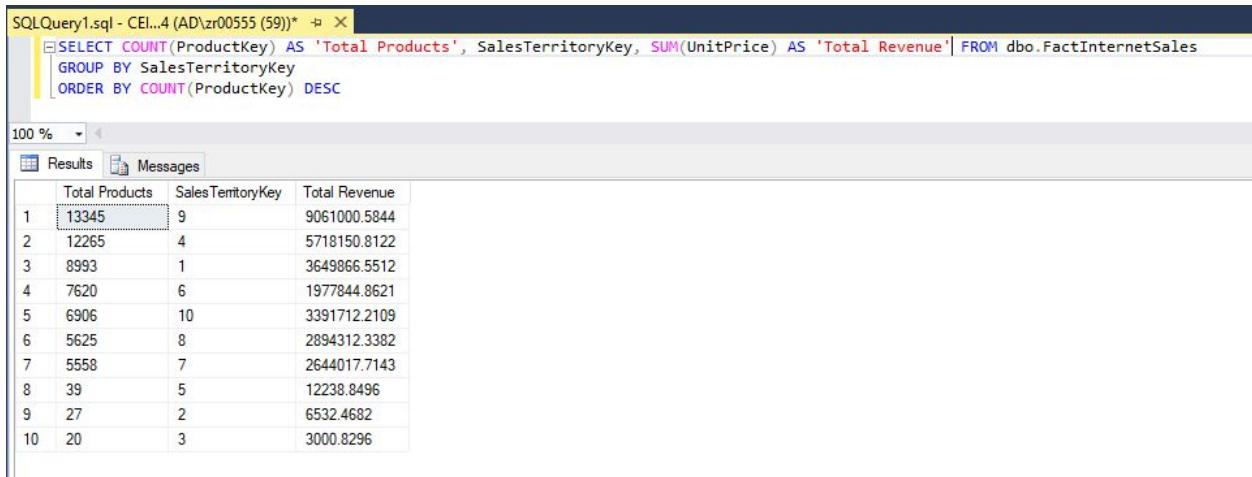
## ETL Process - Sales

3. Here we highlight the conditional split function in SSIS, we can separate data based on certain constraints for further analysis.



## SQL Queries - Zac

1. Which country buys the most products over the internet? And how much have they spent in total?



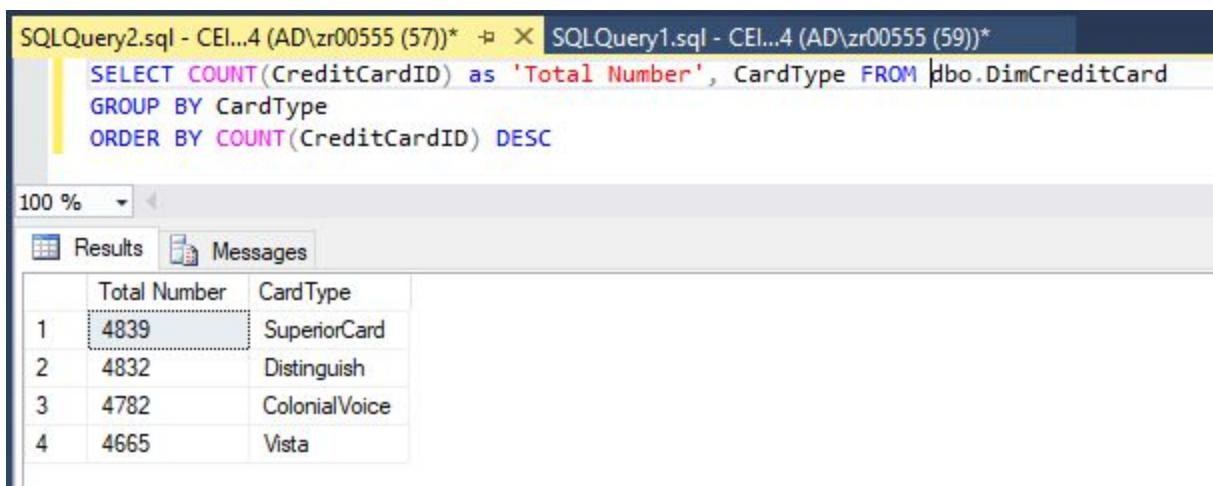
The screenshot shows a SQL query window titled "SQLQuery1.sql - CEI...4 (AD\zr00555 (59))". The query selects the count of ProductKey (Total Products) and the sum of UnitPrice (Total Revenue) from the dbo.FactInternetSales table, grouped by SalesTerritoryKey and ordered by the count of ProductKey in descending order.

```
SELECT COUNT(ProductKey) AS 'Total Products', SalesTerritoryKey, SUM(UnitPrice) AS 'Total Revenue'
FROM dbo.FactInternetSales
GROUP BY SalesTerritoryKey
ORDER BY COUNT(ProductKey) DESC
```

The results grid displays 10 rows of data:

	Total Products	SalesTerritoryKey	Total Revenue
1	13345	9	9061000.5844
2	12265	4	5718150.8122
3	8993	1	3649866.5512
4	7620	6	1977844.8621
5	6906	10	3391712.2109
6	5625	8	2894312.3382
7	5558	7	2644017.7143
8	39	5	12238.8496
9	27	2	6532.4682
10	20	3	3000.8296

2. What is the most popular credit card type?



The screenshot shows a SQL query window titled "SQLQuery2.sql - CEI...4 (AD\zr00555 (57))". The query selects the count of CreditCardID (Total Number) and the CardType from the dbo.DimCreditCard table, grouped by CardType and ordered by the count of CreditCardID in descending order.

```
SELECT COUNT(CreditCardID) as 'Total Number', CardType
FROM dbo.DimCreditCard
GROUP BY CardType
ORDER BY COUNT(CreditCardID) DESC
```

The results grid displays 4 rows of data:

	Total Number	CardType
1	4839	SuperiorCard
2	4832	Distinguish
3	4782	ColonialVoice
4	4665	Vista

3. Who is the top 10% of sales persons by total sales revenue?

```
SQLQuery2.sql - CEI...4 (AD\zr00555 (57))*
SQLQuery1.sql - CEI...4 (AD\zr00555 (59))*
SELECT DISTINCT TOP (10) PERCENT c.salesYTD, c.lastName, c.firstName
FROM
    (SELECT s.FirstName, s.LastName, SUM(f.salesYTD) AS salesYTD
     FROM dbo.FactSalesOrderHeader f JOIN dbo.DimSalesPerson s ON f.SalesPerson_id = s.SalesPerson_id
     GROUP BY S.LastName, S.FirstName) AS c
ORDER BY c.subTotal DESC
```

100 %

	salesYTD	LastName	FirstName
1	3763178.1787	Blythe	Michael
2	4251368.5497	Mitchell	Linda
3	3189418.3662	Carson	Jillian
4	4116871.2277	Pak	Jae
5	3121616.3202	Varkey Chudukatil	Ranjit

4. Which vendors have the most profit when accounting for number of employees (Profit/NumOfEmployees)

```
SQLQuery2.sql - CEI...4 (AD\zr00555 (57))*
SQLQuery1.sql - CEI...4 (AD\zr00555 (59))*
SELECT Name, AnnualRevenue, NumberEmployees FROM dbo.DimVendors
ORDER BY (AnnualRevenue/NumberEmployees) DESC
```

100 %

	Name	AnnualRevenue	NumberEmployees
1	Fun Toys and Bikes	30000.00	2
2	Grand Bicycle Stores	30000.00	2
3	Another Bicycle Company	30000.00	2
4	Professional Cycle Store	30000.00	2
5	A Bike Store	30000.00	2
6	Operational Manufacturing	30000.00	2

## 5. Select top 10 best exchange rate currency

The screenshot shows a SQL Server Management Studio window with two tabs: 'SQLQuery2.sql' and 'SQLQuery1.sql'. The 'SQLQuery2.sql' tab is active, displaying the following query:

```
SELECT DISTINCT TOP (10) AverageRate, ToCurrencyCode FROM Sales.CurrencyRate
```

The results are displayed in a table titled 'Results' with columns 'AverageRate' and 'ToCurrencyCode'. The data is as follows:

	AverageRate	ToCurrencyCode
1	1.00	ARS
2	1.5491	AUD
3	1.9379	BRL
4	1.4641	CAD
5	8.2781	CNY
6	1.8967	DEM
7	0.9697	EUR
8	6.3611	FRF
9	0.6183	GBP
10	104.91	JPY

## SQL Queries - Jonathan

1. The popularity of each product category per territory and how many of each product category was order per territory.

```
Select b.SalesTerritoryKey, b.SalesTerritoryRegion, c.ProductCategoryKey, c.EnglishProductCategoryName, Count(c.ProductCategoryKey) AS NumberOfOrdersForCategory
From dbo.FactSalesOrderHeader as a
join dbo.DimSalesTerritory as b
  On a.SalesTerritoryKey = b.SalesTerritoryKey
join dbo.DimProductCategory as c
  on c.ProductCategoryKey = a.ProductCategoryKey

Group by b.SalesTerritoryKey, b.SalesTerritoryRegion
Order by b.SalesTerritoryKey
```

100 %

	SalesTerritoryKey	SalesTerritoryRegion	ProductCategoryKey	EnglishProductCategoryName	CountOfCategory
1	1	Northwest	2	Components	2629
2	1	Northwest	3	Clothing	2873
3	1	Northwest	4	Accessories	6176
4	1	Northwest	1	Bikes	5187
5	2	Northeast	3	Clothing	1100
6	2	Northeast	4	Accessories	485
7	2	Northeast	2	Components	1806
8	2	Northeast	1	Bikes	2445
9	3	Central	2	Components	1615

\*Comparison to the code need to show the same data between a data warehouse and a regular database

\*database code

```
Select a.TerritoryID as SalesTerritoryKey, b.Name as SalesTerritoryRegion, iden.ProductCategoryID as ProductCategoryKey, iden.Name as EnglishProductCategoryName, Count(iden.ProductCategoryID) as CountOfCategory
From Sales.SalesOrderHeader as a
join Sales.SalesTerritory as b
  On a.TerritoryID = b.TerritoryID
join Sales.SalesOrderDetail as c
  On c.SalesOrderID = a.SalesOrderID
Join (
  Select c.Name, c.ProductCategoryID as ProductCategoryID, a.ProductID
From Production.Product as a
join Production.ProductSubcategory as b
  On a.ProductSubcategoryID = b.ProductSubcategoryID
join Production.ProductCategory as c
  On c.ProductCategoryID = b.ProductCategoryID ) as Iden
  On iden.ProductID = c.ProductID

Group by iden.Name, a.TerritoryID, b.Name, iden.ProductCategoryID
Order by a.TerritoryID
```

10 %

	SalesTerritoryKey	SalesTerritoryRegion	ProductCategoryKey	EnglishProductCategoryName	CountOfCategory
1	1	Northwest	2	Components	2629
2	1	Northwest	3	Clothing	2873
3	1	Northwest	4	Accessories	6176
4	1	Northwest	1	Bikes	5187
5	2	Northeast	3	Clothing	1100
6	2	Northeast	4	Accessories	485
7	2	Northeast	2	Components	1806
8	2	Northeast	1	Bikes	2445
9	3	Central	2	Components	1615
10	3	Central	3	Clothing	1148
11	3	Central	1	Bikes	2569
12	3	Central	4	Accessories	500
13	4	Southwest	4	Accessories	8386
14	4	Southwest	2	Components	4198

## 2. Total number of orders per region

```
Select b.SalesTerritoryKey, b.SalesTerritoryRegion, Count(a.SalesOrderHeader_id) as NumberOfSales
From dbo.FactSalesOrderHeader as a
join dbo.DimSalesTerritory as b
On a.SalesTerritoryKey = b.SalesTerritoryKey

Group by b.SalesTerritoryKey, b.SalesTerritoryRegion
Order by b.SalesTerritoryKey
```

100 %

Results Messages

	SalesTerritoryKey	SalesTerritoryRegion	NumberOfSales
1	1	Northwest	16865
2	2	Northeast	5836
3	3	Central	5832
4	4	Southwest	25644
5	5	Southeast	5976
6	6	Canada	19064
7	7	France	9088
8	8	Germany	7528
9	9	Australia	15058
10	10	United Kingdom	10426

## 3. Total income per territory till the current point in time

```
Select Sum(a.SubTotal) as TotalRevenuePerTerritory, a.SalesTerritoryKey, b.SalesTerritoryRegion, b.SalesTerritoryCountry
From dbo.FactSalesOrderHeader as a
join dbo.DimSalesTerritory as b
On a.SalesTerritoryKey = b.SalesTerritoryKey

Group by a.SalesTerritoryKey, b.SalesTerritoryRegion, b.SalesTerritoryCountry
Order By a.SalesTerritoryKey
```

100 %

Results Messages

	TotalRevenuePerTerritory	SalesTerritoryKey	SalesTerritoryRegion	SalesTerritoryCountry
1	411207275.1256	1	Northwest	US
2	253771294.4446	2	Northeast	US
3	263099145.8509	3	Central	US
4	696896625.8111	4	Southwest	US
5	226427460.2519	5	Southeast	US
6	526969463.2426	6	Canada	CA
7	198158287.8685	7	France	FR
8	92658225.6455	8	Germany	DE
9	70573924.1595	9	Australia	AU
10	187208421.6412	10	United Kingdom	GB

## Production - Keenan

### WIP

WIP Production-Keenan								
Product Dimension								
What-Product Product	By- Product Sub Category ProductSubCategory	What-Location Location	What-ProductDescription ProductDescription	What-TransactionHistory TransactionHistory	What-ProductCategory ProductCategory	What-ProductProduct Photo ProductProductPhoto	What-ProductModel ProductModel	
ProductID	ProductSubCategoryID	LocationID	ProductDescriptionID	TransactionID	ProductCategoryID	ProductID	ProductModelID	
Name	ProductCategoryID	Name	Description	ProductID	Name	ProductPhotoID	Name	
ProductNumber		CostRate	rowguid	ReferenceOrderID	rowguid	Primary	CatalogDescription	
MakeFlag	rowguid	Availability	ModifiedDate	TransactionDate	ModifiedDate	ModifiedDate	Instructions	
FinishedGoodsFlag	ModifiedDate	ModifiedDate		TransactionType	Quantity	rowguid	ModifiedDate	
Color					ActualCost			
SafetyStockLevel					ModifiedDate			
ReorderPoint					ReferenceOrderLineID			
StandardCost								
ListPrice								
Size								
SizeUnitMeasureCode								
WeightUnitMeasureCode								
WeightUnitMeasureCode								
DaysToManufacture								
ProductLine								
Class								
Style								
ProductSubcategoryID								
ProductModelID								
SellStartDate								
SellEndDate								
DiscontinuedDate								
rowguid								
ModifiedDate								

Observations/Measures: QuantityNum=4, CategoryNum=4, TotalTransactions=5	
Preliminary Workload :	
Question/Query:	How many products had atleast 1 finished good flag and the list price was less than \$200?
Question/Query:	How many transactions have been made this year?
Question/Query:	How many products have the same bill of materials?
Question/Query:	Which products are less than \$50?
Question/Query:	What % of orders were sent out on time ?

