Sri Lanka Institute of Information Technology

Assignment 1

Data Warehouse & Business Intelligence

2022

Submitted by:

Wickramaarachchi W.A.K.M

IT20073428

Table of Contents

1. Data Selection and Preparation

The chosen data source is a collection of transactional data. A link to the source data set is provided below.

Dataset: - AmExpert 2019

The data set derived from the source was modified as needed. This dataset is about orders made at the store. The dataset contains information about around 1000k orders placed at several marketplaces along with discounts.

The dataset contains CSV files with information about campaign data, customer data, and product data. Modifications were made to the data set derived from the source as needed. This data set consists of combinations of transactions and promotional initiatives.

- campaign_data.csv – campaign information for each campaign
- customer_demographics_data.csv – customer information for customers
- customer_transaction_data.csv – transaction data for all customers for the duration of campaigns
- item_data.csv – item information for each item

All the data sources are provided as CSV files by the Kaggle. Therefore, in the preparation of data sources, some CSV files were imported to the source database and added columns and separated some data to make other data files.

The final types of data sources are mentioned below:

- **SQL Database**
  - Campaign
  - Customer
  - CustomerAddress
  - Transaction

- **CSV files**
  - item_category.csv
  - item_data.csv

2. Description of the data set

Name: Customer		
Source Type: SQL Database		
Column Name	**Data Type**	**Description**
customer_id	int	Unique identifier of a customer
first_name	nvarchar(50)	First name of the customer
last_name	nvarchar(50)	Last name of the customer
title	nvarchar(10)	Title of the customer
gender	nvarchar(3)	Customer's gender
email	nvarchar(50)	Email address of the customer
phone	nvarchar(20)	Phone number of the customer
age_range	nvarchar(50)	Customer age range
marital_status	nvarchar(20)	Marital status of the customer
rented	bit	Rented – 1 / not – 0
family_size	nvarchar(10)	Size of the customer's family
no_of_children	nvarchar(10)	No. of children in the family
income_bracket	int	Customer income bracket

Name: CustomerAddress		
Source Type: SQL Database		
Column Name	**Data Type**	**Description**
customer_id	int	Customer’s unique Id
city	nvarchar(30)	Customer’s city
country	nvarchar(50)	Customer’s country
country_code	nvarchar(5)	Customer’s country code
latitude	float	Latitude
longitude	float	Longitude
street_address	nvarchar(150)	Customer’s street address
street_name	nvarchar(50)	Customer’s street name
street_number	Int	Customer’s street number
street_suffix	nvarchar(20)	Customer’s street suffix

Name: Campaign		
Source Type: SQL Database		
Column Name	**Data Type**	**Description**
campaign_id	Int	Unique identifier for campaign
campaign_type	nvarchar(5)	Type of the campaign
start_date	datetime	Campaign start date
end_date	datetime	Campaign end date

Name: Transaction		
Source Type: SQL Database		
Column Name	**Data Type**	**Description**
transaction_id	int	Unique identifier of the transaction
customer_id	int	Customer unique identifier
item_id	int	Item unique identifier
date	datetime	Order placement date
quantity	int	Order item quantity
selling_price	money	Item selling rice
other_discount	money	Discounts
coupon_discount	money	Coupon discount

Name: Item Category		
Source Type: CSV file		
Column Name	**Data Type**	**Description**
category_id	int	unique category identifier
category_name	nvarchar(50)	Name of the item category

<table>
<tr><td colspan="3"><b>Name:</b> Item Data</td></tr>
<tr><td colspan="3"><b>Source Type:</b> CSV File</td></tr>
<tr><th>Column Name</th><th>Data Type</th><th>Description</th></tr>
<tr><td>item_id</td><td>int</td><td>Item unique identifier</td></tr>
<tr><td>brand</td><td>int</td><td>Item brand code</td></tr>
<tr><td>brand_type</td><td>nvarchar(50)</td><td>Item brand type</td></tr>
<tr><td>category</td><td>int</td><td>Item category id</td></tr>
</table>

3. ER Diagram

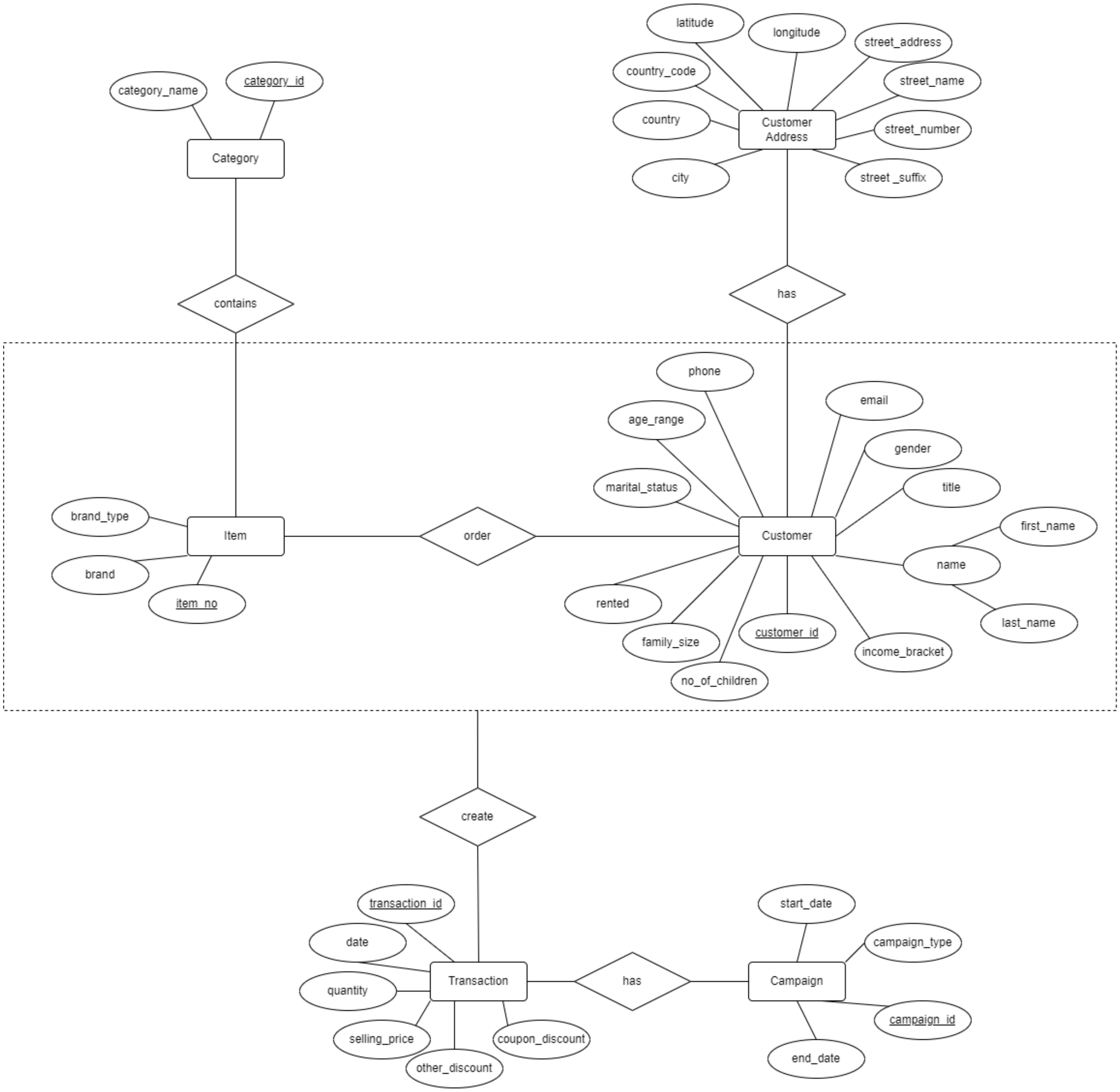


Figure 1: ER Diagram

The above diagram shows the connections between entities

4. Solution Architecture

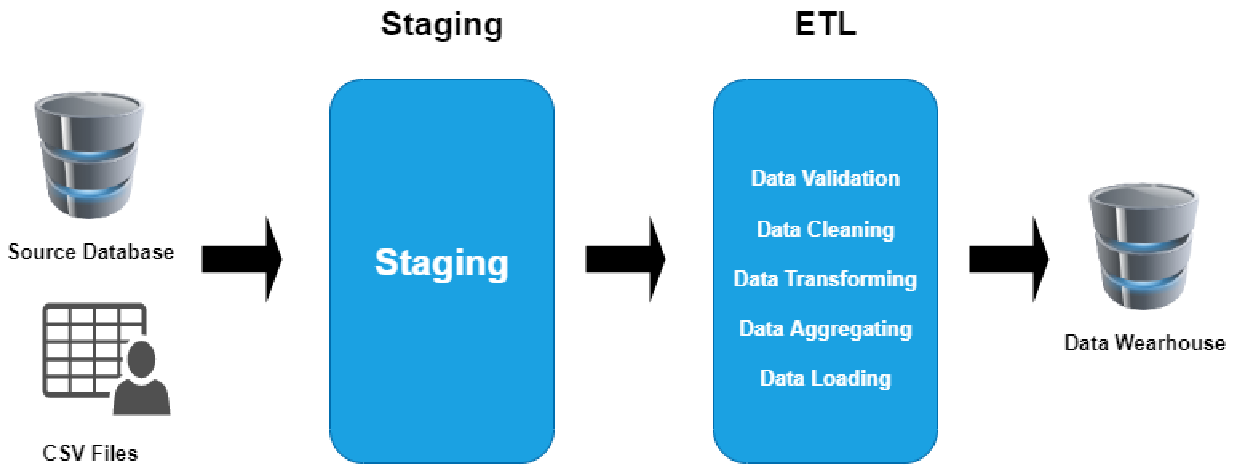


Figure 2: High-Level BI Solution Architecture

In the staging layer;

- StgCustomer
- StgCustomerAddress
- StgCampaign
- StgItem
- StgItemCategory
- StgTransaction tables were created

5. Data Warehouse Design and Development

a) Design

AmExpert_DW is designed according to the snowflake schema with one fact table and four dimension tables, including Date Dimension.

