# Business Intelligence (BI) and Data Warehousing (DW)

## Data Modeling Concepts

Jackline Ssanyu

February 1, 2020

# Introduction

- Data modeling is the **formalization** and **documentation** of **existing processes** and **events** that occur during e.g., application software design and development.
- Data modeling techniques and tools **capture and translate complex system designs** into **easily understood representations of the dataflows and processes**, creating a blueprint for construction and/or re-engineering.
- A data model can be thought of as a **diagram or flowchart** that illustrates the relation- ships between data.
- Data modelers often **use multiple models** to view the same data and ensure that all processes, entities, relationships and data flows have been identified.

# Importance of Data models

Well-documented models:

- allow stake-holders to identify errors and make changes before any programming code has been written.
- a data model can be easily verified because the model is built by using notations and language which are easy to understand and decipher.
- facilitate communication about information requirements.
- serve as a blueprint for the DBA when building the physical database. For example, the model can easily be used to define the key elements, such as the primary keys, foreign keys, and tables that will be used in the design of the data structure.

# Data modeling Techniques

- Data models are about **capturing and presenting information**.
- Every organization has information that is typically either in the **operational form** (such as OLTP applications) or **the informational form** (such as the data warehouse).
- The primary data modeling techniques are **E/R** and **dimensional modeling**.

# E/R modeling

- E/R modeling is a design technique in which we store the data in highly **normalized form** inside a relational database.
- The focus of the E/R model is to **capture the relationships between various entities** of the organization or process for which we design the model.
- The E/R diagram is a tool that can help in the **analysis** of business requirements and in the **design** of the resulting data structure.
- ER models are **normalized**. The E/R model basically focuses on three things, **entities, attributes**, and **relationships**.
- People use E/R modeling primarily when designing for highly **transaction-oriented OLTP applications**.
- For the OLTP applications, the goal of a well-designed E/R data model is to **efficiently and quickly get the data inside** (Insert, Update, Delete) the database.

# Advantages of E/R modeling

- It **eliminates redundant data**, which saves storage space, and better enables enforcement of integrity constraints.
- The **INSERT, UPDATE, and DELETE** commands **executed on a normalized E/R model** are much faster than on a denormalized model because there are fewer redundant sources of the data, resulting in fewer executions.
- The E/R modeling technique **helps capture the interrelationships among various entities** for which you are designing the database. In other words, an E/R model is very good at representing relationships.

# Disadvantage of E/R modeling

- E/R model is **not efficient** when **performing very large queries involving multiple tables**. In other words, an E/R model is good at INSERT, UPDATE, or DELETE processing, but not as good for SELECT processing.

# Dimensional Modeling

- To overcome performance issues for **large queries** in the data warehouse, we use **dimensional models**.
- The **dimensional modeling** approach provides a way to **improve query performance** for summary reports without affecting data integrity.
- A dimensional model is also commonly called a **star schema**. This type of model is very popular in data warehousing because it can **provide much better query performance**, especially **on very large queries**, than an E/R model.
- The dimensional model consists of **two types of tables** having different characteristics. They are:
  - Fact table
  - Dimension table

# Fact table

- A fact table is a table that contains the measures of interest.
- For example, sales amount would be such a measure. This measure is stored in the fact table with the appropriate granularity. For example, it can be sales amount by store by day. In this case, the fact table would contain three columns: A date column, a store column, and a sales amount column.

# Fact table characteristics

- It **contains numeric measurements** (values) that represent a specific business aspect or activity. Example:
  - sales figures are numeric measurements that represent product and/or service sales.
  - Facts commonly used in business data analysis are **units, costs, prices, and revenues**.
- Each fact table **contains the keys to associated dimension tables**. These are called foreign keys in the fact table.
- The information in a fact table has characteristics, such as:
  - It is **numerical and used to generate aggregates and summaries**.
  - Data values can be **additive, semi-additive, or non-additive**, to enable summarization of a large number of values.
  - All **other facts in the table must refer directly to the dimension keys**.This enables access to additional information from the dimension tables.