## BEAMS

### *UnitMeasure Dimension*

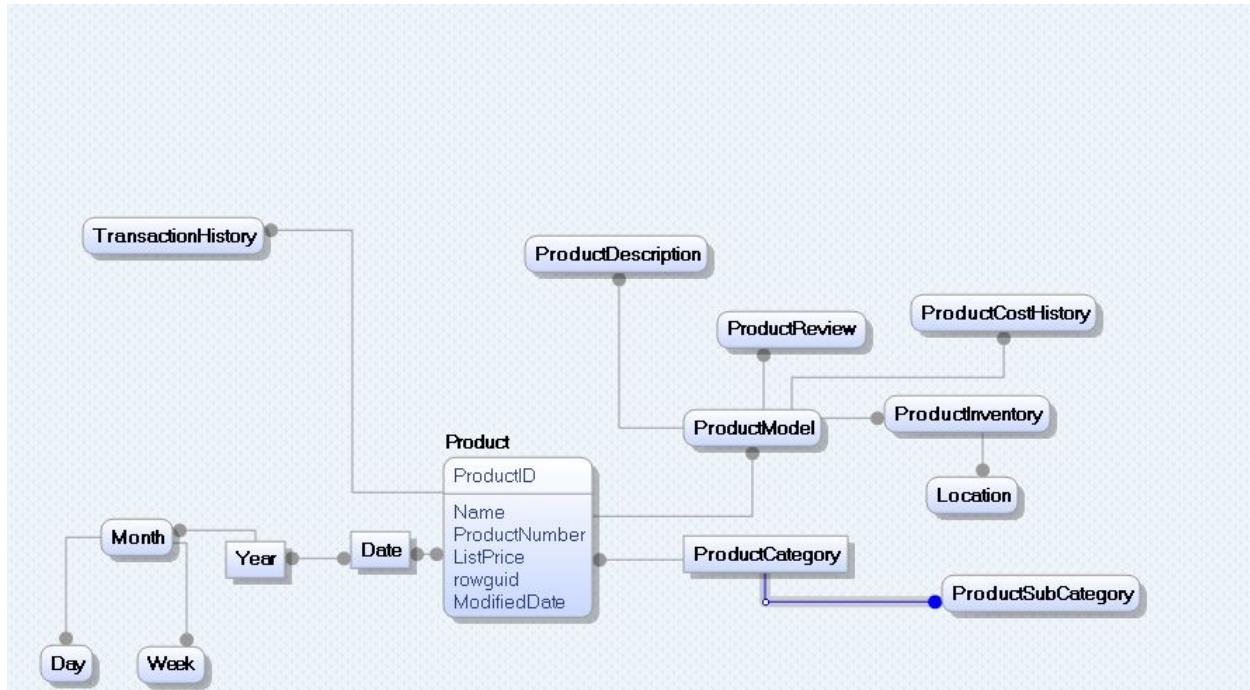
Beam UnitMeasureCode Dimension - Keenan		
	has	updated
UnitMeasureCode	Name	ModifiedDate
MD		
um_code	name	modified_date
Box	Boxes	2008-04-30 0:00:00
BTL	Cottle	2008-04-30 0:00:00
C	Celcius	2008-04-30 0:00:00
CAN	Canister	2008-04-30 0:00:00
CAR	Carton	2008-04-30 0:00:00
CBM	Cubic meters	2008-04-30 0:00:00
CCM	Cubic centimeter	2008-04-30 0:00:00
CDM	Cubic decimeter	2008-04-30 0:00:00
CM	Centimeter	2008-04-30 0:00:00
CM2	Square centimeter	2008-04-30 0:00:00
CR	Crate	2008-04-30 0:00:00
CS	Case	2008-04-30 0:00:00
CTN	Container	2008-04-30 0:00:00
DM	Decometer	2008-04-30 0:00:00
DZ	Dozen	2008-04-30 0:00:00
EA	Each	2008-04-30 0:00:00

## Data Warehouse Design

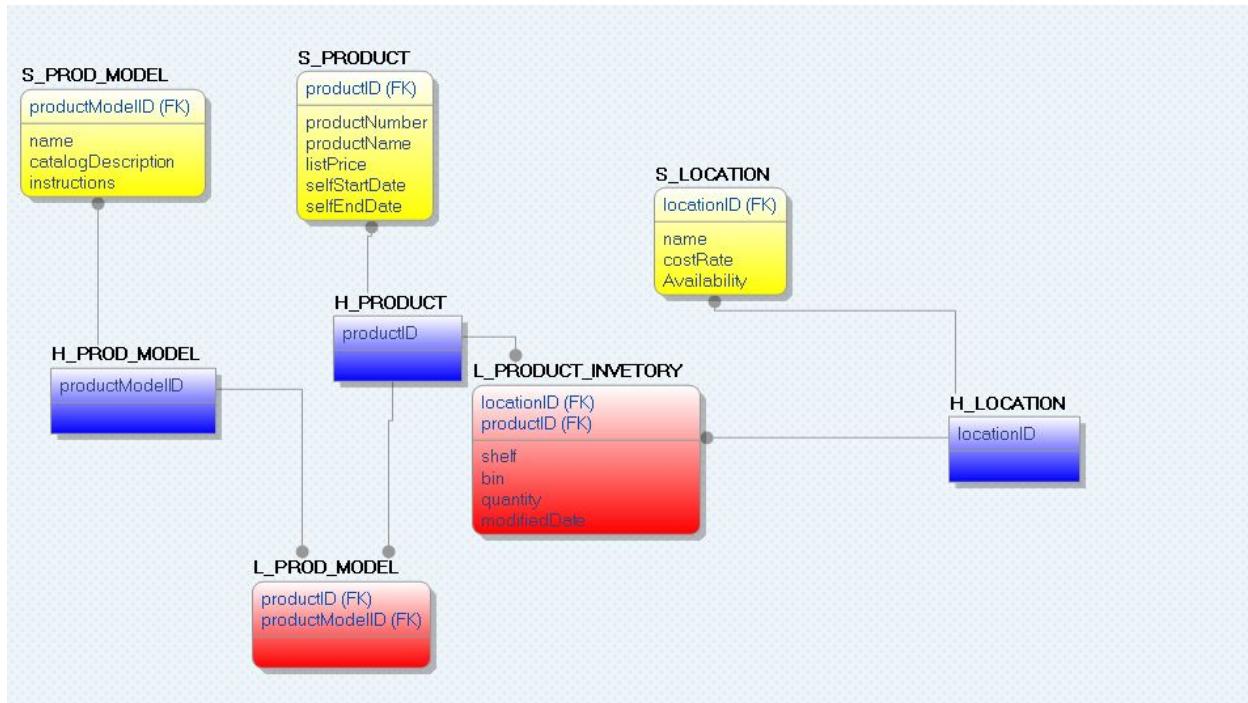
Conceptual Models

Production - Keenan

*Dimensional Fact Model*

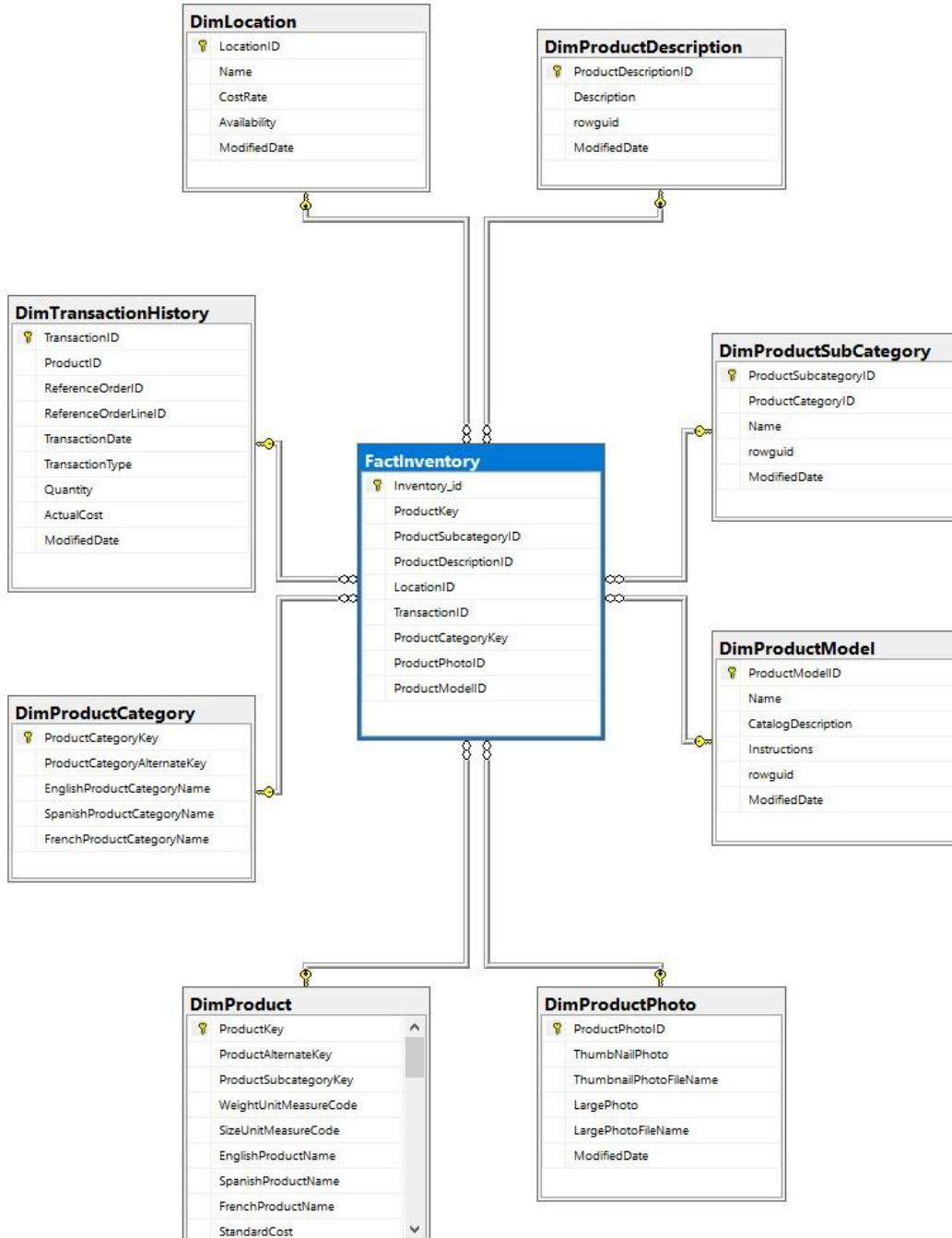


## Data Vault



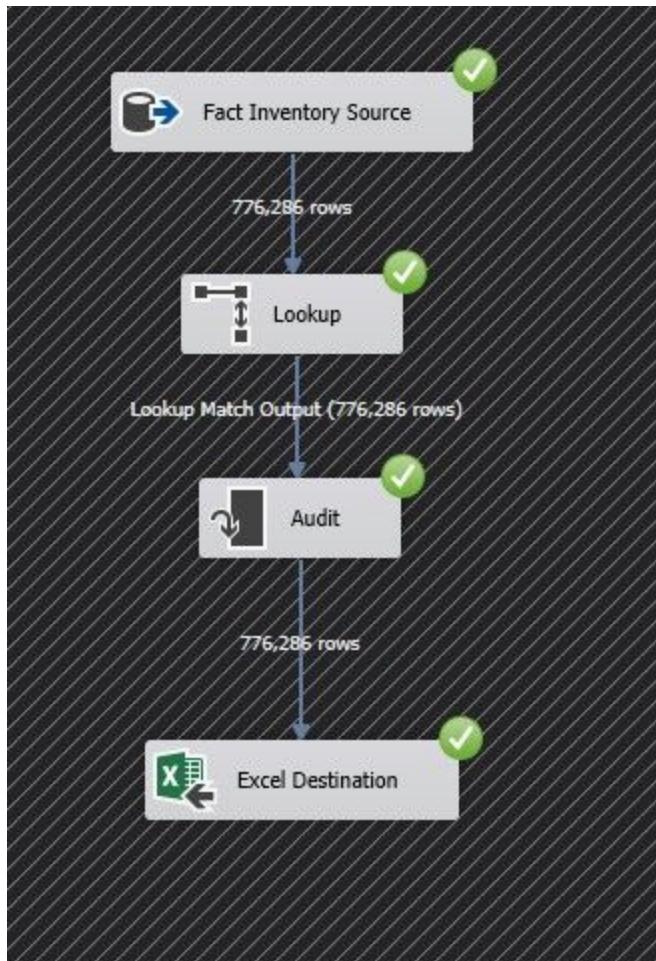
## Logical Model

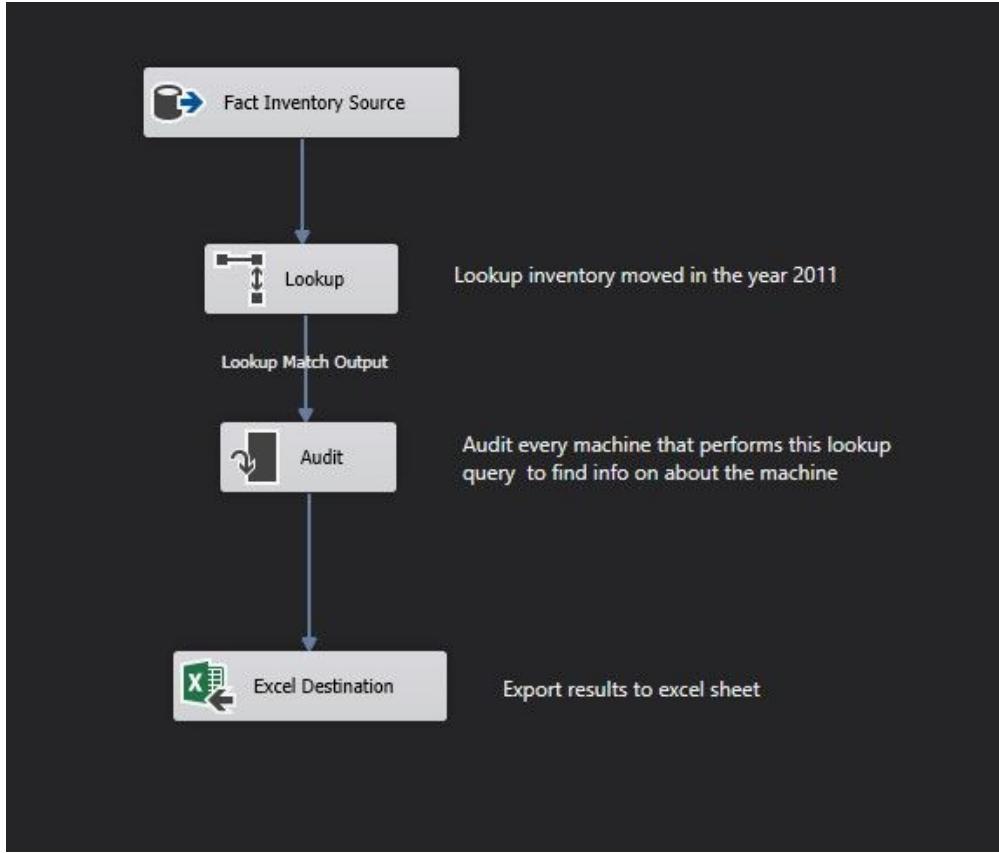
*Production Data Mart*



## ETL Process - Production

1. Makes use of the Audit function to be able to retrieve the machine name, user name, and product ID of each computer that runs a query on the selected source.

