Sri Lanka Institute of Information Technology



**Data Warehouse and Business Intelligence – Assignment 1 Submission**

Name: Andrew Asher

Reg-No: IT20220624

Batch: Year 03 Semester 01

**Contents**

1. Data set selection…………………………………………………3
2. Preparation of data source………………………………………..4
3. Solution Architecture……………………………………………..7
4. Data warehouse design and development………………….……..8
5. Design………………………………………………………8
6. Assumptions………………………………………………..9
7. Slowly changing dimensions……………………………….9
8. ETL Development………………………………………………..10
   1. Data Extraction & Load into Staging tables……………….10
   2. Data Profiling………………………………………………13
   3. Data Transformation and Loading…………………………14

**1.DATA SET SELECTION**

Data Set Name: AmExpert 2019 - **Predicting Coupon Redemption**

Provided by: kaggle.com

Source link: [https://www.kaggle.com/bharath901/amexpert-2019](https://www.kaggle.com/bharath901/amexpert-2019e)

About Dataset:

The selected data source is a collection of transactional data.

Discount marketing and coupon usage are very widely used promotional techniques to attract new customers and to retain & reinforce loyalty of existing customers. The measurement of a consumer’s propensity towards coupon usage and the prediction of the redemption behavior are crucial parameters in assessing the effectiveness of a marketing campaign.

ABC is an established Brick & Mortar retailer that frequently conducts marketing campaigns for its diverse product range. ABC’s promotions are shared across various channels including email, notifications, etc. A number of these campaigns include coupon discounts that are offered for a specific product/range of products. The retailer would like the ability to predict whether customers redeem the coupons received across channels, which will enable the retailer’s marketing team to accurately design coupon construct, and develop more precise and targeted marketing strategies.

Dataset contains five csv files with information about customers, Train, Items, Campaign and Transaction. Modifications were done accordingly to the data set derived from the source This data set reflects combinations between customer transactions and promotion campaigns.

* **train.csv:** Train data containing the coupons offered to the given customers under the campaigns
* **campaign\_data.csv:** Campaign information for each of the campaign
* **customer\_demographics.csv:** Customer demographic information for some customers
* **customer**transaction**data.csv:** Transaction data for all customers for duration of campaigns in the train data
* **item\_data.csv:** Item information for each item sold by the retailer

**2.PREPARATION OF DATA SOURCE**

All the data sources are provided in csv format by the web site. In preparation of data sources, some changes have done for the source format (some columns were added, separated into a another table) of the given files as converting into text files and importing csv files into a source database.

Final State of Preparation of the source data formats before Transforming data =>