- Dimension Tables
    - DimDate
    - DimCustomer
    - DimItem
    - DimItemCategory

- Fact Table
    - DimTransaction

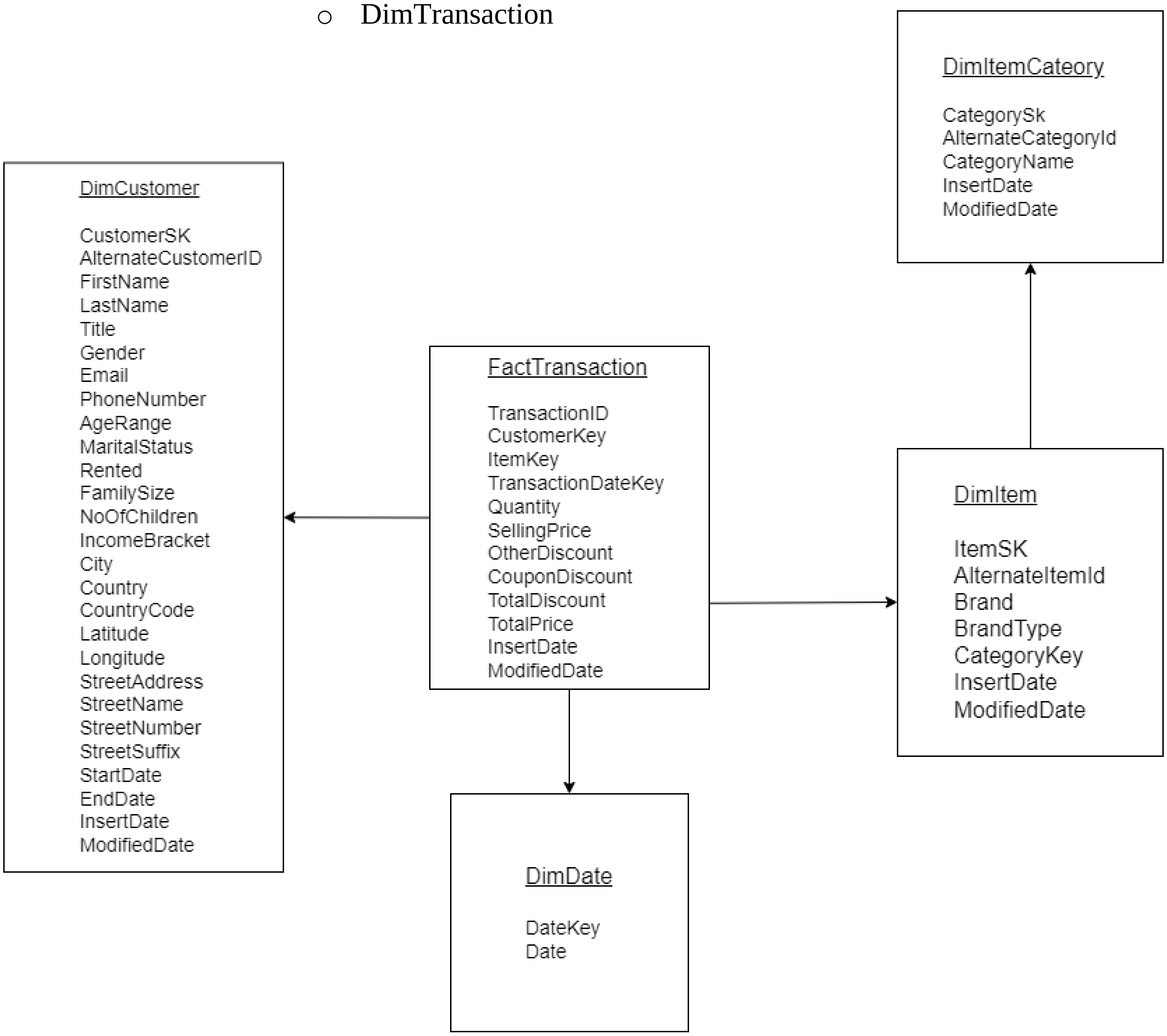


Figure 3: snowflake schema

- Hierarchies

  - DimItemCategory is a hierarchical dimension of DimItem
  - DimCustomer table has hierarchical attributes about customer address

- Calculations
  - Total Discount is calculated in the FactTransaction table
    (([OtherDiscount] + [CouponDiscount]) * [Quantity])

  - Total Price is calculated in the FactTransaction table
    (([SellingPrice]) * [Quantity]) – [TotalDiscount])

b) Assumptions

- Transaction table used for creating fact table
- Transaction per customer was considered as the grain

c) Slowly changing dimensions

- DimCustomer was considered a slowly changing dimension

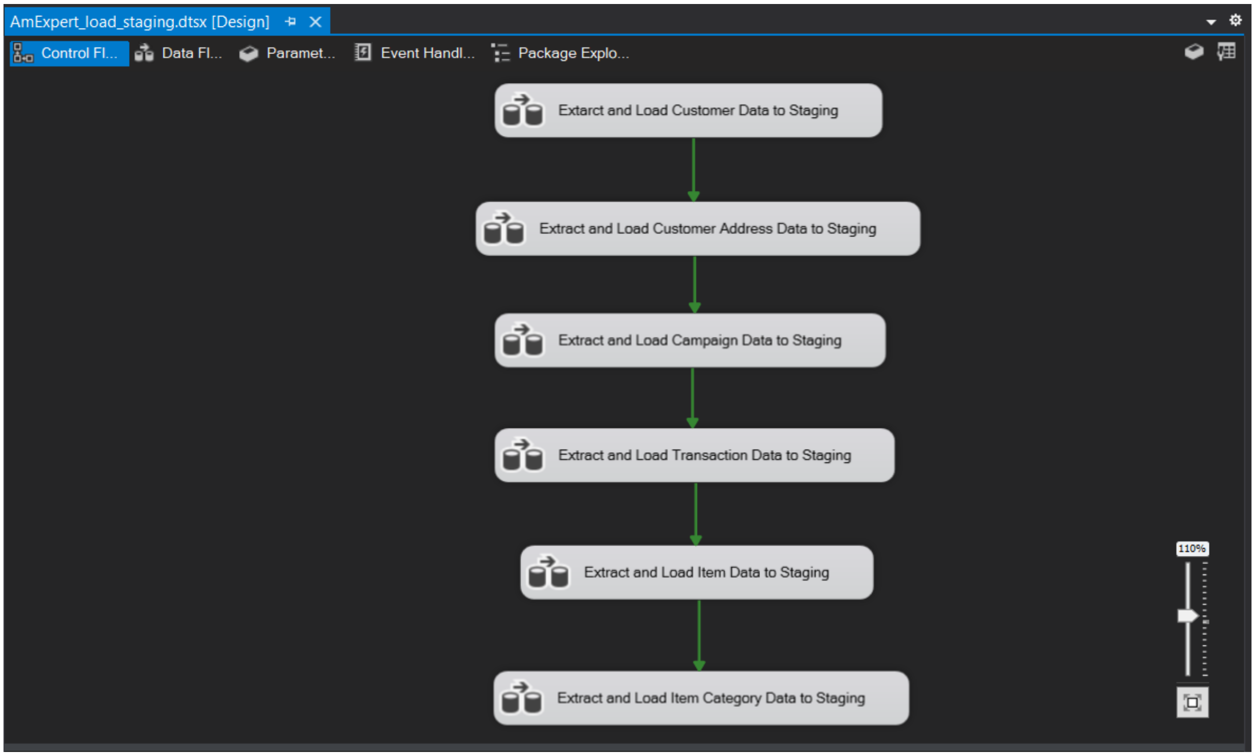
Changing Attributes	**Historical Attributes**
MaritalStatus	AgeRange
PhoneNumber	City
Title	Country
FamilySize	CountryCode
NoOfChildren	Latitude
	Longitude
	StreetAddress
	StreetName
	StreetNumber
	StreetSuffix

