

 Fantastic Technologies Inc

 CS 683 Data Warehouse

Data warehouse Requirements

Ahmed Eltahir

03/07/2018

Table of Contents

[Data Warehouse Requirements (Week 1) 5](#_Toc508219645)

[Overview of an Enterprise 5](#_Toc508219646)

[Data Acquisition 6](#_Toc508219647)

[Source data 6](#_Toc508219648)

[Data staging 6](#_Toc508219649)

[Data Storage 7](#_Toc508219650)

[DW DBMS 8](#_Toc508219651)

[Data marts 8](#_Toc508219652)

[Metadata 9](#_Toc508219653)

[Information Delivery 10](#_Toc508219654)

[MDDB 10](#_Toc508219655)

[Data mining 10](#_Toc508219656)

[OLAP 10](#_Toc508219657)

[Report and query 11](#_Toc508219658)

[Management and Control 11](#_Toc508219659)

[Design Requirements (Week 2) 12](#_Toc508219660)

[Conceptual Design 12](#_Toc508219661)

[Data-driven Conceptual Design 13](#_Toc508219662)

[The Dimensional Modeling 14](#_Toc508219663)

 [17](#_Toc508219664)

 [17](#_Toc508219665)

 [17](#_Toc508219666)

 [17](#_Toc508219667)

 [17](#_Toc508219668)

 [17](#_Toc508219669)

[Logical Design 19](#_Toc508219670)

[ROLAP Systems 20](#_Toc508219671)

[View Materialization 22](#_Toc508219672)

[View Fragmentation 22](#_Toc508219673)

[Physical Design 22](#_Toc508219674)

[Load Data (Week 3) 25](#_Toc508219675)

[Data Staging Design 25](#_Toc508219676)

[Data Population 26](#_Toc508219677)

[Extracting Data 26](#_Toc508219678)

[Data Cleansing 28](#_Toc508219679)

[Transformation 32](#_Toc508219680)

[Loading Data 32](#_Toc508219681)

[ETL Mapping 41](#_Toc508219682)

[Data Analysis (Week 4) 42](#_Toc508219683)

[Maintenance and SQL Script and Conclusion (Week 5) 43](#_Toc508219684)

[Reference 44](#_Toc508219685)

# Data Warehouse Requirements (Week 1)

## Overview of an Enterprise

Fantastic Technologies Inc is a growing, leading provider of technology products. The company sells technological products at an unbeatable price to the consumers, minor business owners and businessmen who visit their stores, the company has operations in the U.S and Canada. The company founded on 21st October 2013, the main headquarter is situated in San Leandro California and has 10 branch bureaus. It has 10,000 customers all the way through the United States of America & Canada. There are 700 laborer par time, 540 proficient, and 320 expertise full-time workers.

Since Fantastic Technologies Inc is having different branches, where everyday sales take place for several products. A higher management is facing a problem with decision making due to non-availability of integrated data, hence, they can’t do a study on their data as per their requirement. So, they requested for a requirement elicitation process for a data warehouse that identifies its information contents, to help them quickly in the decision-making process, identifying the company best and worst customers and to spot buying trends.

Besides, they would like to have analytical views, containing graphs and charts that display the company’s performance broken down by customers and products. Of course, having such a tool would help investors to endorse the products that fall short of selling in hot trading areas.

The ultimate objective is to have 3 data marts: sales, marketing, and operational, so as manager on each division can focus on his area, and at the same time to provide a common view that will support Fantastic Technologies Inc management in their decision. The information delivery will be the company sales products in a monthly report, including the product type, employee name, customer name, suppliers, and the date.

# Data Acquisition

Data acquisition is the method of extracting the data from diverse source (operational databases) systems, integrating the data and transforming the data into a similar format and loading into the target warehouse database. In other words, it’s called as ETL (Extraction, Transformation, and Loading).

## Source data

Data warehouses do not create their own data. Instead, they are always "fed" data from other sources (Microsoft, 2018). In general, a data warehouse principally combines information from several sources into one comprehensive database, allowing analysis, or decision to be made. Data warehouses architect frequently must identify multiple data sources that will be used to populate the warehouse (Baya,2006).

In our case, Fantastic Technologies Inc are storing information within a various format, it lies in sales, marketing, and operational departments. Data within the sales department are stored in a SQL Server database, whereas marketing and operational departments are stored on an excel sheet.

## Data staging

According to Thomas (2016), a data staging area (DSA) is a momentary storage zone between the data sources and a data warehouse. when data has been loaded into the staging area, the staging area is used to join data from various data sources, transformations, data cleansing, and validations. Data is regularly distorted into a star schema before loading it in a data warehouse. Thus, all essential data must be accessible prior data can be integrated into the Data warehouse.