- category.txt

- AmExpert\_SourceDB (Source Database) Tables: -

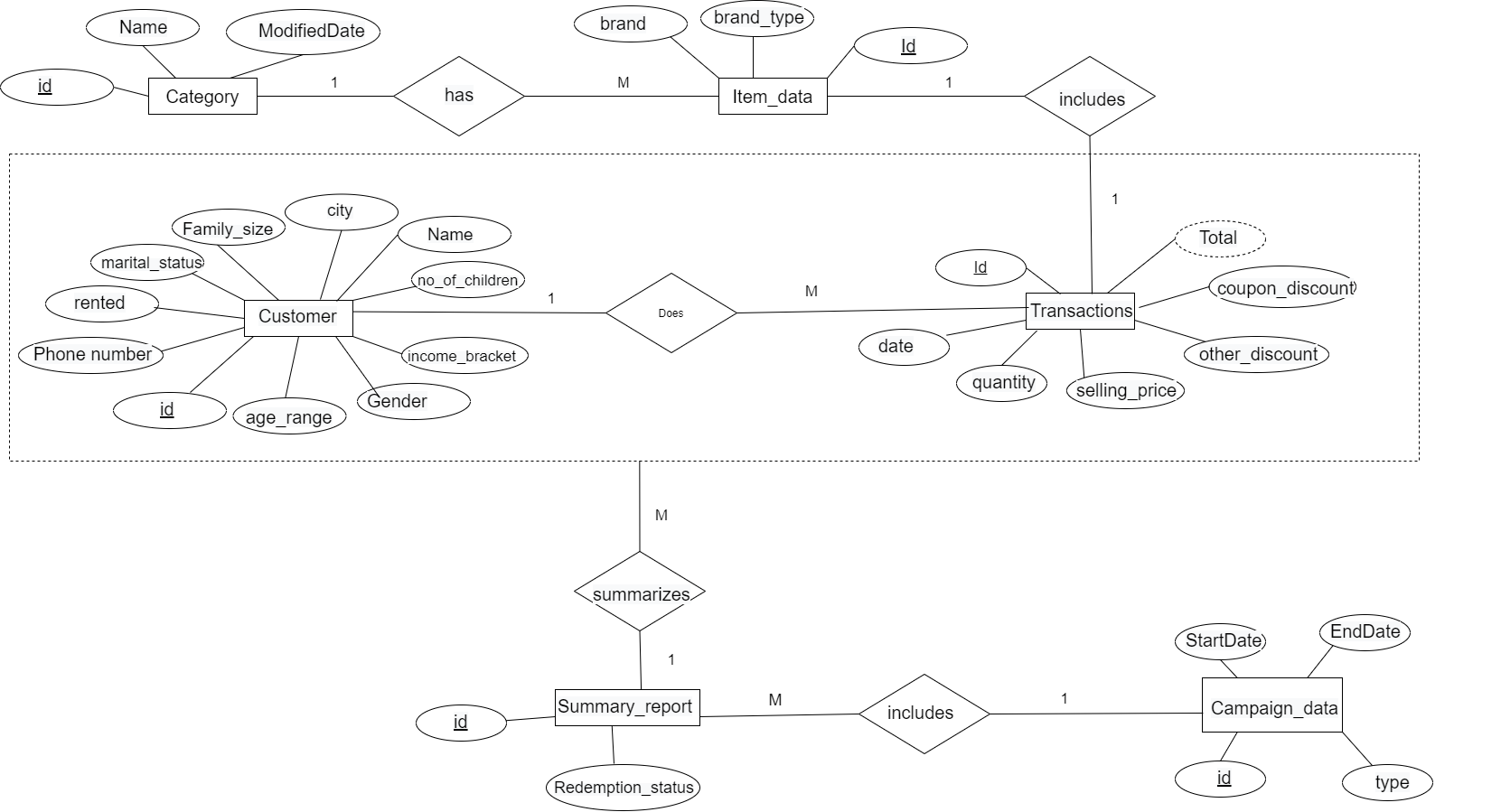
- dbo.Campaign

- dbo.Item

- dbo.Train

- dbo.Transaction

- dbo.CustomerDetails

**ER-Diagram**

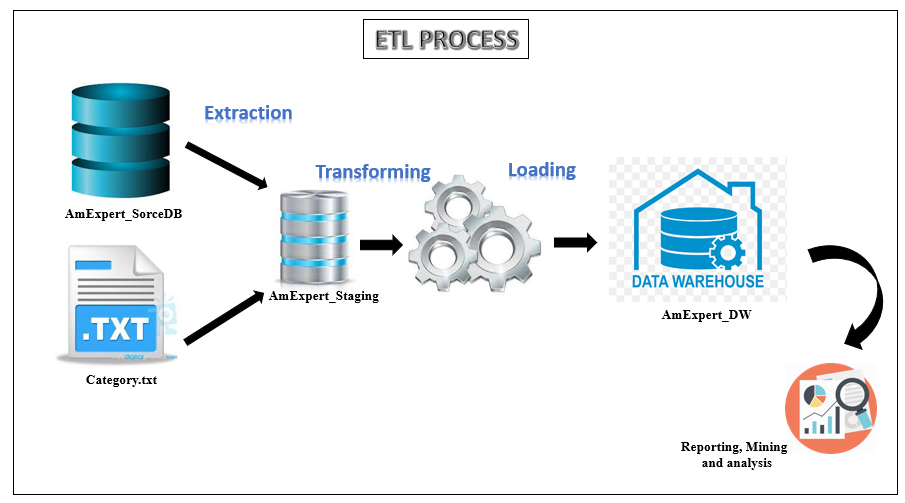
*Figure 1: ER-Diagram*

* The above diagram shows the connection between the entities in the data set.
* Assumptions:
  + The particular transaction includes only a single item .
  + One summary report(train) summarizes many customer transactions.
  + There can be many campaign data sets in a single summary report.
  + One customer can have many transactions.

**Description of the data set**

|  |  |  |
| --- | --- | --- |
| **Source Type** | **Table Name** | **Include** |
| Category.txt | Category | |  |  |  | | --- | --- | --- | | Column | Data type | Description | | CategoryID | int | Category key | | Name | nvarchar(50) | Category name | | ModifiedDate | datetime | Modification date of category | |
| AmExpert\_SourceDB | Campaign | |  |  |  | | --- | --- | --- | | Column | Data type | Description | | CampaignID | int | Campaign key | | CampaignTpe | nvarchar(1) | Anonymized Campaign Type (X/Y) | | StartDate | datetime | Campaign Start date | | EndDate | datetime | Campaign end Date | |
| Item | |  |  |  | | --- | --- | --- | | Column | Data type | Description | | ItemID | int | Unique id for item | | Brand | int | Unique id for item brand | | BrandType | nvarchar(50) | Brand Type (local/Established) | | CategoryID | int | Item CategoryID | |
| Train | |  |  |  | | --- | --- | --- | | Column | Data type | Description | | TrainID | Int | Unique id for coupon customer impression | | CampaignID | Int | Unique id for a discount campaign | | CouponID | Int | Unique id for a discount coupon | | RedemptionStatus | Int | target) (0-Coupon not redeemed,  1 - Coupon redeemed) | |
| Transaction | |  |  |  | | --- | --- | --- | | Column | Data type | Description | | TransactionID | int | Unique id for transaction | | CustomerID | int | Unique id for a customer | | ItemID | int | Unique id for an item | | TrainID | int | Unique id for an train | | Date | datetime | Date of Transaction | | Quantity | int | Quantity of item bought | | SellingPrice | money | Sales value of the transaction | | OtherDiscount | money | Discount from other sources such as manufacturer coupon/loyalty card | | CouponDiscount | money | Discount availed from retailer coupon | |
| Customer | |  |  |  | | --- | --- | --- | | Column | Data type | Description | | CustomerID | int | Unique id for a customer | | Name | nvarchar(20) | Customer name | | Gender | nvarchar(1) | Gender of customer | | AgeRange | nvarchar(50) | Age range of customer family in years | | MaritalStatus | nvarchar(50) | Married/Single | | City | nvarchar(50) | City name | | Phone | nvarchar(25) | Contact number | | Rented | int | 0 - not rented accommodation, 1 - rented accommodation | | FamilySize | int | Number of family members | | NoOfChild | int | Number of children in the family | | IncomeBracket | int | Label Encoded Income Bracket (Higher income corresponds to higher number) | |

