Unit 3

Designing the Target structure
Designing The Target Structure

• we have our entire source structures defined in the data warehouse Builder.
• But before we do anything with them, we need to design what our target data warehouse structure id going to look like
• Which will help to figure out , we can start mapping data from source to the target.
Data warehouse design

• When it comes to the design of a data warehouse, there is basically one option that makes the most sense for how we will structure our database and that is the dimensional model.

• This is way of looking at the data from a business perspective that makes the data simple, understanding and easy to query for the business end user.
• It doesn’t require a database administrator to be able to retrieve data from it.
• A normalized model removes redundancies in data by storing information in discrete tables, and then referencing those tables when needed.
• This is an advantage for a transactional system because information needs to be entered at only one place in the database, without duplicate any information already entered.
• For example in the ACME Toys and gizmos Transactional database each time a transaction is recorded for the sales of an item at a register, a record needs to be added only to the transaction table.

• In the table all details regarding the information to identity the register, the item information and the employee who proceed the transaction do not need to be entered because the information is already stored in separate tables.
Dimensional design

• A dimensional model takes the business rule of our organization and represent them in the database in more understanding way.

• A business manager looking at sales data is naturally going to think more along the lines of how many gizmos did I sell last month in all stores in the south and how does that compare to how many I sold in the last month last year?

• Managers just want to know the result, don’t want to worry about how many tables need to be joined

• Users can think of the data as a cube and edges of the cubes as dimension labeled as stores, product and time frame
Cube and dimension

- The dimensions become the business characteristics about the sales (Cubes are the main objects in online analytic processing (OLAP), a technology that provides fast access to data in a data warehouse. A cube is a set of data that is usually constructed from a subset of a data warehouse and is organized and summarized into a multidimensional structure defined by a set of dimensions and measures.)
- TIME DIMENSIONS : user can look back in time and check various time periods
- STORE DIMENSIONS : information can be retrieved by store and location
- PRODUCT DIMENSION : various product for sales can be broken out.
• Think of the dimension as the edges of a cube and the intersection of the dimension as the measures, we are interested in for the particular combination of time, store, product.

• Think of the width of the cube as the PRODUCT DIMENSIONS, every piece of information in the same row refers to the same product, so there are many rows.

• Think of the height of the cube going up and down as the STORE DIMENSION, every piece of information in a column represents one single store, so there are many columns as there are stores.

• Finally think of the dept of the cube as the time dimensions, so any piece of information in the rows and column at the same dept represents the same point in time.
• The intersection of each of these three dimensions locates a single individual cube is the big cube. In this case its dollar sales for a single product in a single stores at a single point of time
Implementation of a dimensional model in a database

• Now before we finalize our model for the ACME TOY and Gizmos data warehouse.
• Let look the implementation of the model how it get physically represented in database.
• There are two options a relational and multi dimensional implementation.
Relational implementation
(star schema)

• For a relational data warehouse design, the relational characteristics are retained between tables.
• But a design principle is followed to keep the number of levels of foreign key relationship to a minimum.
• It's much faster and easier to understand if we don't have to include multiple levels of referenced tables.
• For this reason, a data warehouse dimensional design that is represented relationally in the database, will have one main table to hold the primary facts, we want to store
• The er-diagram of such an implementation would be shaped somewhat like a star and thus the term star schema is used to refer to this kind of an implementation.

• The main table in the middle is referred to as fact table because it holds the fact, that we are interested in about our organization.

• The tables surrounding the fact tables are known as dimension tables. These are the dimensions of the cube.

• Its is job of data warehouse design to determine what piece of information need to be included.
• For a data warehouse however the query time and simplicity is of paramount importance over the duplication of data.
• Look at an example for ACME toys and gizmos
• Every product in our store is associated with a department
• If we have dimension for product information, one piece of information about the product would be the department it is in.
• We would creating a table department table to store department description with one row for each department and would use short key code to refer to the department record in the product table

• But in our data warehouse we would include that department information descriptions in the product dimensions

• This will result in the same information being duplicated for each product in the department.
To avoid such a problem we will need to de-normalize our data warehouse implementation and might want to include another level basically dimension table referenced by another dimension tables. This is a variation of the star-schema referred to as snowflake schema because with this type of implementation, dimension tables are partially normalized to pull common data out into secondary dimensions tables.
Multidimensional implementation (OLAP)

• A multidimensional implementation requires a database with special features that allow it to store cubes as actual objects in the database.

• It also provides advanced calculation and analytic content built into the database to facilitate advanced analytic querying.

• It is utilized to build a highly specialized data marts or a subset of the data warehouse (A data mart is a simple form of a data warehouse that is focused on a single subject (or functional area), such as Sales, Finance, or Marketing. Data marts are often built and controlled by a single department within an organization.

• The data mart then draws its data to load from the main data warehouse which would be a relational dimensional star schema.