Documenting ETL Rules using CA Erwin Data Modeler

By Sampath Kumar

Abstract

In any data warehouse development project some of the major challenges include

- Effective capture and maintenance of metadata information in data model such as data source ,transformation rules and data synchronization etc
- Effective communication of captured metadata information by data modeler to other teams such as ETL

This document covers features in CA Erwin Data Modeler which can be leveraged for capturing the metadata information such as **E**xtract **T**ransform **L**oad (ETL) rules. This document explains step by step of how to capture the ETL information using Erwin and also covers the generation of reports with the captured information to communicate effectively to other teams.

Introduction

The data warehouse combines information from several **O**n**l**ine **T**ransactional **P**rocessing (OLTP) systems and archive data into a single decision support system. It can be either relational or non relational data source (both structured and unstructured data). In order to keep the data in synch with the operational system it's very essential to capture the data source for each column in the data warehouse and information of when and how the data is updated.

So in a nutshell the following information needs to be captured in any data warehouse environment

- Source of data
- Transformation rules-The method in which the data is getting extracted, transformed and loaded
- Frequency: The frequency and timing of data warehouse updates.

In many organizations it used to be a separate document apart from data model but it becomes very hard to maintain document and data model in synch.

Why it's important

Data modeling is the first step which converts the business rules into a data model and the data modeler is the one who understand the rules from business counterparts (both in a structured and unstructured way). As a result of this, the data modeler captures most of the business rules directly in the data model and some of them (such as data source, transformation rules and frequency above) needs to be passed on to other teams such as DBA and ETL .It's very essential to capture all data related business rules as a part of data modeling effort to avoid getting lost. CA Erwin Data Modeler provides effective way to capture this information and as a data modeler it should be captured as part of modeling efforts.

Approach

This document covers how the above challenges can be addressed using Data Transformation and Data Movement features available in CA Erwin Data Modeler. To explain better there will be a simple running example throughout this document which will navigate step by step.

Overview

The CA Erwin Data Modeler has come up with the following salient features to capture the metadata information effectively.

- Data warehouse Sources Dialog: to define sources of data for your data warehouse
- Columns Editor: to document the data warehouse source assignments and transform the information for each column in the dimensional model in the data source tab.
- Data Movement Rules Editor: to document the data warehouse maintenance processes required to regularly update each table in your dimensional model.

Let's explore these features in detail in the rest of the document using simple example of Customer_Dim.

Customer_Dim

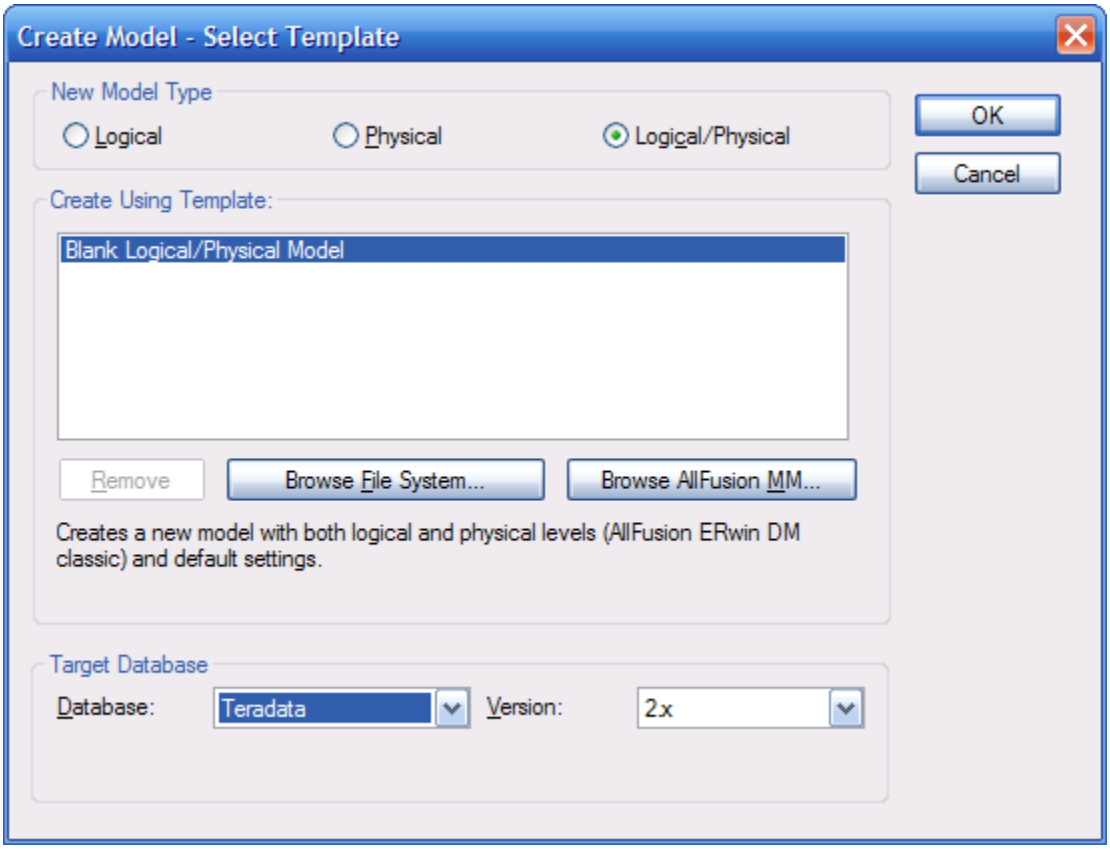
Let's take a fictitious example of an entity Customer Dimension to explain the above features. Let's assume that it's sourced from multiple operational systems (relational DB), attributes having different transformation comments and the frequency of customer information getting updated is daily.

The following are the key attributes in the Customer dimension

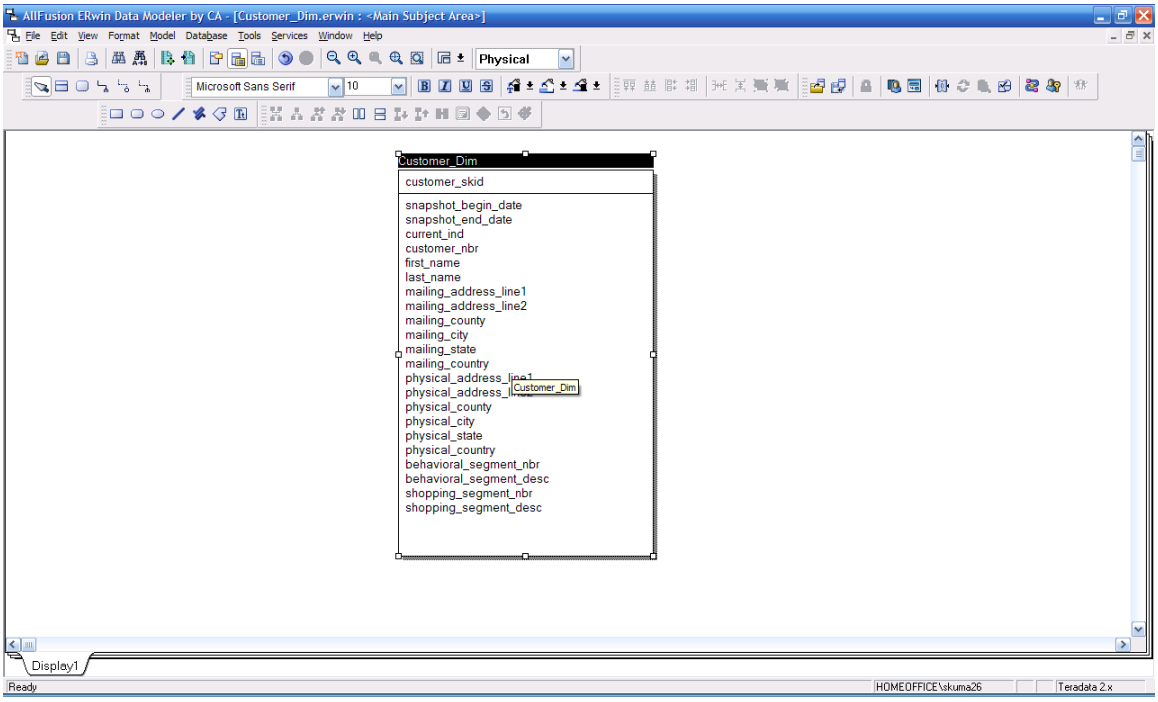
- Snapshot
    - customer_SKID
    - snapshot_Begin_Date
    - snapshot_End_Date
    - current_ind
- Basic Information
    - Customer name
    - Customer Date of Birth
    - Driving License
- Address
    - Mailing Address
    - Physical Address
- Communication
    - Email Address
    - Phone
    - Fax
- Segmentation
    - Shopping
    - Behavior

Capturing Data Source

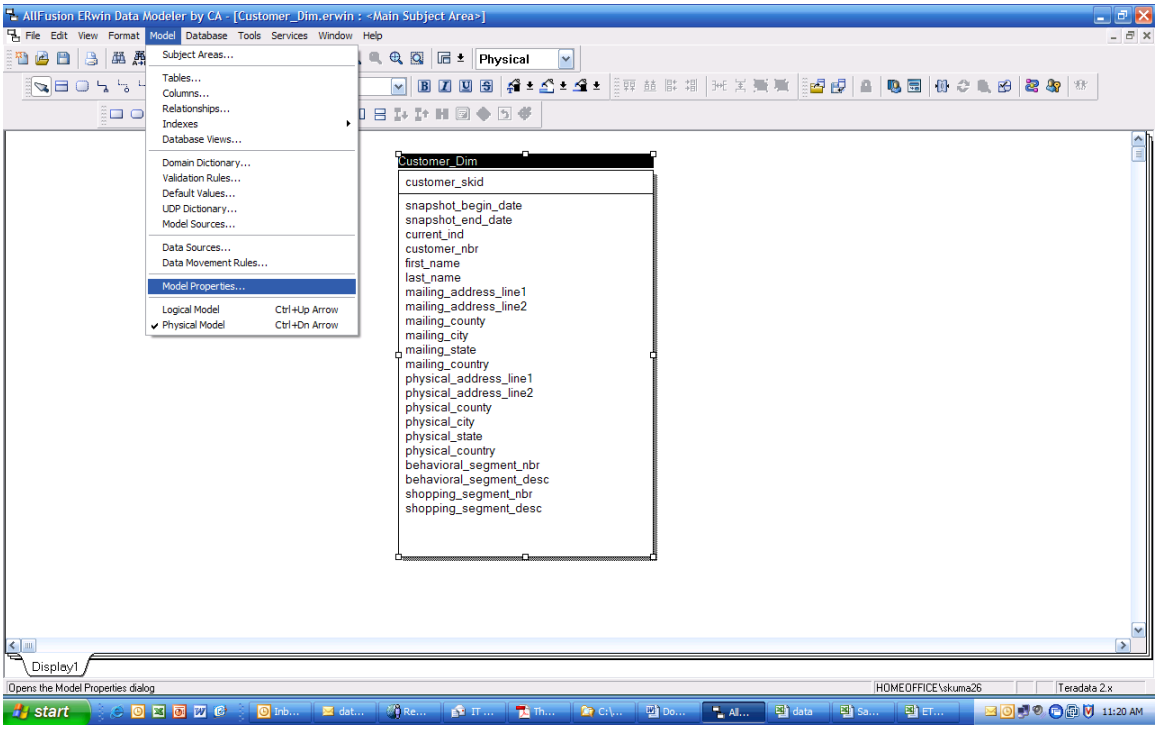
In order to proceed further lets create the empty data model using “Create Model” (File –>New) of the model type Logical/Physical and target database as Teradata