**3.SOLUTION ARCHITECTURE**

 *Figure 2: High-level BI Solution Architecture*

As the figure 2 shows for the ETL processing, initially

**AmExpert\_SourceDB**: Source Database,

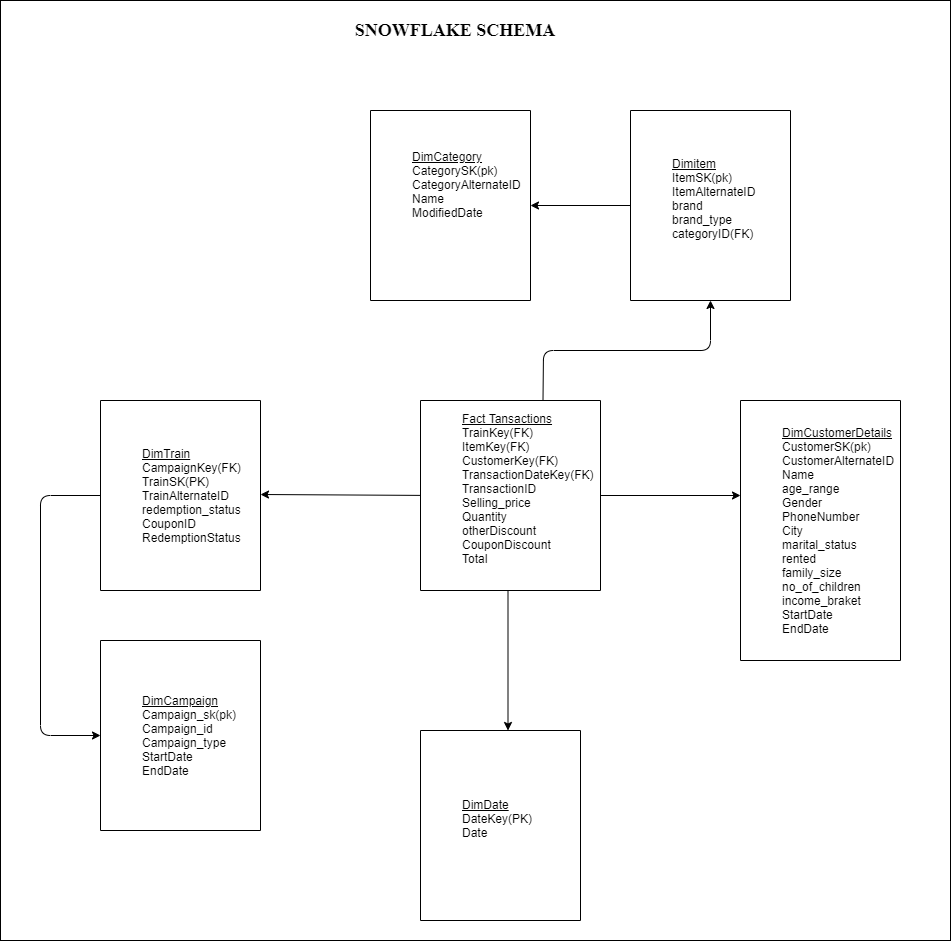
**category.txt**: Text file,

used for the data extraction to the Staging Destination.

**4.DATA WARE HOUSE DESIGN & DEVELOPMENT**

1. **Design**

The AmExpert\_DW (ware house) is designed according to the given below snowflake schema with one fact table (dbo.FactTransaction) and six dimension tables including Date dimension.



*Figure 3: Snowflake schema*

* Hierarchies

- DimCategory is applied as a hierarchical dimension of DimItem table.

- DimCampaign is applied as a hierarchical dimension of DimTrain table.

* Calculation

- Total payment is calculated in dbo.FactTransaction.Total

(([SellingPrice]-([OtherDiscount]+[CouponDiscount]))\*[Quantity])

1. **Assumptions**

* dbo.DimDate is added to the Data Warehouse for better performance.
* dbo.Transaction is used in creating the fact table.
* The transactions per customer was considered as the grain when designing.