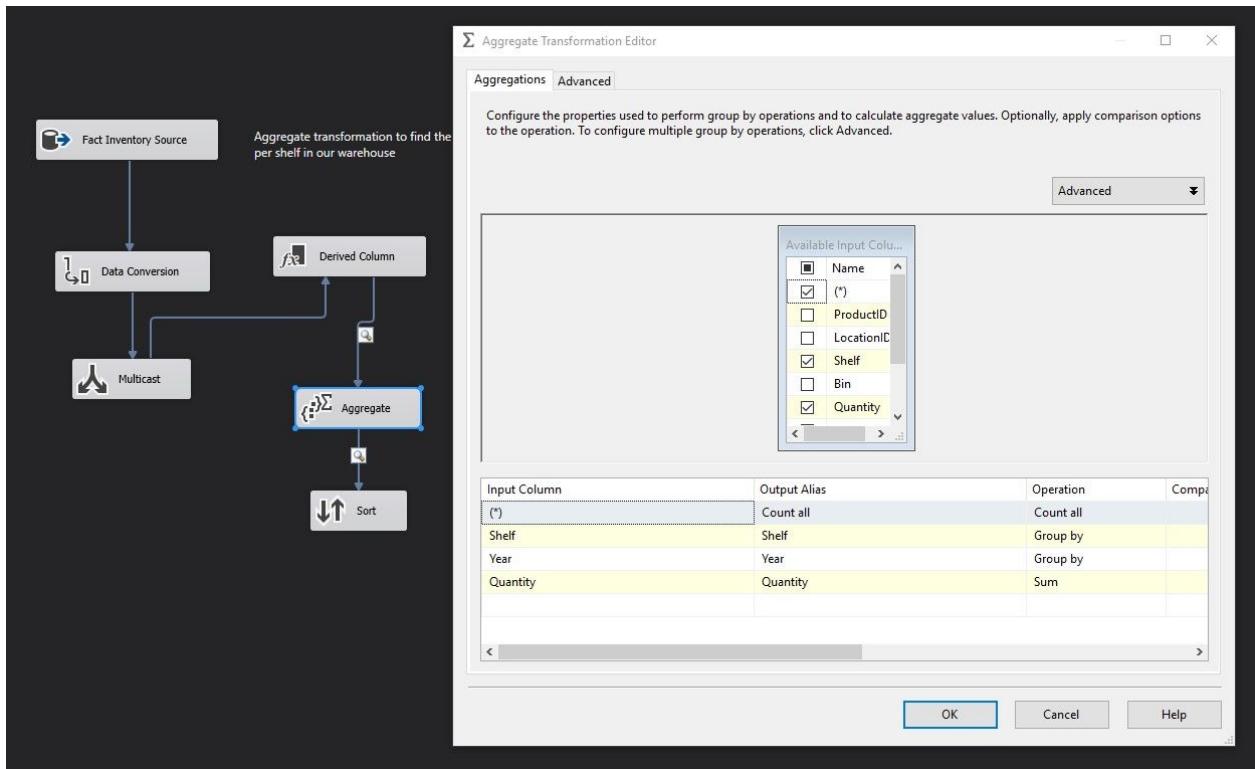
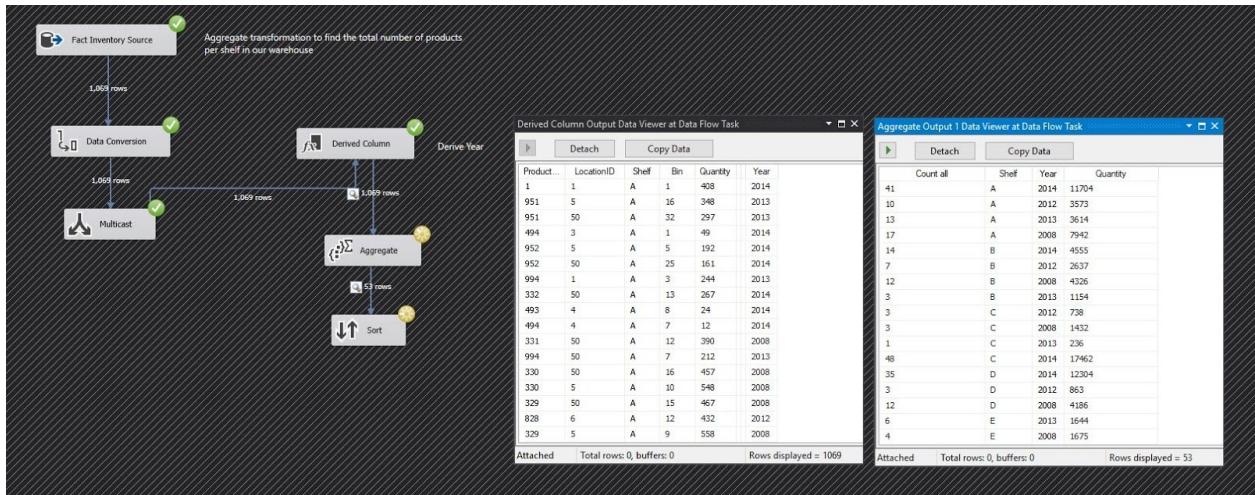




	A	B	C	D	E	F	G	H	I	J
1	PackageID	User name	ProductKe	DateKey	MovementDate	UnitCost	UnitsIn	UnitsOut	UnitsBalance	
2	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110101	2011-01-01	0.32	0	0	875	
3	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110102	2011-01-02	0.32	0	0	875	
4	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110103	2011-01-03	0.32	0	0	875	
5	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110104	2011-01-04	0.32	0	0	875	
6	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110105	2011-01-05	0.32	0	0	875	
7	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110106	2011-01-06	0.32	0	0	875	
8	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110107	2011-01-07	0.32	0	0	875	
9	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110108	2011-01-08	0.32	0	0	875	
10	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110109	2011-01-09	0.32	0	0	875	
11	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110110	2011-01-10	0.32	0	0	875	
12	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110111	2011-01-11	0.32	0	0	875	
13	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110112	2011-01-12	0.32	0	0	875	
14	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110113	2011-01-13	0.32	0	0	875	
15	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110114	2011-01-14	0.32	0	0	875	
16	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110115	2011-01-15	0.32	0	0	875	
17	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110116	2011-01-16	0.32	0	0	875	
18	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110117	2011-01-17	0.32	0	0	875	
19	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110118	2011-01-18	0.33	0	0	875	
20	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110119	2011-01-19	0.33	0	0	875	
21	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110120	2011-01-20	0.32	0	0	875	
22	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110121	2011-01-21	0.32	0	0	875	
23	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110122	2011-01-22	0.32	0	0	875	
24	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110123	2011-01-23	0.32	0	0	875	
25	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110124	2011-01-24	0.32	0	0	875	
26	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110125	2011-01-25	0.32	0	0	875	
27	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110126	2011-01-26	0.32	0	0	875	
28	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110127	2011-01-27	0.32	0	0	875	
29	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110128	2011-01-28	0.32	0	0	875	
30	{6F220AC8-D603-4F69-97C4-54F93E02941E}	KEENAN/Keenan	1	20110129	2011-01-29	0.32	0	0	875	

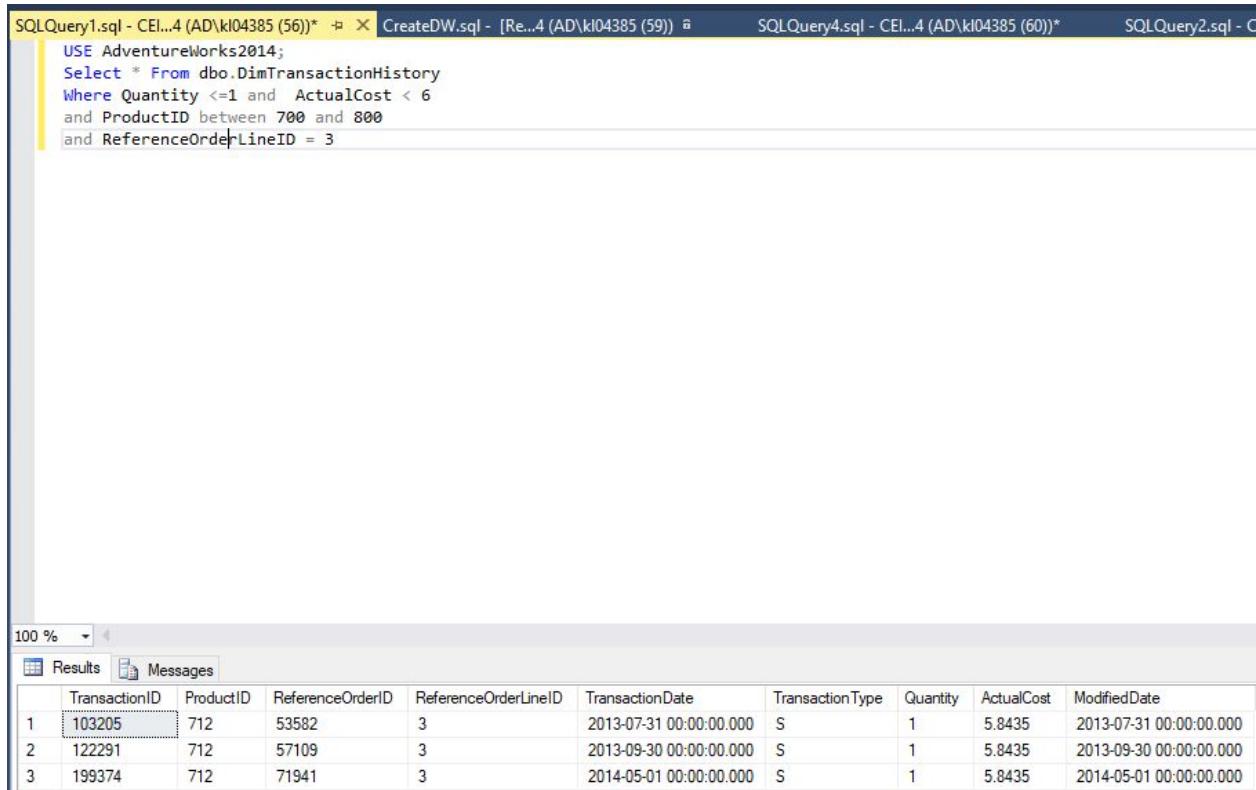
## ETL Process - Production

2. This diagram makes use of the Aggregate data function of SSIS. This function allows us to find specific, aggregated data, such as “What is the quantity of items on each shelf of inventory, grouped by year?”.



## SQL Queries - Production

- 1) How many products had a product ID between 700 and 800 had a cost less than 6 with the same reference order line?



The screenshot shows a SQL Server Management Studio window with three tabs at the top: 'SQLQuery1.sql - CEI..4 (AD\kl04385 (56))' (selected), 'CreateDW.sql - [Re..4 (AD\kl04385 (59))' (disabled), and 'SQLQuery4.sql - CEI..4 (AD\kl04385 (60))' (disabled). Below the tabs is a code editor with the following T-SQL query:

```
USE AdventureWorks2014;
Select * From dbo.DimTransactionHistory
Where Quantity <=1 and ActualCost < 6
and ProductID between 700 and 800
and ReferenceOrderLineID = 3
```

At the bottom, there is a results grid titled 'Results' with three rows of data:

	TransactionID	ProductID	ReferenceOrderID	ReferenceOrderLineID	TransactionDate	TransactionType	Quantity	ActualCost	ModifiedDate
1	103205	712	53582	3	2013-07-31 00:00:00.000	S	1	5.8435	2013-07-31 00:00:00.000
2	122291	712	57109	3	2013-09-30 00:00:00.000	S	1	5.8435	2013-09-30 00:00:00.000
3	199374	712	71941	3	2014-05-01 00:00:00.000	S	1	5.8435	2014-05-01 00:00:00.000

**2) Products with only one finished good flag and a price less than 20**

SQLQuery7.sql - CEI...4 (AD\kl04385 (52)) \* X

```
SELECT FinishedGoodsFlag , ListPrice
FROM dbo.DimProduct
Where FinishedGoodsFlag = 1 and ListPrice < 20
```

100 %

Results Messages

	FinishedGoodsFlag	List Price
1	1	9.50
2	1	9.50
3	1	8.99
4	1	19.99
5	1	13.99
6	1	4.99
7	1	9.99
8	1	8.99
9	1	2.29
10	1	8.99
11	1	8.99
12	1	7.95
13	1	4.99
14	1	3.99
15	1	4.99

### 3. Top 10 percent of products standard cost

The screenshot shows a SQL Server Management Studio (SSMS) interface. The top pane displays a query window titled "SQLQuery2.sql - CEI...4 (AD\kl04385 (55))". The query is:

```
SELECT TOP(10) PERCENT
ProductID, StandardCost
FROM dbo.DimProductCostHistory
```

The bottom pane shows the "Results" tab with the query's output. The table has two columns: "ProductID" and "StandardCost". The data is as follows:

	ProductID	StandardCost
1	707	12.0278
2	707	13.8782
3	707	13.0863
4	708	12.0278
5	708	13.8782
6	708	13.0863
7	709	3.3963
8	710	3.3963
9	711	12.0278
10	711	13.8782
11	711	13.0863
12	712	5.7052
13	712	5.2297
14	712	6.9223
15	713	31.7244
16	713	29.0807
17	713	38.4923
18	714	31.7244
19	714	29.0807

**4. Top 3 percent of work orders with 3 hours of research**

```
SELECT TOP(3) PERCENT WorkOrderID, ActualResourceHrs  
From dbo.DimWorkOrderRouting  
Where ActualResourceHrs = 3
```

	WorkOrderID	ActualResourceHrs
1	13	3.0000
2	14	3.0000
3	39	3.0000
4	40	3.0000
5	41	3.0000
6	42	3.0000
7	65	3.0000
8	66	3.0000
9	67	3.0000
10	68	3.0000
11	69	3.0000
12	70	3.0000
13	71	3.0000
14	72	3.0000
15	73	3.0000
16	74	3.0000
17	75	3.0000
18	76	3.0000
19	77	3.0000

## 5. How products have to same quantity during assembly of 2?

```
SELECT BillOfMaterialsID , PerAssemblyQty  
FROM dbo.DimBillOfMaterials  
WHERE BillOfMaterialsID BETWEEN 10 and 300  
and PerAssemblyQty = 2
```

100 %

Results Messages

	BillOfMaterialsID	PerAssemblyQty
1	262	2.00
2	291	2.00
3	279	2.00
4	273	2.00
5	17	2.00
6	268	2.00
7	12	2.00
8	120	2.00
9	119	2.00
10	153	2.00
11	288	2.00
12	299	2.00
13	285	2.00

## Human Resources - Chris

### WIP

Human Resources Work Information Package (WIP) - Christopher Iverson					
Who - Shift	Who - Employee	Who - Job Candidate	Where - Department	What - Department History	What - Pay History
D1- Shift	D2 - Employee	D3 - JobCandidate	D4 - Department	D5 - DepartmentHistory	D6 - PayHistory
ShiftID	BusinessEntityID	BusinessEntityID	DepartmentID	BusinessEntityID	BusinessEntityID
Name	NationlDNumber	JobCandidateID	Name	DepartmentID	RateChangeDate
StartTime	LoginID	Resume	GroupName	StartDate	Rate
EndTime	ShiftID	ApplicationDate	ModifiedDate	ShiftID	PayFrequency
ModifiedDate	JobTitle	ModifiedDate		EndDate	ModifiedDate
	BirthDate				
	MaritalStatus				
	Gender				
	HireDate				
	SalariedFlag				
	VacationHours				
	SickLeaveHours				
	CurrentFlag				
	ModifiedDate				
Observations/Measures: Rate - \$					
Expected history length of 7 years					
Preliminary Workload					
Question/Query	How many employees have been hired in the last x months?				
Question/Query	Of the job candidates that applied in the past x weeks, how many submitted a resume?				
Question/Query	Which department has the most/least employees?				
Question/Query	Which employee has been with the company the longest?				
Question/Query	List all employees who make more than \$x a year				
Question/Query	See if employee x has any vacation hours left				