Since Fantastic Technologies Inc has multiple data sources, then a staging database will be needed to correlate data from these sources prior the population, and we are going to use ETL tool which is (SSIS) SQL Server Integration Services to copy data directly from the SQL Server Database and the excel sheets.


 Figure 1: Similar example of Fantastic Technologies Inc Data Warehousing Environment

# Data Storage

Of course, the data that is stored in the data warehouse is just as significant as the data warehouse itself ([Exforsys](http://www.exforsys.com/author/Exforsys),2007). By having an essential recognizing of how this data is stored can be convenient in the fruitful implementation and utilization of a data warehouse.

Fantastic Technologies Inc is using a normalized approach to store the data as it is relatively simple to add new data.

## DW DBMS

Indeed, choosing a database management system is important to the success of a data warehouse.

Fantastic Technologies Inc are using Relational Database (RDBMS) for their data warehousing, due to its powers in processing and storing transactions and in developing reports. RDBMS has dominated the entire database market for years (Joseph,2018).

## Data marts

According to Denise (2010), a data mart delivers the main access to the data stored in the data warehouse or operational data store. It is a subsection of data sourced from the data warehouse or operational data store precisely concentrated on a business function or set of associated business functions.

Thus, data marts hold only one subject area, which is going to be sales, marketing, and operational data marts.

Data marts in Fantastic Technologies Inc is dependent and following a Top- Down data warehousing approach.



 Figure 2: Design of Fantastic Technologies Inc Data Warehousing

## Metadata

Metadata is data that defines the data and schema objects and is utilized by applications to fetch and calculate the data correctly. In other words, Metadata defines the relationships between operational databases and the data warehouse and is stored in the warehouse.

Developers are often using the database's metadata to acquaint themselves with the new structure and figure out its construction and the kinds of records it has. Metadata describes the “who, what, when, where, and why” of data.

Fantastic Technologies Inc metadata includes:

1. Source of data
2. Data Target
3. File Name
4. Type
5. Size
6. Creation date and time
7. Last modification date and time
8. Business Rules.
9. Data Standards.

# Information Delivery

## MDDB

A multidimensional database is a special form of database that has been optimized for data warehousing and OLAP (online analytical processing). Basically, there are two types of the multidimensional database: The Logical multidimensional data model which collected by measures, logical cubes, dimensions, levels. hierarchies, and attributes, and the Relational multidimensional data model which is typically a star schema or snowflake schema (i.e. fact tables, dimension tables, and materialized views).

Fantastic Technologies Inc follow a Relational data model.

## Data mining

Data mining is a decision support course in which we look for patterns of information in data. i.e. the finding of meaningful new patterns, relationships and trends in big volumes of data stored in a database or data mart by using pattern recognition knowledge and mathematical methods.

Fantastic Technologies Inc follow descriptive models, which describe patterns in present data to guide the decision-making process.

## OLAP

OLAP is an abbreviation for On- Line Analytical Processing. It is a tactic to quickly offer the answer to analytical inquiries that are dimensional in nature. It is a slice of the wider class of business intelligence, which also contains Extract transform load (ETL), data mining, and relational reporting.

Fantastic Technologies Inc is following Relational OLAP (ROLAP), where the base data and dimension tables are stored as relational tables. Indeed, such an approach permits multidimensional analysis.

## Report and query

Of course, the purpose of query and reporting tools is to let users recover and present data from the data mart to achieve some pre-defined or ad-hoc study. Management at Fantastic Technologies Inc will be able to view the sales products in a monthly report, including the product type, employee name, customer name, suppliers, and the date.

# Management and Control

Within Fantastic Technologies Inc, auditing data regularly is an important part of their quality and control procedure. This will assure that the information from the source systems and reports are matched. In addition, security is essential, data is obtainable to the right operators at the correct time. The staffs of each division will be able to view only the data that is related to their own department, while also permitting top management to view data for all departments and subsidiaries. Similarly, implementing security applications (i.e. firewalls) to prevent & stop unauthorized users from accessing or altering data or theft by hackers. Besides, the company has its own policies and procedures that employee have to follow, such as standard naming conventions to all tables.

# Design Requirements (Week 2)

## Conceptual Design

According to Marseille (2013), building a DW is a very multifaceted task, which requires an accurate planning meant at devising satisfactory answers to organizational and architectural questions.

Conceptual Modeling is a high-level description of a business’s informational needs. It typically includes only the main notions and the main relationships among them. Of course, the conceptual model is needed to understand and capture business knowledge from a data (rather than a process) viewpoint, and by developing the conceptual data model, that will support identify the breadth of the subject area and can aid establish the scope for the project. Some features of conceptual data model include:

1. Highly abstract.
2. Easily to understand.
3. Easily enhanced.
4. Only entities are visible.
5. Abstract relationship

As stated before, Fantastic Technologies Inc is following a Top-Down Approach to building their data warehouse, such a method views the data warehouse as the linchpin of the entire analytic environment. The data warehouse holds atomic or transaction data that is extracted from one or more source systems and integrated within a normalized, enterprise data model. From there, the data is summarized, dimensionalized, and distributed to one or more “dependent” data marts.