1. **Slowly changing dimensions**

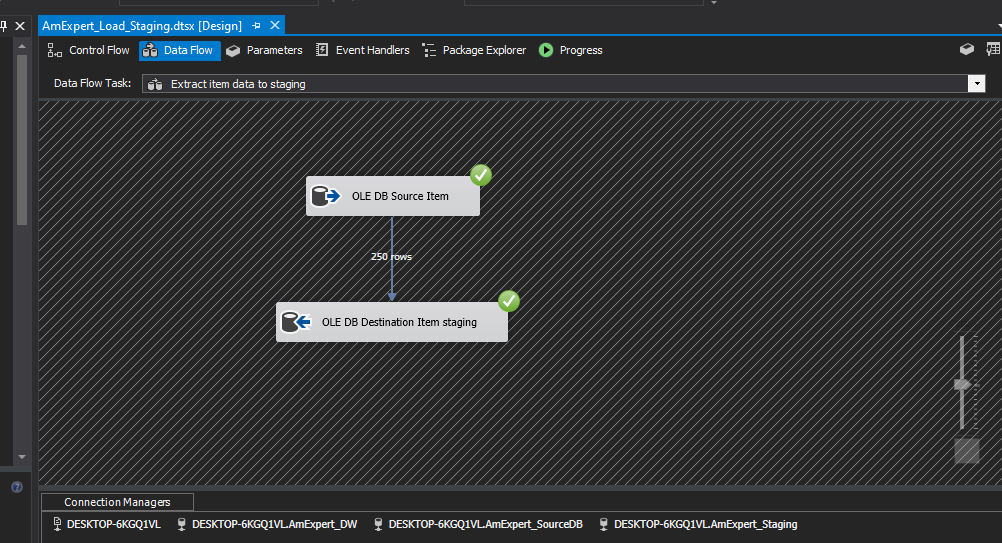
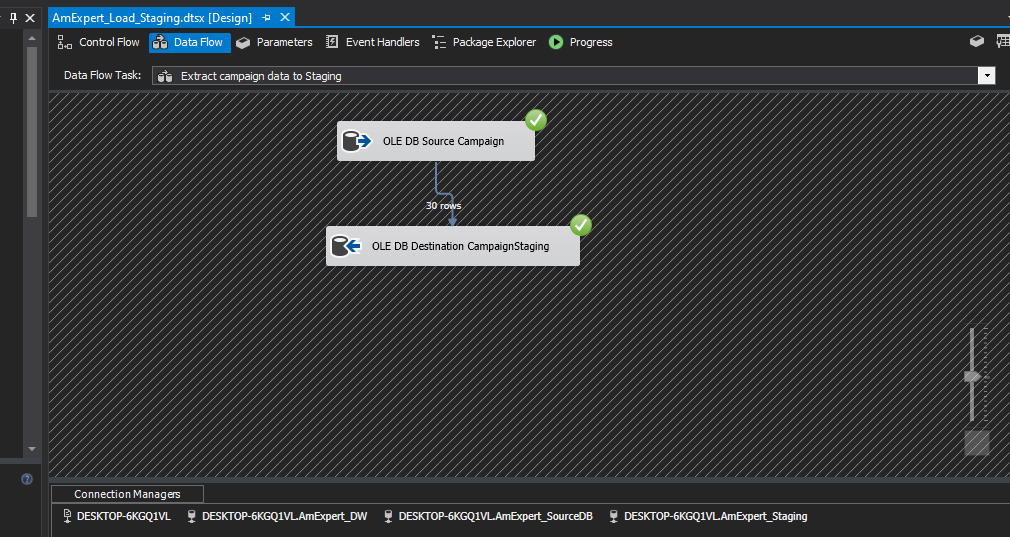
* Customer Details were considered as a slowly changing dimension

|  |  |
| --- | --- |
| **Dimension table** | **Attributes** |
| DimCustomerDeails | MaritalStatus (changing attribute)  PhoneNumber (changing attribute)  AgeRange (Hostorical)  City (Hostorical) |

**5.ETL DEVELOPMENT**

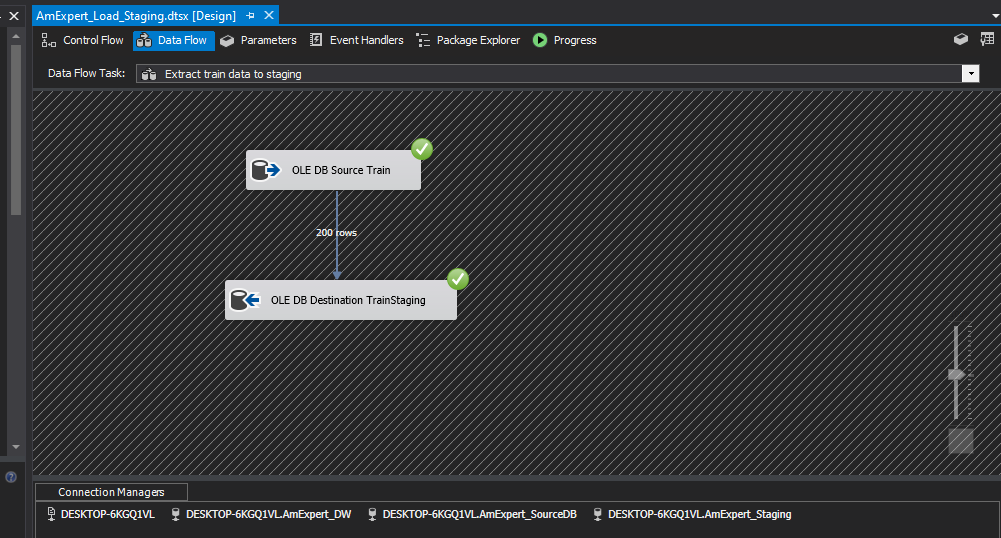
1. Data Extraction & Load into Staging tables

* Data Extraction is done by using the provided data sources mentioned above in Visual Studio 2015 (Data Tool) development environment. The text file and the source database were used here.
* Initially, **OLE DB SOURCE** (for source database) or **FLAT FILE SOURCE** (for flat files txt) is used to extract data for the Staging criteria. In this step developer is able to select the columns what would be included in the Staging from available data columns. As the next step of Staging, **OLE DB DESTINATION** has applied here to storing data in the Staging tables of **AmExpert\_Staging**.

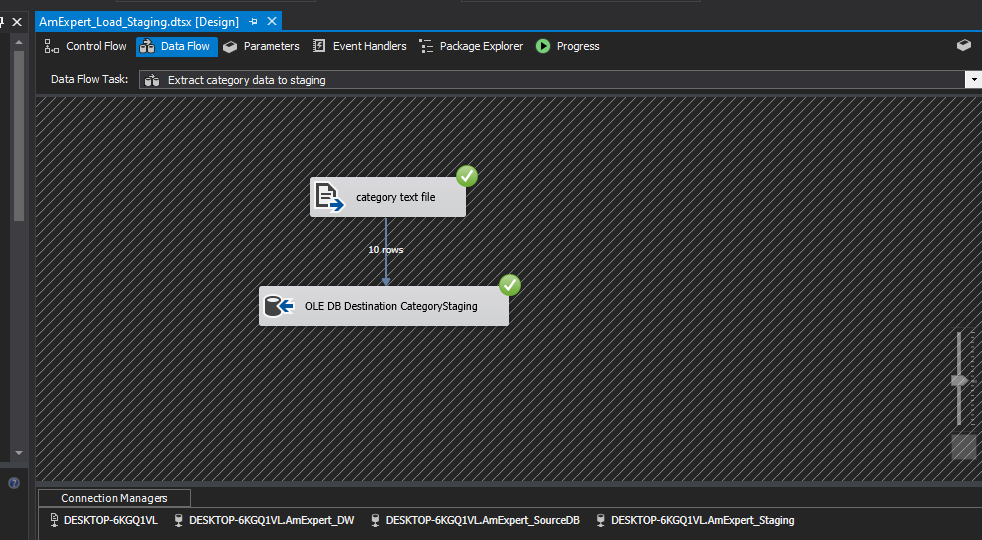


Campaign data is extracted from the Campaign table in the source database and inserted to the CampaignStaging table

