# Milestone3 Deliverables Requirements

## I Overview

### A. What to do First

In this milestone, your team will perform ETL for our pilot DWE (a single MDM for our first star schema) at Rusty's RV. Each team will be given a model bundle and a data bundle (each bundle is for your team only, each team's bundle is different). The model bundle contains the CDM and PDM files for each of the OLTP data models / databases as well as CDM and PDM for the star schema. The data bundle includes scripts and data files needed to actually create the databases, login accounts, and database schemas as well as the files used to load the data for your team's OLTP databases / models. These bundles include the schema details for the new "official" Star_Schema (the data model / and ddl used to create the database and the database account that owns it)—but no data for the star—that is what you will be working on! In the data bundle, the scripts will create a separate database for each OLTP database and one for the OLAP database (our star schema). Each database will be owned and accessed using a separate database login that is created by the data bundle. Each login uses SQL-AUTHENTICATION, this means it is NOT your UST account—you can run the data bundle scripts using your UST account (which has been given the DBA-access needed to create databases and SQL logins). There are four databases created, each with its own separate login. Each database has its own login account, and all four accounts for each team share the same password. Each team's data bundle password details as well as the model bundles themselves will be posted to the blackboard under your team's file exchange area.

This Star_Schema model and database defined in these bundles is the "official" ROLAP data storage for your team's star schema. Your team will use this for milestone 3. It is not identical to your team's M2 MDM, but should be somewhat similar. Regardless, of how similar or different these star schemas are; the files in your M3 bundles are the one we will use (**NOT YOUR TEAM'S M2 SUBMISSION FILES**).

### B. Definitive Source Mapping for M3

In order to perform ETL in M3, you need to determine the **definitive source** for each DATT defined in your team's official star schema. Although we cannot be truly metadata driven for this step (it is not the tightly integrated situation we discussed as the "ideal" solution in lecture), we can be metadata driven to some extent. In other words, we can choose to manually create the mapping spreadsheet, or we can chose to generate at least some parts of it automatically. Regardless, I recommend that you **first** capture the technical metadata details in an excel spreadsheet (called the mapping spreadsheet), and then use the spreadsheet to plan / design your mapping requirements. This in turn can be used to plan / coordinate your implementation of the actual ETL that implements these mapping.

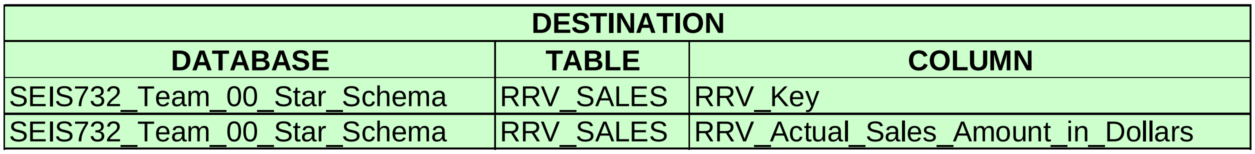
To start creating this spreadsheet, you can access metadata from one or more of our tools. Since we are concerned about mapping PHYSICAL database, table, and column details we need to consider tools that contain technical metadata about our PHYSICAL schemas (the sources and destination). Be careful, the names are CASE-SENSITIVE, and use the right metadata!

In your groups file exchange area; I have posted the M3 model bundle and data bundle. The model bundle includes spreadsheets with the relevant metadata for the OLTP models and the relevant OLAP model. These spreadsheets are essentially the export from the conceptual and physical data models created from a tool (e.g. power designer) for each OLTP source model as well as the destination star schema model. In other words, think of these as if you had created physical reports in power designer for each model and then cut and paste the column names from the "per table dialogs / reports" to make it easier to generate the mapping spreadsheet details. Alternatively (and I think more easily), spreadsheets like this could have been created for each PDM using the list-dialogs under the "Model" menu in the power designer tool and copy details from there (if we change the columns displayed before and you can access more of the metadata!). We could even generate the database schema and select the details from the data dictionary in MS SQL to get the table and column name details for spreadsheets such as those in your model bundle.

Here is a sample format for the mapping spreadsheet (an example will also be posted to the blackboard as an excel spreadsheet).