### BEAMS

#### *Shift Dimension*

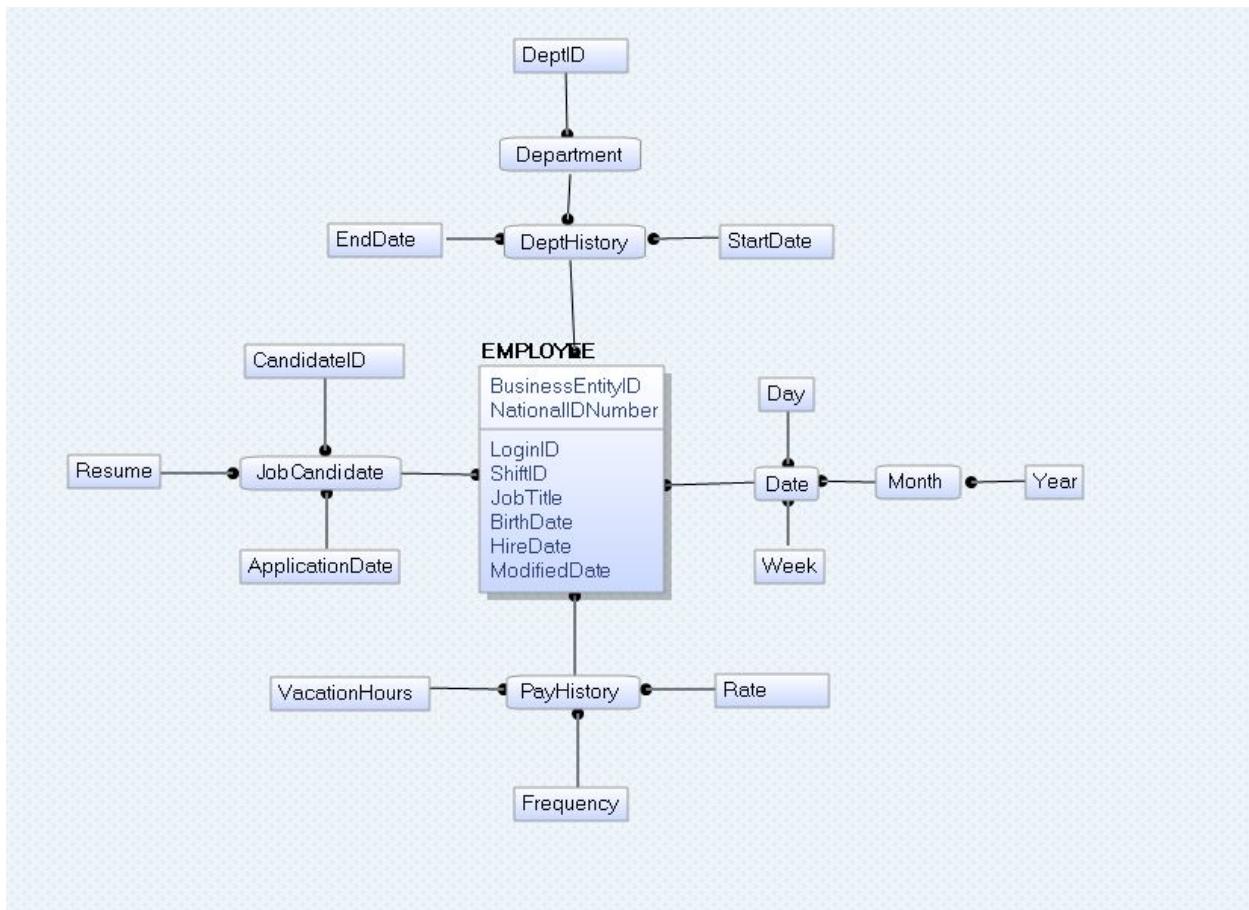
Human Resources Shift Dimension					
SHIFT	HV				
	worked by	started on	ended on	modified on	
SHIFT_ID	EMPLOYEE_NAME	START_TIME	END_TIME	MODIFIED_DATE	
MD, NN	MD, NN				
{PRD_ID}	{PRD_NAME}	{PRD_START}	{PRD_END}	{PRD_MOD}	
8457745	Nathan Smith	8:00AM 7-Feb-19	4:30PM 7-Feb-19		8-Feb-19
8254763	Austin Spade	9:00AM 7-Feb-19	4:00PM 7-Feb-19		10-Jan-19
8459741	David Williams	10:00AM 7-Feb-19	6:00PM 7-Feb-19		13-Dec-18
8991631	Ryan Trommer	8:22AM 7-Feb-19	7:00PM 7-Feb-19		23-Jan-19
8779663	Brennan McCoy	4:00AM 8-Feb-19	5:30PM 8-Feb-19		15-Dec-18
8453215	Nick Muller	5:00AM 8-Feb-19	4:44PM 8-Feb-19		3-Nov-18
8964863	Katie Walden	5:22AM 9-Feb-19	2:22PM 9-Feb-19		3-Nov-18
8945765	Matt Payne	5:59AM 9-Feb-19	12:00PM 9-Feb-19		12-Jan-19
8678515	Stephen Gowen	6:00AM 9-Feb-19	9:00PM 9-Feb-19		26-Jan-19

## Data Warehouse Design

Conceptual Models

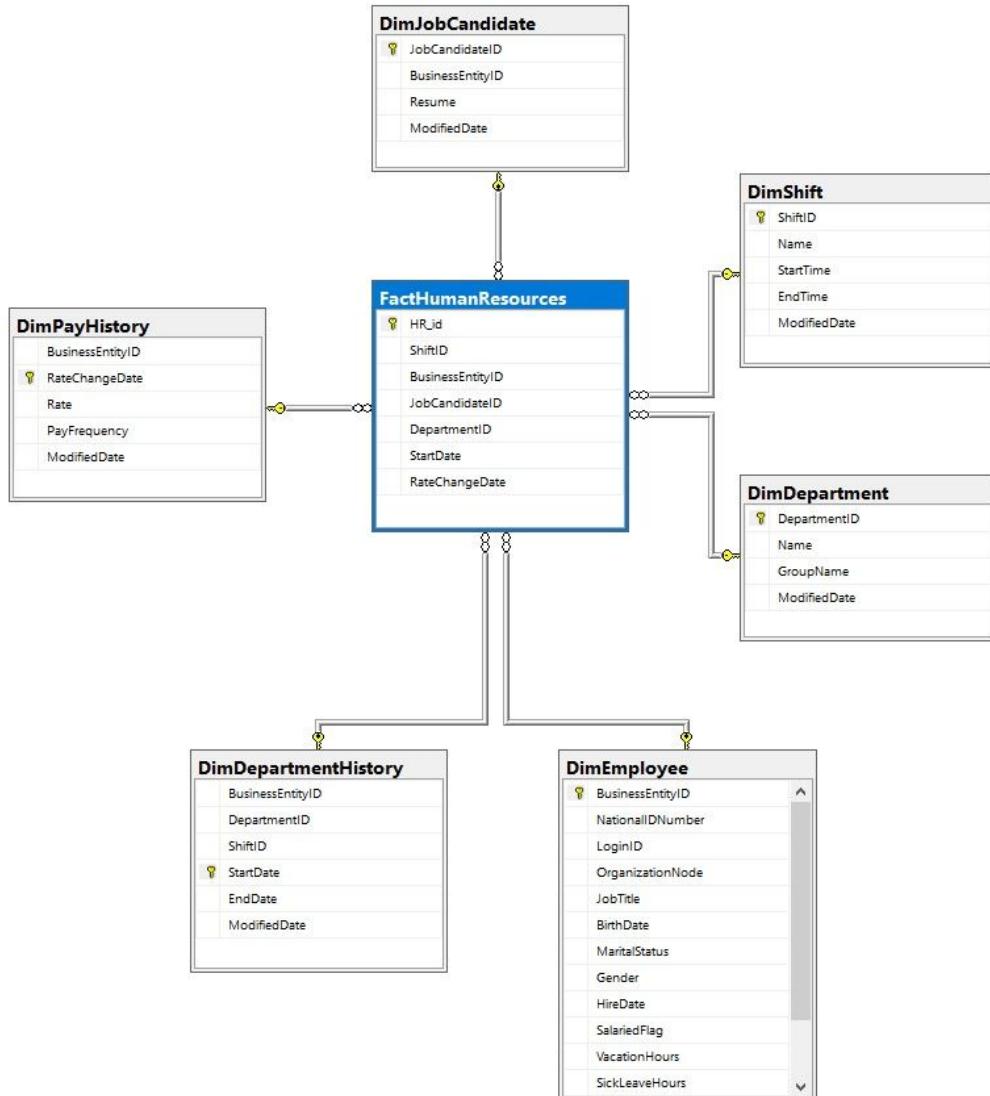
Human Resources - Chris

*Dimensional Fact Model*



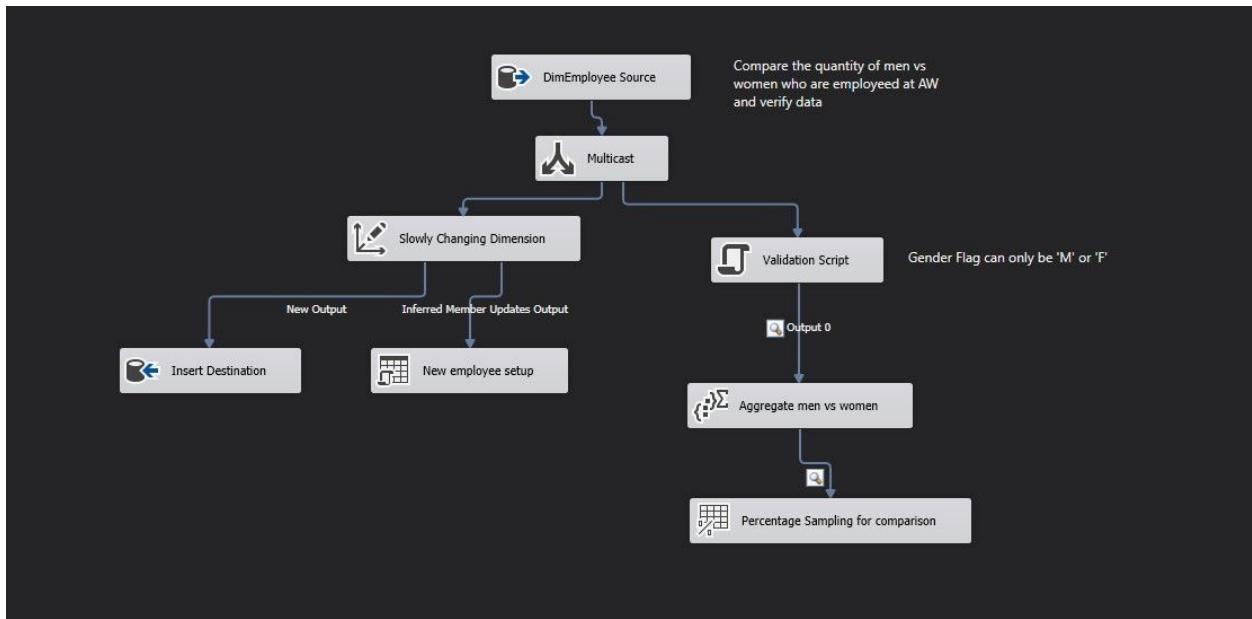
## Logical Model

### Production Data Mart

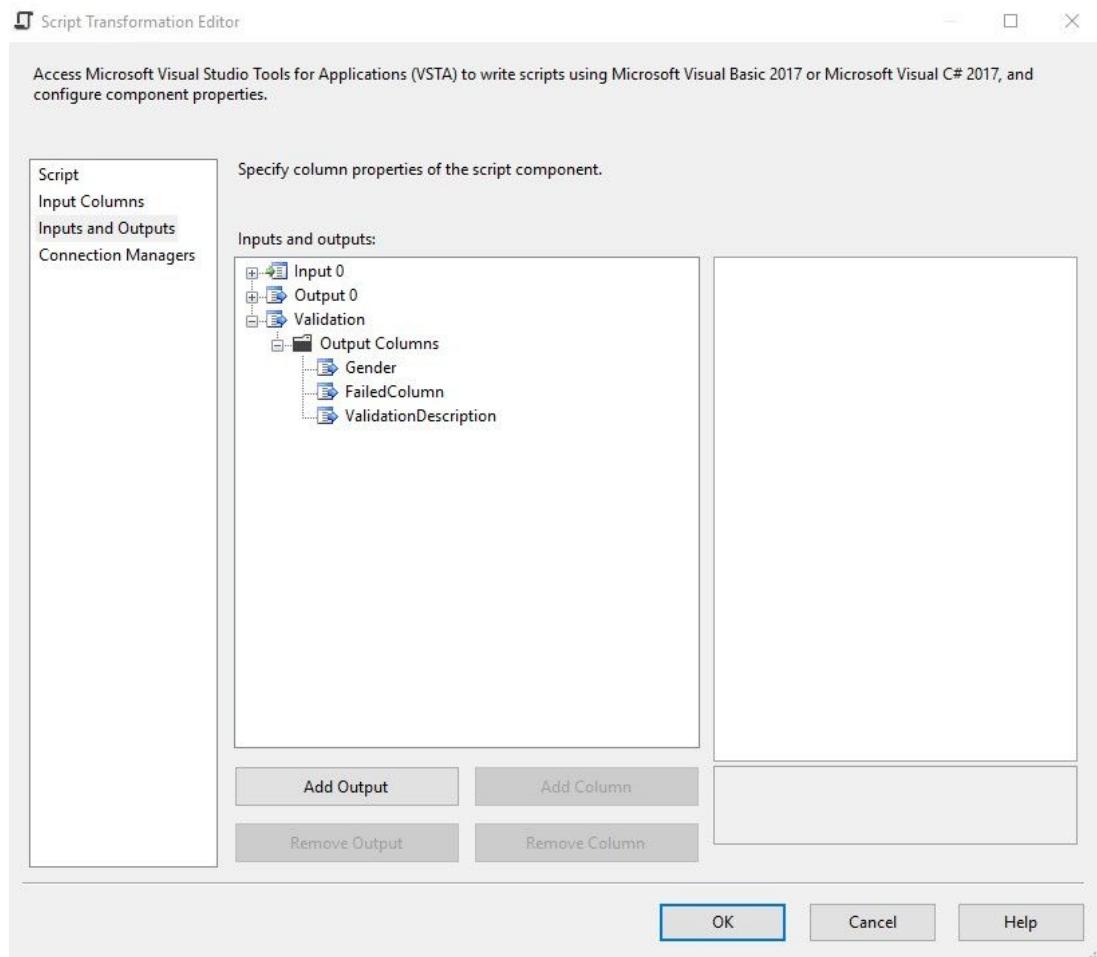


## ETL Process - Human Resources

- In this diagram, we show how we can use scripts to validate incoming data. This ensures us that the data we are using is safe, secure, and correct. The script constraints the Gender row to only include 'M' and 'F' entries, all other throw an error.