According to Golfarelli, & Rizzi (2009), three fundamental methodological approaches can be taken to data mart design: the data-driven method, the requirement-driven approach, and the mixed approach. In the data-driven approach, it is extremely interesting to study how you can derive your conceptual schema from those schemata that define a relational database.

Our methodology may be applied to conceptual Entity-Relationship schemata or relational logical schemata along with minor changes. Naturally, conceptual Entity-Relationship schemata are more expressive than relational logical schemata. For this reason, they are generally considered as a better design resource.

### ****Data-driven Conceptual Design****

The technique used for the Dimensional Fact Model (DFM)–compliant conceptual design of a data mart based on an operational source Entity-Relationship schema includes the following steps:

1. Define facts.

2. For each fact:

a. Build an attribute tree

b. Prune and graft the attribute tree

c. Define dimensions

d. Define measures

e. Create a fact schema

.

### The Dimensional Modeling

DFM was first projected in 1998 by Golfarelli & Rizzi and continuously enriched and advanced during the following years in order to optimally suit the diversity of modeling circumstances that may be encountered in real projects of small to large complexity.

The DFM is a graphical conceptual model, precisely devised for multidimensional design, intended at:

 • Efficiently supporting conceptual design.

 • Providing an environment in which supporting the dialogue among the designer and the end-users to improve the specification of requirements.

 • Generating a stable platform to ground logical design.

• Providing an expressive and non-ambiguous design documentation.

Thus, The Dimensional Fact Model (DFM)is a conceptual model created specifically to function as data mart design support, and it is essentially graphic and based on the multidimensional model.

Dimensional Fact Model includes fact schemata be created for every fact of primary importance for users. A fact schema can graphically represent all multidimensional model concepts, such as facts, measures, dimensions, and hierarchies. It can also include a set of advanced constructs that can accurately represent different conceptual shades distinguishing the most complex real-world scenarios.

A fact is an emphasis of interest for the enterprise; a dimension determines the granularity adopted for representing facts; a hierarchy determines how fact instances may be aggregated and selected significantly for the decision-making process. A fact scheme is structured as a tree whose root is a fact.

The resulting group of fact schemata is the conceptual schema for our data mart.

Thereby, with the Dimensional modeling two kinds of tables involved:

1. Dimension tables which used to define the data we need to store.
2. Fact tables which comprise the data we desire to include in reports, aggregated based on values within the related dimension tables.

Since a dimensional model is visually characterized as a fact table surrounded by dimension tables, it is frequently called a star schema.

**Product**

**Store**

 **Sales**

 **Employee**

**Time**

 Figure 3: Fantastic Technologies Inc Conceptual Design

**Product**

 **Store**

Product

**Employee**

 **(0, n)**

 **(0, n)**

 **(0, n)**

Sales

Employee

Store

 **(0, n)**

 **Time**

Time

**Sales Amount**

 **Items Sold**

 Figure 4: The Entity-Relationship schema

##

##

##

1

M

N

1

Product

Is Sold

Sell

Employee

##

N

M

##

Sold On

Sold to

1

N

N

M

1

1

1

1

Day

Week

Month

Year

Store

 Figure 5: Star ER Model

This model syndicates star structure with constructs of ER model. The star ER comprises facts, entities, relationships, and attributes. This model has the following concepts:

1. Fact set.
2. Entity set.
3. Relationship set.

The model presented above encompasses of one fact table and four-dimension tables. The tables in the model are:

1. Fact Sales: This table holds references to the dimension tables plus two facts (price and quantity sold). Note that all four foreign keys together form the primary key of the table.
2. Dimension Employee: This is an employee dimension table that stores basic employee attributes, such as employee ID and employee name.
3. Dimension Product: This is a product dimension table with only two attributes (other than the primary key): product name and product type.
4. Dimension Store: store dimension table contain basic store attributes location, such as store ID, city, and country.
5. Dimension Time: This table handles the time dimension.

A hint to the ERD translation in the case of the star schema is that the fact table has no primary key of its own but inherits the primary keys of all related dimension tables which are concatenated to form its unique identifier.

.

## Logical Design

The logical design stage contains a set of steps that, starting from the conceptual schema, make it doable to outline the logical schema of a data mart (Golfarelli, & Rizzi,2009).