# Additive Facts

- The most flexible and useful facts are **fully additive**; additive measures can be summed across any of the dimensions associated with the fact table.
- **For example,**
  - Consider the following three **dimensions**: Time, Store and Product, and the **facts**: sales and costs. The purpose is to store sales made and costs incurred on each product. In this case sales and costs are additive facts. Because you can sum them up across all the three dimensions. For example, the sum of sales for all 7 days in a week represents the total sales amount for that week.

# Semi-additive Facts

- **Semi-additive** measures can be summed across some dimensions, but not all (especially time);
- **Examples,**
  - Consider two **dimensions**: Time and BankAccount, and two **facts** Account_balance and Profit_Margin.
    - Current_Balance is a semi-additive fact, as it makes sense to add them up for all accounts (what's the total current balance for all accounts in the bank?), but it does not make sense to add them up through time (adding up all current balances for a given account for each day of the month does not give us any useful information).
    - Profit_Margin is a non-additive fact, for it does not make sense to add them up for the account level or the day level.
  - Stock levels are also common semi-additive facts because they are additive across all dimensions except time. If you had 100 in stock yesterday, and 50 in stock today, your total stock is 50, not 150. It doesn't make sense to add up the measures over time, you need to find the most recent value.
  - Others: salary, test results etc.

# Non-additive Facts

- Some measures are completely **non-additive**, such as ratios. i.e. cannot be summed up for any of the dimensions present in the fact table. A good approach for non-additive facts is, where possible, to store the fully additive components of the non-additive measure and sum these components into the final answer set before calculating the final non-additive fact. This final calculation is often done in the BI layer or OLAP cube.

# Types of Fact Tables

Based on the above classifications, there are two types of fact tables:

- **Cumulative:** This type of fact table describes what has happened over a period of time. For example, this fact table may describe the total sales by product by store by day. The facts for this type of fact tables are mostly additive facts.
- **Snapshot:** This type of fact table describes the state of things in a particular instance of time, and usually includes more semi-additive and non-additive facts.

# Dimension table

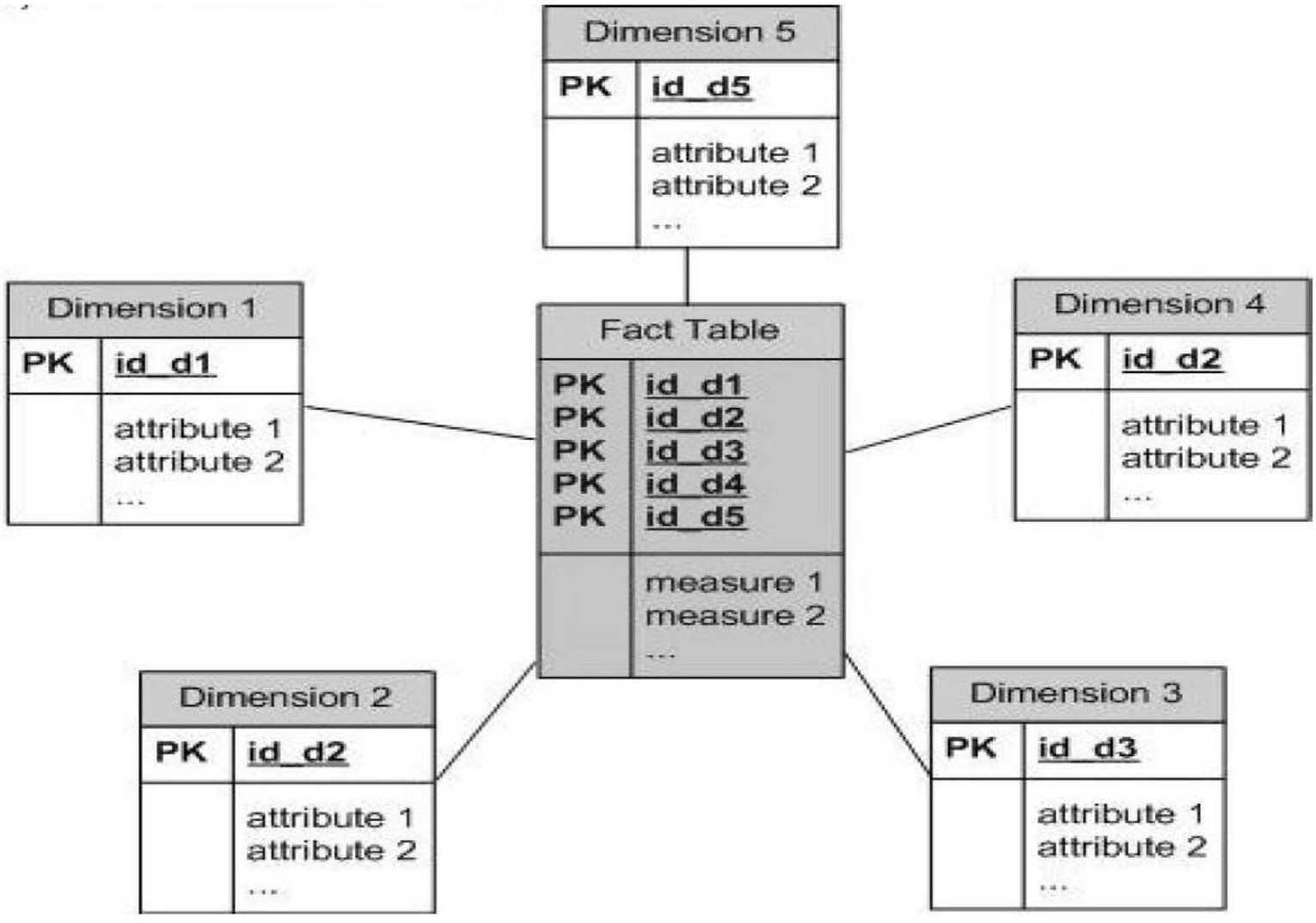
A dimension table contains dimension keys, values and attributes. For example:

- The time dimension would contain every hour, day, week, month, quarter and year that has occurred since you started your business operations.
- Product dimension could contain a name and description of products you sell, their unit price, color, weight and other attributes as applicable.

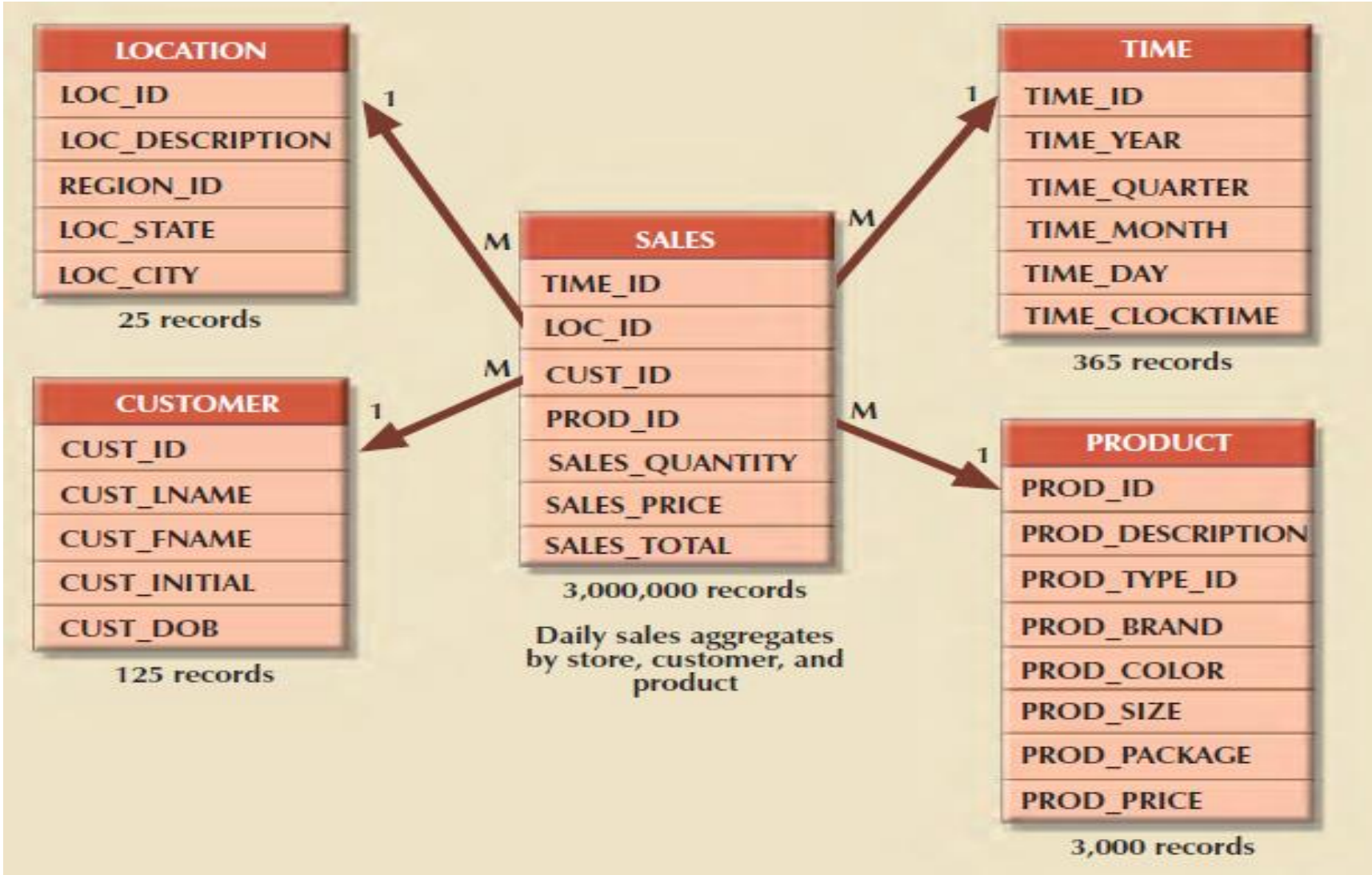
# Dimension table characteristics

- Contain characteristics (attributes) that provide additional perspectives to a given fact.
- Attributes include:
  - A primary key (equivalent foreign key in the fact Table).
  - The **hierarchy attributes**- Consider a business hierarchy – pin-code to city to district to state to country for location dimension. This means that each hierarchy element will be an attribute.
  - **Textual as well as the code attributes**- Location code as well as the name of the location. This is required, because both could be used for different reasons by different users.
  - **Include all parallel hierarchies** Consider the different criteria for analyzing a particular data and include all attributes that can make it possible.