6. ETL Development

I. Data Extraction and Loading into Staging Tables

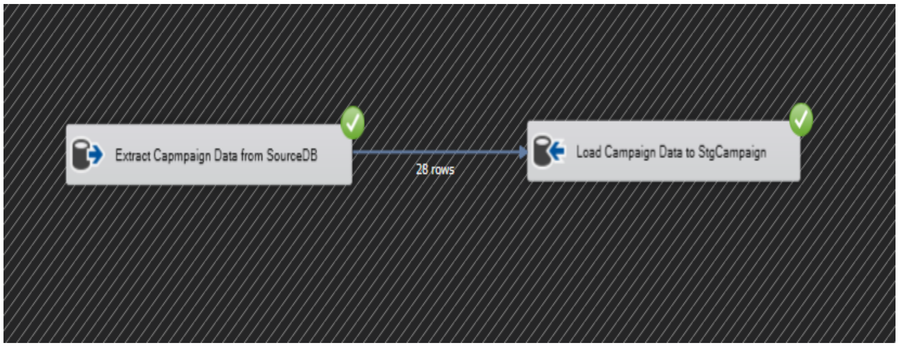
- Data extraction is done by using Visual Studio Data Tools. The CSV files and database are used as data sources.
- OLE DB SOURCE is used to extract data from the database, and FLAT FILE SOURCE is used to extract data from CSV files.
- For the load data to the staging area, OLE DB DESTINATION was used
- Use EXECUTE SQL TASK to truncate Staging tables before load data. This will prevent data from being duplicated in tagging tables

Control Flow


AmExpert_load_staging.dtsx [Design]
Control Fl...
Data Fl...
Paramet...
Event Handl...
Package Explo...
Extarct and Load Customer Data to Staging
Extract and Load Customer Address Data to Staging
Extract and Load Campaign Data to Staging
Extract and Load Transaction Data to Staging
Extract and Load Item Data to Staging
Extract and Load Item Category Data to Staging
110%

Attachment 1: Control Flow

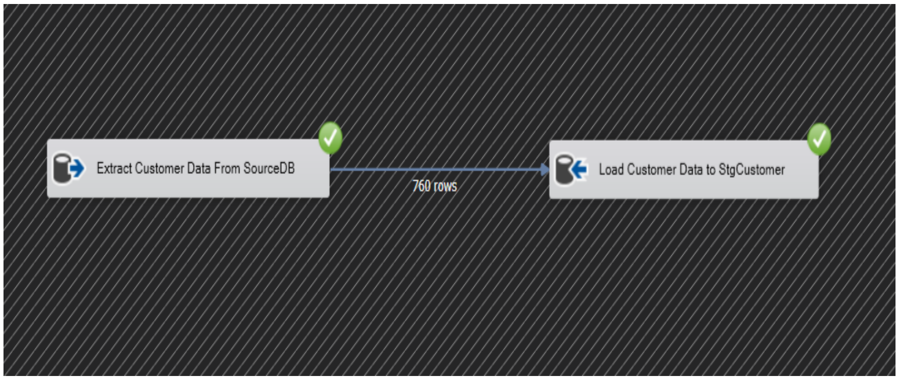
Staging Campaign Details


Extract Capmpaign Data from SourceDB
28 rows
Load Campaign Data to StgCampaign

Campagn data is extracted from AmExpert SourceDB and inserted into Campaign staging table.

Attachment 2: Staging campaign data

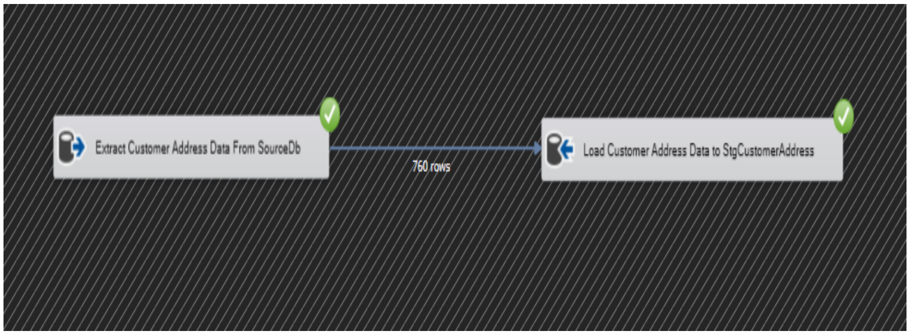
Staging Customer Details


Extract Customer Data From SourceDB
760 rows
Load Customer Data to StgCustomer

Customer data is extracted from AmExpert SourceDB and inserted into Customer staging table.

Attachment 3: Staging customer details

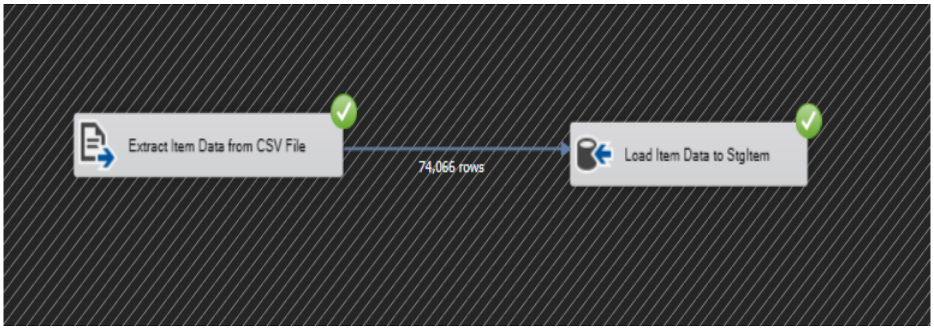
Staging Customer Address Details


Extract Customer Address Data From SourceDb
760 rows
Load Customer Address Data to StgCustomerAddress

Customer Address data is extracted from AmExpert SourceDB ans insert into Customer Address staging table

Attachment 4: Staging customer address data

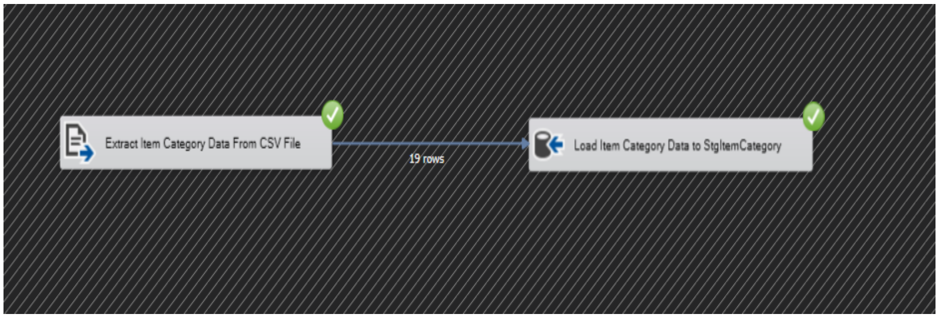
Staging Item Details



Attachment 5: Staging item details

Item details is extracted from CSV file and insert into Item staging table

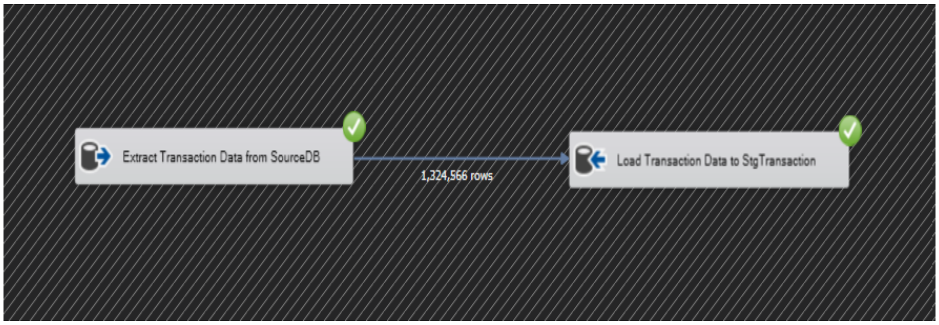
Staging Item Category Details


Extract Item Category Data From CSV File
19 rows
Load Item Category Data to StgItemCategory

Attachment 6: Staging item category details

Item Category data is extracted from CSV file and insert into Item Category staging table