In the logical design, we emphasize on the information requirements and save the implementation particulars for the physical stage. Features associated with the logical design include:

1. More detailed than the conceptual model.
2. Presence of attributes for each entity.
3. Non- key attributes.
4. User- friendly attributes names.
5. Database agnostic (i.e. not specific)
6. Bit more efforts required to enhance.

Indeed, in the dimensional modeling, we classify which information belongs to a central fact table and which information belongs to its related dimension tables. Also, detect business subjects or fields of data, express relationships between business subjects, and name the attributes for each subject.

According to Golfarelli, & Rizzi (2009), initially, from the conceptual schema of a data mart, it’s doable to get a logical schema that can be directly implemented in a relational DBMS. The key steps in this method are:

1. Translating fact schemata into logical schemata: star, snowflake, and constellationschemata.
2. Materializing views.
3. Fragmenting Fact table

We are going to implement the Star schema, as its the simplest model used in DWH. The name of Star came due to the reality that the Fact table is in the center of the schema with dimension tables around it, it looks unevenly like a star. Besides, the only logical schema used in relational online analytical processing (ROLAP) systems is the star schema—or one of its derivatives—consisting of a fact table, whose primary key consists of the foreign keys referencing one or more dimension tables.

Fundamentally, the star schema comprises of a set of relations called dimension tables, and of course the fact table(FT) which referencing all the dimension tables.

In general, three different logical models can be used to represent a multidimensional data structure: the relational model, used in the so-called Relational OLAP (ROLAP) systems; the multidimensional model, used in Multidimensional OLAP (MOLAP) systems; and the hybrid model called Hybrid OLAP (HOLAP). In our approach, we are using the ROLAP.



 Figure 6: ROLAP, MOLAP, & HOLAP

### ****ROLAP Systems****

ROLAP systems adopt the well-known relational model to represent multidimensional data. The usage of a model based on a bidimensional element (relations have rows and columns) for multidimensional modeling seems to be strained. The advantage of using such model that the database industry standard and every professional database designer is familiar with it. Furthermore, relational DBMSs have been evolving for 30 years since they were originally marketed, making for highly sophisticated and optimized tools.

**DimProduct**

**DimStore**

**ProductID (PK)**

**StoreID (PK)**

**Product \_Type**

**Product\_ Name**

**Country**

**City**

 **FactSales**

**ProductID (FK)**

**StoreID (FK)**

**Time\_Key (FK)**

**EmployeeID (FK)**

**Items\_Sold**

**DimEmployee**

**DimTime**

**Sales\_Amount**

**EmployeeID (PK)**

**Time\_ Key (PK)**

**Employee\_ Name**

**Day**

**Day \_of\_ Week**

**Month**

**Year**

**Quarter**

 Figure 7: Logical Design “Star Schema”

### View Materialization

The term view materialization refers to the selection process of a set of secondary views that aggregate primary view data (Gupta and Mumick, 1998). According to the definition, there are two vital parts in the process: setting materialization goals and choosing a selection technique. The materialization goals are partially linked to the architectural model used for our data warehouse (Golfarelli, & Rizzi, 2009).

### View Fragmentation

According to Golfarelli, & Rizzi (2009), the term fragmentation means the subdivision of a table into multiple tables, known as fragments, to improve system performance. Generally, A table can be fragmented in two ways: The first is called horizontal fragmentation, which divides a relation into multiple parts, each of which has all the attributes but only one subset of tuples of the original relation. The second way is vertical fragmentation, which divides a relation into multiple parts, each of which contains all the tuples and a subset of the attributes of the original relation.

## Physical Design

The Physical design implements the choices made in the previous design phases and gives data mart schemata their final shape. In general, the best results are achieved when the logical design phase and the physical design phase are executed together because they are closely interdependent. For instance, materializing a view or creating a fact table index are two competing alternatives whose common goal is to improve system performance, but they are separated due to their complexity. Processes like optimization and indexes to improve the performance will be implemented in this phase.

Basically, the steps for physical data model design are as follows:

1. Alter entities into tables.
2. Change relationships into foreign keys.
3. Convert attributes into columns.
4. Adjust the physical data model based on physical constraints/requirements.