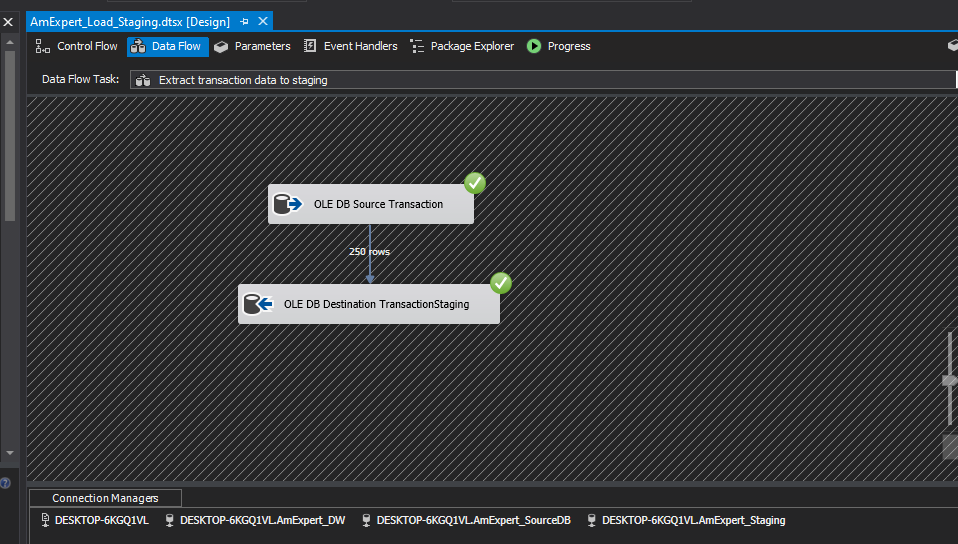
Item data is extracted from the Item table in the source database and inserted to the ItemStaging table



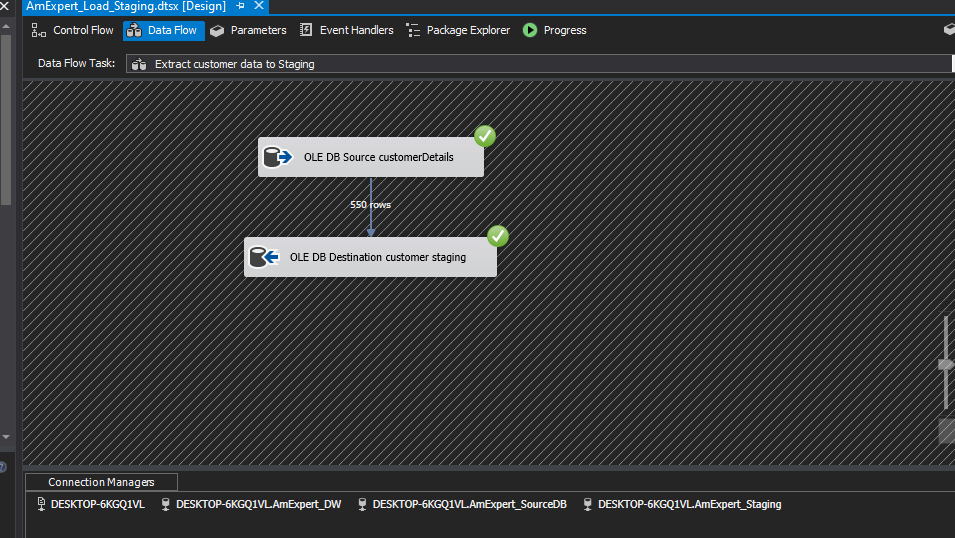
Train data is extracted from the Train table in the source database and inserted to the TrainStaging table



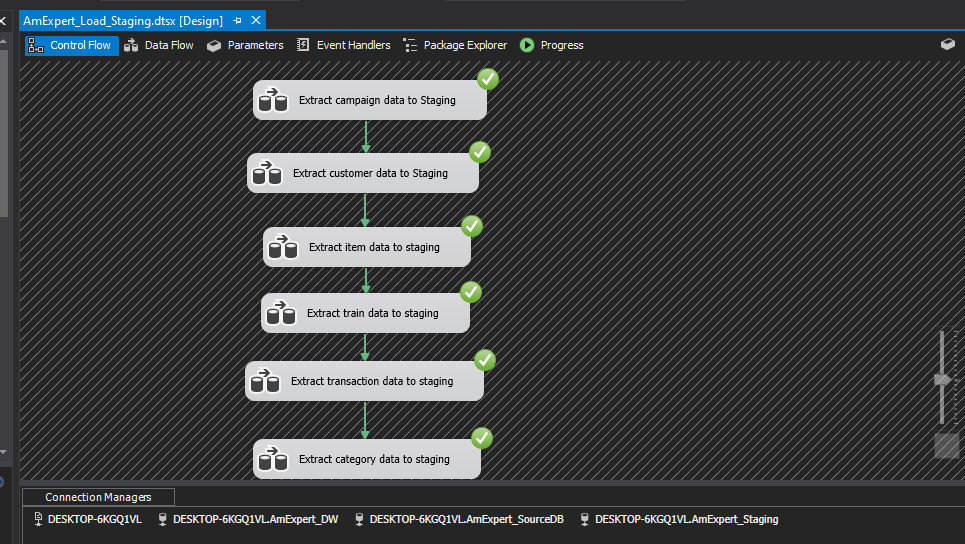
Category data is extracted from the category table in the text file and inserted to the CategoryStaging table