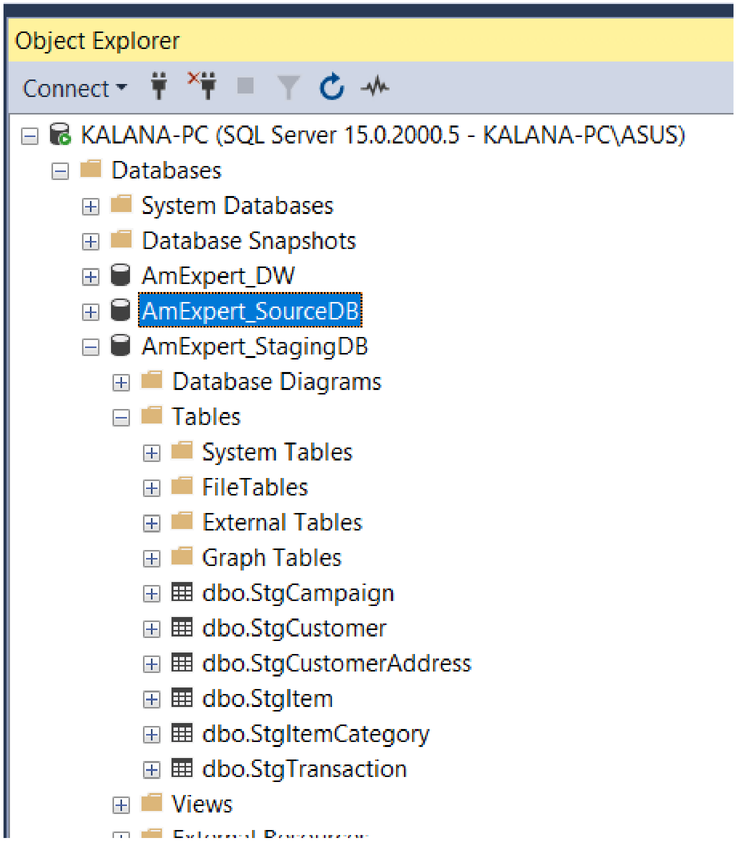
Staging Transaction Details


Extract Transaction Data from SourceDB
1,324,566 rows
Load Transaction Data to StgTransaction

Attachment 7: Staging transaction details

Transaction data is extracted from AmExpert SourceDb and insert into Transaction staging table

Created Staging Tables



Object Explorer

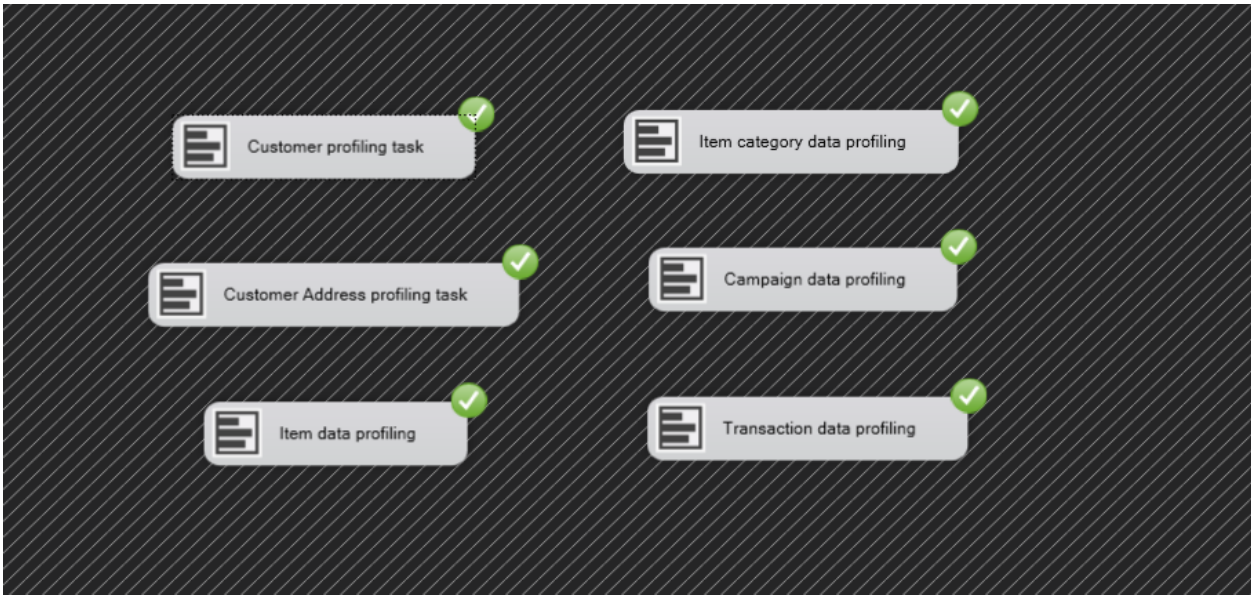
- KALANA-PC (SQL Server 15.0.2000.5 - KALANA-PC\ASUS)
  - Databases
    - System Databases
    - Database Snapshots
    - AmExpert_DW
    - AmExpert_SourceDB
    - AmExpert_StagingDB
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - External Tables
        - Graph Tables
        - dbo.StgCampaign
        - dbo.StgCustomer
        - dbo.StgCustomerAddress
        - dbo.StgItem
        - dbo.StgItemCategory
        - dbo.StgTransaction
      - Views

Attachment 8: Created staging tables

Staging tabales were created and values inserted

II. Data Profiling

Data Profiling enables the analysis of enormous amounts of data using various procedures. Null values, repeated values, and data quality are all examined in this step.

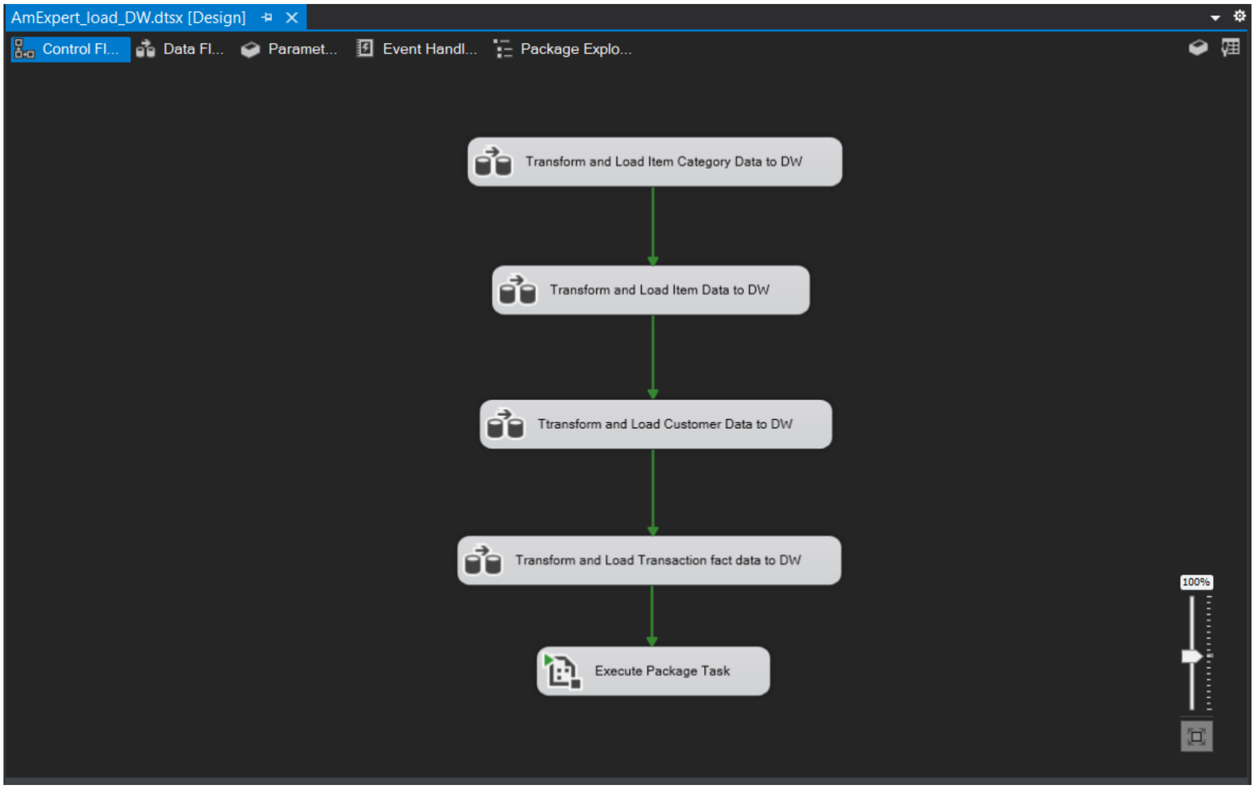

Customer profiling task
Item category data profiling
Customer Address profiling task
Campaign data profiling
Item data profiling
Transaction data profiling

Attachment 9: Data profiling task

- Every staging table had run a profiling task and saved output in a selected location
- By referring these data profiles, the developer is able to identify the issues with staging data

III. Data Transforming and Loading

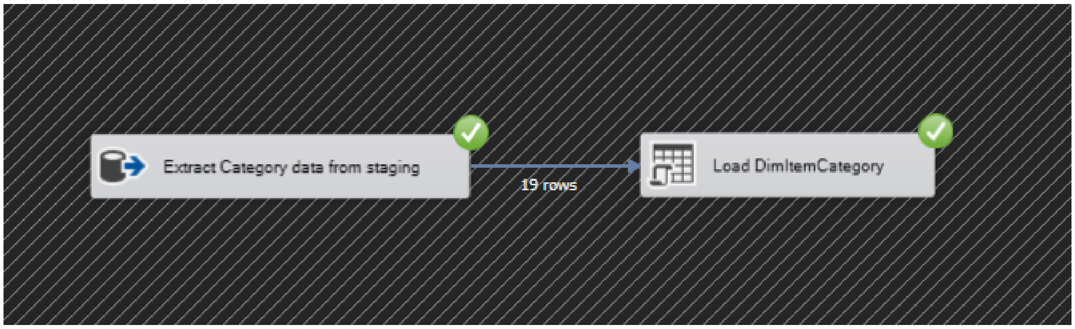
- Data transformation is developed according to the dimension modelling designed above (figure 3)
- Dimension tables are loaded with data from relevant staging tables in this step.