## Setting Gender as a validation row



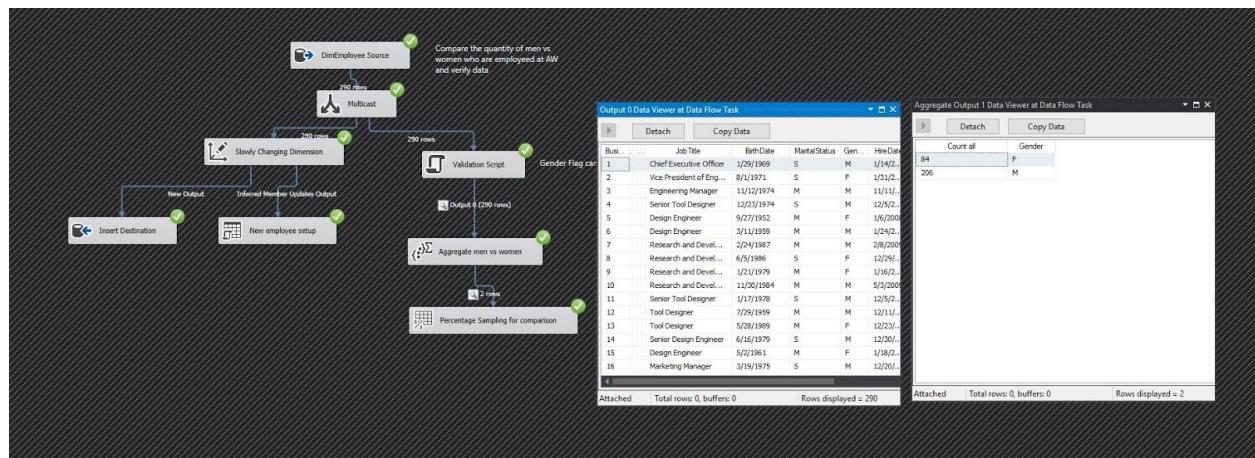
## Script

```

SC_d4589abdd61d4ed3a54ddeb00a565cb7 ScriptMain
1 Help: Introduction to the Script Component
2
3 Namespaces
4
5 //////////////////////////////////////////////////////////////////
6 // This is the class to which to add your code. Do not change the name, attributes, or parent
7 // of this class.
8 //////////////////////////////////////////////////////////////////
9 [Microsoft.SqlServer.Dts.Pipeline.SSIScriptComponentEntryPointAttribute]
10 public class ScriptMain : UserComponent
11 {
12     Help: Using Integration Services variables and parameters
13
14     Help: Using Integration Services Connection Managers
15
16     Help: Firing Integration Services Events
17
18     /// <summary>
19     /// This method is called once, before rows begin to be processed in the data flow.
20     ///
21     /// You can remove this method if you don't need to do anything here.
22     /// </summary>
23     public override void PreExecute()...
24
25     /// <summary>
26     /// This method is called after all the rows have passed through this component.
27     ///
28     /// You can delete this method if you don't need to do anything here.
29     /// </summary>
30     public override void PostExecute()...
31
32     /// <summary> This method is called once for every row that passes through the component.
33     public override void Input0_ProcessInputRow(Input0Buffer Row)
34     {
35         if (!string.Equals(Row.Gender, "F") || !string.Equals(Row.Gender, "M"))
36         {
37             ValidationBuffer.AddRow();
38             ValidationBuffer.ValidationDescription = "Only Male and Female are valid";
39             ValidationBuffer.FailedColumn = "Gender";
40
41             /*Map existing Columns*/
42             ValidationBuffer.Gender = Row.Gender;
43         }
44     }
45 }

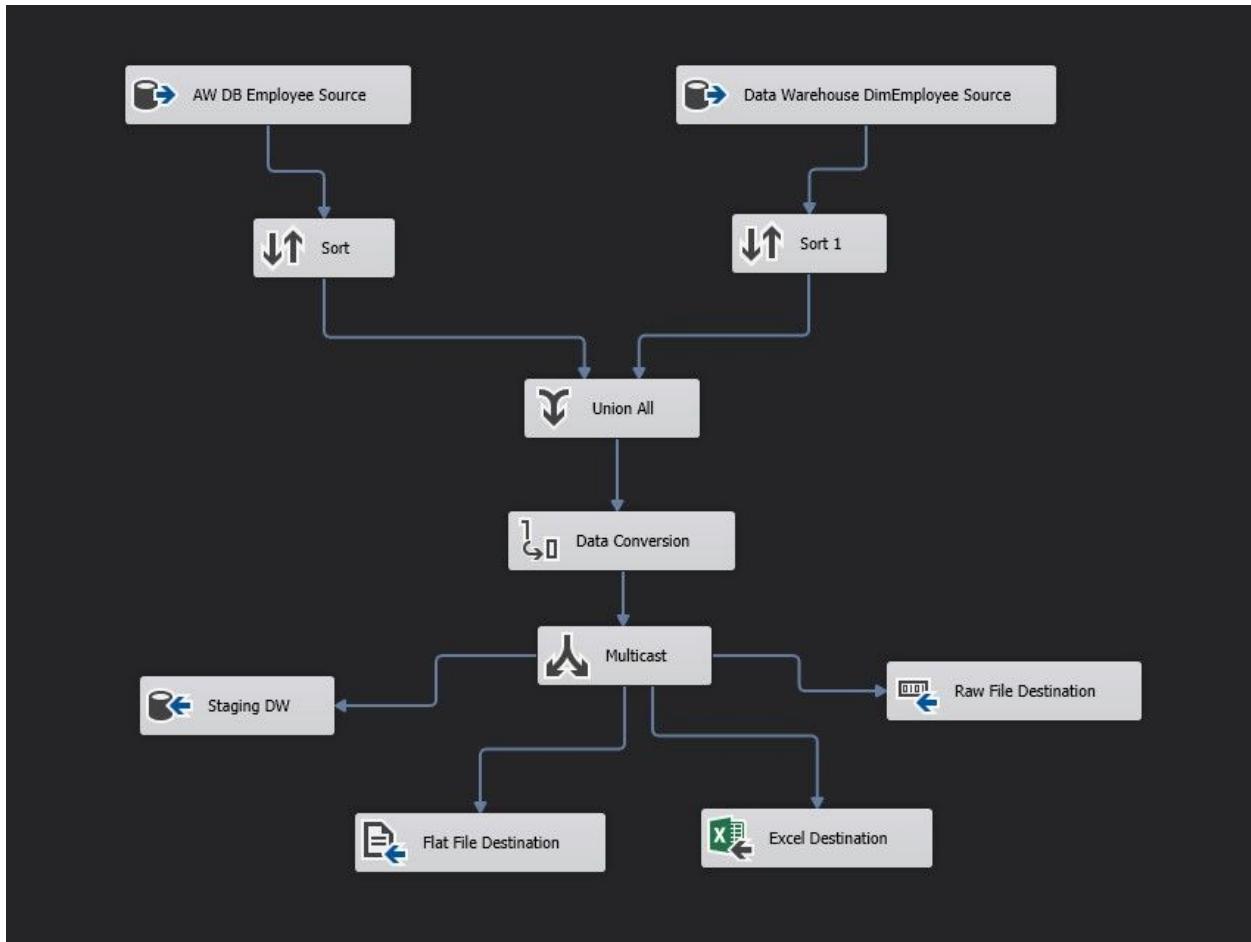
```

Data viewer shows us that the script worked successfully and was parsed with aggregate



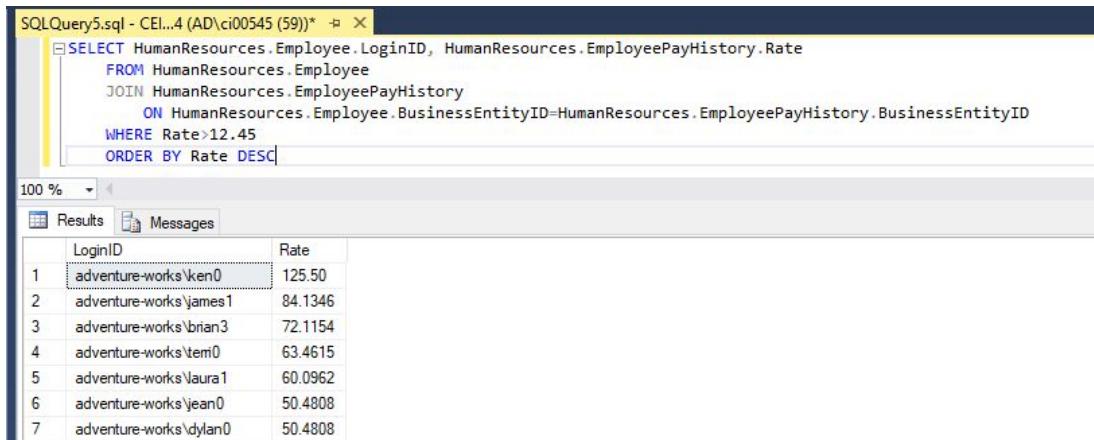
## ETL Process - Human Resources

- Shows the full extends of SSIS with exporting and loading data to destination sources



## SQL Queries - Human Resources

- Which Employees make more than \$12.45 an hour?



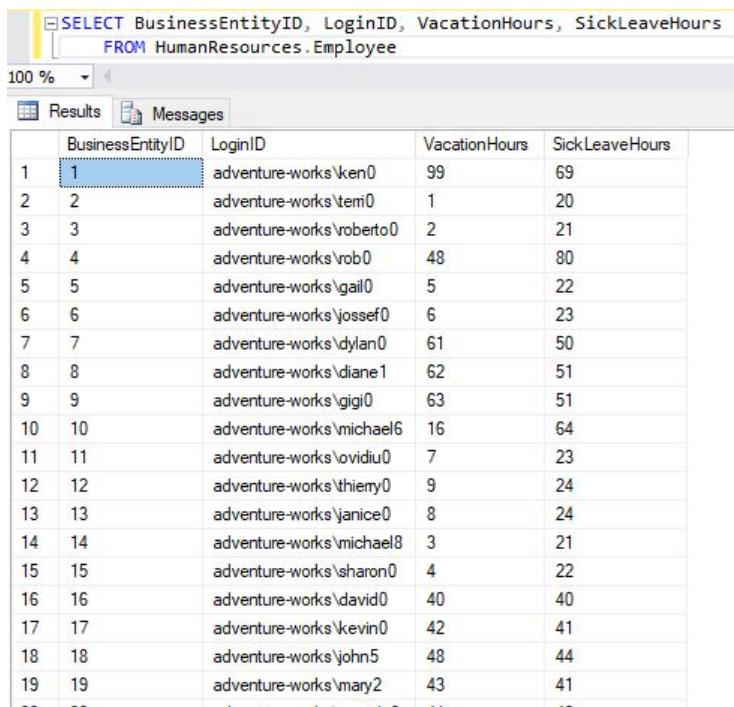
The screenshot shows a SQL Server Management Studio window titled "SQLQuery5.sql - CEI...4 (AD\ci00545 (59))". The query is:

```
SELECT HumanResources.Employee.LoginID, HumanResources.EmployeePayHistory.Rate
FROM HumanResources.Employee
JOIN HumanResources.EmployeePayHistory
    ON HumanResources.Employee.BusinessEntityID=HumanResources.EmployeePayHistory.BusinessEntityID
WHERE Rate>12.45
ORDER BY Rate DESC
```

The results grid displays the following data:

	LoginID	Rate
1	adventure-works\ken0	125.50
2	adventure-works\james1	84.1346
3	adventure-works\brian3	72.1154
4	adventure-works\tim0	63.4615
5	adventure-works\aura1	60.0962
6	adventure-works\jean0	50.4808
7	adventure-works\dylan0	50.4808

- Check how many vacation hours and sick leave hours an employee has left



The screenshot shows a SQL Server Management Studio window titled "SQLQuery5.sql - CEI...4 (AD\ci00545 (59))". The query is:

```
SELECT BusinessEntityID, LoginID, VacationHours, SickLeaveHours
FROM HumanResources.Employee
```

The results grid displays the following data:

	BusinessEntityID	LoginID	VacationHours	SickLeaveHours
1	1	adventure-works\ken0	99	69
2	2	adventure-works\tim0	1	20
3	3	adventure-works\roberto0	2	21
4	4	adventure-works\rob0	48	80
5	5	adventure-works\gail0	5	22
6	6	adventure-works\jossef0	6	23
7	7	adventure-works\dylan0	61	50
8	8	adventure-works\diane1	62	51
9	9	adventure-works\gigi0	63	51
10	10	adventure-works\michael6	16	64
11	11	adventure-works\ovidiu0	7	23
12	12	adventure-works\thierry0	9	24
13	13	adventure-works\janice0	8	24
14	14	adventure-works\michael8	3	21
15	15	adventure-works\sharon0	4	22
16	16	adventure-works\david0	40	40
17	17	adventure-works\kevin0	42	41
18	18	adventure-works\john5	48	44
19	19	adventure-works\mary2	43	41

- Select all Departments in a department Group

```

SELECT Name, GroupName
FROM HumanResources.Department
WHERE GroupName = 'Executive General and Administration'

```

.00 % ▾

Results Messages

	Name	GroupName
1	Human Resources	Executive General and Administration
2	Finance	Executive General and Administration
3	Information Services	Executive General and Administration
4	Facilities and Maintenance	Executive General and Administration
5	Executive	Executive General and Administration

#### 4. How many shifts does each employee currently have scheduled?

```

SELECT LoginID, COUNT(ShiftID) AS 'Amount of Shifts'
FROM HumanResources.EmployeeDepartmentHistory
JOIN HumanResources.Employee
    ON HumanResources.Employee.BusinessEntityID = HumanResources.EmployeeDepartmentHistory.BusinessEntityID
GROUP BY LoginID
ORDER BY 'Amount of Shifts' DESC

```

10 % ▾

Results Messages

LoginID	Amount of Shifts
adventure-works\sheela0	3
adventure-works\vob0	2
adventure-works\william0	2
adventure-works\david0	2
adventure-works\laura1	2
adventure-works\linda0	1
adventure-works\linda1	1
adventure-works\linda2	1
adventure-works\linda3	1
adventure-works\lionel0	1
adventure-works\volan0	1
adventure-works\vor0	1
adventure-works\vor1	1

#### 5. Which Departments have the most/least number of employees?

```

SELECT Department.DepartmentID, MAX(Department.Name) AS 'Name', COUNT(4) AS 'Number of Employees'
FROM HumanResources.Department
JOIN HumanResources.EmployeeDepartmentHistory ON Department.DepartmentID = EmployeeDepartmentHistory.DepartmentID
GROUP BY Department.DepartmentID
ORDER BY DepartmentID ASC

```

% ▾

Results Messages

DepartmentID	Name	Number of Employees
1	Engineering	7
2	Tool Design	4
3	Sales	18
4	Marketing	10
5	Purchasing	13
6	Research and Development	4
7	Production	180
8	Production Control	6
9	Human Resources	6
10	Finance	11
11	Information Services	10
12	Document Control	5
13	Quality Assurance	7
14	Facilities and Maintenance	7
15	Shipping and Receiving	6
16	Executive	2

## Purchasing - Wayne

### WIP

WIP Purchasing-Shawein Smith				
Who-Vendor D1-Vendor	What-Purchase Order Detail D2-PurchaseOrderDetail	What-Purchase Order Header D3-PurchaseOrderHeader	What-Ship Method D4-ShipMethod	What-Product Vendor D5-ProductVendor
<b>BusinessEntityID</b>	<b>PurchaseOrderID</b>	<b>PurchaseOrderID</b>	<b>ShipMethodID</b>	<b>BusinessEntityID</b>
AccountNumber	<b>PurchaseOrderDetailID</b>	<b>ShipMethodID</b>	Name	<b>ProductID</b>
Name	DueDate	VendorID	ShipBase	AverageLeadTime
CreditRating	OrderQty	RevisionNumber	ShipRate	StandardPrice
Preferred VendorStatus	ProductID	Status	rowguid	LastReceiptCost
ActiveFlag	UnitPrice	OrderDate	ModifiedDate	LastReceiptDate
PurchasingWebServiceURL	LineTotal	ShipDate		MinOrderQty
ModifiedDate	ReceivedQty	SubTotal		MaxOrderQty
	RejectedQty	TaxAmt		UnitMeasureCode
	StockedQty	Freight		ModifiedDate
	ModifiedDate	TotalDue		
		ModifiedDate		
<b>Observations/Measures:</b> Account Number-, CreditRating-, OrderQty-, UnitPrice-\$, RevisionNumber-, SubTotal-\$, TaxAmt-\$, StandardPrice-\$				
<b>Preliminary Workload</b>				
Question/Query	What is the average ship rate?			
Question/Query	How much was the largest order made?			
Question/Query	Sort the vendors by credit rating from largest to smallest			
Question/Query	What are the order statuses of orders shipped out on XX date?			
Question/Query	What are the URLs of all the vendors?			

