Dimension Tables

Based on Chapter 05 Dimension Tables – the Nouns of the Data Warehouse in *Object-Oriented Data Warehouse Design: Building a Star Schema* by Wm. Giovinazzo

Star Schema

- Allows the creation of a multidimensional space within a relational database

Star Schema Tables

- Two types of tables

- Fact Table(s)
- Dimension Tables(s)

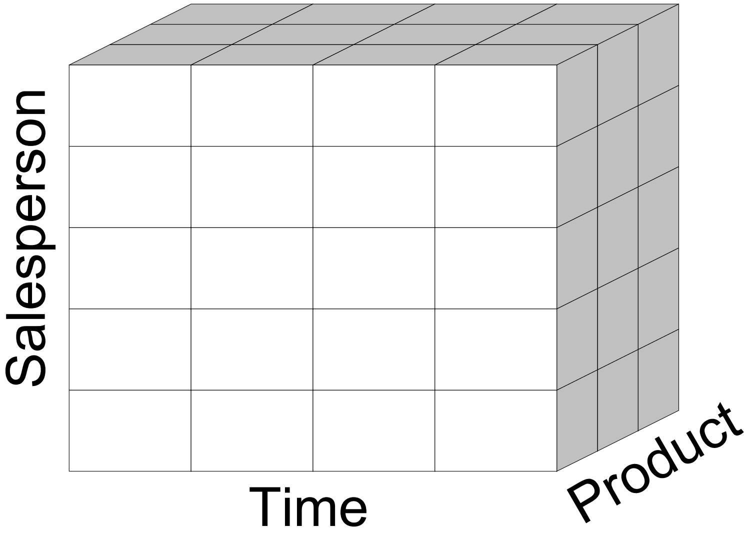
- Look at Dimension Tables first

Dimension Tables

- Dimension tables hold nouns, the objects
- Hold attributes that used to define the set of objects for analysis
  - These are analytical data elements
- Dimension tables are not normalized, typically
  - There are exceptions

Analytical Data Elements

- analytical data elements are the dimensions that the analyst wants to use to form the cubes
- For example:
  - Salesperson
  - Time
  - Product



Analytical Data Elements (Grouping)

- Analytical data elements group objects together
- Group records in the fact table based on some attribute
  - a dimension attribute
- Examples:
  - Good (Why?) Grouping Attribute: Car Engine Size
  - Poor (Why?) Grouping Attribute: Car Serial Number

    If an attribute cannot be used for analysis by grouping or by adding more detail, then should it be included in the data warehouse?

Analytical Data Elements - Finite Domain Characteristic

- The domain of the attributes should be finite
  - That is, there should not be uncountably many possibilities for values of dimension attributes
- The values for attributes are for the most part numeric fields with a limited number of possible values or text fields with a limited number of possible values
  - That is, categorical data or nominal data
- Thus, free format text fields are not very useful
  - difficult to group on

Slowly Changing Dimensions

- A fact is a fact
  - Facts are not volatile

- Objects -- represented in the dimension tables -- may change over time
  - Usually the change over time is slow
  - If it is not slow, then the object may not be suitable for data mining purposes

- Problem with dimensions that change
  - How do we allow change without losing the history

- Author suggests 5 ways to handle slowly changing dimensions

1. Change the Record

- Change the record -- lose the history of the change
  - Problem: history is changed as a result
- When is this a good choice?
  - To correct errors in the data
  - When the view of the past and the future changes permanently

    E.g., the company re-organizes its sales divisions into four rather than the previous three
- When is it not a good choice?
  - E.g. 15 years ago a woman made a purchase and she was single; now she makes a purchase and she is married – should the old data change from single to married? A: It depends, but probably not

2. Create a New Dimension Record

- Add a new record and keep the older record also
  - Maintains history
  - Accurately reflects current state
  - Problem: may hide valuable information
- When is this a good choice?
  - When trends related to the changing attribute are not considered valuable information
- When is it not a good choice?
  - When change in attribute could provide valuable relevant information
  - E.g., how do buying habits change when a person moves from single to married to married-with-children status

3. Alter the Structure of the Dimension Table

- Modify the structure if the dimension table so that it can capture the change in the value of an attribute
  - Create current_status and past_status fields within one record plus date of change
  - Problem: must be done for every attribute where slow change is expected and where it is something we want to track – may significantly increase table size
  - Problem: how many changes are to be captured? Past to current? What if the value changes again? More fields needed.