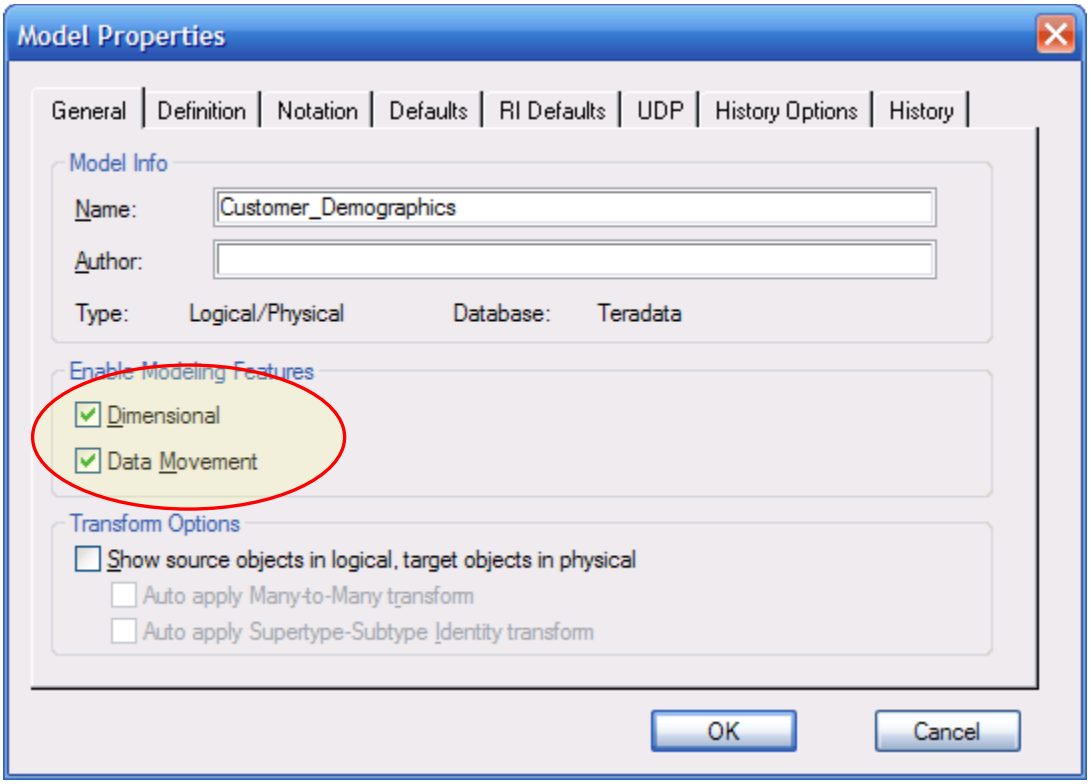


Let’s create the Customer_Dim table and add the attributes

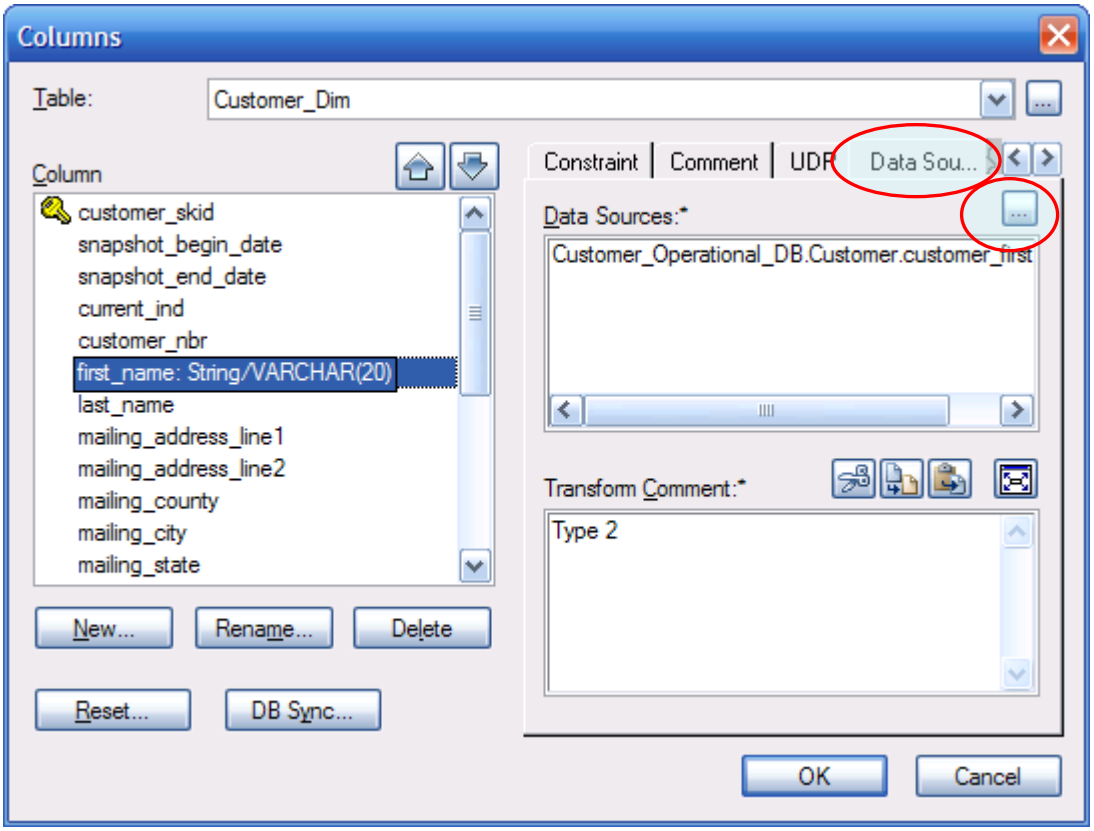


In order to make this model as dimensional model and to capture the data movement rules **goto Model->Model Properties** and select the check box for Dimensional and Data Movement

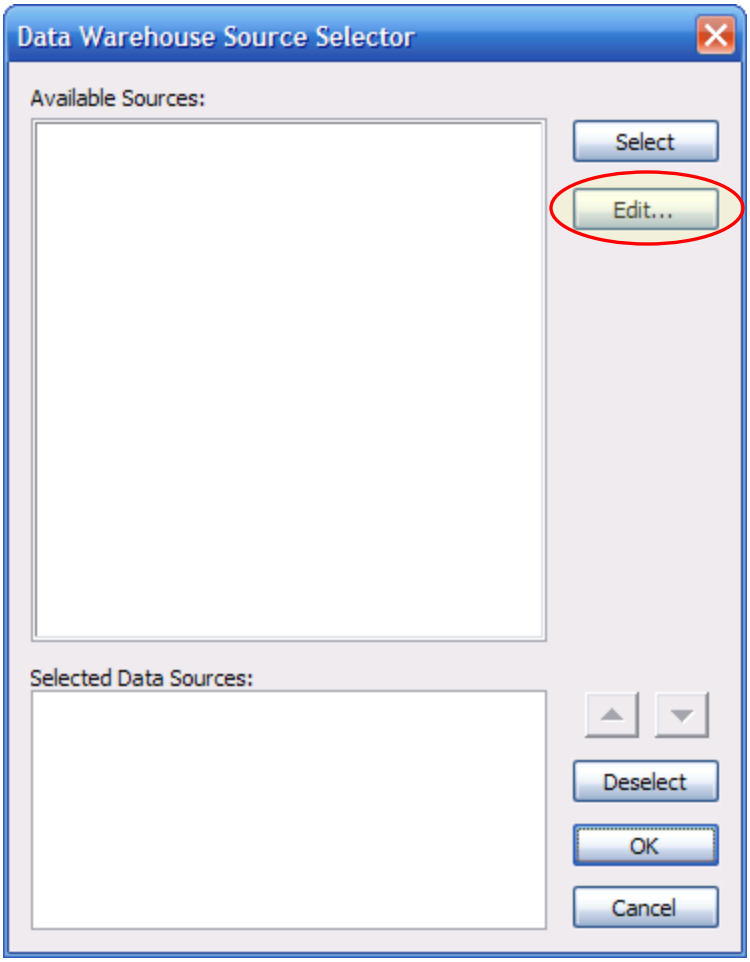




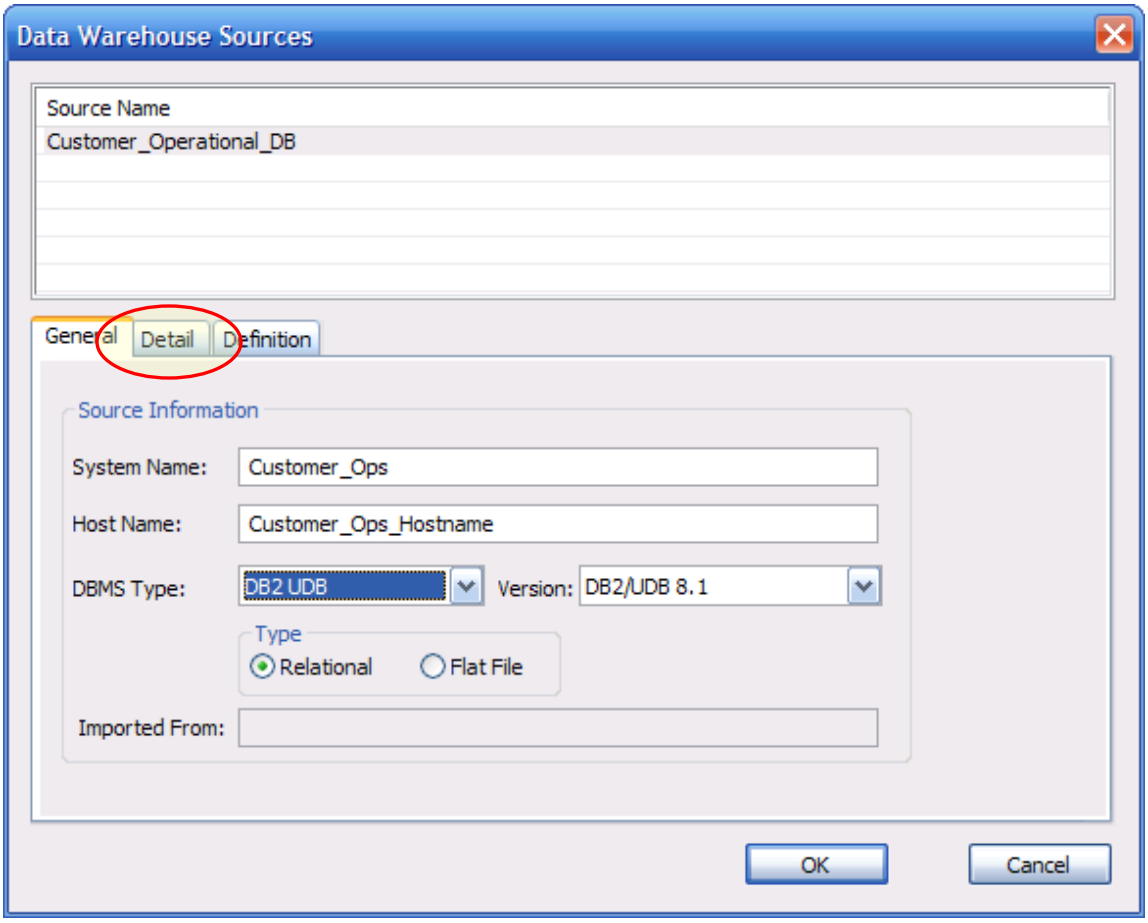
You can observe that in the columns wizard “Data Source” tab will be enabled since the “Data Movement” was selected.



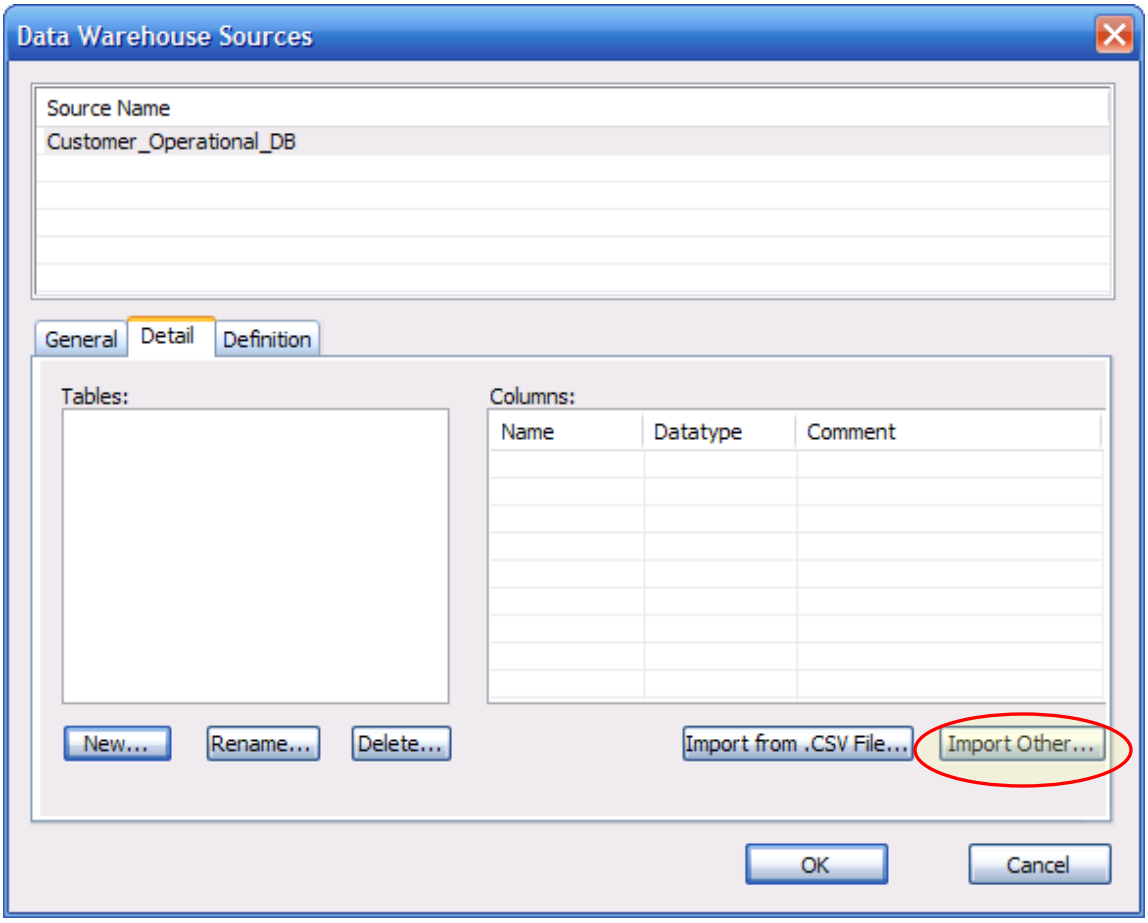
Click on the button (with 3 dots) to open the “Data sources” wizard to create the data source which can be used for mapping the columns



Click the edit button of “Data Warehouse Source Selector”, to open the below screen.
In the source information, provide the Operational Source System name, operational database host name, operational DBMS type. Select “Type” as Relational if the operational system is relational or flat file if it’s from the file feed.