### BEAMS

#### *ShipMethod Dimension*

ShipMethod Dimension - Shawein					
ShipMethod	HV				
ShipMethodID	has	has	has	updated on	
MD	<b>Name</b>	<b>ShipBase</b>	<b>ShipRate</b>	<b>ModifiedDate</b>	
1	XRQ - TRUCK GROUND		3.95	0.99	2008-04-30 0:00:00
2	ZY - EXPRESS		9.95	1.99	2008-04-30 0:00:00
3	OVERSEAS - DELUXE		29.95	2.99	2008-04-30 0:00:00
4	OVERNIGHT J-FAST		21.95	1.29	2008-04-30 0:00:00
5	CARGO TRANSPORT 5		8.99	1.49	2008-04-30 0:00:00

## Vendor Dimension

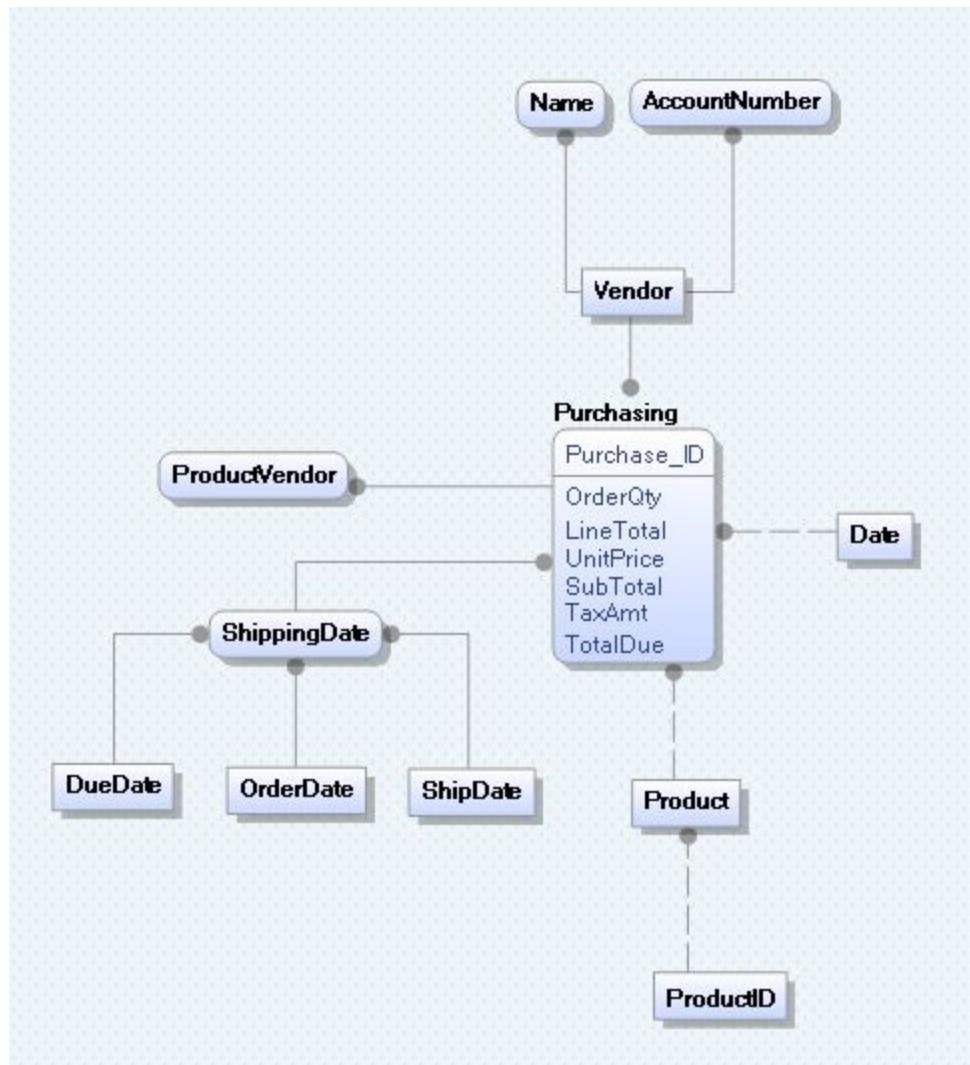
Vendor Dimension - Shawein								
Vendor	HV							
	has AccountNumber	has Name	has CreditRating	has PreferredVendorStatus	has ActiveFlag	has PurchasingWebServiceURL	updated on ModifiedDate	NN
BusinessEntityID	NN	NN	NN	NN	NN	NN	ModifiedDate	NN
MD								
1492	AUSTRALI0001	Australia Bike Retailer		1	1	1	2011-12-23 0:00:00	
1494	ALLENSON0001	Allenson Cycles		2	1	1	2011-04-25 0:00:00	
1496	ADVANCED0001	Advanced Bicycles		1	1	1	2011-04-25 0:00:00	
1498	TRIKE50001	Trikes, Inc.		2	1	1	2012-02-03 0:00:00	
1500	MORGANB0001	Morgan Bike Accessories		1	1	1	2012-02-02 0:00:00	
1502	CYCLING0001	Cycling Master		1	1	1	2011-12-24 0:00:00	

## Data Warehouse Design

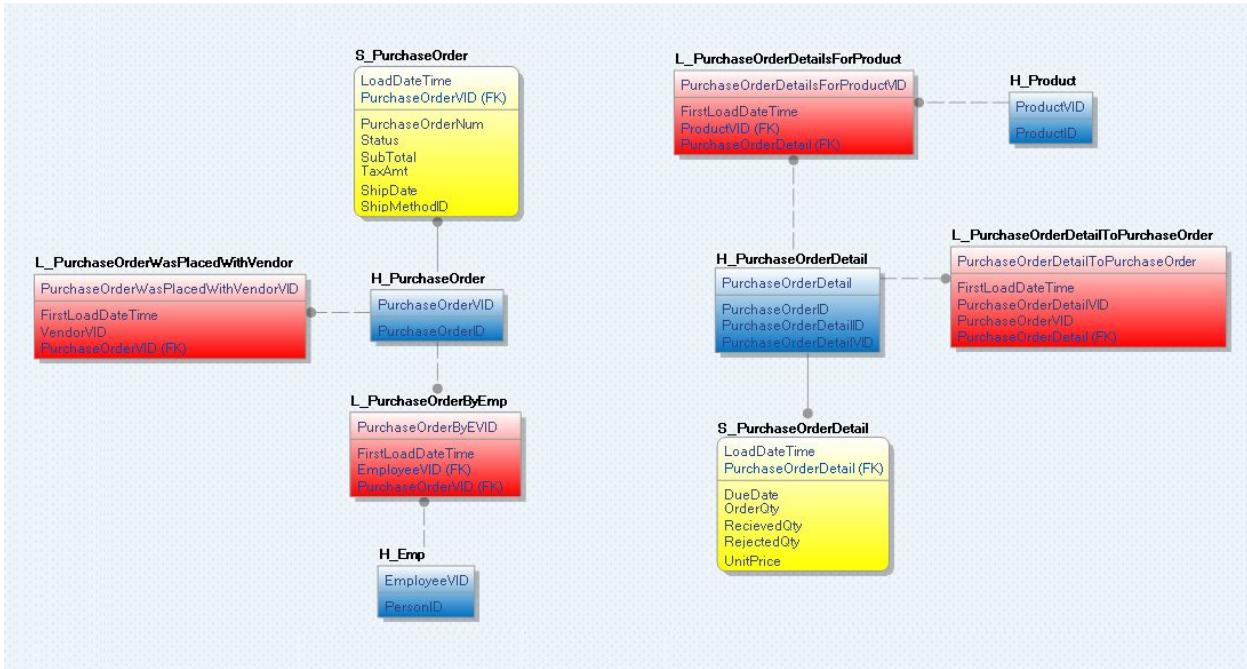
Conceptual Models

Purchasing - Wayne

Dimensional Fact Model

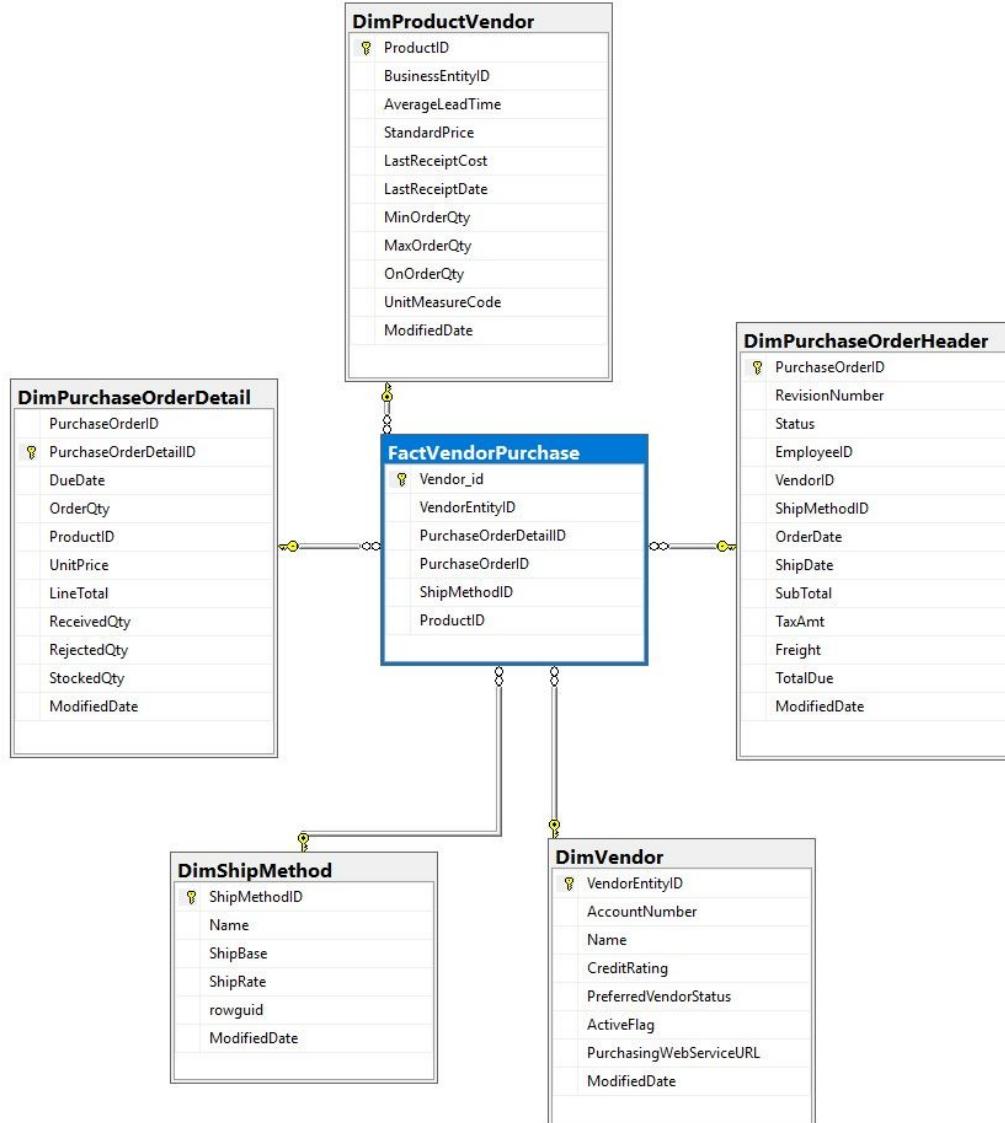


## vData Vault - Purchasing



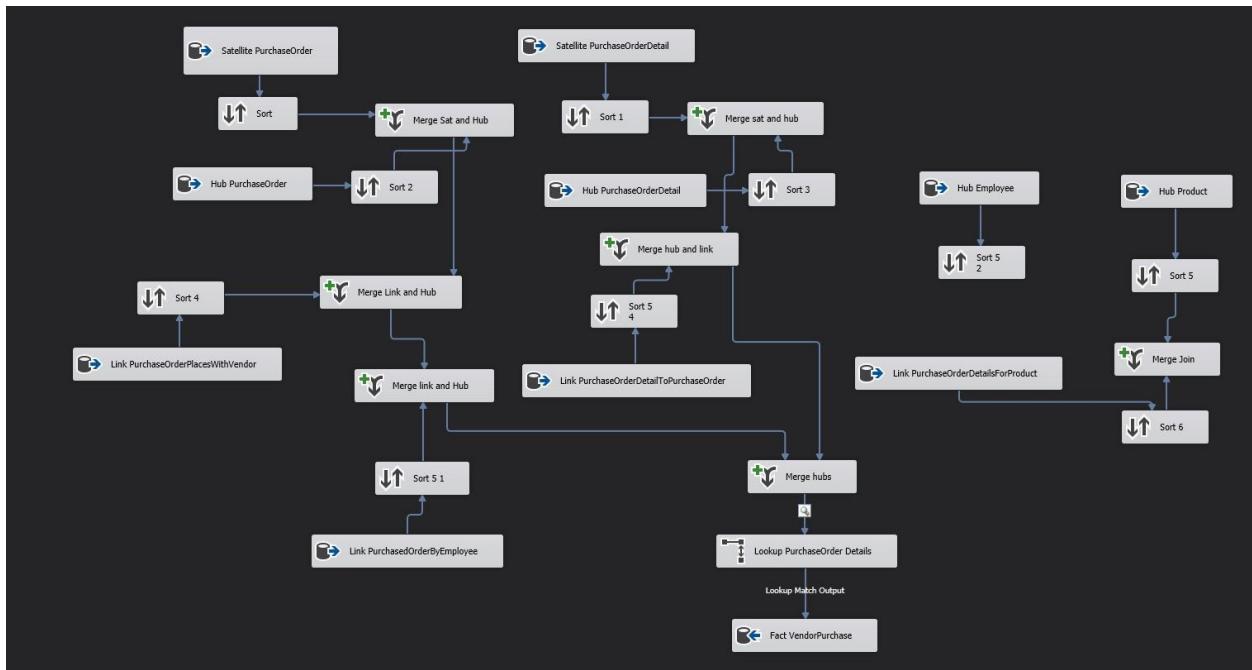
## Conceptual Model

### Purchasing Data Mart



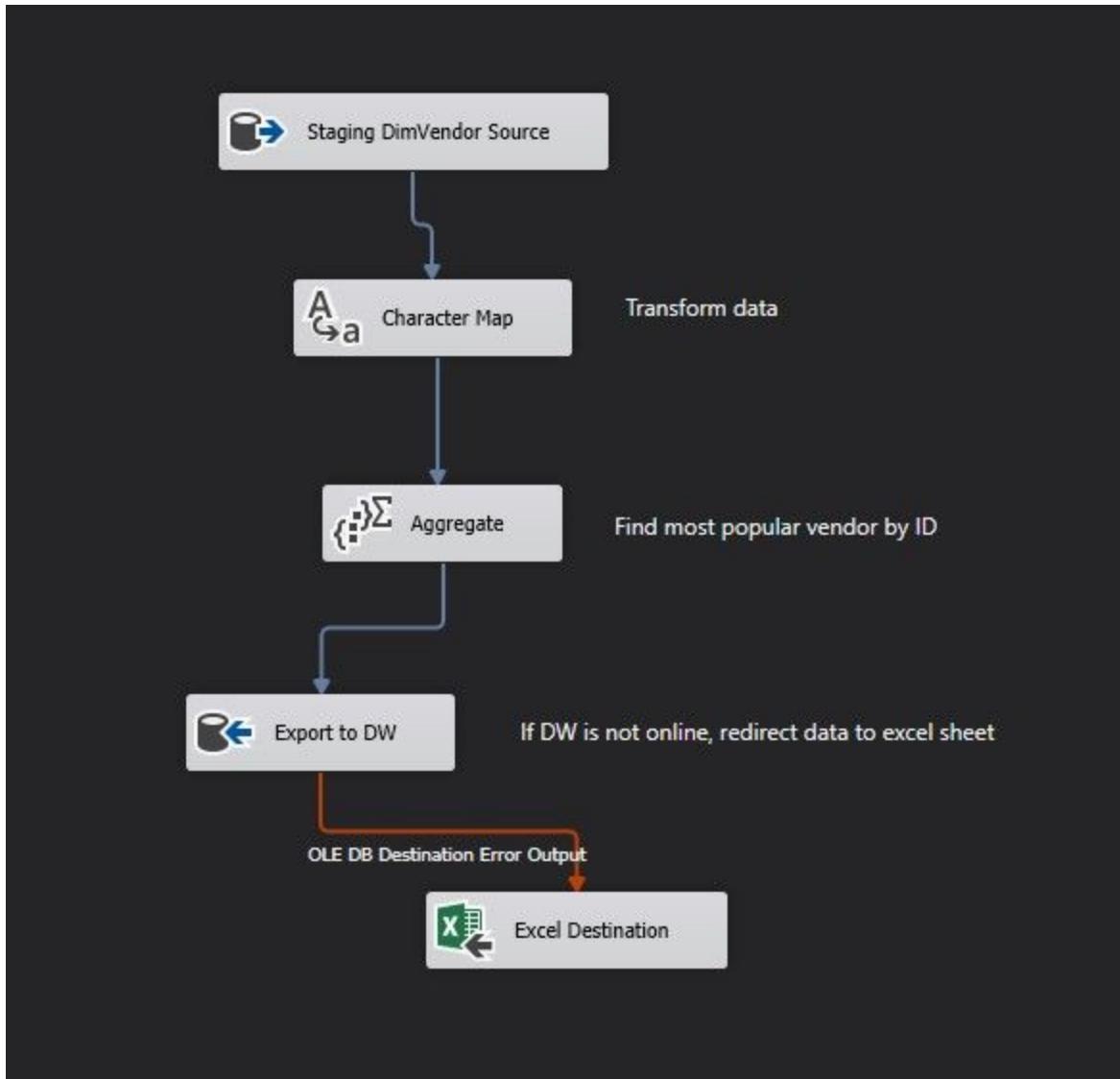
## ETL Process - Purchasing

1. This diagram shows how SSIS is capable of modeling the flow of a model



## ETL Process - Purchasing

- Shows the ETL process when the destination data source is not connected



## SQL Queries

## Closing Remarks

We have learned an exceptional amount about the inner workings of a data warehouse throughout the course of this semester. The WIP and BEAM tables helped us out an extreme amount when creating our data marts. The logical model was a blessing when it came to helping understand the constraints of our data warehouse. The biggest takeaway from the entire project has to be how powerful a tool like SSIS with Visual Studio is. SSIS holds the power to transform any data into any form you can possibly think of, package it up nicely, and send it anywhere you would like. The skills learned from SSIS can be applied to so many things. The data flow model helped us build the final underlying parts of our data warehouse to give it that final polish.