# General form of Fact Table and Dimension Tables



# Example of Fact Table and Dimension Tables

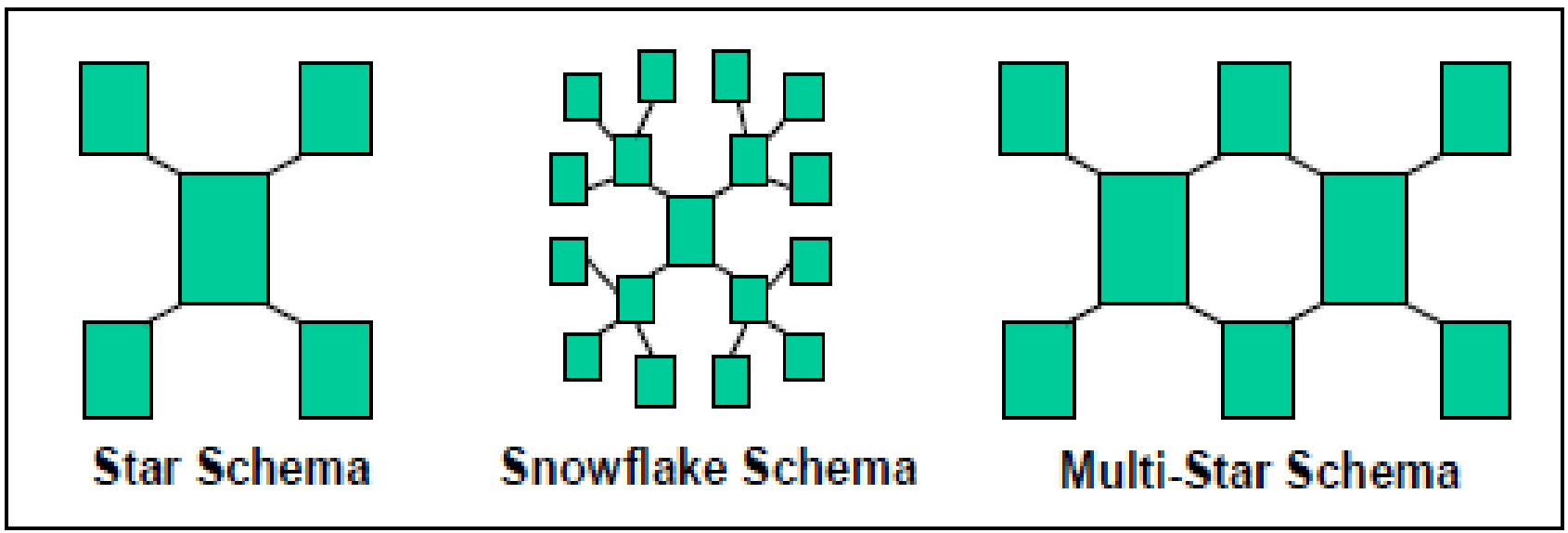


# Types of dimensional models

There are three basic types of dimensional models, and they are:

- Star model - has **one fact table** and **several dimension** tables. The **dimension tables are not denormalized**.
- Snowflake model - **normalizes and expands dimensions** to eliminate redundancy. That is, the dimension data has been grouped into multiple tables instead of one large table.
- Multi-star (Fact constellation) model - consists of **multiple fact tables**, joined together through dimensions.

# Types of Dimension models



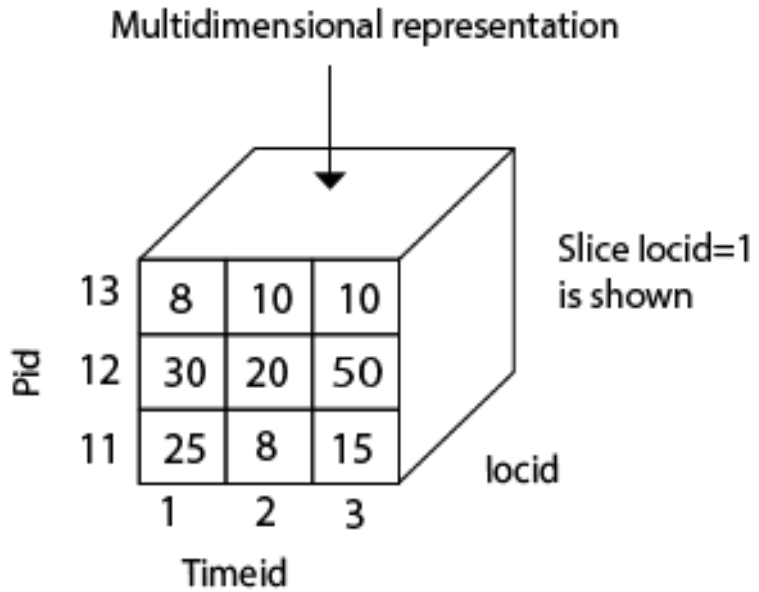
# What is Multi-Dimensional Data Model?

- A multidimensional model views data in the form of a data-cube. A data cube enables data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.
- A multidimensional data model is organized around a central theme, for example, sales. This theme is represented by a fact table. Facts are numerical measures. The fact table contains the names of the facts or measures of the related dimensional tables.