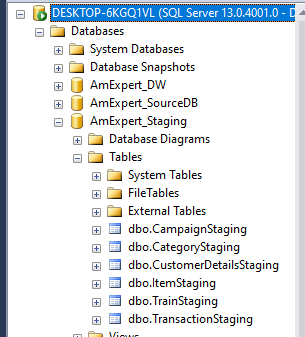
transaction data is extracted from the Transaction table in the source database and inserted to the TransactionStaging table



Customer data is extracted from the Customer Details table in the source database and inserted to the CustomerStaging table



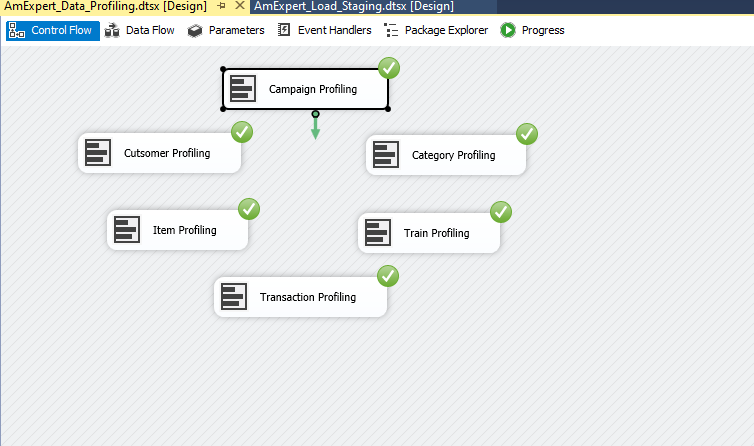
The Control Flow of ‘Extract Data and Load into Staging’ Step can illustrate as the give figure.



Staging Tables created and values inserted

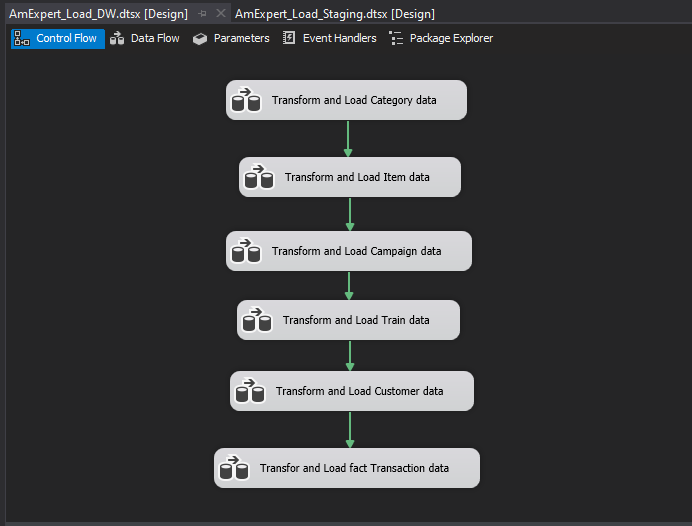
1. Data Profiling

Data Profiling provides the means of analyzing large amount of data using different kind of processes. In this step, null values, repeated values and quality of the data is checked.

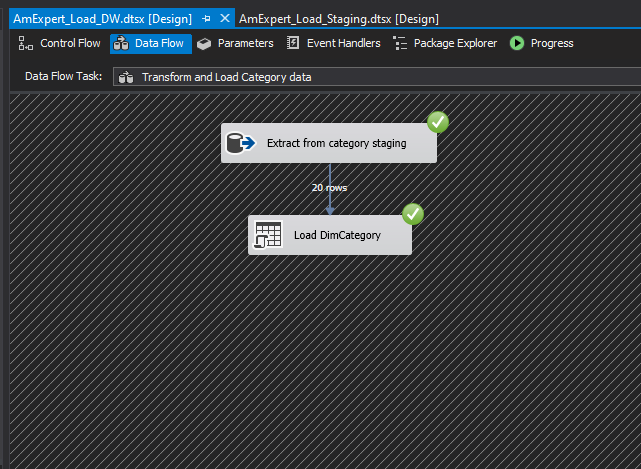
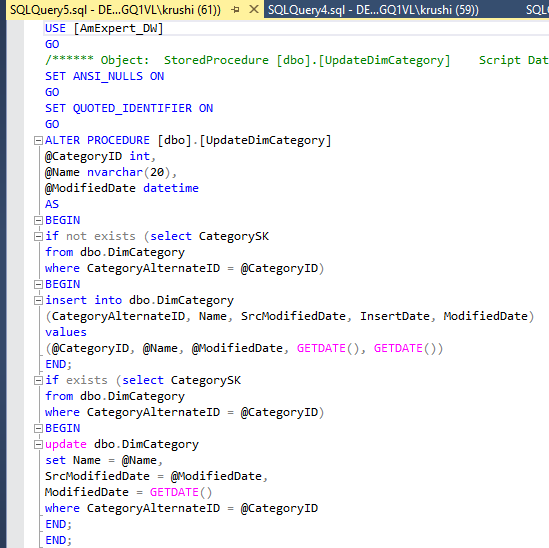