Being, in Georgia, the demand for data warehousing is skyrocketing currently. The amount of people who know how to run a data warehouse is dwarfed by the growing industry and demand for intelligent data warehousing. The only thing that we would have changed about our project was to add indexing and clusters within our dimensions to make our data warehouse perform better. Overall the performance was average, nothing amazing, but definitely enough to get the job done when it came to ETL. This project really brings the whole course together and adds another notch on the belt of database and data warehouse storage.

## DW Generate Script

```
USE [AdventureWorksDW]
GO
/***** Object: Table [dbo].[DimCreditCard]  Script Date: 4/27/2019 3:12:28 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimCreditCard](
    [CreditCardID] [int] IDENTITY(1,1) NOT NULL,
    [CardType] [nvarchar](50) NOT NULL,
    [CardNumber] [nvarchar](25) NOT NULL,
    [ExpMonth] [tinyint] NOT NULL,
    [ExpYear] [smallint] NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
    CONSTRAINT [PK_CreditCard_CreditCardID] PRIMARY KEY CLUSTERED
(
    [CreditCardID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

```
***** Object: Table [dbo].[DimCustomer]  Script Date: 4/27/2019 3:12:28 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimCustomer](
    [CustomerKey] [int] IDENTITY(1,1) NOT NULL,
    [GeographyKey] [int] NULL,
    [CustomerAlternateKey] [nvarchar](15) NOT NULL,
    [Title] [nvarchar](8) NULL,
    [FirstName] [nvarchar](50) NULL,
    [MiddleName] [nvarchar](50) NULL,
    [LastName] [nvarchar](50) NULL,
    [NameStyle] [bit] NULL,
    [BirthDate] [date] NULL,
    [MaritalStatus] [nchar](1) NULL,
    [Suffix] [nvarchar](10) NULL,
    [Gender] [nvarchar](1) NULL,
    [EmailAddress] [nvarchar](50) NULL,
    [YearlyIncome] [money] NULL,
    [TotalChildren] [tinyint] NULL,
    [NumberChildrenAtHome] [tinyint] NULL,
    [EnglishEducation] [nvarchar](40) NULL,
    [SpanishEducation] [nvarchar](40) NULL,
    [FrenchEducation] [nvarchar](40) NULL,
    [EnglishOccupation] [nvarchar](100) NULL,
    [SpanishOccupation] [nvarchar](100) NULL,
    [FrenchOccupation] [nvarchar](100) NULL,
    [HouseOwnerFlag] [nchar](1) NULL,
    [NumberCarsOwned] [tinyint] NULL,
    [AddressLine1] [nvarchar](120) NULL,
    [AddressLine2] [nvarchar](120) NULL,
    [Phone] [nvarchar](20) NULL,
    [DateFirstPurchase] [date] NULL,
    [CommuteDistance] [nvarchar](15) NULL,
CONSTRAINT [PK_DimCustomer_CustomerKey] PRIMARY KEY CLUSTERED
(
    [CustomerKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimDate]  Script Date: 4/27/2019 3:12:28 AM *****/
```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimDate](
    [DateKey] [int] NOT NULL,
    [FullDateAlternateKey] [date] NOT NULL,
    [DayNumberOfWeek] [tinyint] NOT NULL,
    [EnglishDayNameOfWeek] [nvarchar](10) NOT NULL,
    [SpanishDayNameOfWeek] [nvarchar](10) NOT NULL,
    [FrenchDayNameOfWeek] [nvarchar](10) NOT NULL,
    [DayNumberOfMonth] [tinyint] NOT NULL,
    [DayNumberOfYear] [smallint] NOT NULL,
    [WeekNumberOfYear] [tinyint] NOT NULL,
    [EnglishMonthName] [nvarchar](10) NOT NULL,
    [SpanishMonthName] [nvarchar](10) NOT NULL,
    [FrenchMonthName] [nvarchar](10) NOT NULL,
    [MonthNumberOfYear] [tinyint] NOT NULL,
    [CalendarQuarter] [tinyint] NOT NULL,
    [CalendarYear] [smallint] NOT NULL,
    [CalendarSemester] [tinyint] NOT NULL,
    [FiscalQuarter] [tinyint] NOT NULL,
    [FiscalYear] [smallint] NOT NULL,
    [FiscalSemester] [tinyint] NOT NULL,
CONSTRAINT [PK_DimDate_DateKey] PRIMARY KEY CLUSTERED
(
    [DateKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimDepartment]  Script Date: 4/27/2019 3:12:28 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimDepartment](
    [DepartmentID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [varchar](255) NOT NULL,
    [GroupName] [varchar](255) NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_Department_DepartmentID] PRIMARY KEY CLUSTERED
(

```

```

[DepartmentID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimDepartmentHistory]  Script Date: 4/27/2019 3:12:28 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimDepartmentHistory](
    [BusinessEntityID] [int] NOT NULL,
    [DepartmentID] [int] NOT NULL,
    [ShiftID] [int] NOT NULL,
    [StartDate] [date] NOT NULL,
    [EndDate] [date] NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_StartDate] PRIMARY KEY CLUSTERED
(
    [StartDate] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimEmployee]  Script Date: 4/27/2019 3:12:28 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimEmployee](
    [BusinessEntityID] [int] NOT NULL,
    [NationalIDNumber] [nvarchar](15) NOT NULL,
    [LoginID] [nvarchar](256) NOT NULL,
    [OrganizationNode] [int] NULL,
    [JobTitle] [nvarchar](50) NOT NULL,
    [BirthDate] [date] NOT NULL,
    [MaritalStatus] [nchar](1) NOT NULL,
    [Gender] [nchar](1) NOT NULL,
    [HireDate] [date] NOT NULL,
    [SalariedFlag] [int] NOT NULL,
    [VacationHours] [smallint] NOT NULL,
    [SickLeaveHours] [smallint] NOT NULL,
    [CurrentFlag] [int] NOT NULL,

```

```

[rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_Employee_BusinessEntityID] PRIMARY KEY CLUSTERED
(
    [BusinessEntityID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimJobCandidate]  Script Date: 4/27/2019 3:12:28 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimJobCandidate](
    [JobCandidateID] [int] IDENTITY(1,1) NOT NULL,
    [BusinessEntityID] [int] NULL,
    [Resume] [varchar](500) NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_JobCandidate_JobCandidateID] PRIMARY KEY CLUSTERED
(
    [JobCandidateID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimLocation]  Script Date: 4/27/2019 3:12:28 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimLocation](
    [LocationID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [varchar](255) NOT NULL,
    [CostRate] [smallmoney] NOT NULL,
    [Availability] [decimal](8, 2) NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_Location_LocationID] PRIMARY KEY CLUSTERED
(
    [LocationID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
***** Object: Table [dbo].[DimPayHistory]  Script Date: 4/27/2019 3:12:28 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimPayHistory](
    [BusinessEntityID] [int] NOT NULL,
    [RateChangeDate] [datetime] NOT NULL,
    [Rate] [money] NOT NULL,
    [PayFrequency] [tinyint] NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_EmployeePayHistory_RateChangeDate] PRIMARY KEY CLUSTERED
(
    [RateChangeDate] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimProduct]  Script Date: 4/27/2019 3:12:28 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimProduct](
    [ProductKey] [int] IDENTITY(1,1) NOT NULL,
    [ProductAlternateKey] [nvarchar](25) NULL,
    [ProductSubcategoryKey] [int] NULL,
    [WeightUnitMeasureCode] [nchar](3) NULL,
    [SizeUnitMeasureCode] [nchar](3) NULL,
    [EnglishProductName] [nvarchar](50) NOT NULL,
    [SpanishProductName] [nvarchar](50) NOT NULL,
    [FrenchProductName] [nvarchar](50) NOT NULL,
    [StandardCost] [money] NULL,
    [FinishedGoodsFlag] [bit] NOT NULL,
    [Color] [nvarchar](15) NOT NULL,
    [SafetyStockLevel] [smallint] NULL,
    [ReorderPoint] [smallint] NULL,
    [ListPrice] [money] NULL,
    [Size] [nvarchar](50) NULL,
    [SizeRange] [nvarchar](50) NULL,
    [Weight] [float] NULL,
    [DaysToManufacture] [int] NULL,

```

```

[ProductLine] [nchar](2) NULL,
[DealerPrice] [money] NULL,
[Class] [nchar](2) NULL,
[Style] [nchar](2) NULL,
[ModelName] [nvarchar](50) NULL,
[LargePhoto] [varbinary](max) NULL,
[EnglishDescription] [nvarchar](400) NULL,
[FrenchDescription] [nvarchar](400) NULL,
[ChineseDescription] [nvarchar](400) NULL,
[ArabicDescription] [nvarchar](400) NULL,
[HebrewDescription] [nvarchar](400) NULL,
[ThaiDescription] [nvarchar](400) NULL,
[GermanDescription] [nvarchar](400) NULL,
[JapaneseDescription] [nvarchar](400) NULL,
[TurkishDescription] [nvarchar](400) NULL,
[StartDate] [datetime] NULL,
[EndDate] [datetime] NULL,
[Status] [nvarchar](7) NULL,
CONSTRAINT [PK_DimProduct_ProductKey] PRIMARY KEY CLUSTERED
(
    [ProductKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
CONSTRAINT [AK_DimProduct_ProductAlternateKey_StartDate] UNIQUE NONCLUSTERED
(
    [ProductAlternateKey] ASC,
    [StartDate] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
***** Object: Table [dbo].[DimProductCategory]  Script Date: 4/27/2019 3:12:28 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimProductCategory](
    [ProductCategoryKey] [int] IDENTITY(1,1) NOT NULL,
    [ProductCategoryAlternateKey] [int] NULL,
    [EnglishProductCategoryName] [nvarchar](50) NOT NULL,
    [SpanishProductCategoryName] [nvarchar](50) NOT NULL,
    [FrenchProductCategoryName] [nvarchar](50) NOT NULL,
CONSTRAINT [PK_DimProductCategory_ProductCategoryKey] PRIMARY KEY CLUSTERED

```

```

(
    [ProductCategoryKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
CONSTRAINT [AK_DimProductCategory_ProductCategoryAlternateKey] UNIQUE
NONCLUSTERED
(
    [ProductCategoryAlternateKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimProductDescription]  Script Date: 4/27/2019 3:12:28 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimProductDescription](
    [ProductDescriptionID] [int] IDENTITY(1,1) NOT NULL,
    [Description] [nvarchar](400) NOT NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_ProductDescription_ProductDescriptionID] PRIMARY KEY CLUSTERED
(
    [ProductDescriptionID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimProductModel]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimProductModel](
    [ProductModelID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [varchar](255) NOT NULL,
    [CatalogDescription] [varchar](500) NULL,
    [Instructions] [varchar](500) NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_ProductModel_ProductModelID] PRIMARY KEY CLUSTERED
(

```

```

[ProductModelID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimProductPhoto] Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimProductPhoto](
    [ProductPhotoID] [int] IDENTITY(1,1) NOT NULL,
    [ThumbNailPhoto] [varbinary](max) NULL,
    [ThumbnailPhotoFileName] [nvarchar](50) NULL,
    [LargePhoto] [varbinary](max) NULL,
    [LargePhotoFileName] [nvarchar](50) NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_ProductPhoto_ProductPhotoID] PRIMARY KEY CLUSTERED
(
    [ProductPhotoID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
***** Object: Table [dbo].[DimProductSubCategory] Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimProductSubCategory](
    [ProductSubcategoryID] [int] IDENTITY(1,1) NOT NULL,
    [ProductCategoryID] [int] NOT NULL,
    [Name] [varchar](255) NOT NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_ProductSubcategory_ProductSubcategoryID] PRIMARY KEY CLUSTERED
(
    [ProductSubcategoryID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimProductVendor] Script Date: 4/27/2019 3:12:29 AM *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimProductVendor](
    [ProductID] [int] NOT NULL,
    [BusinessEntityID] [int] NOT NULL,
    [AverageLeadTime] [int] NOT NULL,
    [StandardPrice] [money] NOT NULL,
    [LastReceiptCost] [money] NULL,
    [LastReceiptDate] [datetime] NULL,
    [MinOrderQty] [int] NOT NULL,
    [MaxOrderQty] [int] NOT NULL,
    [OnOrderQty] [int] NULL,
    [UnitMeasureCode] [nchar](3) NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_ProductVendor_ProductID] PRIMARY KEY CLUSTERED
(
    [ProductID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimPurchaseOrderDetail]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimPurchaseOrderDetail](
    [PurchaseOrderID] [int] NOT NULL,
    [PurchaseOrderDetailID] [int] IDENTITY(1,1) NOT NULL,
    [DueDate] [datetime] NOT NULL,
    [OrderQty] [smallint] NOT NULL,
    [ProductID] [int] NOT NULL,
    [UnitPrice] [money] NOT NULL,
    [LineTotal] AS (isnull([OrderQty]*[UnitPrice],(0.00))),
    [ReceivedQty] [decimal](8, 2) NOT NULL,
    [RejectedQty] [decimal](8, 2) NOT NULL,
    [StockedQty] AS (isnull([ReceivedQty]-[RejectedQty],(0.00))),
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_PurchaseOrderDetail_PurchaseOrderDetailID] PRIMARY KEY CLUSTERED
(
    [PurchaseOrderDetailID] ASC
)

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimPurchaseOrderHeader]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimPurchaseOrderHeader](
    [PurchaseOrderID] [int] IDENTITY(1,1) NOT NULL,
    [RevisionNumber] [tinyint] NOT NULL,
    [Status] [tinyint] NOT NULL,
    [EmployeeID] [int] NOT NULL,
    [VendorID] [int] NOT NULL,
    [ShipMethodID] [int] NOT NULL,
    [OrderDate] [datetime] NOT NULL,
    [ShipDate] [datetime] NULL,
    [SubTotal] [money] NOT NULL,
    [TaxAmt] [money] NOT NULL,
    [Freight] [money] NOT NULL,
    [TotalDue] AS (isnull(([SubTotal]+[TaxAmt])+[Freight],(0))) PERSISTED NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_PurchaseOrderHeader_PurchaseOrderID] PRIMARY KEY CLUSTERED
(
    [PurchaseOrderID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimSalesPerson]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimSalesPerson](
    [SalesPerson_id] [int] IDENTITY(1,1) NOT NULL,
    [FirstName] [varchar](255) NULL,
    [LastName] [varchar](255) NULL,
    [JobTitle] [varchar](255) NULL,
    [gender] [varchar](255) NULL,
    [HiringDate] [int] NULL,
    [BirthDate] [int] NULL,