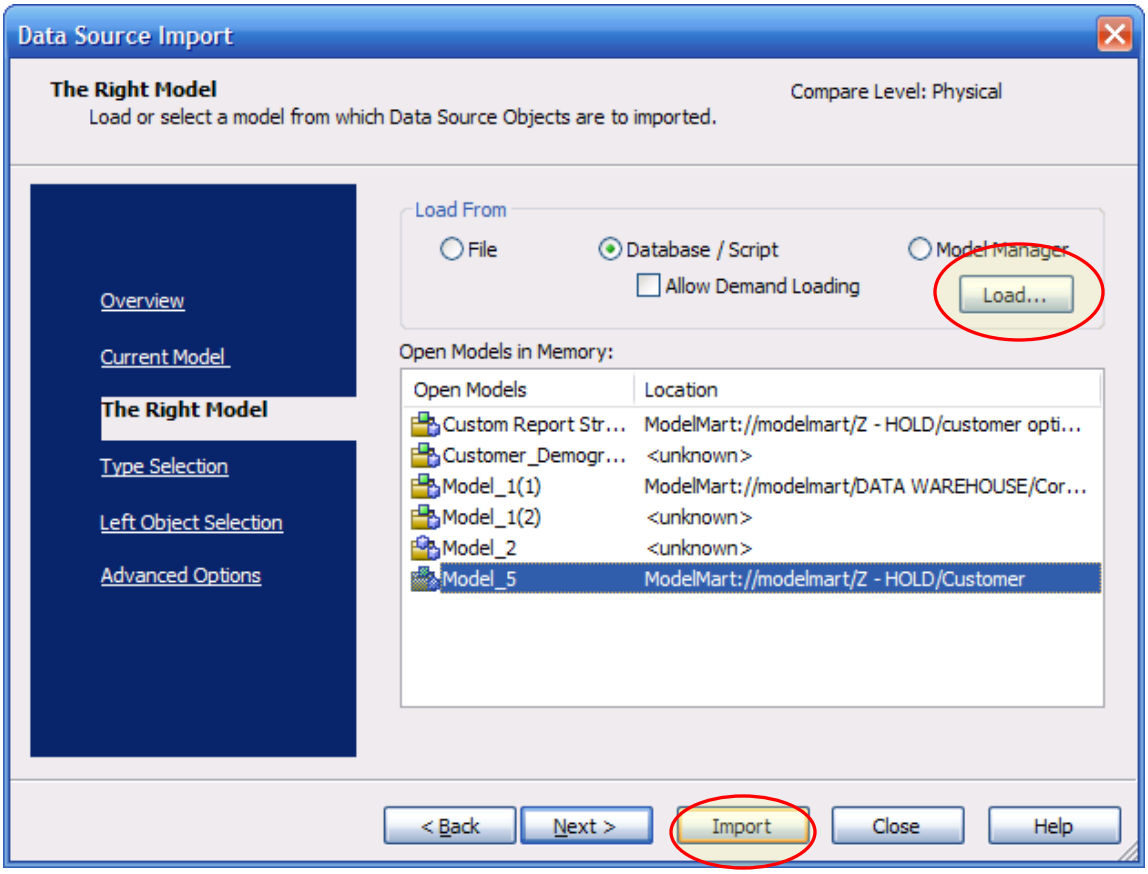


Open the tab “Detail” which will provide the options of **“Import from CSV”** and **“Import other”.** Click the **“Import Other”**

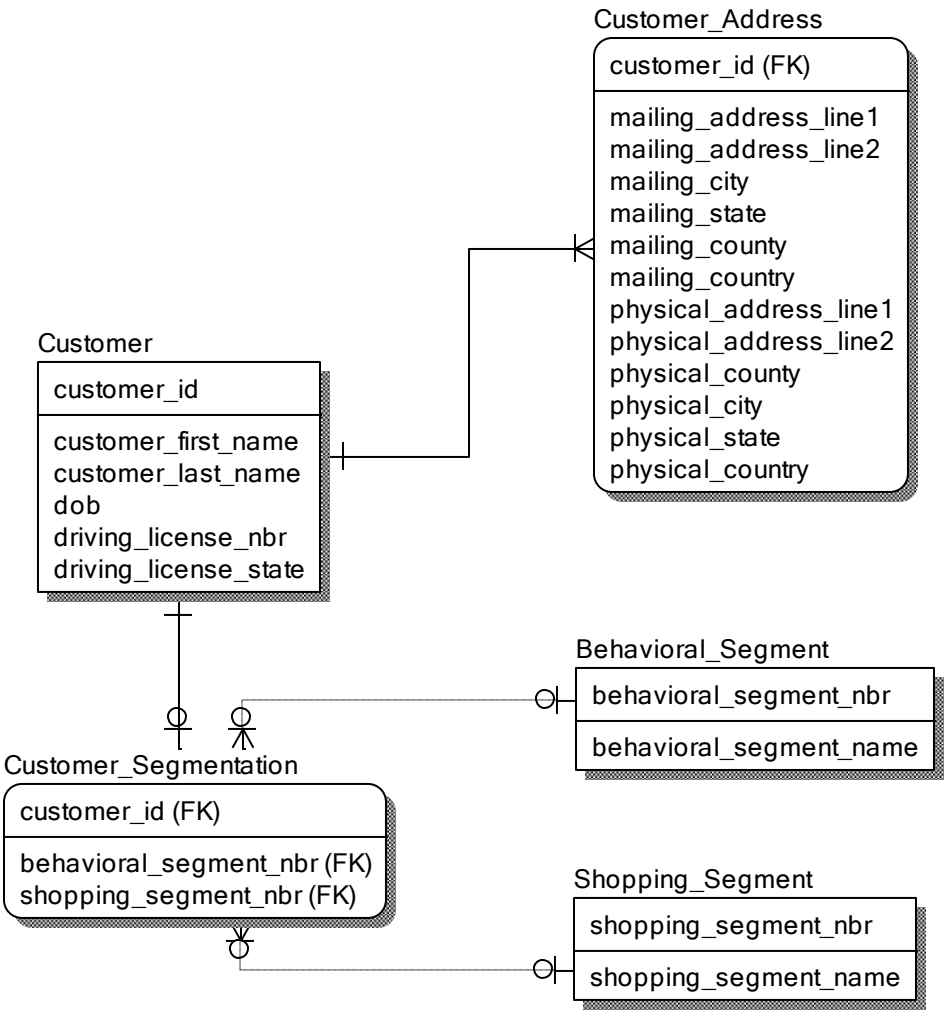


The “Import other” provides three options to import the table structure

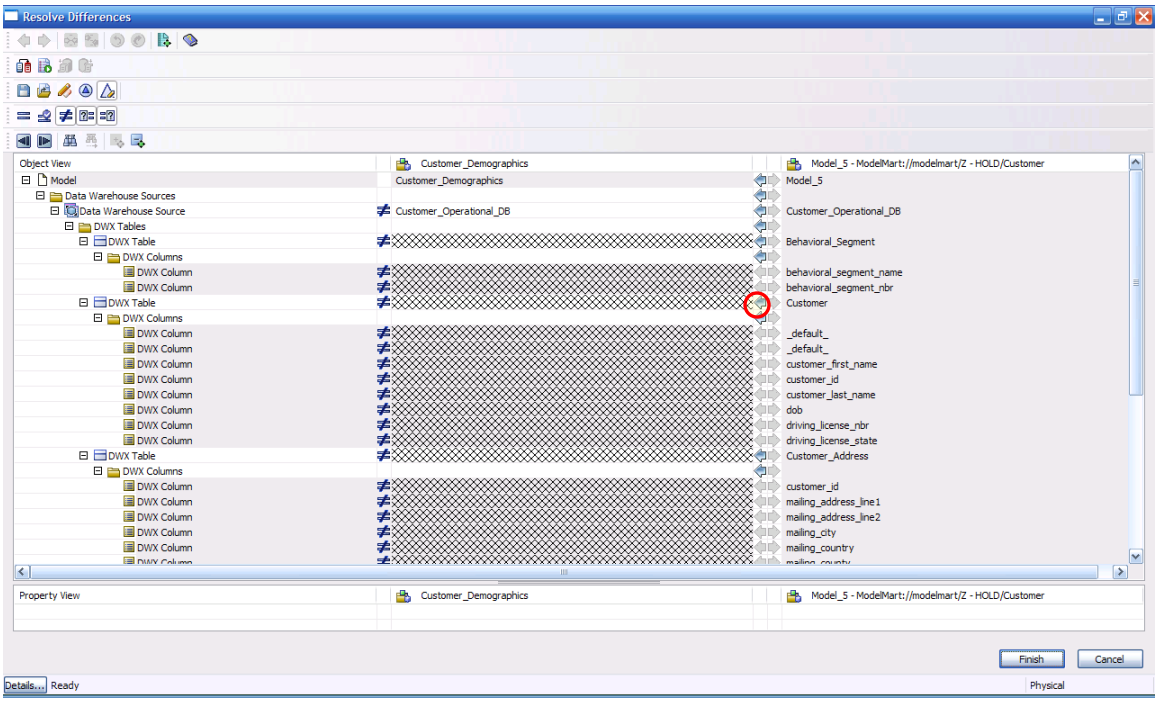
- Flat File
- Database/Script
- Model Manager