# Multidimensional data model - Example

Tabular representation

| Pid | Timeid | locid | Sales |
|---|---|---|---|
| 11 | 1 | 1 | 25 |
| 11 | 2 | 1 | 8 |
| 11 | 3 | 1 | 15 |
| 12 | 1 | 1 | 30 |
| 12 | 2 | 1 | 20 |
| 12 | 3 | 1 | 50 |
| 13 | 1 | 1 | 8 |
| 13 | 2 | 1 | 10 |
| 13 | 3 | 1 | 10 |
| 11 | 1 | 2 | 35 |



# Data Warehouse Modeling

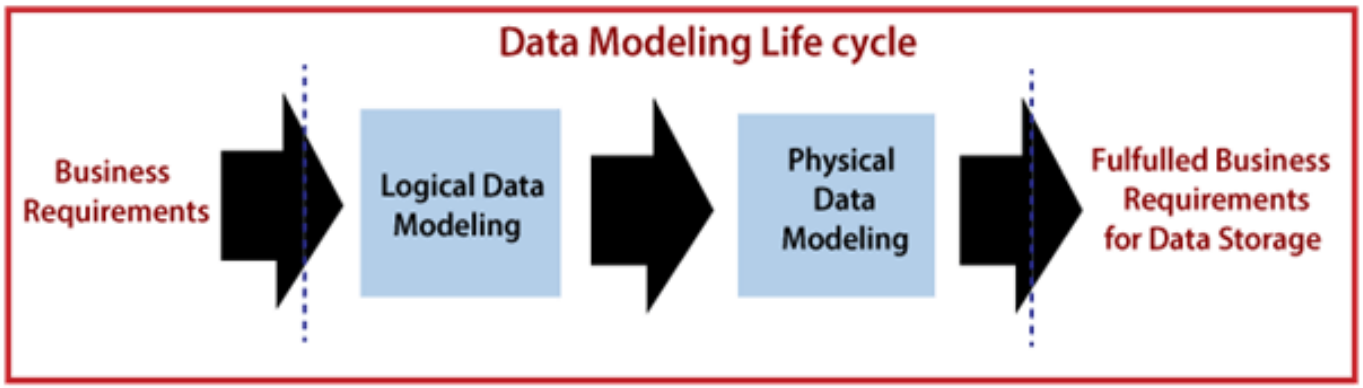
- Data warehouse modeling is the process of designing the schemas of the detailed and summarized information of the data warehouse.
- Data warehouse modeling is an essential stage of building a data warehouse for two main reasons.
  - through the schema, data warehouse clients can visualize the relationships among the warehouse data, to use them with greater ease.
  - a well-designed schema allows an effective data warehouse structure to emerge, to help decrease the cost of implementing the warehouse and improve the efficiency of using it.

# Data Warehouse Modeling (Cont.)

- Data modeling in data warehouses is different from data modeling in operational database systems.
  - The primary function of data warehouses is to support DSS processes. Thus, the objective of data warehouse modeling is to make the data warehouse efficiently support complex queries on long term information.
  - In contrast, data modeling in operational database systems targets efficiently supporting simple transactions in the database such as retrieving, inserting, deleting, and changing data.

# Data Modeling Life Cycle

A straight forward process of transforming the business requirements to fulfill the goals for storing, maintaining, and accessing the data within IT systems. The result is a logical and physical data model for an enterprise data warehouse.