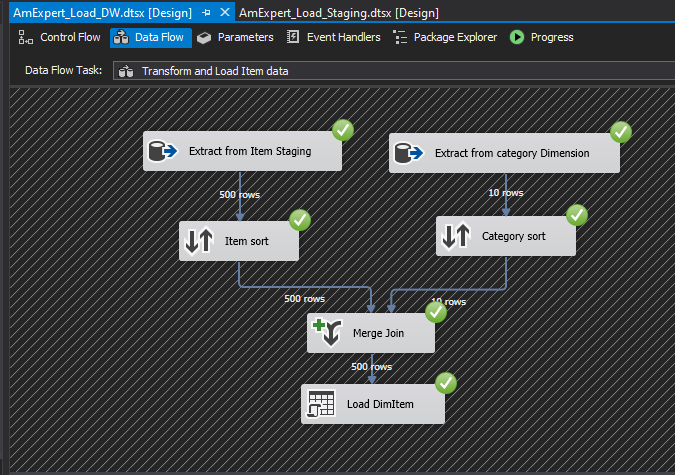
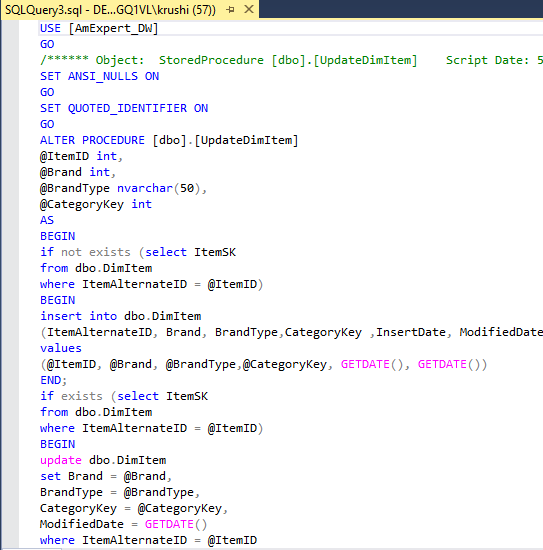
* Every staging table is profiled and saved in a selected location.
* As the figure shows, after the Staging step doing this task shows the things what the developer has to consider about the data which are stored in staging table and the developer is able to identify the issues with staging data by data profiling (such as null values).
* The given figure illustrated the complete part of Data Profiling relevant to the Staging.

1. Data Transformation and Loading

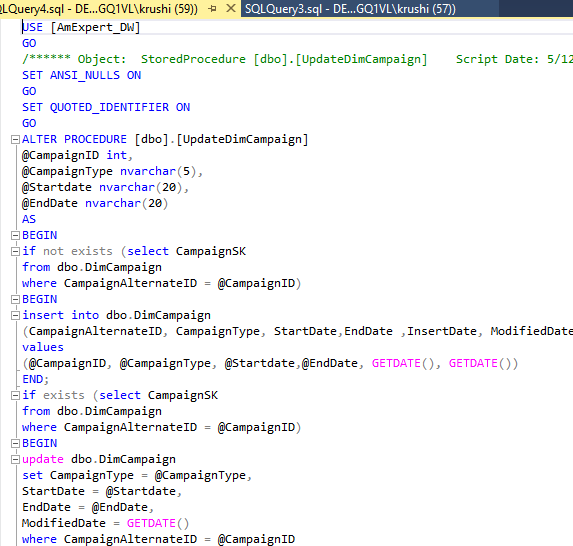
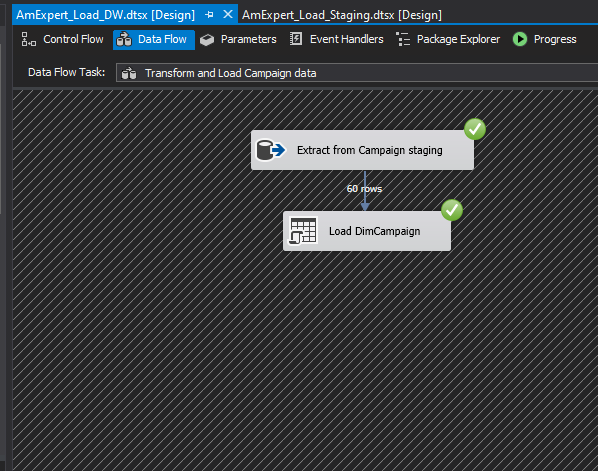
* Data Transformation is developed according to the dimensional modeling designed above.