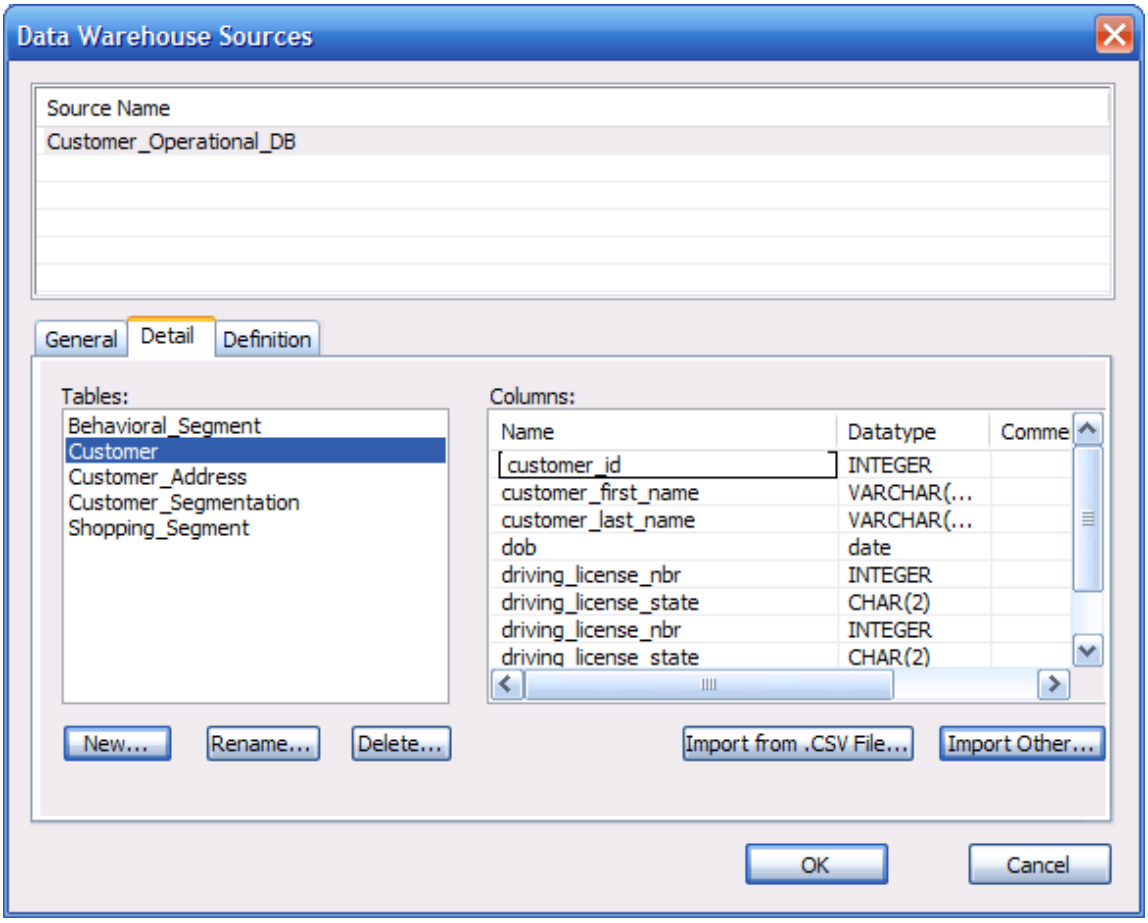
In our example let's assume that we are importing the table structure from the "Model Manager" .Select **"Model Manager"** and click on **"Load"** button open the Model Mart to load the model from the operation system. Let's assume the following is the high level model of Customer data in the operational system and we want to map these tables to dimensions in data warehouse.



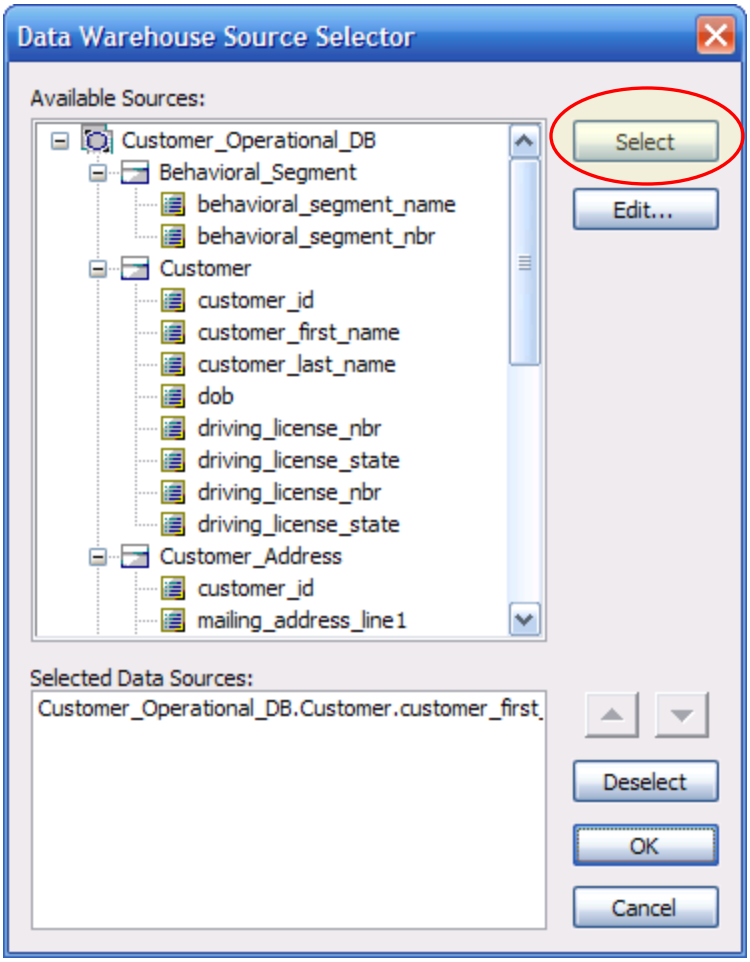
Click on **"Import"** to import the data model from the model mart which will display all tables in that data model .Select the Customer, Customer_Address, Customer_Segmentation, Behavioral_Segment and Shopping_Segment

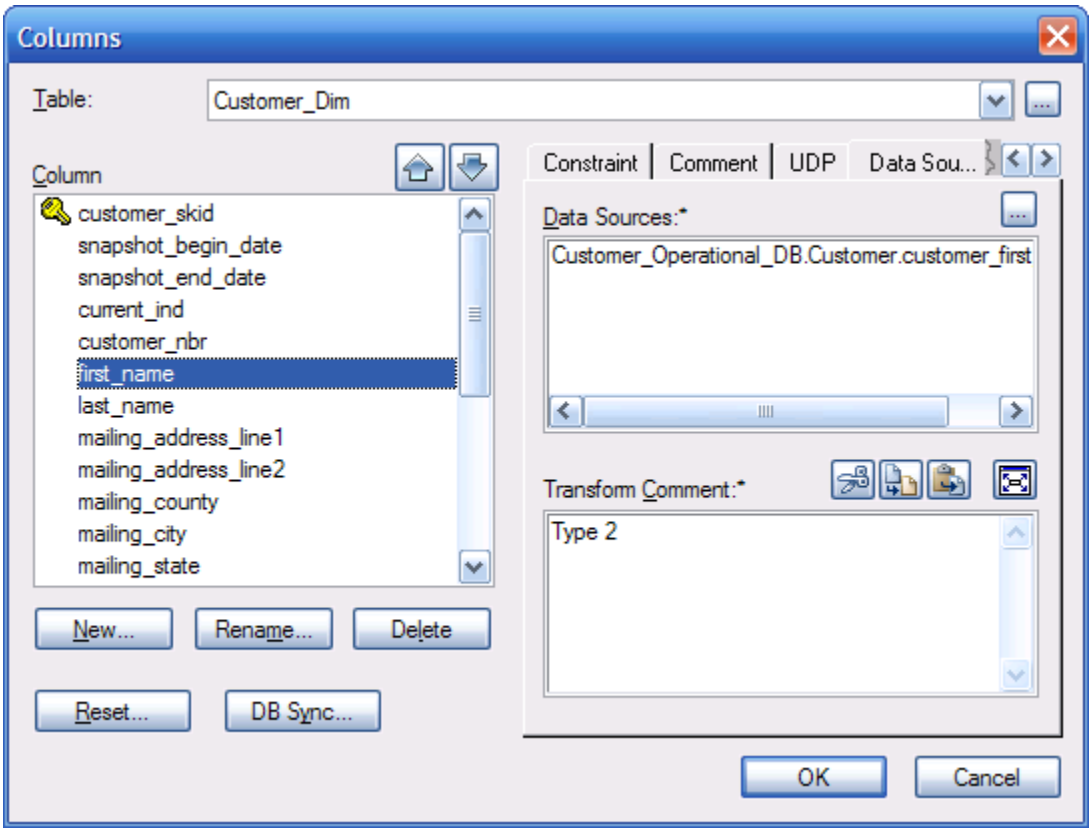


Click on the small arrow (highlighted in red circle) in the above diagram to select the required tables from the model mart. Once the required operational system tables are selected; it will be available for mapping as below.



Click **"Ok"** to see the source tables available for selection.

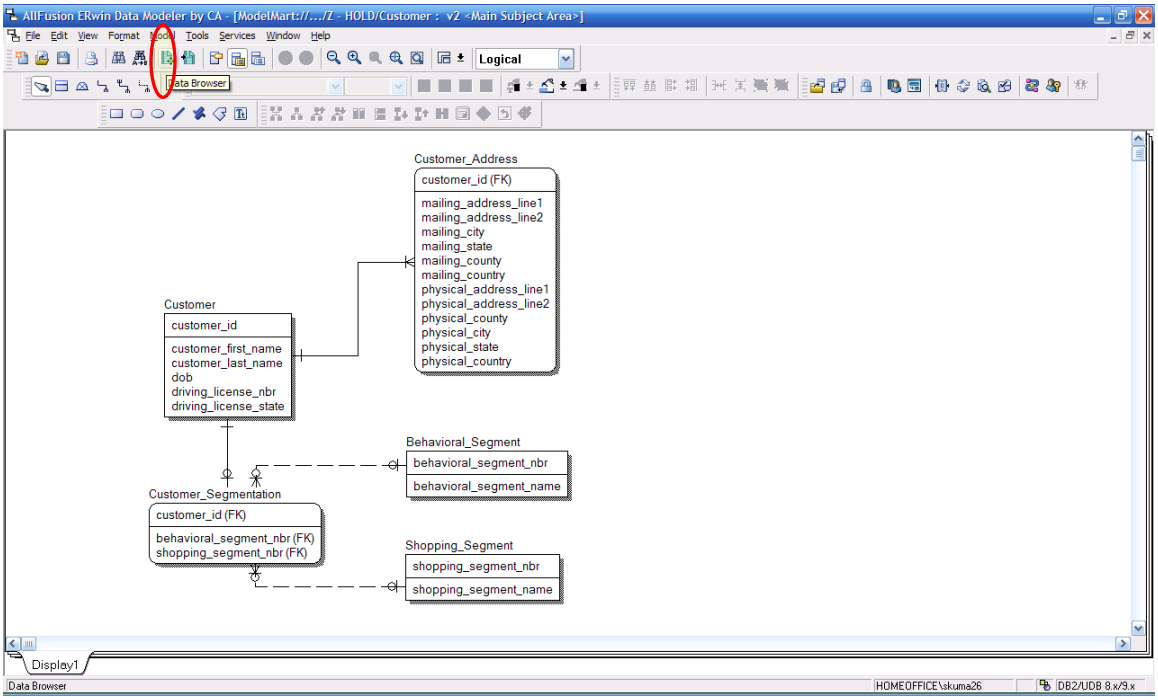

Data Warehouse Source Selector
Available Sources:
Customer_Operational_DB
Behavioral_Segment
behavioral_segment_name
behavioral_segment_nbr
Customer
customer_id
customer_first_name
customer_last_name
dob
driving_license_nbr
driving_license_state
driving_license_nbr
driving_license_state
Customer_Address
customer_id
mailing_address_line1
Select
Edit...
Selected Data Sources:
Customer_Operational_DB.Customer.customer_first_
Deselect
OK
Cancel



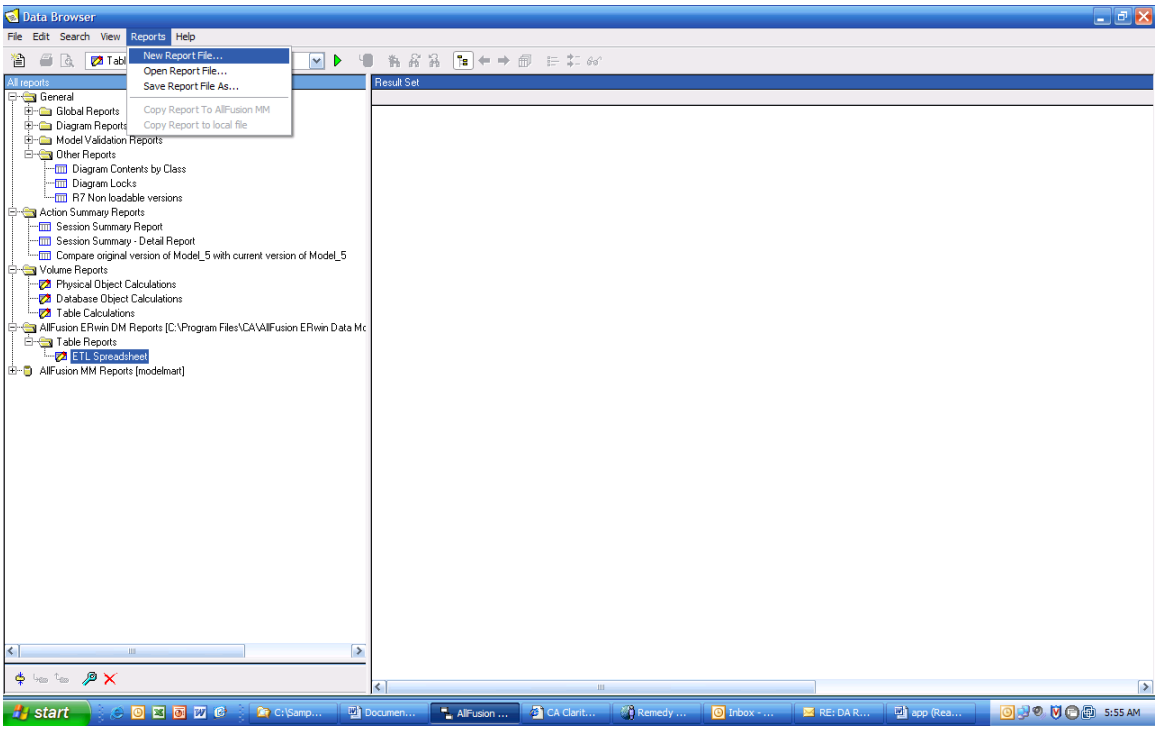
Once all the “Data Source” and “Transformation Comments” are entered, ETL Spreadsheet can be generated .Please follow the below steps to generate the ETL Spreadsheet.