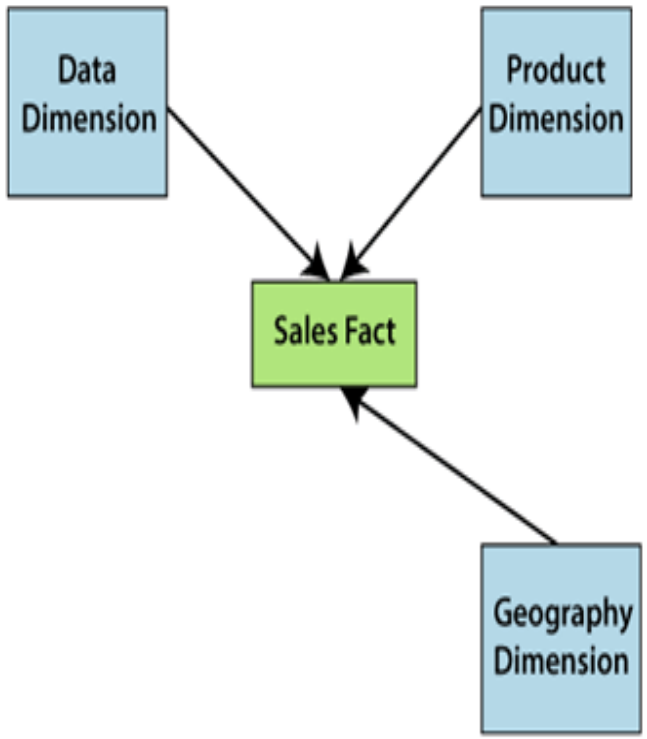

A generic data modeling life cycle

# Conceptual modeling

- A conceptual data model recognizes the highest-level relationships between the different entities.
- **represents an early basis for design reviews**, including confirmation that the business requirements are sufficiently described and that there is an available solution.
- Features of conceptual data model include:
  - Includes the **important entities and the relationships** among them.
  - **No attribute** is specified.
  - **No primary key** is specified.
- From this point **starts the logical data modeling** which transforms the business requirements into the context of the data/information necessary to be stored, accessed, and maintained.

# Conceptual Model - Example

The only data shown via the conceptual data model is the entities that define the data and the relationships between those entities. No other data.




**Example of Conceptual Data Model**

# Logical modeling

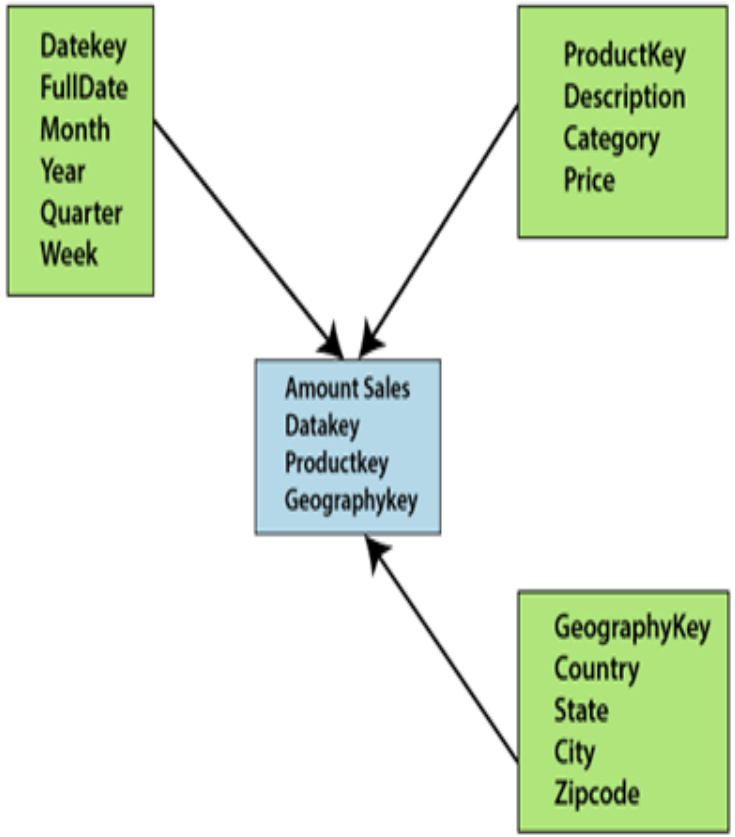
- A logical data model defines the information in as much structure as possible, without observing how they will be physically achieved in the database.
- The primary objective of logical data modeling is to document the business data structures, processes, rules, and relationships by a single view - the logical data model.
- Features include:
  - Includes all entities and relationships among them.
  - All attributes for each entity are specified.
  - The primary key for each entity is specified.
  - Foreign keys (keys identifying the relationship between different entities) are specified.