**DimStore**

**DimProduct**

**ProductID int PK**

**StoreID int PK**

**Product\_Name varchar(50)**

**City varchar(50)**

**Product \_Type varchar(50)**

**Country varchar(50)**

 **FactSales**

**StoreID int PK FK**

**EmployeeID int PK FK**

**ProductID int PK FK**

**Time\_Key int PK FK**

**Items\_Sold int**

**Sales\_Amount decimal(8,2)**

**DimEmployee**

**DimTime**

**EmployeeID int PK**

**Time\_ Key int PK**

**Employee\_ Name varchar(50)**

**Day date**

**Day\_of\_Week int**

**Quarter int**

**Month int**

**Year int**

 Figure 8: Physical Design

# Load Data (Week 3)

## Data Staging Design

The staging area seems to be one of the more ignored elements of a data warehouse architecture, and still, it is an essential part of the ETL factor design, as the data staging layer hosts the ETL processes that extract, integrate, and clean data from operational sources to provide the data warehouse layer with data. According to Golfarelli, & Rizzi (2009), it’s very recommended to follow a three-layer architecture, since populating data marts directly is a very composite task and it frequently leads to poor outcomes. Basically, the population process is separated into two stages: from the operational sources to the reconciled database, and from the reconciled database to the data mart. Certainly, the importance of a reconciled database proves that data mart designers have ultimately gained accurate knowledge of their operational data sources, which is essential for data marts to be properly populated (Golfarelli, & Rizzi, 2009, p 276).

Obviously staging area gives a lot of elasticity during data loading, as ETL process includes multifaceted data transformations that need additional space to momentarily stage the data.

Of course, a well- designed ETL is extremely important since every single information piece stored in a data mart is directly associated with one or more source attributes. Generally, ETL is the extraction of data from the input source, transforming it into an appropriate format for loading and loading it into the target database (Laberge, 2011, p 258).

According to Laberge (2011), two types of ETL process involved in the data warehouse, the first one is from source to a preparation and holding area, which is called data population. The second ETL portion in a data warehouse is from the holding area to the data marts, which is called data distribution. Basically, the ETL process contains four distinct stages: extraction (or capture), cleansing (or cleaning or scrubbing), transformation, and lastly loading.

## Data Population

According to Laberge (2011), the data population contains much processing with an importance on understanding the required data, the technical landscape, the variances between the source, which is classically operational online transactional processing system (OLTP), and the data warehouse, which is an online analytical processing system (OLAP).



 The Population Process ETL (Golfarelli & Rizzi, 2009, p 277)

###  Extracting Data

Extracting data is the primary stage of ETL, during that, data is collected from one or more data sources and detained in temporary storage where the succeeding two stages can be executed. According to Golfarelli, & Rizzi (2009), we can classify the data in our source into three types: transient data, semi- periodic data, and temporal data. Transient data means when the source database remain only a snapshot of the present data and no overwrite is feasible. Semi- periodic data occur when our database saves a restricted number of former data versions without knowing exactly for how long the information will be obtainable. Lastly, Temporal data when our database retains track of all data changes for an obviously defined interval. In our case with Fantastic Technologies Inc, the data is usually transient since the data is overwritten as soon as new goods arrive.

Static extraction is the easiest method for data extraction, such a technique will totally scan all the data in operational sources (Golfarelli & Rizzi, 2009, p 278). Indeed, static extraction is compulsory when we start up our system, but we can also use it each time the reduced size of data permits for it to be carried out within a satisfactory time frame.

The techniques based on the incremental extraction and can be categorized as immediate and delayed. The immediate methods record the changes to data once the operational database is restructured. The delayed methods delay this operation.

Fantastic Technologies Incis using Application-assisted extraction which is an immediate-extraction practice that generates a set of functions in the operational applications to store data variations to the staging area without altering the exterior actions of applications. According to Golfarelli, & Rizzi (2009), such technique can similarly be utilized for new-generation operational systems for which a layer of primitives (application programming interfaces, or APIs) are consistently used by all the applications to get data and to be DBMS-independent.



Application-assisted extraction. (Golfarelli & Rizzi, 2009, p 279)

### Data ****Cleansing****

Data cleansing, data cleaning, and data scrubbing refer to the set of operations intended at assuring the reliability and accuracy of the data in the reconciled database (Golfarelli & Rizzi, 2009, p 286). Indeed, such a phase is vital in a data warehouse system since it is meant to advance data quality and its proficiency. The subsequent list comprises the most recurrent mistakes and discrepancies that make data “dirty”:

1. Duplicate data, for example, an employee is recorded many times in a different store.
2. Unreliable values for a single entity because unlike methods were utilized.
3. Missing data, like a product type or name.
4. Unforeseen use of fields, for instance, a Social Security Number field could be used inappropriately to store company fax numbers.

According to Golfarelli, & Rizzi (2009), any kind of problem needs different techniques for its solution. Techniques such as Dictionary- based technique, Approximate Merging, and Ad- hoc technique.

As mentioned earlier, the ETL tool used by Fantastic Technologies Inc is SSIS (SQL Server Integration Services). The source of our data on the sales, marketing, and operational departments. We are going to demonstrate how to extract, transform, and load data using SSIS. Below are the data that we are going extracted (i.e. Excel sheet format).