```

```

    [SalesPersonKey] [int] NULL,
PRIMARY KEY CLUSTERED
(
    [SalesPerson_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[DimSalesTerritory]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimSalesTerritory](
    [SalesTerritoryKey] [int] IDENTITY(1,1) NOT NULL,
    [SalesTerritoryAlternateKey] [int] NULL,
    [SalesTerritoryRegion] [nvarchar](50) NOT NULL,
    [SalesTerritoryCountry] [nvarchar](50) NOT NULL,
    [SalesTerritoryGroup] [nvarchar](50) NULL,
    [SalesTerritoryImage] [varbinary](max) NULL,
CONSTRAINT [PK_DimSalesTerritory_SalesTerritoryKey] PRIMARY KEY CLUSTERED
(
    [SalesTerritoryKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY],
CONSTRAINT [AK_DimSalesTerritory_SalesTerritoryAlternateKey] UNIQUE NONCLUSTERED
(
    [SalesTerritoryAlternateKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/***** Object: Table [dbo].[DimShift]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimShift](
    [ShiftID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [varchar](255) NOT NULL,
    [StartTime] [time](7) NOT NULL,
    [EndTime] [time](7) NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,

```

```

CONSTRAINT [PK_Shift_ShiftID] PRIMARY KEY CLUSTERED
(
    [ShiftID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimShipMethod]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimShipMethod](
    [ShipMethodID] [int] IDENTITY(1,1) NOT NULL,
    [Name] [varchar](255) NOT NULL,
    [ShipBase] [money] NOT NULL,
    [ShipRate] [money] NOT NULL,
    [rowguid] [uniqueidentifier] ROWGUIDCOL NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_ShipMethod_ShipMethodID] PRIMARY KEY CLUSTERED
(
    [ShipMethodID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimTransactionHistory]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimTransactionHistory](
    [TransactionID] [int] IDENTITY(100000,1) NOT NULL,
    [ProductID] [int] NOT NULL,
    [ReferenceOrderID] [int] NOT NULL,
    [ReferenceOrderLineID] [int] NOT NULL,
    [TransactionDate] [datetime] NOT NULL,
    [TransactionType] [nchar](1) NOT NULL,
    [Quantity] [int] NOT NULL,
    [ActualCost] [money] NOT NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_TransactionHistory_TransactionID] PRIMARY KEY CLUSTERED
(

```

```

[TransactionID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[DimVendor]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DimVendor](
    [VendorEntityID] [int] NOT NULL,
    [AccountNumber] [int] NOT NULL,
    [Name] [varchar](255) NOT NULL,
    [CreditRating] [tinyint] NOT NULL,
    [PreferredVendorStatus] [varchar](255) NOT NULL,
    [ActiveFlag] [varchar](255) NOT NULL,
    [PurchasingWebServiceURL] [nvarchar](1024) NULL,
    [ModifiedDate] [datetime] NOT NULL,
CONSTRAINT [PK_Vendor_VendorEntityID] PRIMARY KEY CLUSTERED
(
    [VendorEntityID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[FactHumanResources]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[FactHumanResources](
    [HR_id] [int] IDENTITY(1,1) NOT NULL,
    [ShiftID] [int] NULL,
    [BusinessEntityID] [int] NULL,
    [JobCandidateID] [int] NULL,
    [DepartmentID] [int] NULL,
    [StartDate] [date] NULL,
    [RateChangeDate] [datetime] NULL,
PRIMARY KEY CLUSTERED
(
    [HR_id] ASC
)

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[FactInventory]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[FactInventory](
    [Inventory_id] [int] IDENTITY(1,1) NOT NULL,
    [ProductKey] [int] NULL,
    [ProductSubcategoryID] [int] NULL,
    [ProductDescriptionID] [int] NULL,
    [LocationID] [int] NULL,
    [TransactionID] [int] NULL,
    [ProductCategoryKey] [int] NULL,
    [ProductPhotoID] [int] NULL,
    [ProductModelID] [int] NULL,
PRIMARY KEY CLUSTERED
(
    [Inventory_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
***** Object: Table [dbo].[FactSalesOrderHeader]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[FactSalesOrderHeader](
    [SalesOrderHeader_id] [int] IDENTITY(1,1) NOT NULL,
    [SalesTerritoryKey] [int] NULL,
    [ProductCategoryKey] [int] NULL,
    [ProductKey] [int] NULL,
    [salesPerson_id] [int] NULL,
    [DateKey] [int] NULL,
    [CreditCardID] [int] NULL,
    [CustomerKey] [int] NULL,
PRIMARY KEY CLUSTERED
(
    [SalesOrderHeader_id] ASC
)

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[FactVendorPurchase]  Script Date: 4/27/2019 3:12:29 AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[FactVendorPurchase](
    [Vendor_id] [int] IDENTITY(1,1) NOT NULL,
    [VendorEntityID] [int] NULL,
    [PurchaseOrderDetailID] [int] NULL,
    [PurchaseOrderID] [int] NULL,
    [ShipMethodID] [int] NULL,
    [ProductID] [int] NULL,
PRIMARY KEY CLUSTERED
(
    [Vendor_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[FactHumanResources] WITH CHECK ADD CONSTRAINT
[fk_BusinessEntityID] FOREIGN KEY([BusinessEntityID])
REFERENCES [dbo].[DimEmployee] ([BusinessEntityID])
GO
ALTER TABLE [dbo].[FactHumanResources] CHECK CONSTRAINT [fk_BusinessEntityID]
GO
ALTER TABLE [dbo].[FactHumanResources] WITH CHECK ADD CONSTRAINT
[fk_DepartmentID] FOREIGN KEY([DepartmentID])
REFERENCES [dbo].[DimDepartment] ([DepartmentID])
GO
ALTER TABLE [dbo].[FactHumanResources] CHECK CONSTRAINT [fk_DepartmentID]
GO
ALTER TABLE [dbo].[FactHumanResources] WITH CHECK ADD CONSTRAINT
[fk_JobCandidateID] FOREIGN KEY([JobCandidateID])
REFERENCES [dbo].[DimJobCandidate] ([JobCandidateID])
GO
ALTER TABLE [dbo].[FactHumanResources] CHECK CONSTRAINT [fk_JobCandidateID]
GO
ALTER TABLE [dbo].[FactHumanResources] WITH CHECK ADD CONSTRAINT
[fk_RateChangeDate] FOREIGN KEY([RateChangeDate])

```

```
REFERENCES [dbo].[DimPayHistory] ([RateChangeDate])
GO
ALTER TABLE [dbo].[FactHumanResources] CHECK CONSTRAINT [fk_RateChangeDate]
GO
ALTER TABLE [dbo].[FactHumanResources] WITH CHECK ADD CONSTRAINT [fk_ShiftID]
FOREIGN KEY([ShiftID])
REFERENCES [dbo].[DimShift] ([ShiftID])
GO
ALTER TABLE [dbo].[FactHumanResources] CHECK CONSTRAINT [fk_ShiftID]
GO
ALTER TABLE [dbo].[FactHumanResources] WITH CHECK ADD CONSTRAINT [fk_StartDate]
FOREIGN KEY([StartDate])
REFERENCES [dbo].[DimDepartmentHistory] ([StartDate])
GO
ALTER TABLE [dbo].[FactHumanResources] CHECK CONSTRAINT [fk_StartDate]
GO
ALTER TABLE [dbo].[FactInventory] WITH CHECK ADD CONSTRAINT [fk_LocationID]
FOREIGN KEY([LocationID])
REFERENCES [dbo].[DimLocation] ([LocationID])
GO
ALTER TABLE [dbo].[FactInventory] CHECK CONSTRAINT [fk_LocationID]
GO
ALTER TABLE [dbo].[FactInventory] WITH CHECK ADD CONSTRAINT [fk_ProductCategoryKey]
FOREIGN KEY([ProductCategoryKey])
REFERENCES [dbo].[DimProductCategory] ([ProductCategoryKey])
GO
ALTER TABLE [dbo].[FactInventory] CHECK CONSTRAINT [fk_ProductCategoryKey]
GO
ALTER TABLE [dbo].[FactInventory] WITH CHECK ADD CONSTRAINT
[fk_ProductDescriptionID] FOREIGN KEY([ProductDescriptionID])
REFERENCES [dbo].[DimProductDescription] ([ProductDescriptionID])
GO
ALTER TABLE [dbo].[FactInventory] CHECK CONSTRAINT [fk_ProductDescriptionID]
GO
ALTER TABLE [dbo].[FactInventory] WITH CHECK ADD CONSTRAINT [fk_ProductKey]
FOREIGN KEY([ProductKey])
REFERENCES [dbo].[DimProduct] ([ProductKey])
GO
ALTER TABLE [dbo].[FactInventory] CHECK CONSTRAINT [fk_ProductKey]
GO
ALTER TABLE [dbo].[FactInventory] WITH CHECK ADD CONSTRAINT [fk_ProductModelID]
FOREIGN KEY([ProductModelID])
REFERENCES [dbo].[DimProductModel] ([ProductModelID])
```

GO  
ALTER TABLE [dbo].[FactInventory] CHECK CONSTRAINT [fk\_ProductModelID]  
GO  
ALTER TABLE [dbo].[FactInventory] WITH CHECK ADD CONSTRAINT [fk\_ProductPhotoID]  
FOREIGN KEY([ProductPhotoID])  
REFERENCES [dbo].[DimProductPhoto] ([ProductPhotoID])  
GO  
ALTER TABLE [dbo].[FactInventory] CHECK CONSTRAINT [fk\_ProductPhotoID]  
GO  
ALTER TABLE [dbo].[FactInventory] WITH CHECK ADD CONSTRAINT  
[fk\_ProductSubcategoryID] FOREIGN KEY([ProductSubcategoryID])  
REFERENCES [dbo].[DimProductSubCategory] ([ProductSubcategoryID])  
GO  
ALTER TABLE [dbo].[FactInventory] CHECK CONSTRAINT [fk\_ProductSubcategoryID]  
GO  
ALTER TABLE [dbo].[FactInventory] WITH CHECK ADD CONSTRAINT [fk\_TransactionID]  
FOREIGN KEY([TransactionID])  
REFERENCES [dbo].[DimTransactionHistory] ([TransactionID])  
GO  
ALTER TABLE [dbo].[FactInventory] CHECK CONSTRAINT [fk\_TransactionID]  
GO  
ALTER TABLE [dbo].[FactSalesOrderHeader] WITH CHECK ADD CONSTRAINT [fk\_CreditCard]  
FOREIGN KEY([CreditCardID])  
REFERENCES [dbo].[DimCreditCard] ([CreditCardID])  
GO  
ALTER TABLE [dbo].[FactSalesOrderHeader] CHECK CONSTRAINT [fk\_CreditCard]  
GO  
ALTER TABLE [dbo].[FactSalesOrderHeader] WITH CHECK ADD CONSTRAINT [fk\_Customer]  
FOREIGN KEY([CustomerKey])  
REFERENCES [dbo].[DimCustomer] ([CustomerKey])  
GO  
ALTER TABLE [dbo].[FactSalesOrderHeader] CHECK CONSTRAINT [fk\_Customer]  
GO  
ALTER TABLE [dbo].[FactSalesOrderHeader] WITH CHECK ADD CONSTRAINT [fk\_Date]  
FOREIGN KEY([DateKey])  
REFERENCES [dbo].[DimDate] ([DateKey])  
GO  
ALTER TABLE [dbo].[FactSalesOrderHeader] CHECK CONSTRAINT [fk\_Date]  
GO  
ALTER TABLE [dbo].[FactSalesOrderHeader] WITH CHECK ADD CONSTRAINT [fk\_Product]  
FOREIGN KEY([ProductKey])  
REFERENCES [dbo].[DimProduct] ([ProductKey])  
GO

```
ALTER TABLE [dbo].[FactSalesOrderHeader] CHECK CONSTRAINT [fk_Product]
GO
ALTER TABLE [dbo].[FactSalesOrderHeader] WITH CHECK ADD CONSTRAINT
[fk_ProductCategory] FOREIGN KEY([ProductCategoryKey])
REFERENCES [dbo].[DimProductCategory] ([ProductCategoryKey])
GO
ALTER TABLE [dbo].[FactSalesOrderHeader] CHECK CONSTRAINT [fk_ProductCategory]
GO
ALTER TABLE [dbo].[FactSalesOrderHeader] WITH CHECK ADD CONSTRAINT
[fk_SalesPerson_id] FOREIGN KEY([salesPerson_id])
REFERENCES [dbo].[DimSalesPerson] ([SalesPerson_id])
GO
ALTER TABLE [dbo].[FactSalesOrderHeader] CHECK CONSTRAINT [fk_SalesPerson_id]
GO
ALTER TABLE [dbo].[FactSalesOrderHeader] WITH CHECK ADD CONSTRAINT [fk_TerritoryID]
FOREIGN KEY([SalesTerritoryKey])
REFERENCES [dbo].[DimSalesTerritory] ([SalesTerritoryKey])
GO
ALTER TABLE [dbo].[FactSalesOrderHeader] CHECK CONSTRAINT [fk_TerritoryID]
GO
ALTER TABLE [dbo].[FactVendorPurchase] WITH CHECK ADD CONSTRAINT [fk_ProductID]
FOREIGN KEY([ProductID])
REFERENCES [dbo].[DimProductVendor] ([ProductID])
GO
ALTER TABLE [dbo].[FactVendorPurchase] CHECK CONSTRAINT [fk_ProductID]
GO
ALTER TABLE [dbo].[FactVendorPurchase] WITH CHECK ADD CONSTRAINT
[fk_PurchaseOrderDetailID] FOREIGN KEY([PurchaseOrderDetailID])
REFERENCES [dbo].[DimPurchaseOrderDetail] ([PurchaseOrderDetailID])
GO
ALTER TABLE [dbo].[FactVendorPurchase] CHECK CONSTRAINT [fk_PurchaseOrderDetailID]
GO
ALTER TABLE [dbo].[FactVendorPurchase] WITH CHECK ADD CONSTRAINT
[fk_PurchaseOrderID] FOREIGN KEY([PurchaseOrderID])
REFERENCES [dbo].[DimPurchaseOrderHeader] ([PurchaseOrderID])
GO
ALTER TABLE [dbo].[FactVendorPurchase] CHECK CONSTRAINT [fk_PurchaseOrderID]
GO
ALTER TABLE [dbo].[FactVendorPurchase] WITH CHECK ADD CONSTRAINT
[fk_ShipMethodID] FOREIGN KEY([ShipMethodID])
REFERENCES [dbo].[DimShipMethod] ([ShipMethodID])
GO
ALTER TABLE [dbo].[FactVendorPurchase] CHECK CONSTRAINT [fk_ShipMethodID]
```

GO

```
ALTER TABLE [dbo].[FactVendorPurchase] WITH CHECK ADD CONSTRAINT  
[fk_VendorEntityID] FOREIGN KEY([VendorEntityID])  
REFERENCES [dbo].[DimVendor] ([VendorEntityID])
```

GO

```
ALTER TABLE [dbo].[FactVendorPurchase] CHECK CONSTRAINT [fk_VendorEntityID]
```

GO