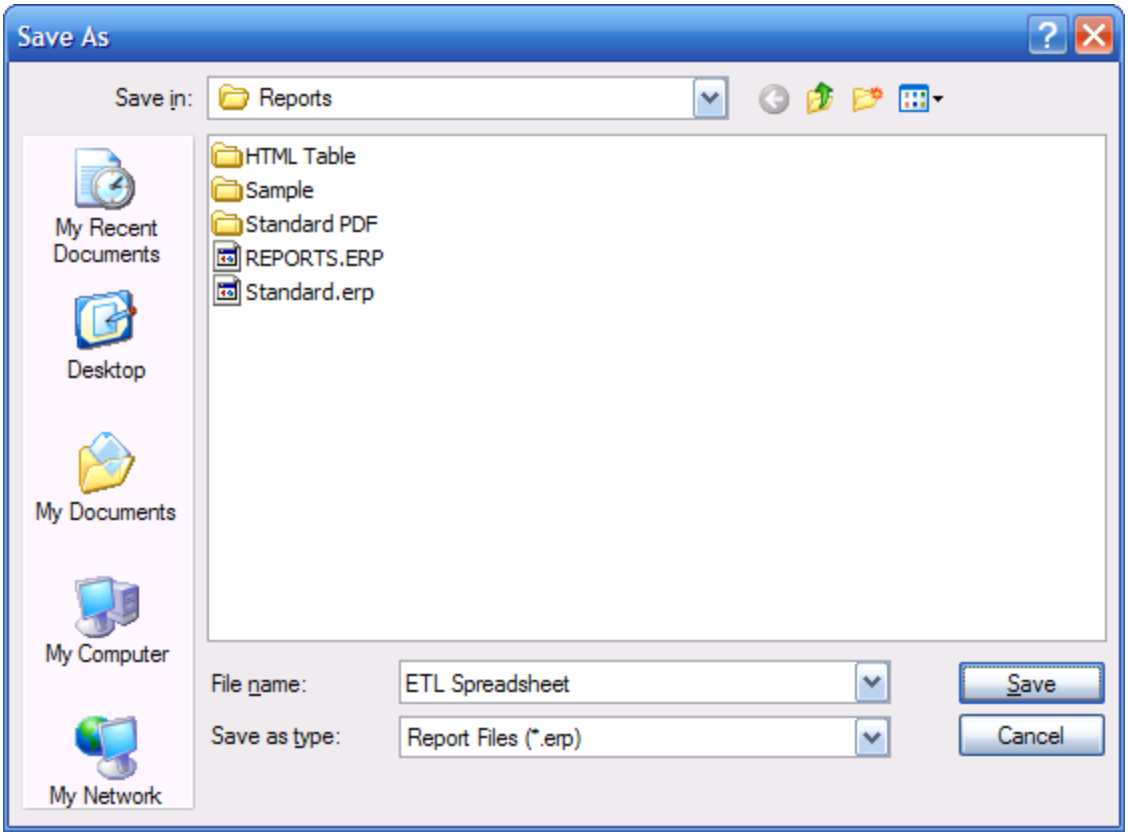
1. Highlight the tables for which you want generate the ETL spreadsheet
2. Click on the “Data Browser” icon which will open the Data browser window for all possible reports which can be generated.

Create new template “**ETL Spreadsheet.erp”** report using “Data Browser”.

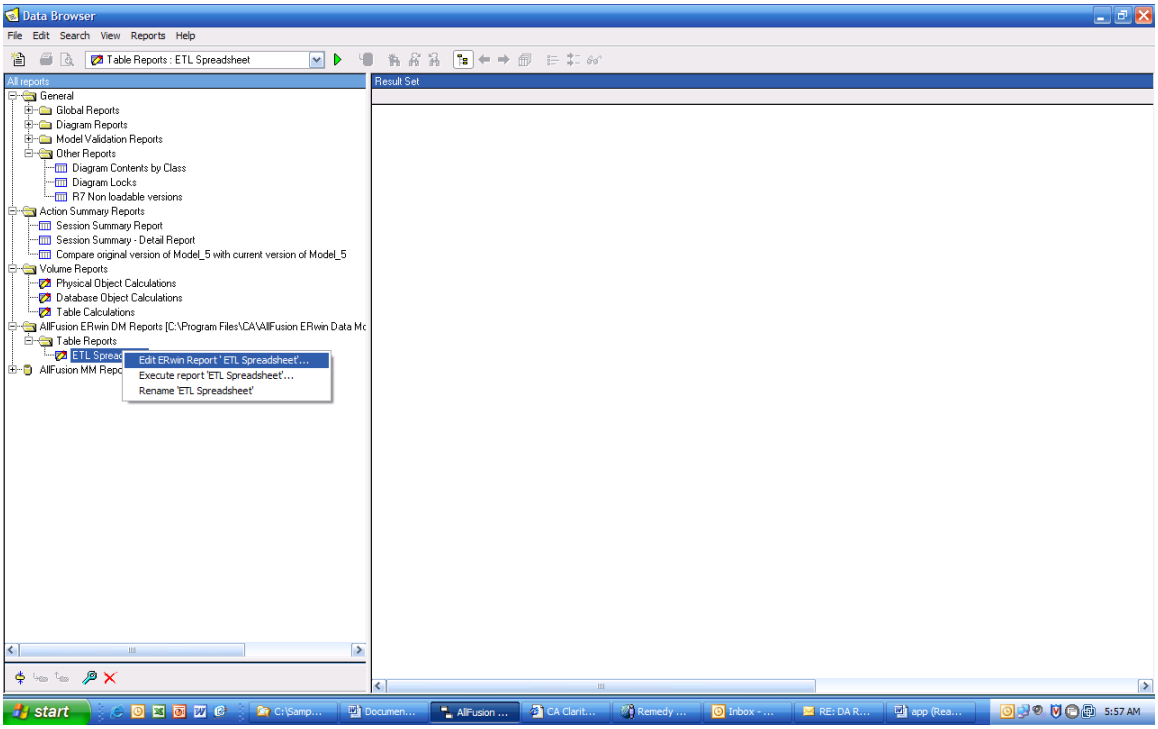


Open the “Reports” in the menu and select “New Reports File” .Create the report named “ETL Spreadsheet.erp“

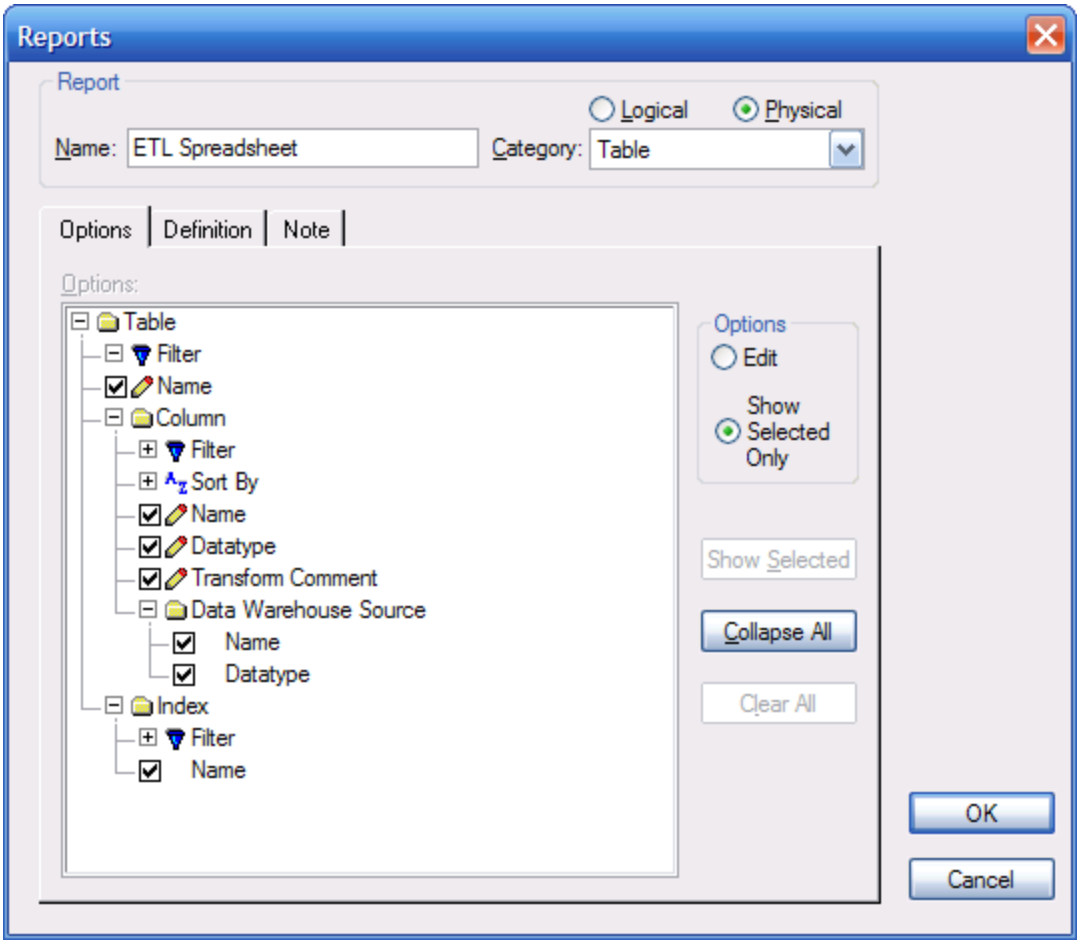




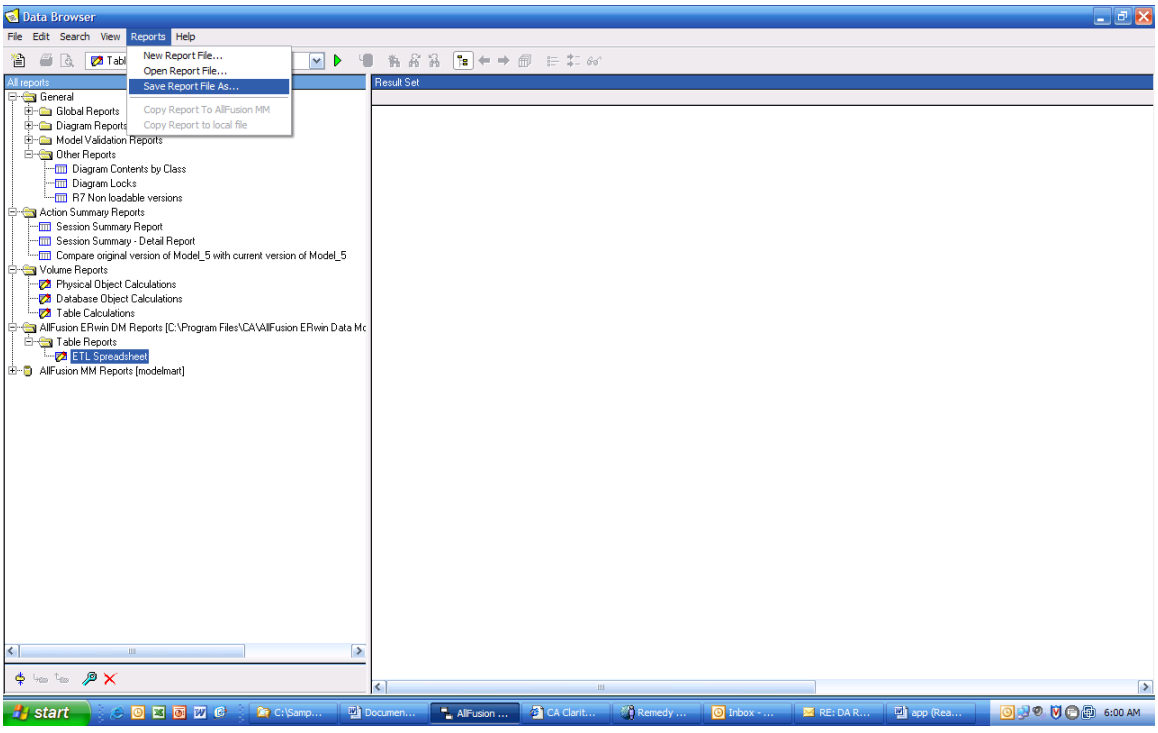
Right click the newly generated report click Edit **ERwin Report 'ETL Spreadsheet'**