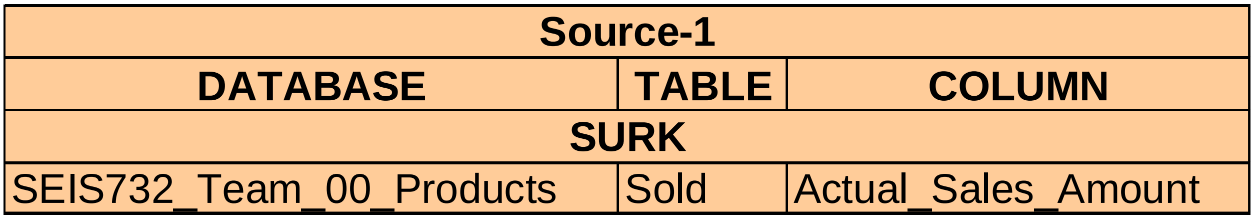
- The Mapping metadata for each Dimension, must be stored in a separate tab within a single mapping excel file. The tab should be named the same as the Dimension table.
- Each table (Destination or Source) will use the same format (template / boilerplate), but contain different details. In addition to the table and column details, we also need capture the **database name** for each source and destination. We will not specify the server name details (since that will not be fixed for us—we will always use the "." machine).
- The Database names follow a simple naming convention: "**SEIS732_Team_xx_...**", where the xx is replaced by your team number. Even though we are not going to attempt to use this mapping spreadsheet for metadata-driven activities in our project, we want to design and populate it in such a way that it COULD be used easily for this purpose.
- One simple example of how we do this: the names must be the actual names (e.g. we for Team_00's mapping spreadsheet, we would not populate the database name in the destination mapping details "Star Schema". Instead, we ensure that we always populate this value with "SEIS732_Team_00_Star_Schema"—the actual name needed to connect to the right database on the server.)
- Similarly, because there is always one destination, and potentially one or more definitive source, we will place the destination details in the FIRST (leftmost) columns of the spreadsheet. This improves usability for the user somewhat, but also makes scripting easier (even though, we will **not** actually attempt to script the metadata driven parts).

Here is a partial example of what this format looks like for the destination details:



| DESTINATION | | |
|---|---|---|
| DATABASE | TABLE | COLUMN |
| SEIS732_Team_00_Star_Schema | RRV_SALES | RRV_Key |
| SEIS732_Team_00_Star_Schema | RRV_SALES | RRV_Actual_Sales_Amount_in_Dollars |

To the right of this information, we would include similar details for each definitive source:



| Source-1 | | |
|---|---|---|
| DATABASE | TABLE | COLUMN |
| SURK | | |
| SEIS732_Team_00_Products | Sold | Actual_Sales_Amount |

For this example, since the RRV_Key is a surrogate key, we made it obvious in the mapping source ("SURK" with no database, table, or column name). By contrast, the Fact listed in this example is mapped from the Actual_Sales_Amount column in the Sold table, and SEIS732_Team_00_Products database.

I did this for the FACT table in this example, but **your team will do this for the DIMENSION tables. This is MAPPING REQUIREMENTS information (NOT implementation) do not include VIEW, STORED PROCEDURE, SQL EXPRESSIONS or other implementation details**.

If there are multiple source columns needed (either in combination with each other or as alternatives to each other depending upon the data characteristics) we simply list them as separate sources "Source-1, Source-2, …, Source-n" within this single, DEFINITIVE SOURCE. This is **NOT** the candidate source technique; it is listing all the sources needed to provide this ONE source for the destination. For example, consider the merging scenarios in Lecture_09-B; let's use slides 21-22 as an example here. The Product_Flavor destination column for the table shown on slide 22 would have two sources needed by it. The names "Source-1", "Source-2", etc. are merely used to clarify which source we are talking about—there is no ordering or ranking the way that candidate source does things. In this example, we would need to list table "**Prod**" with column "**FLAVOR**" as one source (e.g. Source-1), and table "**PRD**" with column "**PRDTYP**" as another source (Source-2).

Obviously, it makes sense to use a reasonable ordering when populating these sources (e.g. always considering a particular database / table order). Consider a counter example: if we populated Source-1 and Source-2 using the product database followed by the corporate database for one DATT, and then populated the corporate database followed by the product database for the same DIM table. If we did this, there would be no benefit, but it would make the mapping details a little harder to read / evaluate quickly.

### *C. The Data Bundle*

The data bundle also consists of a single zip file (the data bundle zip file) which in turn contains several files and subdirectories. Unzip this zip to the **local hard drive** in a location that does **NOT** contain spaces in the path name.