From Sales Department, data about the products

|  |  |  |
| --- | --- | --- |
| Product ID | Product Name | Product Type |
| ITM-001 | I Phone 7 | Mobile Phone |
| ITM-002 | Samsung S 7 | Mobile Phone |
| ITM-003 | Samsung 65 inch | TV |
| ITM-004 | LG OLED 55 inch | TV |
| ITM-005 | Netgear N 600 | Router |

From Marketing department, information about the company stores

|  |  |  |
| --- | --- | --- |
| Store ID | City | Country |
| Loc-A1 | San Francisco | USA |
| Loc- A2 | New York | USA |
| Loc- A3 | Denver | USA |
| Loc- A4 | Los Angles | USA |
| Loc- A5 | Toronto | Canada |

And, from the operational department, data about the employees

|  |  |  |
| --- | --- | --- |
| Employee ID | Employee First Name | Employee Last Name |
| Emp1 | Jack | King |
| Emp2 | Maya | Ali |
| Emp3 | Adam | Smith |
| Emp4 | John | Taha |
| Emp5 | Sara | Moe |

SSIS or SQL Server Integration Services is an expansion instrument and runtime that is enhanced for structuring ETL processes, it’s nice because it allows us to design our ETL processes in a graphical way. According to Ruiter (2012), by using SSIS we can design our ETL process through data flows and control flows. Data flows in SSIS are sort of control flow that let us extract data from an external data sources, flow that data over a number of transformations like filtering, sorting, merging it with other data and altering data types, and lastly store the outcome at a destination, which is the data warehouse.





 

 Prior choosing and constructing the source and the destination, connections for both must be created. Since the Excel worksheet is the source, the Excel Connection will be developed, and for the destination, will use the OLE DB connection. The two connections can be formed by right click on the Connection Manager in the Solution Explorer and selecting the New Connection Manager from the drop-down list.

When the connections between the source and the destination are created, the Data Flow Tasks can be identified and configured in the Data Flow window. Using the SSIS Toolbox, the Excel Source, and the OLE DB Destination tasks must be imported into the Data flow window, as shown in the image underneath



For each data source, we have two operations, truncate the table and then retrieve data through a data flow task. As it’s a simple tactic to revive the data in our data warehouse. We have to empty the table (which holds data from the previous ETL run) using an SQL TRUNCATE TABLE statement, and then reload the whole table from the data source using a data flow. When the data is extracted, the ETL process will go the transformation stage.

### Transformation

The next phase of the ETL process is transformation. After data is extracted, it must be transported to the target destination and converted into the suitable format. This data transformation might contain operations like joining, cleaning, and validating data or developing designed data based on existing values.

In our case here, we are going to use the Merge transformation tool, since in our previous Dim Employee design we have only one column which calls Employee\_ Name, but in the excel sheet that we got from the operational department, we have the First name in one column and Last name in another column. Thus, it will be convenient to use the Merge transformation to merge these two columns into one column (i.e. Employee\_Name).

Of course, The Merge transformation also requires that the merged columns in its inputs have matching metadata, but with the SSIS Designer, the user interface for the Merge transformation routinely maps columns that have the same metadata. So, we can then manually map other columns that have well-matched data types. Thus, this transformation has two inputs and one output, and it does not support an error output.

### Loading Data

Loading data to the target multidimensional structure is the last ETL step. According to Laberge (2011), data distribution is the real loading into the data marts. During this stage, extracted and transformed data is printed into the dimensional structures retrieved by the end users and application structures. Besides, loading step contains both loading dimension tables and loading fact tables.



 3NF to Star. (Laberge, 2011, p 286)

According to Golfarelli, & Rizzi (2009), the kind of loading process relies on the method used in the extraction stage and might differ rendering to temporal or non-temporal slices of the reconciled database to be populated. Hereby, in our approach which is static extraction technique, the data in with the reconciled database will be totally removed and substituted with new data. The same source structure will be applied to the data warehouse.

Lets first create the Data Warehouse, creating 4-dimension tables and 1 fact table in the data warehouse: DimEmployee, DimStore, DimProduct, DimTime, and FactSales. Indeed, the dimension tables must be populated at the beginning prior we populate the fact tables since we want the surrogate keys on the dimension tables to translate the fact table natural keys.