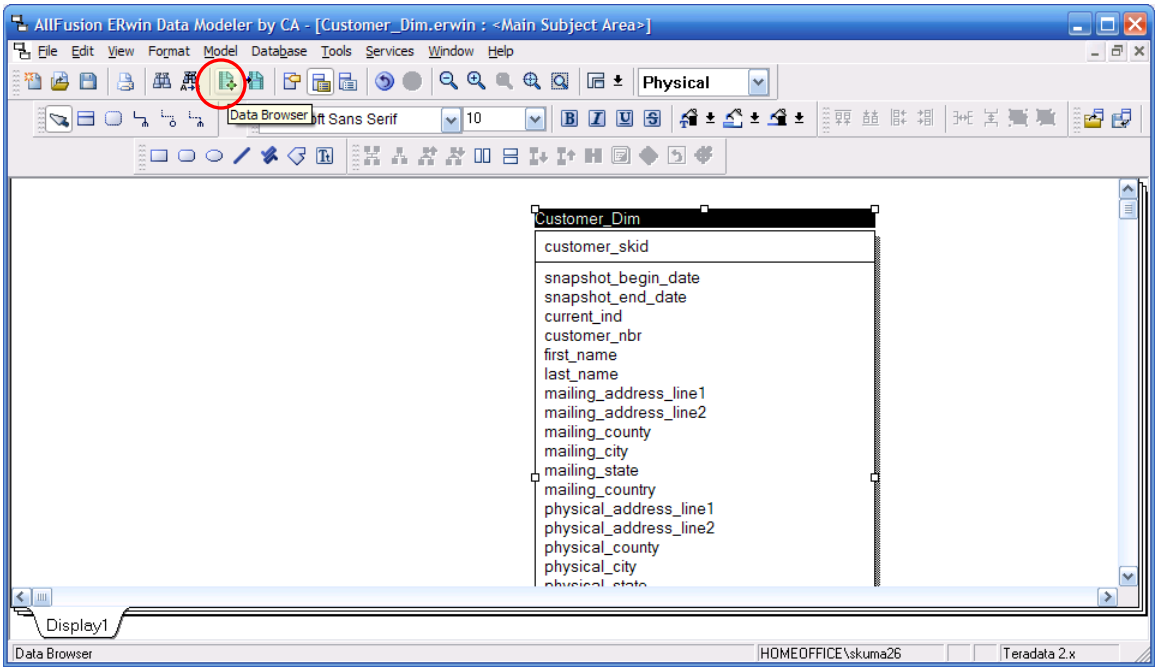


Make sure only the following options are selected which is relevant for ETL.

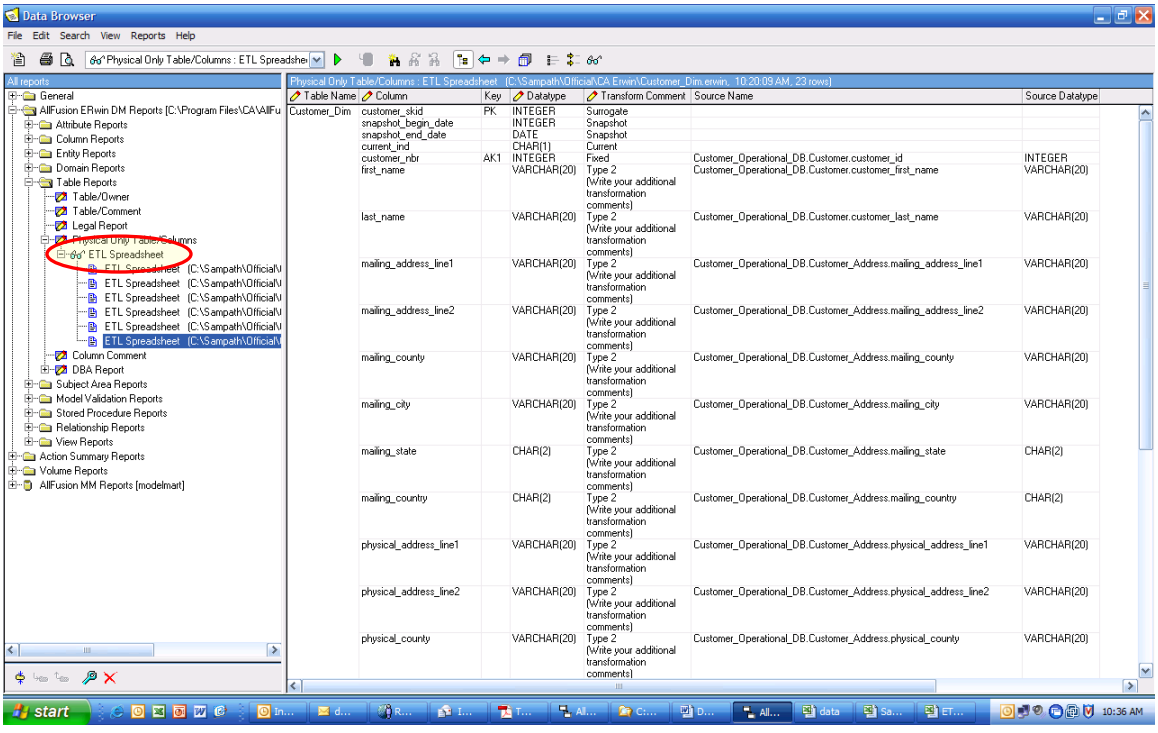


Save the report to make sure selected columns are stored in the report template.

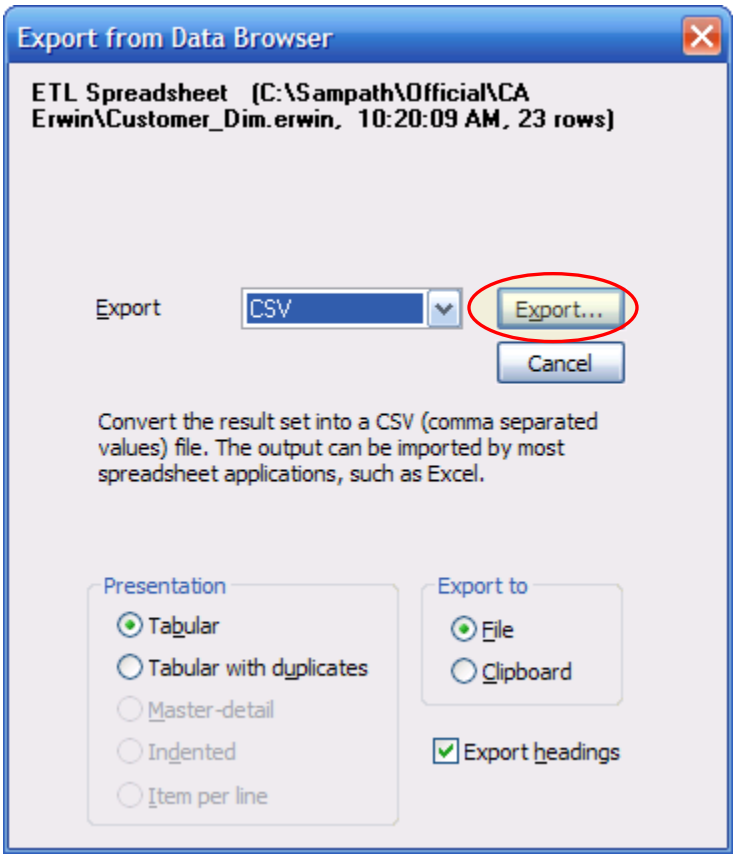


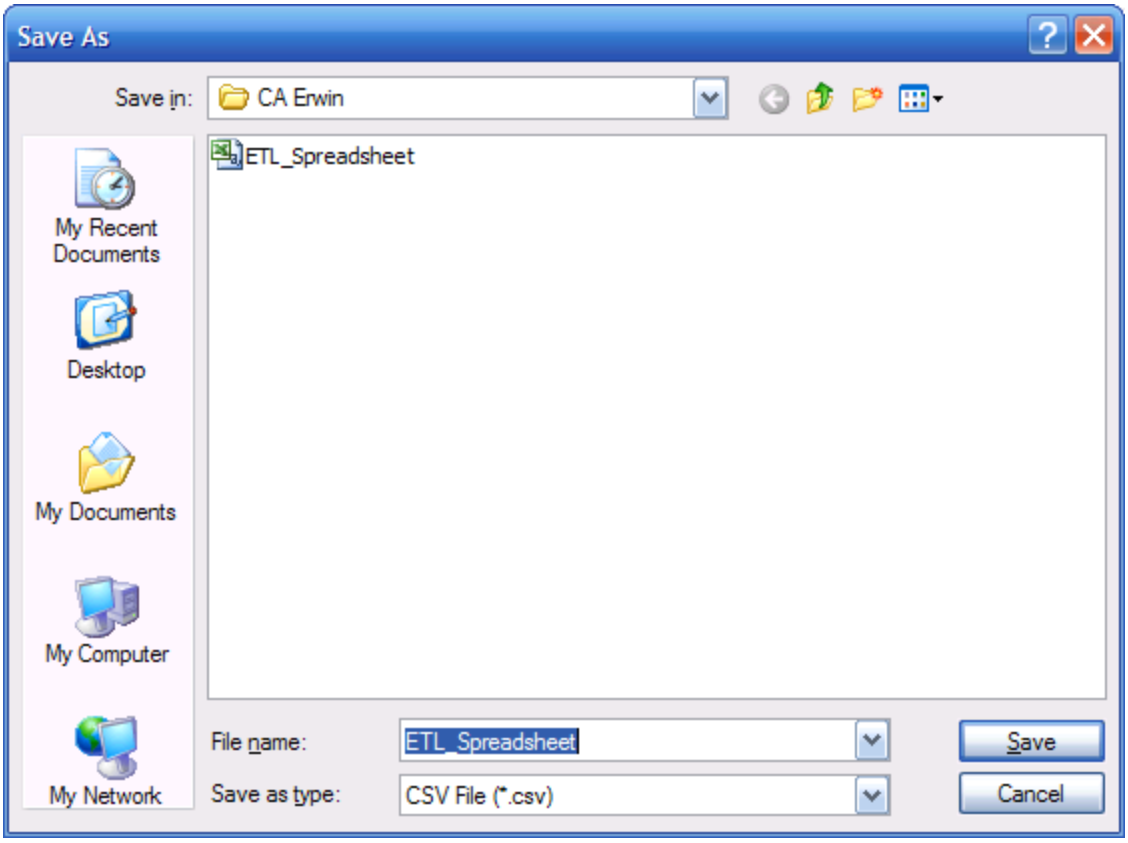


Expand "CA Erwin DM Reports" ->Expand "Table Reports" ->Expand "Physical only Table/Columns" ->Double click on the "ETL Spreadsheet" which will generate new report