- When is this a good choice?
  - Limited number of slowly changing attributes
  - Slowly changing attributes with few change points to be recorded

- When is it not a good choice?
  - Many slowly changing attributes to track
  - Many change points to be recorded

4. Use Record Versioning

- Create a new record for each change - introduce a revision number that can be made part of the key (append to the PK)
  - PK portion identifies like instances
  - Revision number creates an ordered record of change
  - Also add a binary field indicator for which record is the current one (enhances look-up of this record)
  - Problem: complicates the key that joins dimensions to facts

5. Use Record Linking.

- Create a new record with a new PK
  - another attribute is used to link the records (i.e. identify them as belonging to the same "individual"
  - E.g., Use clientID as link

- When is this a good choice?

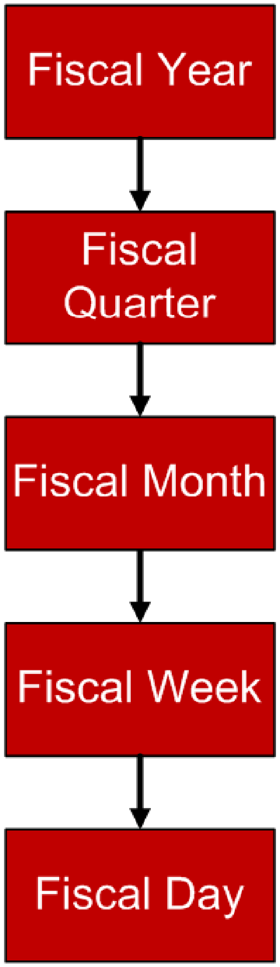
- When is it not a good choice?
  - Sometimes clientID may not be a consistent value due to bringing data in from disparate systems

Time Dimension Table

Why have a dimension table for time?

Analyze trends over time

Time Hierarchy



Fiscal Year

Fiscal Quarter

Fiscal Month

Fiscal Week

Fiscal Day

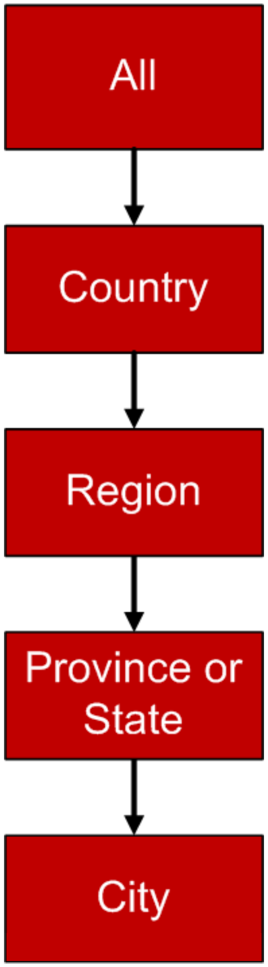
- May need flag for:
  - Holidays
  - Weekends
  - seasons

Location Dimension Table

Why have a dimension table for location?