I will emphasize this again. Do **NOT** use the "U:" drive. I also do **NOT** recommend using a Thumb drive. On campus, you might be able to do this using a folder on your Desktop, but ensure that there are **NO** spaces in the pathname (e.g. **NOT C:\Documents and Settings\...\Desktop**). It is usually best to create a directory on the "C:\Users\<username>" or perhaps on the "D:\" drive called something clear like "Team_xx_M3" (ensure there are **NO** spaces in the pathnames, and ensure that the pathnames are **NOT** too long). Unzip the data bundle in this "nice location", and remember to delete that location's directory when you leave the lab!

For example:

- Open a command prompt (Start→Run→cmd).

- Inside the command box, change to the drive and directory you chose to use. (e.g. for Team_00, I could use D:\Team_00_M3, or D:\Team_00_M3_<my-initials-or-name>)

- In the command prompt, type "DIR" you should see a file named MakeAll.cmd.

- If you don't see MakeAll.cmd, then you are not in the right directory.
  If you do see it, type MakeAll and follow the prompts.

## *D. What happens after running the Data Bundle Scripts?*

If you are at this point, then you have just created the starting point for the actual M3 ETL implementation. You should periodically repeat this process as you develop things, or in the very least, you must run this on a clean machine as part of your final testing before submitting the final deliverable (if you don't you'll be sorry!).

If they are not there already, separate "MS SQL 2012 ETL" documents (or if time permits, screen-casts) will be posted to the blackboard. They will describes how you can get started implementing your own ETL package(s). You will implement the tasks needed to extract, transform and load the data into the star schema database from the three OLTP databases. You will ultimately create **a single M3 Project directory (containing the actual ETL implementation package), and a single mapping spreadsheet** (Excel). These files will be zipped up into a file named Team_XX_M3_Final.zip and posted to the assignment area.

The mapping spreadsheet you create in M3's first steps should serve as a guide / roadmap for dividing work among the team members, as well as estimating how difficult / complex a given dimension will be to extract, transform, and load.

**If you change things in the actual ETL implementation, make sure you update the spreadsheet to accurately reflect what you are REALLY DOING wrt transformations**!

In other words, remember our discussion about what happens when shared metadata is not kept in sync, and do NOT allow that to happen to you!

# II Deliverables Required

## A. Deliverable-1: (Project Status)

1. Use the project tracking tool / spreadsheet you documented in Milestone 1 to capture the necessary project tasks, estimates, effort, and status for your team.

2. Create a separate PDF report for the current project status at the end of each week. Save the report for each week separately in a single PDF file named "**Txx_M3_D1_Project_Status_Week_Ending_YYYY_MM_DD.PDF**". Notice there are NO SPACES in the filename (use underscores).

   Substitute your two-digit team number for the xx, the four digit year for **yyyy**, the two digit month for **mm**, and two digit day for **dd**.

3. Include all project support files (spreadsheets, etc. using a sensible naming convention) in a subdirectory named "**Txx_M3_D1_Project_Status_files**" within your deliverables zip for this milestone.

### Suggestions:

Separate spreadsheet files would make this more modular than trying to edit a single spreadsheet as a team.

In the PDF file, be aware of the formatting, and you might also want to play with the "Page Setup" to see if Portrait or Landscape works better for you.

If information needs to span multiple pages, use page breaks and repeated column headings / row headings to make the presentation of the information usable.

You **DON'T** need to include any "backup files" or previous versions (just the final version of the deliverable files).

## *B. Deliverable-2: Mapping Spreadsheet (for Dimension Tables ONLY)*

Create a **single** Excel Spreadsheet file (with **one worksheet for each DIM Table**) to identify the mapping of "source to destination" for all databases, tables and columns used by you in this milestone. See the example spreadsheet to use as a guide, but aside from that, you are free to format this anyway you like as long as it is readable and contains the following information.