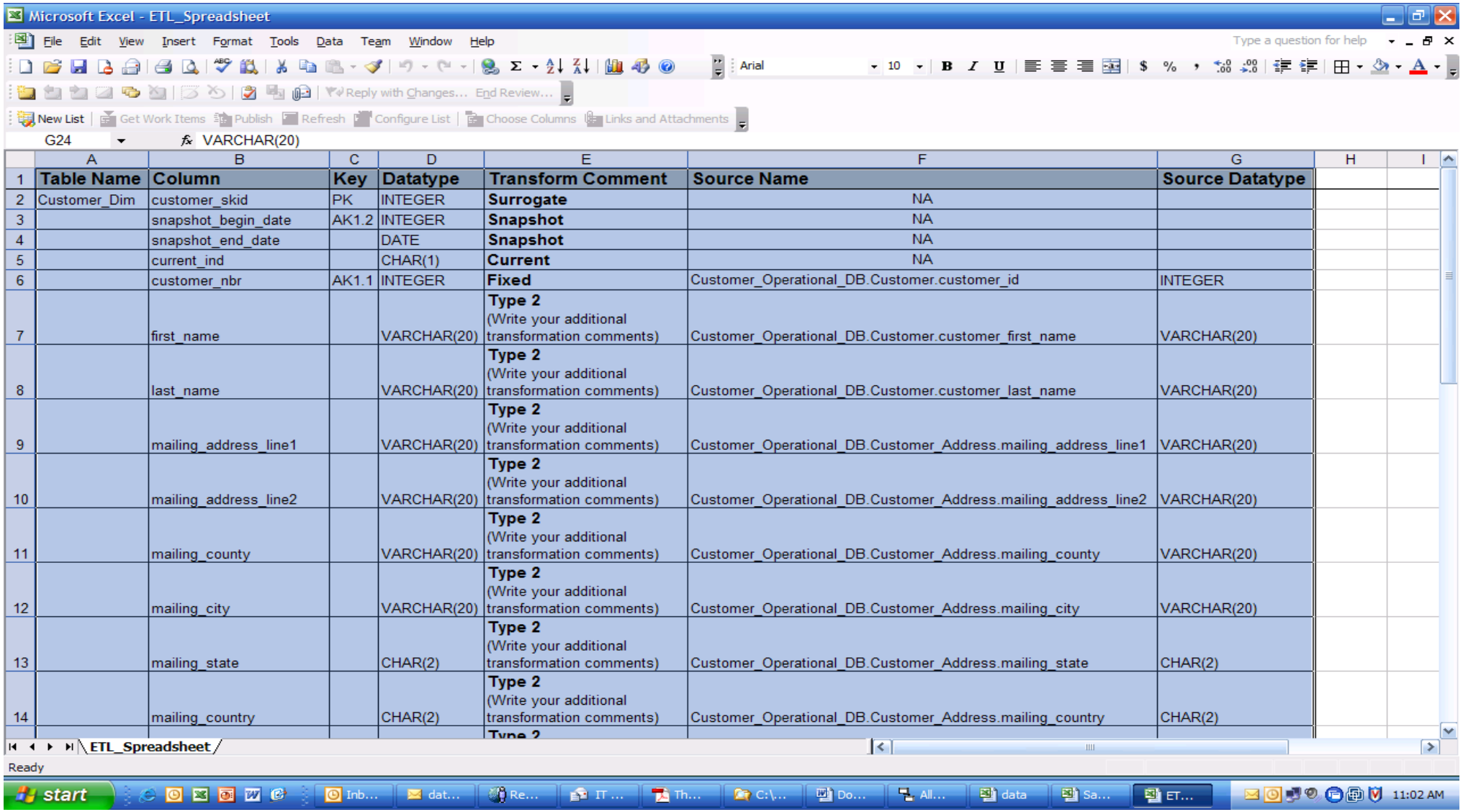


Right click again and select “Export result set ETL Spreadsheet” which will open the “Export from Data Browser” window .Select “CSV” in the Export




Save As
Save in:
CA Erwin
ETL_Spreadsheet
My Recent Documents
Desktop
My Documents
My Computer
My Network
File name:
ETL_Spreadsheet
Save as type:
CSV File (*.csv)
Save
Cancel

The final ETL Spreadsheet will look like the following which will be used as deliverable to the ETL team. You can also highlight the important details after generating the report from the tool like the following .Save it as .**xls** type so that all your custom made changes will be retained when you open again.



Microsoft Excel - ETL_Spreadsheet

File Edit View Insert Format Tools Data Team Window Help

G24 fx VARCHAR(20)

	A	B	C	D	E	F	G	H	I
1	**Table Name**	**Column**	**Key**	**Datatype**	**Transform Comment**	**Source Name**	**Source Datatype**		
2	Customer_Dim	customer_skid	PK	INTEGER	**Surrogate**	NA			
3		snapshot_begin_date	AK1.2	INTEGER	**Snapshot**	NA			
4		snapshot_end_date		DATE	**Snapshot**	NA			
5		current_ind		CHAR(1)	**Current**	NA			
6		customer_nbr	AK1.1	INTEGER	**Fixed**	Customer_Operational_DB.Customer.customer_id	INTEGER		
7		first_name		VARCHAR(20)	**Type 2** (Write your additional transformation comments)	Customer_Operational_DB.Customer.customer_first_name	VARCHAR(20)		
8		last_name		VARCHAR(20)	**Type 2** (Write your additional transformation comments)	Customer_Operational_DB.Customer.customer_last_name	VARCHAR(20)		
9		mailing_address_line1		VARCHAR(20)	**Type 2** (Write your additional transformation comments)	Customer_Operational_DB.Customer_Address.mailing_address_line1	VARCHAR(20)		
10		mailing_address_line2		VARCHAR(20)	**Type 2** (Write your additional transformation comments)	Customer_Operational_DB.Customer_Address.mailing_address_line2	VARCHAR(20)		
11		mailing_county		VARCHAR(20)	**Type 2** (Write your additional transformation comments)	Customer_Operational_DB.Customer_Address.mailing_county	VARCHAR(20)		
12		mailing_city		VARCHAR(20)	**Type 2** (Write your additional transformation comments)	Customer_Operational_DB.Customer_Address.mailing_city	VARCHAR(20)		
13		mailing_state		CHAR(2)	**Type 2** (Write your additional transformation comments)	Customer_Operational_DB.Customer_Address.mailing_state	CHAR(2)		
14		mailing_country		CHAR(2)	**Type 2** (Write your additional transformation comments)	Customer_Operational_DB.Customer_Address.mailing_country	CHAR(2)		

ETL_Spreadsheet

Ready

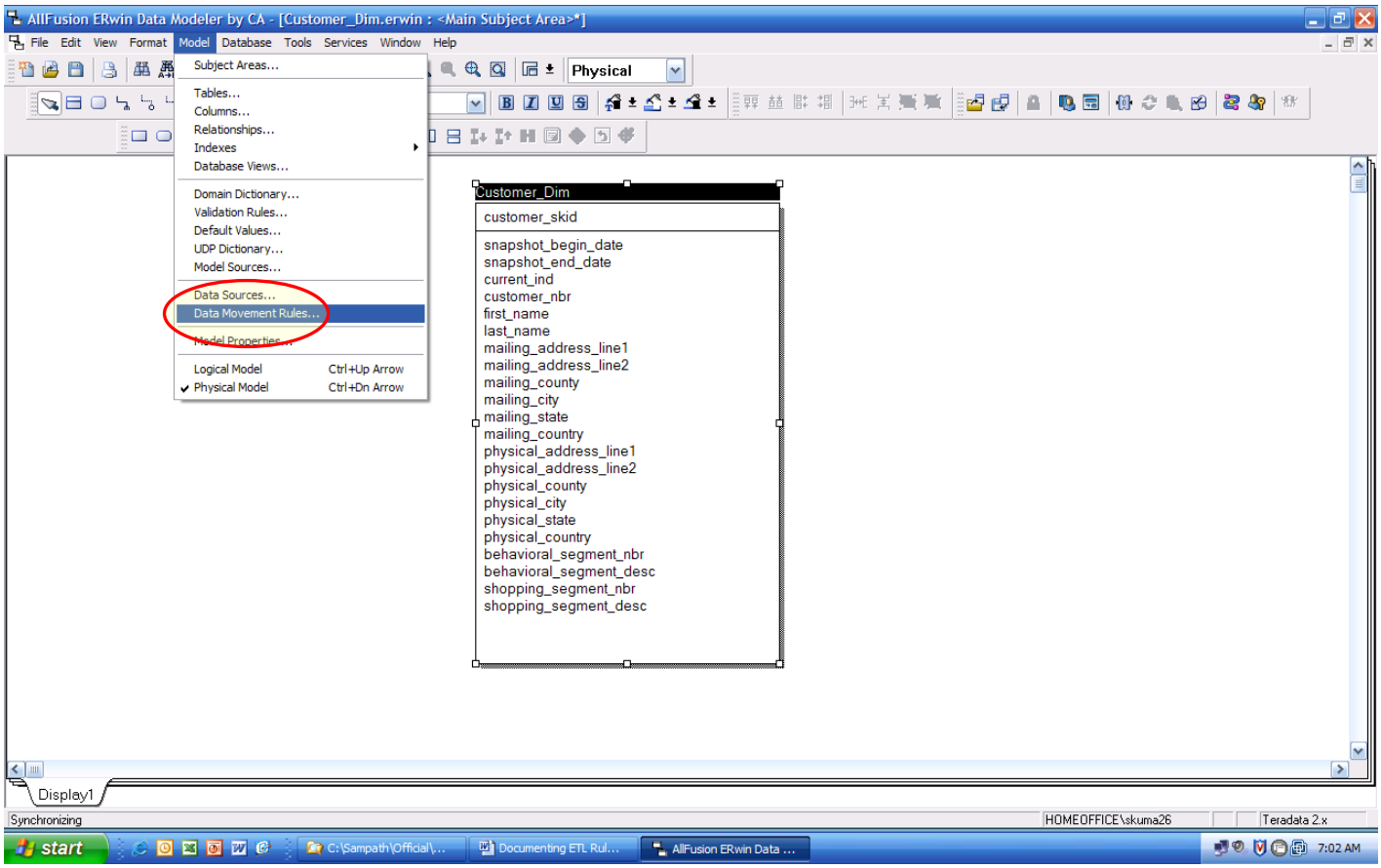
start 11:02 AM

Data Movement Rules in Erwin

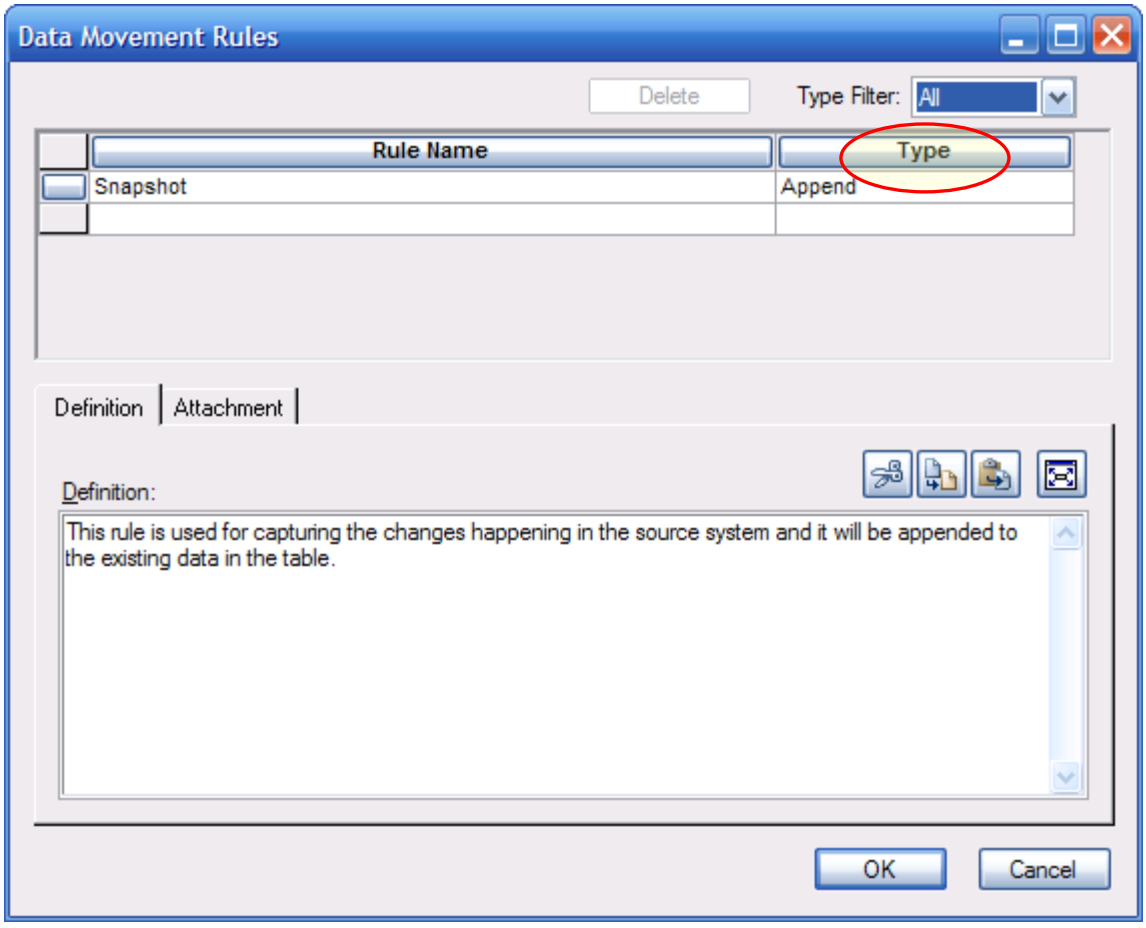
Data movement rules in the Erwin Data Modeler enable you to maintain processes required to regularly update all tables in the model. In our example it's used for keeping the Data warehouse and the operational source system in synch, the various management rules used to manage the information supported by Erwin are:

- Refresh: Replaces existing data.
- Append: Updates the existing information with changes and additional information.
- Backup: Creates a copy of the information to make it available for recovery.
- Recovery: Stores information from the backup information, the recovery process is required when the data is lost due to hardware or network failure.
- Archiving: Extracts information from tables based on criteria and saves the information in a file for future reference.
- Purge: Extracts information based on criteria but does not save the information.