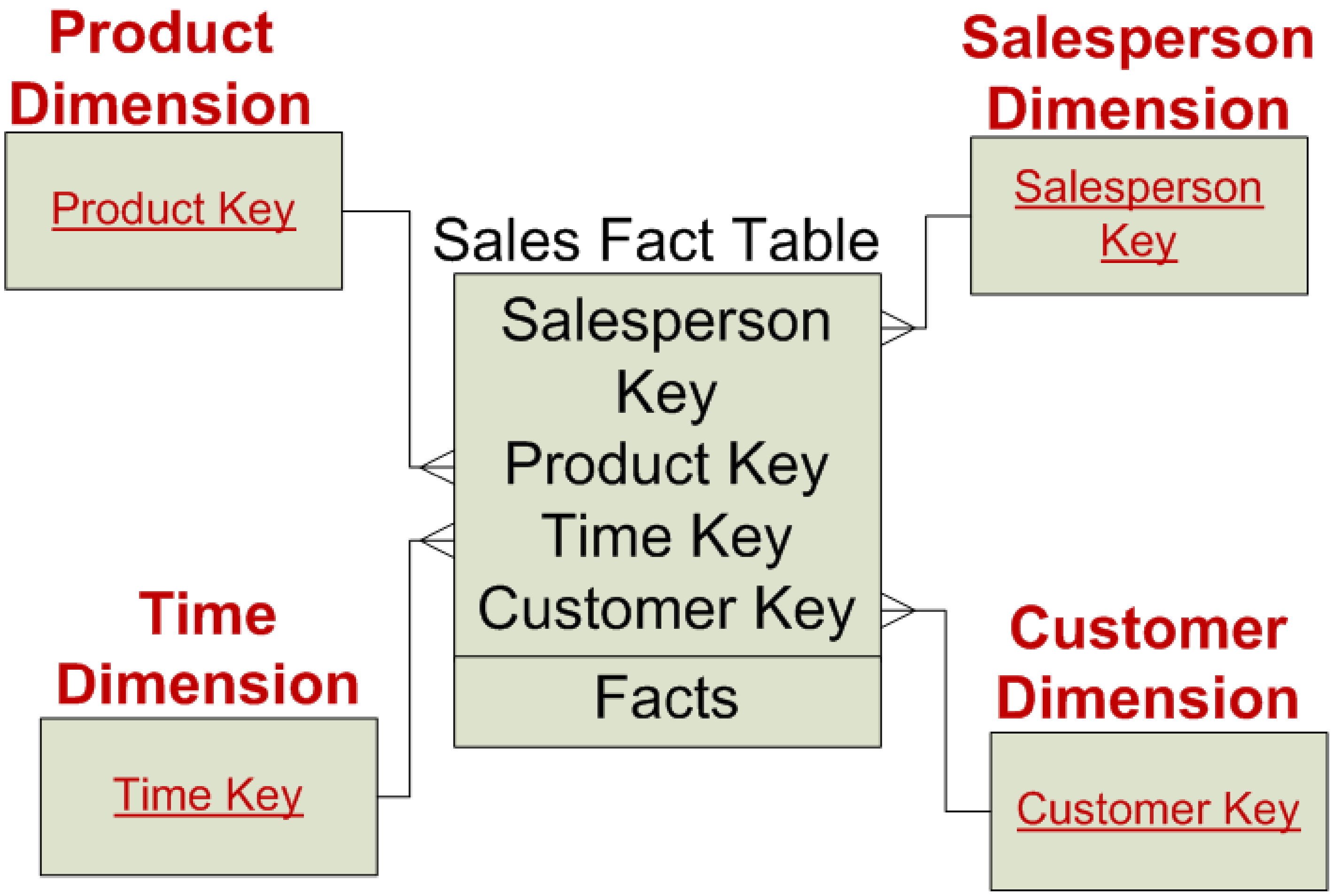
Analyze trends by location

Location Hierarchy



- Other possibilities denoting something about location
  - Zipcode
  - Time zone
  - Lattitude/longitude