**-- Create the data warehouse**

create database FantasticTechnologiesIncDW

go

use FantasticTechnologiesIncDW

go

**-- Create Employee dimension**

create table DimEmployee

(EmployeeID int not null identity(1,1) primary key,

 Employee\_Name varchar(50) not null

)

go

|  |  |  |  |
| --- | --- | --- | --- |
|   | EmployeeID | EmployeeName |   |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

**-- Create Store dimension**

create table DimStore

(StoreID int not null identity(1,1) primary key,

 City varchar(50) not null,

Country varchar (50) not null

)

go

|  |  |  |  |
| --- | --- | --- | --- |
|   | StoreID | City | Country |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

**-- Create Product dimension**

create table DimProduct

(ProductID int not null identity(1,1) primary key,

 Product\_Name varchar(50) not null,

Product\_Type varchar (50) not null

)

go

|  |  |  |  |
| --- | --- | --- | --- |
|   | ProductID | ProductName | ProductType |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

**-- Create Time Dimension**

create table DimTime

( Time\_Key int not null primary key,

 [Year] int, [Month] int, [Date] date, DateString varchar(10))

go

**-- Populate Time Dimension**

-- We can populate Time Dimension using the following script

truncate table DimTime

go

declare @i int, @Date date, @StartDate date, @EndDate date, @DateKey int,

 @DateString varchar(10), @Year varchar(4),

 @Month varchar(7), @Date1 varchar(20)

set @StartDate = '2018-01-01'

set @EndDate = '2028-12-31'

set @Date = @StartDate

insert into DimTime (DateKey, [Year], [Month], [Date], DateString)

 values (0, 'Unknown', 'Unknown', '01/01/0001', 'Unknown') --The unknown row

while @Date <= @EndDate

begin

 set @DateString = convert(varchar(10), @Date, 20)

 set @DateKey = convert(int, replace(@DateString,'-',''))

 set @Year = left(@DateString,4)

 set @Month = left(@DateString, 7)

 insert into DimTime (DateKey, [Year], [Month], [Date], DateString)

 values (@DateKey, @Year, @Month, @Date, @DateString)

 set @Date = dateadd(d, 1, @Date)

end

go

select \* from DimTime

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|   | Time\_Key | Year | Month | Date | DateString |
| 1 | 0 | Unknown | Unknown | 01/01/0001 | Unknown |
| 2 | 20180101 | 2018 | 2018-01 | 1/1/2018 | 1/1/2018 |
| 3 | 20180102 | 2018 | 2018-01 | 2/1/2018 | 2/1/2018 |
| 4 | 20180103 | 2018 | 2018-01 | 3/1/2018 | 3/1/2018 |
| 5 | 20180104 | 2018 | 2018-01 | 4/1/2018 | 4/1/2018 |
| 6 | 20180105 | 2018 | 2018-01 | 5/1/2018 | 5/1/2018 |

**create table FactSales**

(StoreID int not null,

 ProductID int not null, EmployeeID int not null, Time\_Key int not null, --Dimension Keys

 Item\_Sold int not null,

 Sales\_Amount decimal(8,2)

)

Go

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| StoreID | ProductID | EmployeeID | Time\_Key | Item\_Sold | Sales\_Amount |

Now, we are going to use SSIS to populate the dimension tables. Certainly, the simplest method to import the data into a data warehouse is using the Import Data wizard from SQL Server Management Studio. Succeeding the steps through the wizard, we can pick the source file, and the destination table to import the data.

1. Create new integration Service Project, create a new
package say “ImportMultipleExcelFiles.dtsx.
2. Create a Folder **ExcelFiles** (C:\ExcelFiles) and then create multiple
Excel files in folder C:\ExcelFiles but make sure all Excel files have an identical schema (format)
3. Right- Click on Control Flow
Window, Select Variables. Add a variable “**FileName**”
at Package Level having data type string.
4. Go to Control Flow add a “**For Each Loop Container**” Component.
5. Right-click on the Foreach Loop container and select Edit.
Then, Tick on Collection “Collection” tab. Assign folder path and file type



1. **After that, go to “Variable
Mappings “tab and map variable created above**
2. Slog a “Data Flow Task“inside
“Foreach Loop Container”, double- click on Data flow task. Drag one “Excel Source” task, double click on this to get “Excel Source Editor “Window.
3. Now drag “OLE DB Destination”
task, connect “Excel Source” to “OLE DB Destination”. Point this connection to your database and create a new table or use an existing table (we already created our dimension tables, so we are going to choose them). Map both.



1. Go to the Properties of “Excel Connection Manager”. Choose “Connection String” property and assign a value of Expression.
2. Click on the button **“**Evaluate Expression**”** at the bottom left of above window to check for any errors
3. Set property DelayValidation=TRUE on the Data Flow task.
4. Execute your Package



Now we can populate the Fact table, we can follow the below steps:

1. Get Employee ID from the excel sheet we get the Employee ID. We then go to the Employee Dimension to get the Employee ID.
2. Get the product ID
3. Get the Store ID
4. Get Time Key
5. Populate FactSales: when we get all the needed dimension keys, we insert the rows into the FactSales table.

We need to create a Lookup transformation and set to Full Cache, OLE DB connection manager, Fail component, like below:



According to Rainardi (2012), when using a Full cache, SSIS reads all rows from Dim tables into memory, and SSIS doesn’t need to read from the Dimensions anymore. As all lookup operations are done in memory.