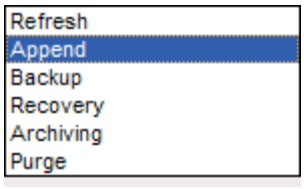
Open "Model" in the main menu and select "Data Movement Rules"



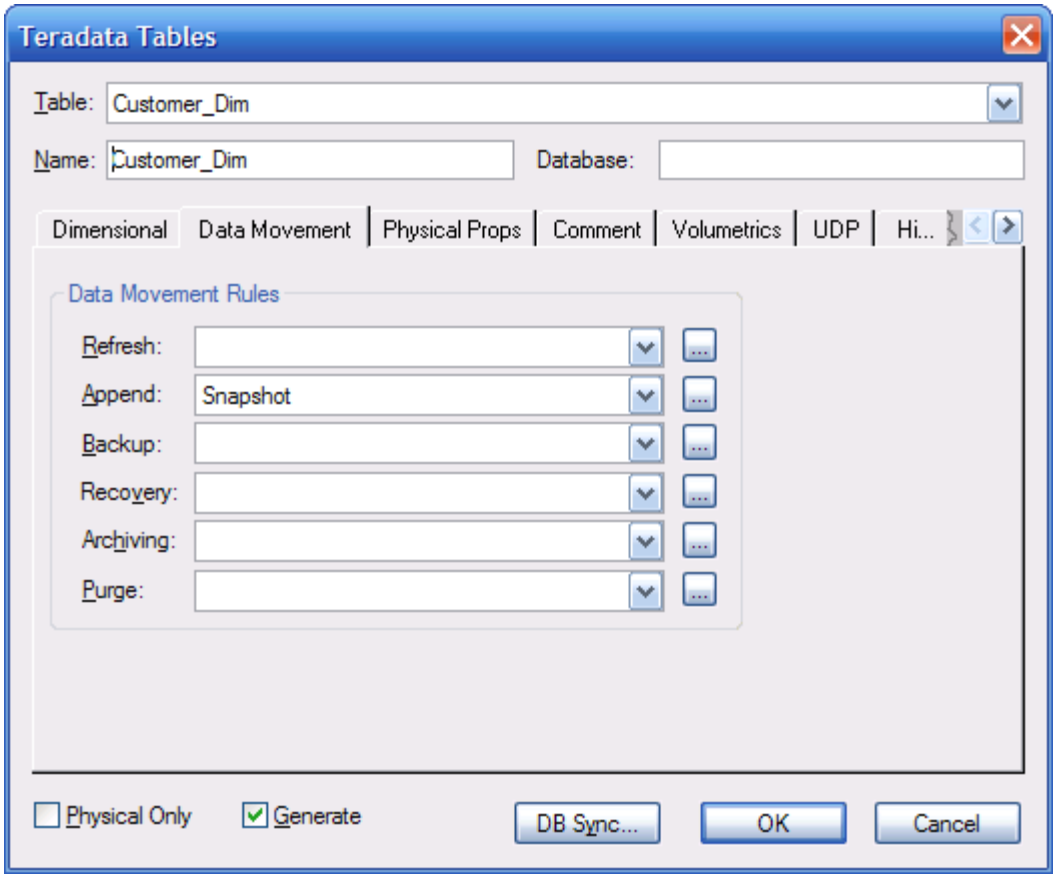
In the "Data Movement Rules" window define the rule name as "Snapshot" and type as "Append" (Refresh, Append, Backup, Recovery, Archive and Purge) .In the definition tab explain the meaning of the rule and how it needs to be attached to tables in the Data warehouse.



The “Type” is a drop down and it will have following options.

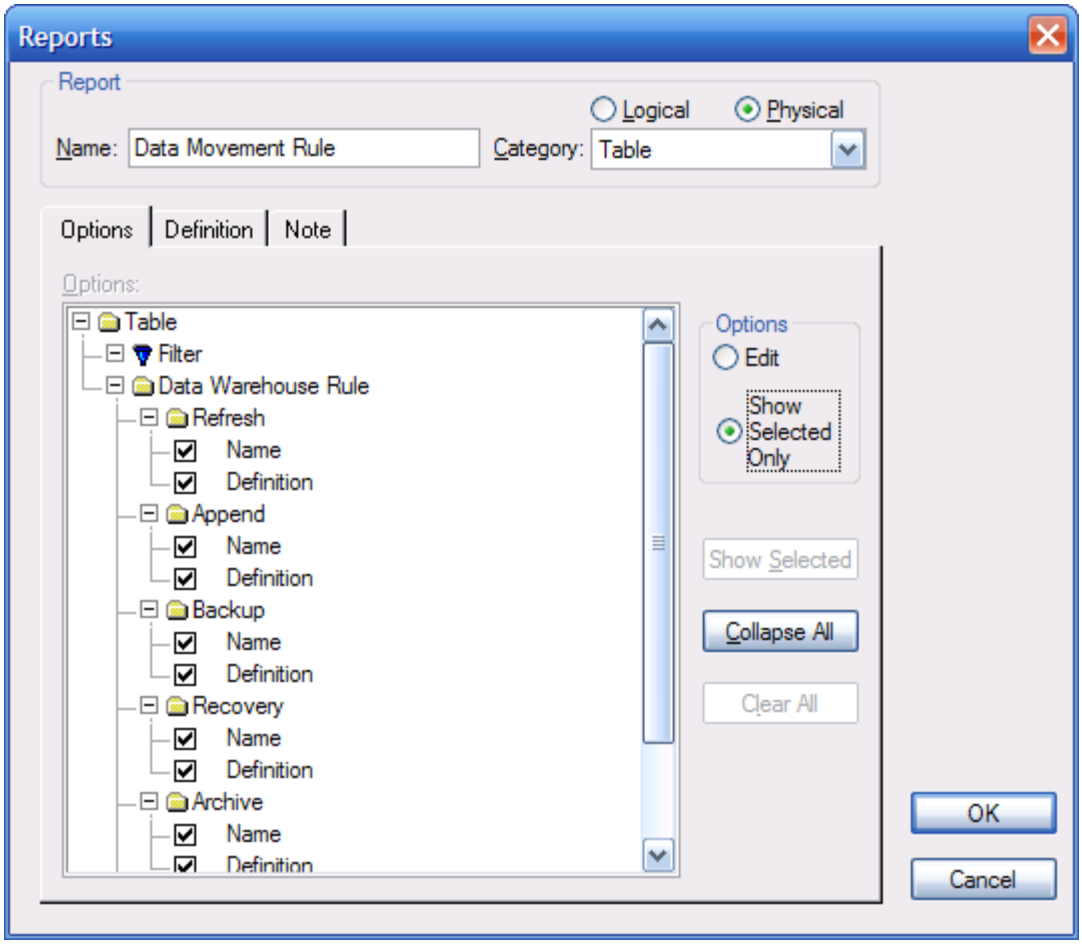


Once data movement rules are defined, attach the rule to the table

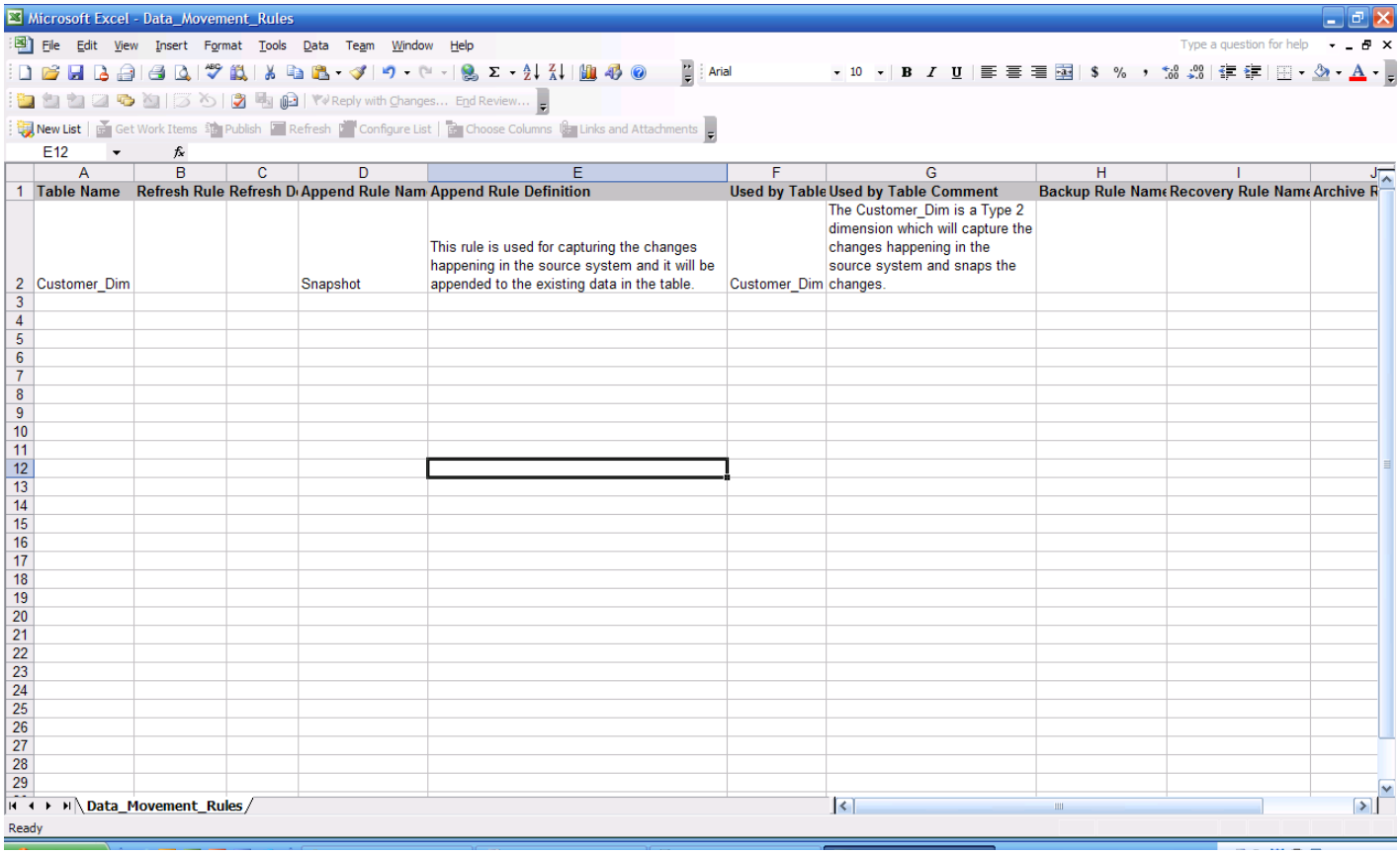


In our Customer_Dim we want to capture the changes happening in the source system as Type 2 dimension so we have selected the “Append” rule named Snapshot. For Type 1 dimensions use “Refresh” rule.

To retrieve this data in the form of report goto “Data Browser” select “File-New Report” .It should be generated separately apart from the “ETL Spreadsheet” to capture the table level information.



Export the report in the form spreadsheet



Conclusion

The metadata information such as “Data Source”, “Transformations rules” and “Data Movement rules” are very important for any Data warehousing efforts and it’s very critical to capture the correct information. It will be the guideline for the ETL team to create mappings to source system and load as per the rules. So try to provide as much as useful metadata information which will enhance effective implementation.

Note:

Please don’t maintain the spreadsheet separately from the data model, keep all your “Data Source”,” Transformation Comments” and “Data Movement Rules” in the data model itself and generate the report as and when changes.