Example – Car Dealership



Product Dimension

Product Key

Sales Fact Table

Salesperson Key

Product Key

Time Key

Customer Key

Facts

Salesperson Dimension

Salesperson Key

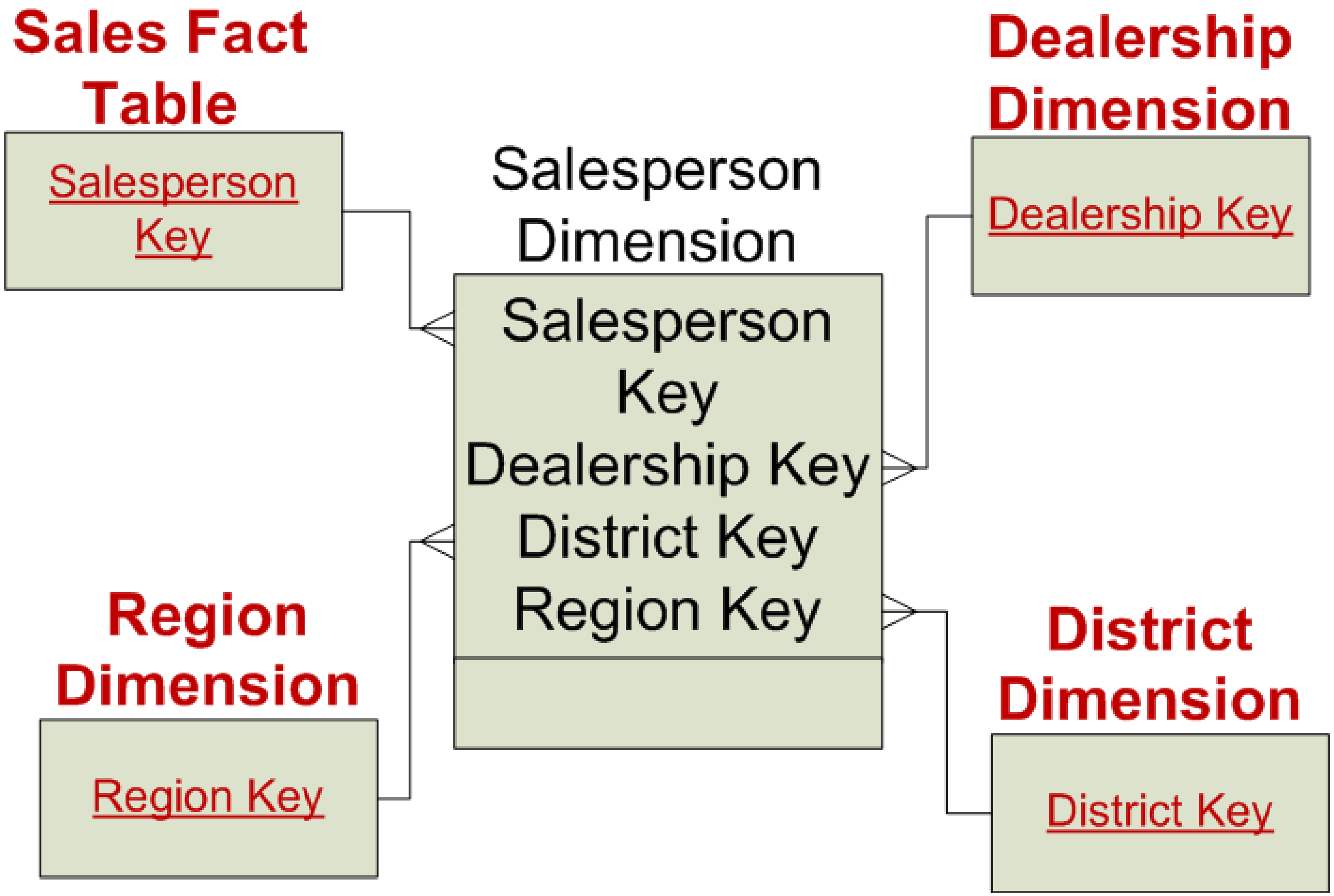
Time Dimension

Time Key

Customer Dimension

Customer Key

Example – Hierarchy Embedded in Salesperson Dimension



Dimension Tables

Based on Chapter 05 Dimension Tables – the Nouns of the Data Warehouse in *Object-Oriented Data Warehouse Design: Building a Star Schema* by Wm. Giovinazzo