AmExpert_load_DW.dtsx [Design]
Control Fl...
Data Fl...
Paramet...
Event Handl...
Package Explo...
Transform and Load Item Category Data to DW
Transform and Load Item Data to DW
Ttransform and Load Customer Data to DW
Transform and Load Transaction fact data to DW
Execute Package Task
100%

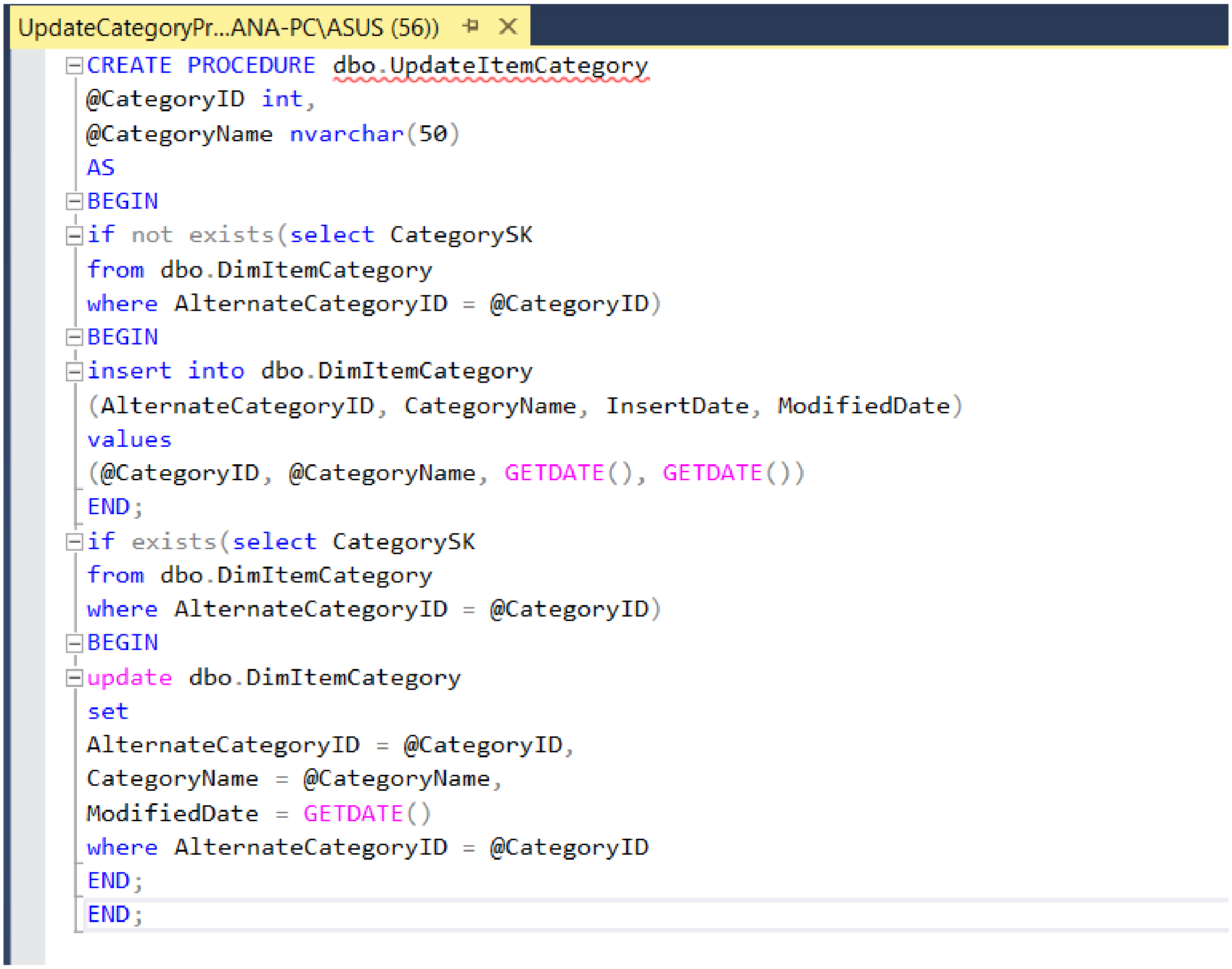
Attachment 10: Control Flow Data Wearhouse

Transform and Load Item Category Data

- Item Category data is loaded into DimCategory table
- UpdateItemCategory Procedure is used to identify data should be insert or not



Attachment 11: Load DImCategory

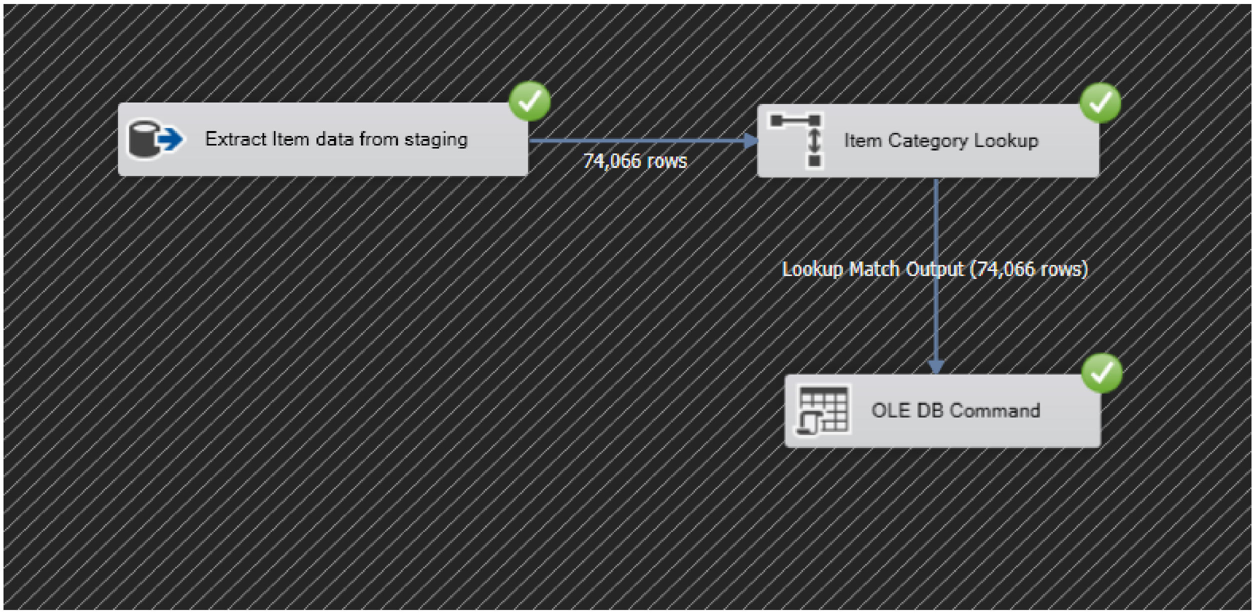


```sql
CREATE PROCEDURE dbo.UpdateItemCategory
@CategoryID int,
@CategoryName nvarchar(50)
AS
BEGIN
if not exists(select CategorySK
from dbo.DimItemCategory
where AlternateCategoryID = @CategoryID)
BEGIN
insert into dbo.DimItemCategory
(AlternateCategoryID, CategoryName, InsertDate, ModifiedDate)
values
(@CategoryID, @CategoryName, GETDATE(), GETDATE())
END;
if exists(select CategorySK
from dbo.DimItemCategory
where AlternateCategoryID = @CategoryID)
BEGIN
update dbo.DimItemCategory
set
AlternateCategoryID = @CategoryID,
CategoryName = @CategoryName,
ModifiedDate = GETDATE()
where AlternateCategoryID = @CategoryID
END;
END;
```