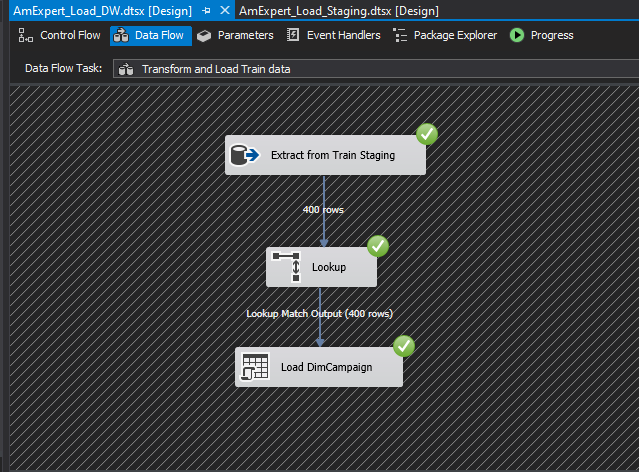
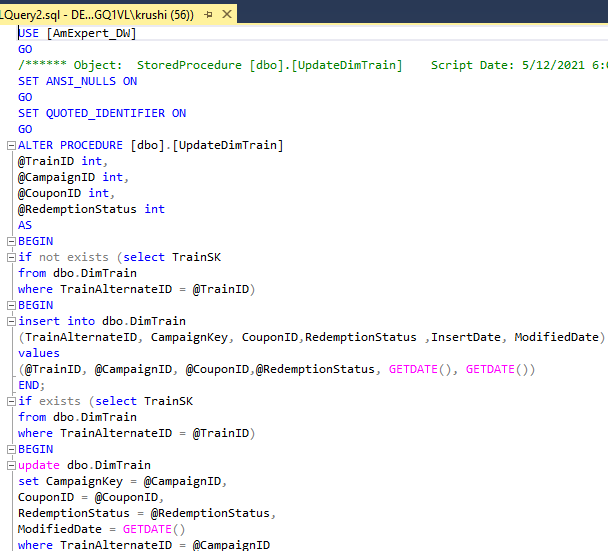
In this step, the Dimension Tables created in AmExpert\_DW are loaded with the data of relevant staging tables.

* Category data is loaded to the DimCategory
* UpdateDimCategory procedure is used to check whether the data inserted or not.

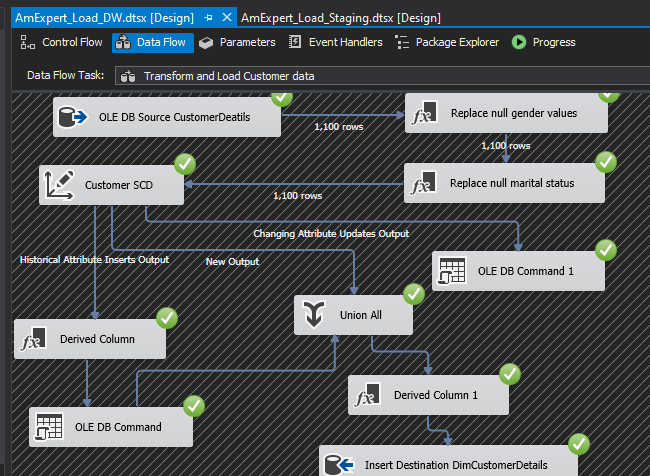


* UpdateDimItem procedure is used to check whether the data inserted or not.
* Sort and merge transformation tasks are used.
* Item data sorted according to the category ID , and category data extracted from the category dimensional table and sorted according to the category ID and merged by Mergr Join component.
* Item data is loaded to the DimItem table.



* Campaign data is loaded to the DimCampaign table.
* UpdateDimCampaign procedure is used to check whether the data inserted or not.
* LookUp transformation task is used.
* Train data is loaded to the DimTrain table.
* UpdateDimTrain procedure is used to check whether the data inserted or not.

**Loading Slowly Changing Dimension**

* DimCustomerDetails is the slowly changing dimension in this dimensional modeling.
* In order to load data to Dimension table, the slowly changing dimensions (historical) have two specific columns as StartDate & EndDate to ensure that the data is valid at the moment.
* slowly changing dimension wizard let the developer to select the Dimension table, Business keys of the dimension and what would be the slowly changing attributes.
* Initially data cleansing is done in order to remove null values from the data source table.
* Based on data profiling result, null values from gender column, and marital status column were removed.
* As mentioned earlier under assumptions, customer details were considered as slowly changing details.
* The below mentioned columns were set as changing attributes:
  + - 1. Marital status (changing)
      2. Phone number (changing)
      3. Age range (historical)
      4. City (historical)
* After extracting data from the Customer staging table, it was sorted according to the customer id and as it was identified as a slowly changing dimension, it was connected as shown above and loaded data to the Customer dimension table.

**Load data to Fact table**

* The final step of Transformation & Loading is load data to fact table. According to the dimensional model, TransactionStaging table is used to insert values into DimTransaction table.
* FactTransaction table has one date key which are related to Date Dimension as TransactionDateKey.
* After loading to all the dimensions, lastly data was loaded to the fact table. The below steps were followed:

1. Data extracted from the customer transaction staging

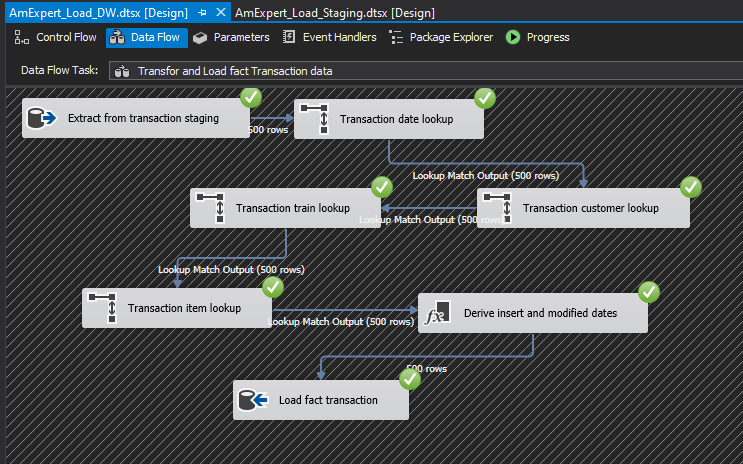
2. Join operation is done for the date using look up.

3. Join operation is done for the customer using look up.

4. Join operation is done for the train using look up.

5. Join operation is done for the Item using look up.

6. insert and modified date were derived.

7. Fact details loaded to the DimTransaction table.

* Fact details were added to the FactTransaction table.