The OLE DB connection manager will be the tables in the warehouse (i.e. DimEmployee, DimProduct, DimStore, and DimTime).

Then, we can populate the FactSales, which it’s the OLE DB destination. On the Connection Manager pane, specify FactSales as the target table.

On the Mapping window, the source column has the same name as the destination column, except for the Time\_Key. The 4-dimension key columns are populated with the surrogate key from the respective dimension.

Below is the overall workflow:



 Get EmployeeID

 Lookup Match Output

 Get ProductID

Lookup Match Output

  Populate FactSales

 Get StoreID

## ETL Mapping

According to Laberge (2011), a crucial tool utilized in data modeling for the ETL process specifications is the mapping document. The notion is that the document should signify the data flow from the source system to the final data mart and BI tool place, if different from the data mart. An example will include source system name, since there can be multiple source systems, and the source system fields, or table and column names, along with data types.

# Data Analysis (Week 4)

# Maintenance and SQL Script and Conclusion (Week 5)

# Reference

FYI Center. (2008). **What is data Acquisition Process? Retrieved from** <http://sqa.fyicenter.com/FAQ/ETL-Extract-Transform-Load/What_is_Data_Acquisition_Process_ETL.html>

Microsoft. (2018). Data Sources (DW)---a Technical Reference Guide for Designing Mission-Critical DW Solutions. Retrieved from

[https://technet.microsoft.com/en-us/library/hh393517(v=sql.110).aspx](https://technet.microsoft.com/en-us/library/hh393517%28v%3Dsql.110%29.aspx)

Baya, D. (2006, Feb 10). [Case Study of Building a Data Warehouse with Analysis Services (Part One)](http://www.informit.com/articles/article.aspx?p=443594). Retrieved from:

<http://www.informit.com/articles/article.aspx?p=443594&seqNum=2>

Thomas, C (2016, May 19). What is a Data Staging Area? Retrieved from:

<https://support.timextender.com/hc/en-us/articles/210438083-What-is-a-Data-Staging-Area->

DW Staging Area (2018). Guide to Data Warehousing and Business Intelligence. Retrieved from: <https://data-warehouses.net/architecture/staging.html>

[Exforsys](http://www.exforsys.com/author/Exforsys). (2007, Feb 11). How Data Is Stored Within a Data Warehouse. Retrieved from: <http://www.exforsys.com/tutorials/data-warehousing/how-data-is-stored-within-a-data-warehouse.html>

Joseph. F (2018). Comparison of Data Warehousing DBMS Platforms. Retrieved from:

<http://www.odbms.org/wp-content/uploads/2013/11/illuminate-Comparison.pdf>

Denise, R (2010, Nov 23). 3 Fundamental Questions about Data Marts… and their answers. Retrieved from: <https://www.databasejournal.com/sqletc/article.php/3913826/3-Fundamental-Questions-about-Data-Martsand-Their-Answers.htm>

Techopedia Inc. (2018). Multidimensional Database (MDB). Retrieved from:

<https://www.techopedia.com/definition/14710/multidimensional-database-mdb>

Marseille. A (2013). Data Warehouse/Data Mart Conceptual Modeling and Design. Retrieved from: <http://www.lsis.org/espinasseb/Supports/DWDM/4-DWConcepDesign-4p.pdf>

 Golfarelli, M., & Rizzi, S. (2009). Data warehouse design: modern principles and methodologies. New York: McGraw-Hill.

Abello, A., Samos, J., & Saltor, F. (2006). YAM2 : a multidimensional conceptual model extending UML. Information System, 31(6), 541-567.

S. Chaudhuri, and U. Dayal. An overview of data warehousing and OLAP technology. SIGMOD Record, vol. 26, n. 1, pp. 65-74, 1997.

Ruiter, A. (Microsoft 2012, Dec 5). Designing an ETL process with SSIS: two approaches to extracting and transforming data. Retrieved from: <https://blogs.msdn.microsoft.com/andreasderuiter/2012/12/05/designing-an-etl-process-with-ssis-two-approaches-to-extracting-and-transforming-data/>

Laberge, R. (2011). The data warehouse mentor: practical data warehouse and business intelligence insights. New York: McGraw-Hill.

Kantor, H. (2018, Feb 19). When (And How) to use Surrogate Keys. Received from: <https://www.sisense.com/blog/when-and-how-to-use-surrogate-keys/>

Rainardi, V. (2012, May 26). How to Populate a Fact Table using SSIS (part1). Retrieved from:

<https://dwbi1.wordpress.com/2012/05/26/how-to-populate-a-fact-table-using-ssis-part1/>