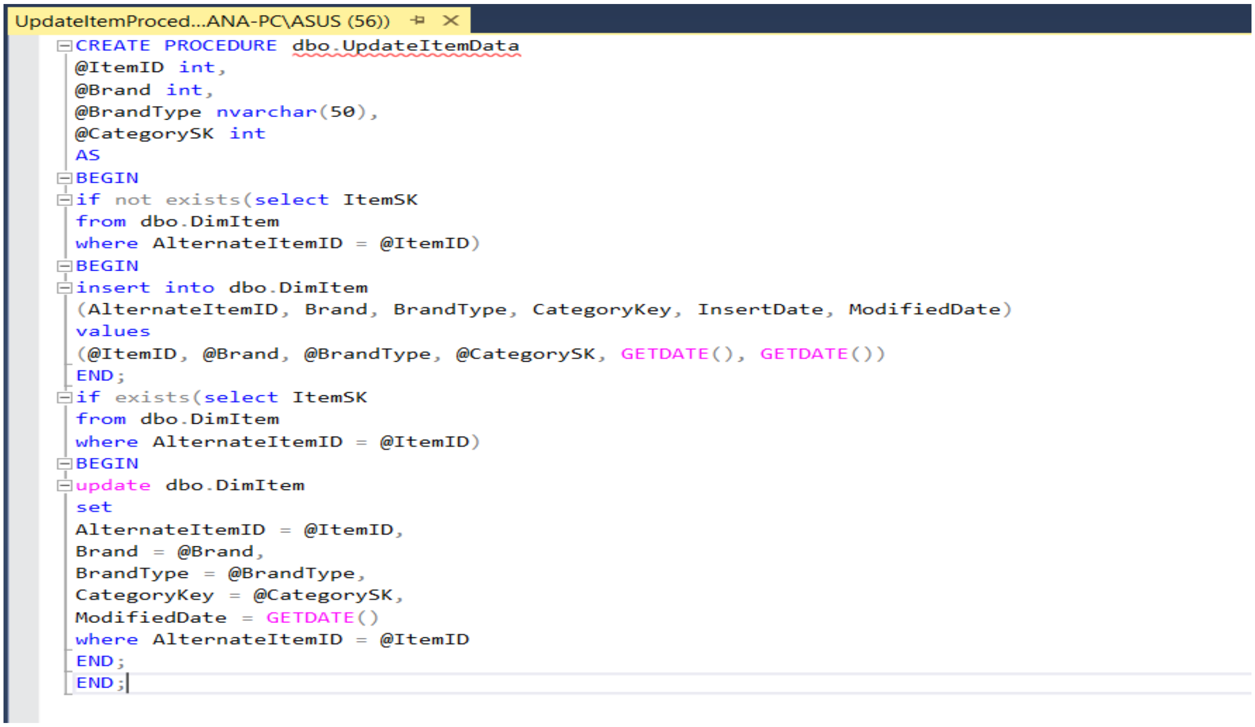
Attachment 12: UpdateItemCategory Procedure

Transform and Load Item Data

- Item data is loaded into DimItem table
- Lookup transformation task used for extract CategoryKey from DimCategory
- UpdateItem Procedure is used to identify data should be insert or not


Extract Item data from staging
74,066 rows
Item Category Lookup
Lookup Match Output (74,066 rows)
OLE DB Command

Attachment 13: Load DimItem

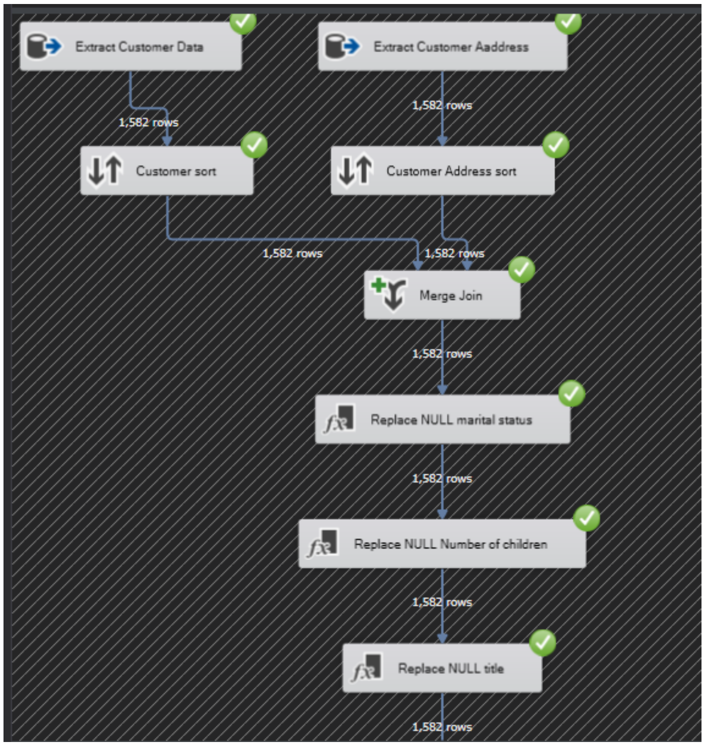


```
CREATE PROCEDURE dbo.UpdateItemData
@ItemID int,
@Brand int,
@BrandType nvarchar(50),
@CategorySK int
AS
BEGIN
if not exists(select ItemSK
from dbo.DimItem
where AlternateItemID = @ItemID)
BEGIN
insert into dbo.DimItem
(AlternateItemID, Brand, BrandType, CategoryKey, InsertDate, ModifiedDate)
values
(@ItemID, @Brand, @BrandType, @CategorySK, GETDATE(), GETDATE())
END;
if exists(select ItemSK
from dbo.DimItem
where AlternateItemID = @ItemID)
BEGIN
update dbo.DimItem
set
AlternateItemID = @ItemID,
Brand = @Brand,
BrandType = @BrandType,
CategoryKey = @CategorySK,
ModifiedDate = GETDATE()
where AlternateItemID = @ItemID
END;
END;
```

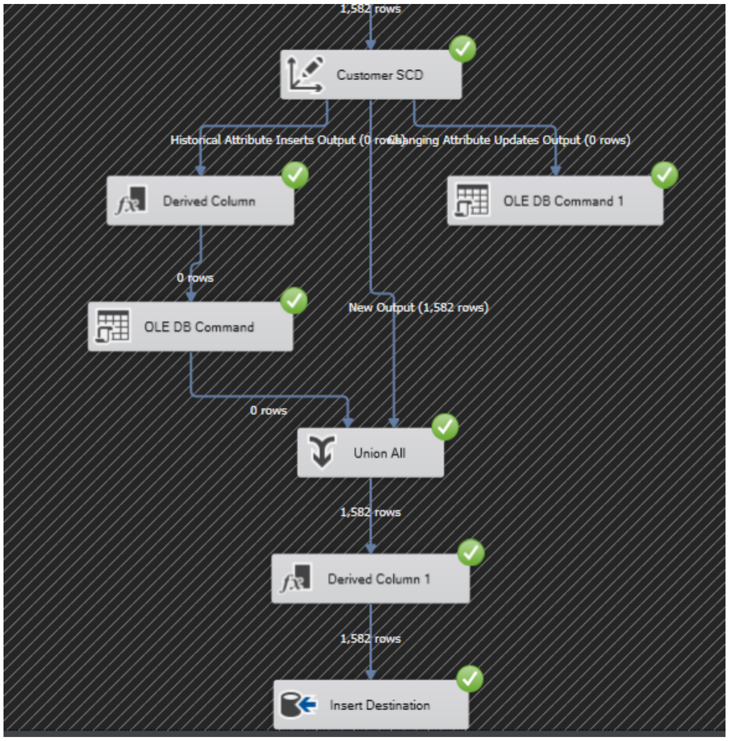
Attachment 14: UpdateItemData Procedure

Transform and Customer Data (Slowly Changing Dimension)

- DimCustomer is the Slowly Changing Dimension (SCD) in this modelling
- StartDate and EndDate columns ensure that the data valid at the moment
- Slowly changing dimension wizard used to implement DimCustomer model
- Derived Column Transformations are used to fill null values (marital status, number of children, title)