1. You only need to map and load the Dimension tables. **Not the fact table**

2. You must map **ALL Ten (10) Dimension tables:**
   - **CUSTOMER**
   - **DEALER**
   - **INCENTIVE_PLAN**
   - **MANUFACTURE_DATE**
   - **MANUFACTURING_PLANT**
   - **MSA**
   - **PACKAGE**
   - **PRODUCT**
   - **PURCHASE_DATE**
   - **SALES_ORG**

3. Create a **separate sheet** (tab) for each DIM. Then, (on the left-hand side of each sheet) list the Database name, Table name, and Column name for each Column in the Physical DIM table in the Star Schema database. We list the TARGET (destination) first, because there is always a destination – we might have multiple sources for some DATTs but there is always only one destination. We list the ACTUAL database names: e.g., SEIS732_Team_00_Star_Schema. In the real world, we would try to dynamically generate this spreadsheet and also use the spreadsheet to dynamically generate other things. Recall "metadata driven" discussions. We are not required to do that for M3, but we will populate the metadata correctly for this purpose.

4. Next, (repeating this as many times as necessary when there are multiple source columns being used) List the Database name, Table name, and Column names for each Physical OLTP database-table-column that is to be mapped (extracted, transformed and loaded) into that Column of the DIM Table in the Star Schema. (Do **NOT** include any details about views, staging tables, or temporary tables). In other words, identify **exactly** which OLTP databases, tables, and columns are mapped into each star schema dimension table and column.

5. If there are any notes, or additional information for a particular mapping, feel free to document that in a different spreadsheet or in a different file and feel free to include such things in the deliverable bundle, but do NOT include it in the OFFICIAL mapping file. For example when multiple columns map to the same star schema column it might be useful to indicate in some other file / spreadsheet why or how we use a particular source.

6. Name the file Team_xx_Mapping (using your team number and the correct file extension) then be sure to include the Excel File in the Final Zip

## *C. Deliverable-3: Mapping Document*

1. From your spreadsheet, make a readable version of all the mapping details for each DIM. In other words, look at the pagination and adjust things to make the printable version **useable and readable** and then

   Print the mapping sheet for each dim as a separate PDF file, named after that DIM
   **DO NOT PRINT THIS ON PAPER,**
   **PRINT IT TO A PDF FILE!**

2. Include these ten **(10) PDF files** in the Final Zip.

## *D. Deliverable-4: ETL Project Directory*

1. When you follow the approach mentioned in other project ETL documents / screen-casts (and / or the approach I will demonstrate in class), each team creates a directory with one solution and one package for all the ETL needed. If you use multiple directories / files while developing, you should merge them for the final and submit a single zip file named Team_xx_ETL.zip.

2. Include this Team_xx_ETL.zip file in the Final Zip

3. You must create a single ETL Package that implements the ETL to Extract Transform and Load data from the OLTP databases into the DIM tables (specified here) within your team's star schema database.

   **You ONLY need to implement ETL for the following (8) Dimensions:**

   - CUSTOMER
   - DEALER
   - INCENTIVE_PLAN
   - MANUFACTURING_PLANT
   - MSA
   - PACKAGE
   - PRODUCT
   - SALES_ORG

# III Final packaging for M3 submissions:

**WHAT TO DO for the Final M3 submission:**

Create a directory named Team_xx and place all files and subdirectories within it. Create a SINGLE ReadMe.txt file, briefly listing what files and file formats are included in submission as well as any other relevant details if necessary (such as the team password). Create a SINGLE CoverSheet.txt file, briefly listing the Milestone number, Team number, CLASS-IDs of all team members who worked on this milestone, and the submission date. Place the directories and files named in the deliverables in this Team_xx directory and then use 7zip to zip the Team_xx directory into a SINGLE ZIP ARCHIVE FILE named "Team_xx_M3_FINAL_Deliverable.zip" for the final submission.

**Inside the SINGLE Zip file that you submit, it should look something like this:**

```
📁Team_01
    |
    +—📁 T01_M3_D1_Project_Status_files
    |       +— T01_M3_D1_Project_Status_Week_Ending_2017_04_09.PDF
    |       |       (one for each week of progress)
    |       +—(any other supporting files)
    |
    +—📁 T01_M3_D2_Mapping_Spreadsheet
    |       +—T01_M2_D2_Mapping.xlsx (or whatever the extension is)
    |
    +—📁 T01_M3_D3_Mapping_Document
    |       +—T01_M2_D2_Mapping.pdf
    |
    +—📁 T01_M3_D4_ETL
    |       +—T01_M2_ETL.zip
    |
    +—📁 T01_M3_Other_Support_Files
    |               (if you have any other support files)
    |
    +—CoverSheet.txt
    |
    +—ReadMe.txt
```