Attachment 15 - i: Load DimCustomer

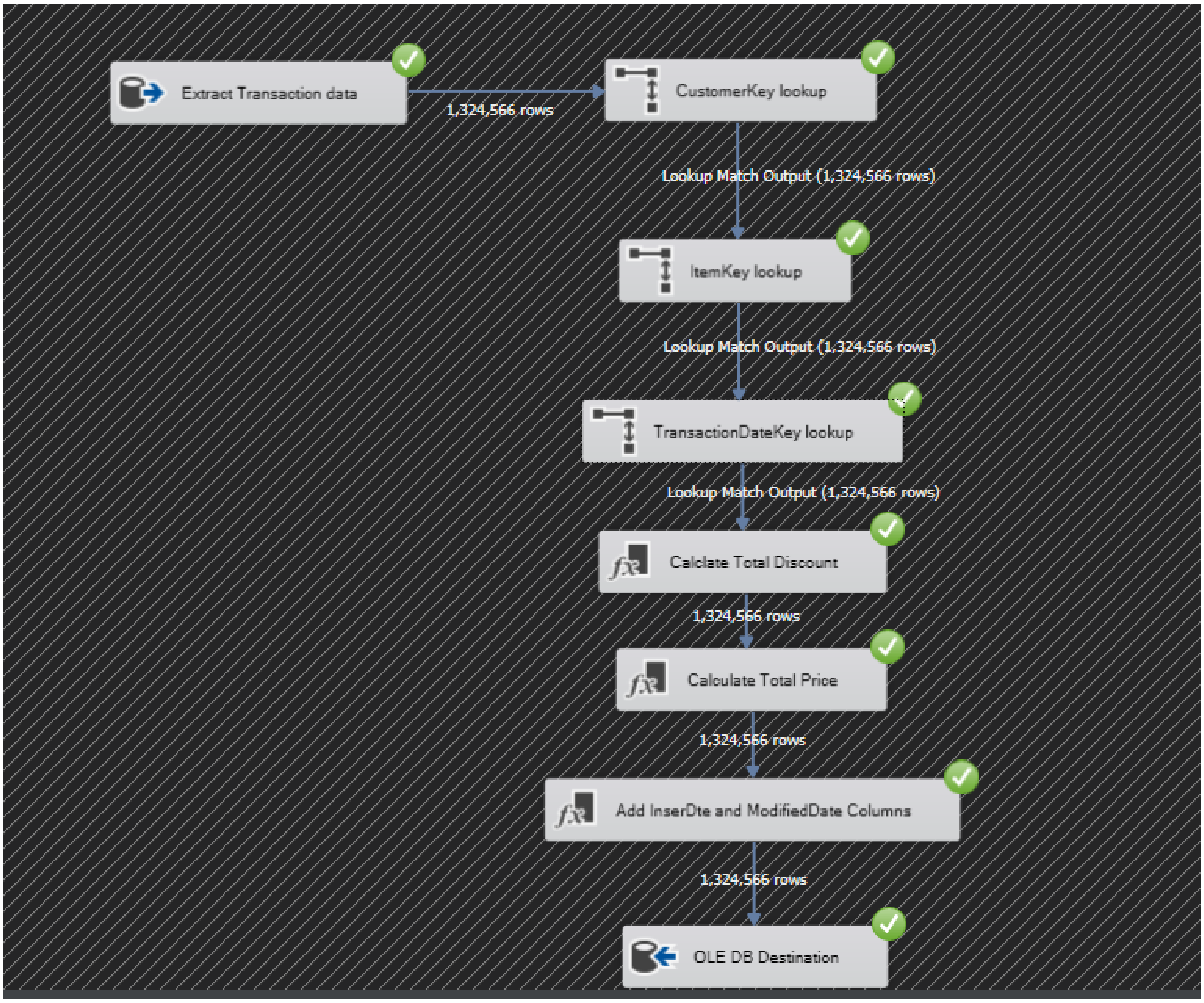


Attachment 15 - ii:Load DimCustomer

Load Data to Fact Table

- According to the dimension model, StgTransaction table used to insert values into FactTransaction table
- After loading all the dimension tables, lastly data was inserted in to the fact table. Below steps were followed

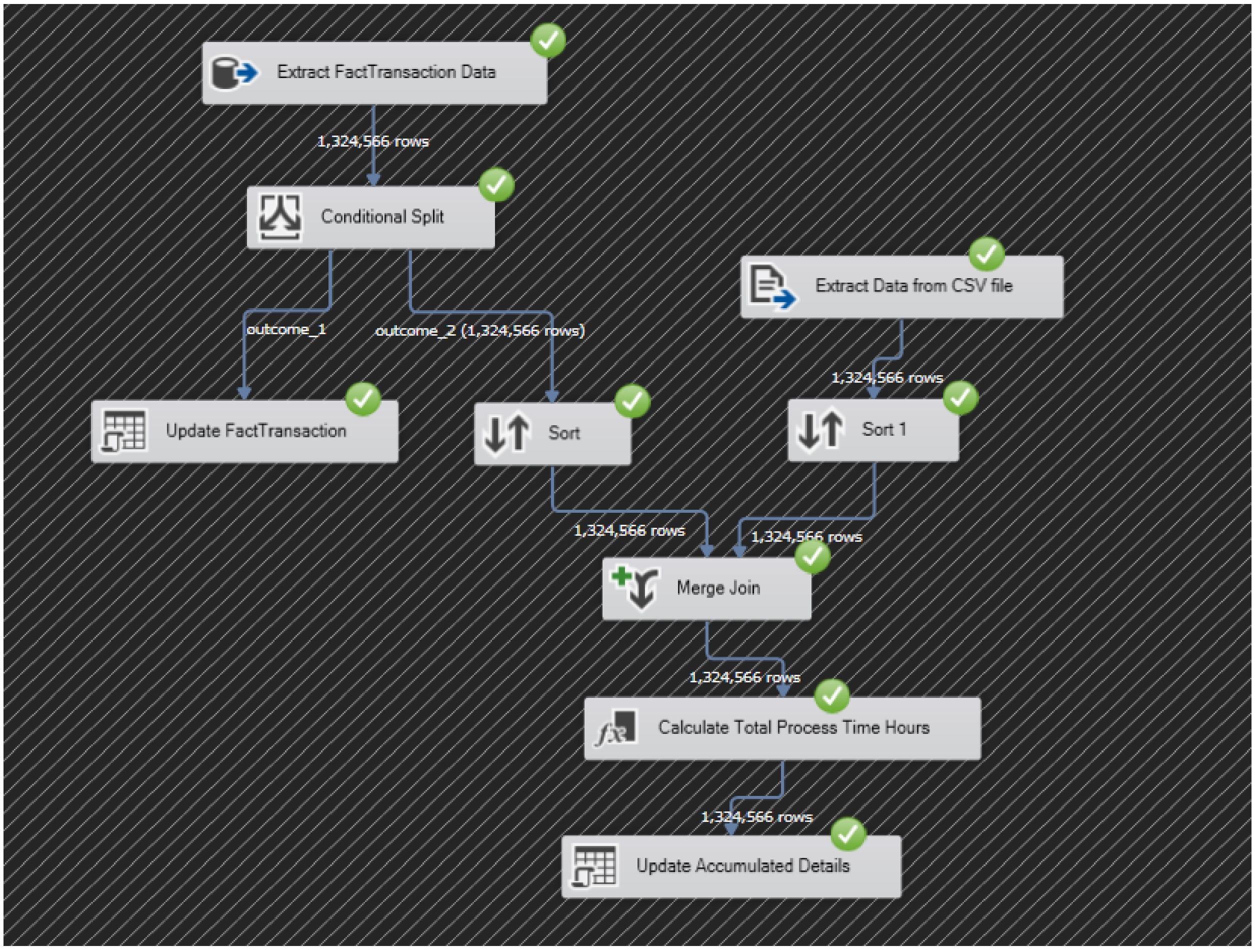
1. Data extracted from the StgTransaction staging table
2. Join operation done for the CustomerKey using lookup
3. Join operation done for the ItemKey using lookup
4. Join operation done for the TransactionDateKey using lookup
5. Join operation done for the CustomerKey using lookup
6. Calculate Total Discount and Total Price using derived column transformations
7. Insert and Modified dates were derived
8. Load FactTransaction table



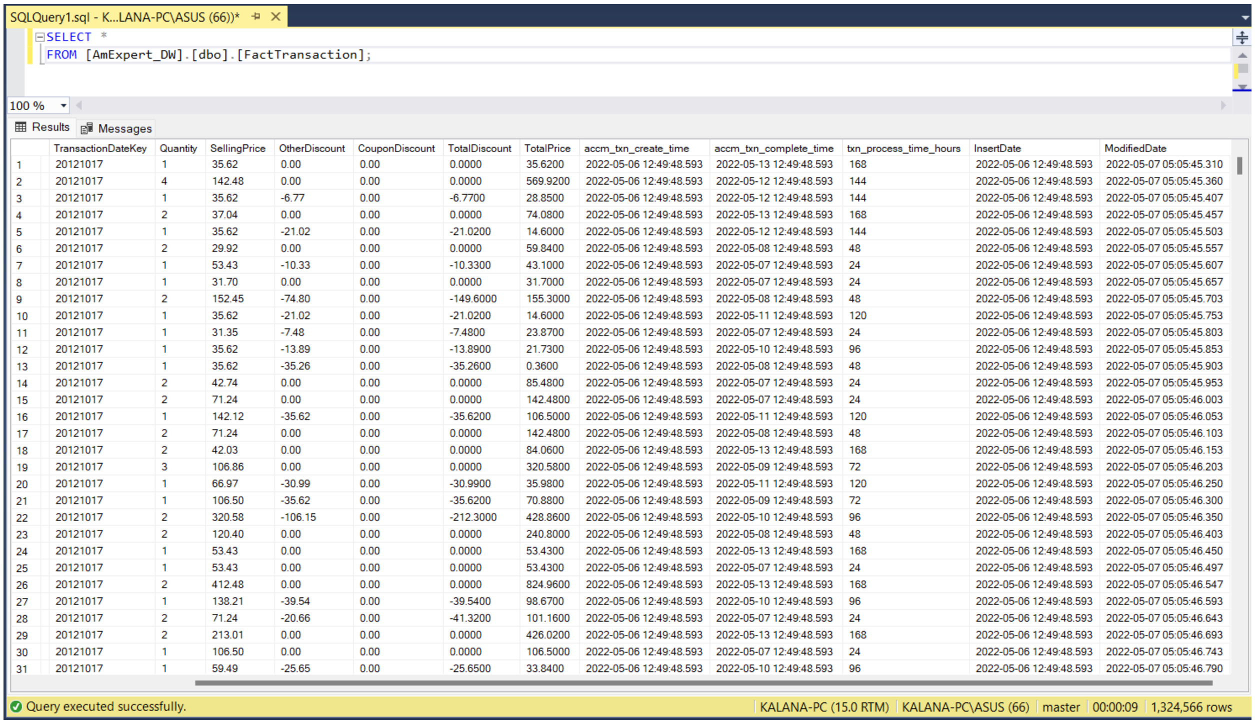
Attachment 16: Load FactTransaction

Load Accumulated Data to Fact Table

- In life cycle of transaction process, Fact table should be updated with current status of transaction
- Therefore developers use accumulated type fact table to implement the solution. Below steps were followed
    1. Extract current Fact data
    2. Use conditional split to check whether accumulated data already exist or not
    3. If not exists, extract new accumulated data from source and join new data with fact data using merge join
    4. Done calculations using derived column transformation
    5. Update Fact table



Attachment 17: Load Accumulated data



SQLQuery1.sql - K...LANA-PC\ASUS (66))*

```
SELECT *
FROM [AmExpert_DW].[dbo].[FactTransaction];
```

	TransactionDateKey	Quantity	SellingPrice	OtherDiscount	CouponDiscount	TotalDiscount	TotalPrice	accm_txn_create_time	accm_txn_complete_time	txn_process_time_hours	InsertDate	ModifiedDate
1	20121017	1	35.62	0.00	0.00	0.0000	35.6200	2022-05-06 12:49:48.593	2022-05-13 12:49:48.593	168	2022-05-06 12:49:48.593	2022-05-07 05:05:45.310
2	20121017	4	142.48	0.00	0.00	0.0000	569.9200	2022-05-06 12:49:48.593	2022-05-12 12:49:48.593	144	2022-05-06 12:49:48.593	2022-05-07 05:05:45.360
3	20121017	1	35.62	-6.77	0.00	-6.7700	28.8500	2022-05-06 12:49:48.593	2022-05-12 12:49:48.593	144	2022-05-06 12:49:48.593	2022-05-07 05:05:45.407
4	20121017	2	37.04	0.00	0.00	0.0000	74.0800	2022-05-06 12:49:48.593	2022-05-13 12:49:48.593	168	2022-05-06 12:49:48.593	2022-05-07 05:05:45.457
5	20121017	1	35.62	-21.02	0.00	-21.0200	14.6000	2022-05-06 12:49:48.593	2022-05-12 12:49:48.593	144	2022-05-06 12:49:48.593	2022-05-07 05:05:45.503
6	20121017	2	29.92	0.00	0.00	0.0000	59.8400	2022-05-06 12:49:48.593	2022-05-08 12:49:48.593	48	2022-05-06 12:49:48.593	2022-05-07 05:05:45.557
7	20121017	1	53.43	-10.33	0.00	-10.3300	43.1000	2022-05-06 12:49:48.593	2022-05-07 12:49:48.593	24	2022-05-06 12:49:48.593	2022-05-07 05:05:45.607
8	20121017	1	31.70	0.00	0.00	0.0000	31.7000	2022-05-06 12:49:48.593	2022-05-07 12:49:48.593	24	2022-05-06 12:49:48.593	2022-05-07 05:05:45.657
9	20121017	2	152.45	-74.80	0.00	-149.6000	155.3000	2022-05-06 12:49:48.593	2022-05-08 12:49:48.593	48	2022-05-06 12:49:48.593	2022-05-07 05:05:45.703
10	20121017	1	35.62	-21.02	0.00	-21.0200	14.6000	2022-05-06 12:49:48.593	2022-05-11 12:49:48.593	120	2022-05-06 12:49:48.593	2022-05-07 05:05:45.753
11	20121017	1	31.35	-7.48	0.00	-7.4800	23.8700	2022-05-06 12:49:48.593	2022-05-07 12:49:48.593	24	2022-05-06 12:49:48.593	2022-05-07 05:05:45.803
12	20121017	1	35.62	-13.89	0.00	-13.8900	21.7300	2022-05-06 12:49:48.593	2022-05-10 12:49:48.593	96	2022-05-06 12:49:48.593	2022-05-07 05:05:45.853
13	20121017	1	35.62	-35.26	0.00	-35.2600	0.3600	2022-05-06 12:49:48.593	2022-05-08 12:49:48.593	48	2022-05-06 12:49:48.593	2022-05-07 05:05:45.903
14	20121017	2	42.74	0.00	0.00	0.0000	85.4800	2022-05-06 12:49:48.593	2022-05-07 12:49:48.593	24	2022-05-06 12:49:48.593	2022-05-07 05:05:45.953
15	20121017	2	71.24	0.00	0.00	0.0000	142.4800	2022-05-06 12:49:48.593	2022-05-07 12:49:48.593	24	2022-05-06 12:49:48.593	2022-05-07 05:05:46.003
16	20121017	1	142.12	-35.62	0.00	-35.6200	106.5000	2022-05-06 12:49:48.593	2022-05-11 12:49:48.593	120	2022-05-06 12:49:48.593	2022-05-07 05:05:46.053
17	20121017	2	71.24	0.00	0.00	0.0000	142.4800	2022-05-06 12:49:48.593	2022-05-08 12:49:48.593	48	2022-05-06 12:49:48.593	2022-05-07 05:05:46.103
18	20121017	2	42.03	0.00	0.00	0.0000	84.0600	2022-05-06 12:49:48.593	2022-05-13 12:49:48.593	168	2022-05-06 12:49:48.593	2022-05-07 05:05:46.153
19	20121017	3	106.86	0.00	0.00	0.0000	320.5800	2022-05-06 12:49:48.593	2022-05-09 12:49:48.593	72	2022-05-06 12:49:48.593	2022-05-07 05:05:46.203
20	20121017	1	66.97	-30.99	0.00	-30.9900	35.9800	2022-05-06 12:49:48.593	2022-05-11 12:49:48.593	120	2022-05-06 12:49:48.593	2022-05-07 05:05:46.250
21	20121017	1	106.50	-35.62	0.00	-35.6200	70.8800	2022-05-06 12:49:48.593	2022-05-09 12:49:48.593	72	2022-05-06 12:49:48.593	2022-05-07 05:05:46.300
22	20121017	2	320.58	-106.15	0.00	-212.3000	428.8600	2022-05-06 12:49:48.593	2022-05-10 12:49:48.593	96	2022-05-06 12:49:48.593	2022-05-07 05:05:46.350
23	20121017	2	120.40	0.00	0.00	0.0000	240.8000	2022-05-06 12:49:48.593	2022-05-08 12:49:48.593	48	2022-05-06 12:49:48.593	2022-05-07 05:05:46.403
24	20121017	1	53.43	0.00	0.00	0.0000	53.4300	2022-05-06 12:49:48.593	2022-05-13 12:49:48.593	168	2022-05-06 12:49:48.593	2022-05-07 05:05:46.450
25	20121017	1	53.43	0.00	0.00	0.0000	53.4300	2022-05-06 12:49:48.593	2022-05-07 12:49:48.593	24	2022-05-06 12:49:48.593	2022-05-07 05:05:46.497
26	20121017	2	412.48	0.00	0.00	0.0000	824.9600	2022-05-06 12:49:48.593	2022-05-13 12:49:48.593	168	2022-05-06 12:49:48.593	2022-05-07 05:05:46.547
27	20121017	1	138.21	-39.54	0.00	-39.5400	98.6700	2022-05-06 12:49:48.593	2022-05-10 12:49:48.593	96	2022-05-06 12:49:48.593	2022-05-07 05:05:46.593
28	20121017	2	71.24	-20.66	0.00	-41.3200	101.1600	2022-05-06 12:49:48.593	2022-05-07 12:49:48.593	24	2022-05-06 12:49:48.593	2022-05-07 05:05:46.643
29	20121017	2	213.01	0.00	0.00	0.0000	426.0200	2022-05-06 12:49:48.593	2022-05-13 12:49:48.593	168	2022-05-06 12:49:48.593	2022-05-07 05:05:46.693
30	20121017	1	106.50	0.00	0.00	0.0000	106.5000	2022-05-06 12:49:48.593	2022-05-07 12:49:48.593	24	2022-05-06 12:49:48.593	2022-05-07 05:05:46.743
31	20121017	1	59.49	-25.65	0.00	-25.6500	33.8400	2022-05-06 12:49:48.593	2022-05-10 12:49:48.593	96	2022-05-06 12:49:48.593	2022-05-07 05:05:46.790

Query executed successfully. | KALANA-PC (15.0 RTM) | KALANA-PC\ASUS (66) | master | 00:00:09 | 1,324,566 rows

Attachment 18: Final FactTransaction table