# Logical modeling (Cont.)

The phase for designing the logical includes:

- Specify primary keys for all entities.
- List the relationships between different entities.
- List all attributes for each entity.
- Normalization.
- No data types are listed

# Logical Model - Example




**Example of Logical Data Model**

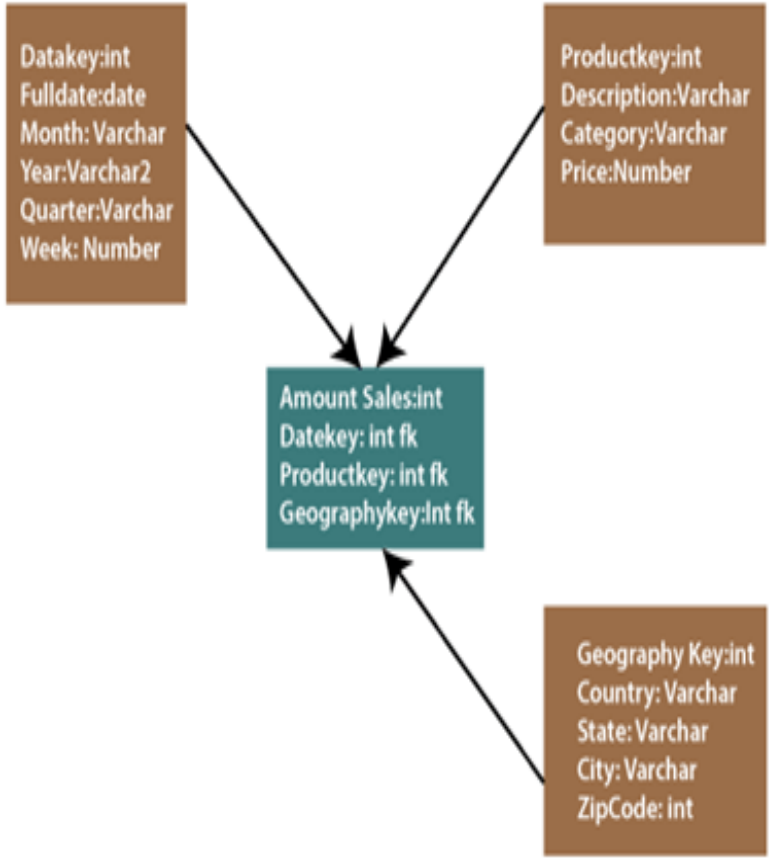
# Physical modeling

- Physical data model describes how the model will be presented in the database.
- maps the **logical data model to the target database management system** (DBMS) in a manner that meets the system performance and storage volume requirements.
- A physical database model demonstrates all table structures, column names, data types, constraints, primary key, foreign key, and relationships between tables.
- The purpose of physical data modeling is the mapping of the logical data model to the physical structures of the RDBMS system hosting the data warehouse.
- This contains defining physical RDBMS structures, such as tables and data types to use when storing the information. It may also include the definition of new data structures for enhancing query performance.

# Physical modeling (Cont.)

- Characteristics of a physical data model
  - Specification all tables and columns.
  - Foreign keys are used to recognize relationships between tables.
- The steps for physical data model design include:
  - Convert entities to tables.
  - Convert relationships to foreign keys.
  - Convert attributes to columns.

# Physical Model - Example




**Example of Physical Data Model**

# Dimensional model Design Process

A dimensional model design life cycle (DMDL) which consists of the following phases:

- Identify **business process requirements**
- Identify the **grain**
- Identify the **dimensions**
- Identify the **facts**
- **Verify** the model
- **Physical design** considerations
- **Meta data** management

# Identify business process requirements

- Involves selection of the business process for which the dimensional model will be designed.
- Based on the selection, the requirements for the business process are gathered.
- Selecting a single business process, out of all those that exist in a company, often requires prioritizing the business processes according to criteria, such as business process significance, quality of data in the source systems, and the feasibility and complexity of the business processes.
- In a manufacturing company, the business processes may include: inventory management, procurement, distribution, production, accounting, sales invoicing.
- The output of this phase results in the requirements gathering report.

# Identify the grain

- Identify the grain definition for the business process.
- The grain of the dimensional model is the finest level of detail (most atomic level) that is implied when the fact and dimension tables are joined.
- The grain conveys the level of detail that is associated with the fact table measurements. The grain of a sales fact table might be stated as "total sales by day by product by store".
- If more detail is included, the level of granularity is lower. If less detail is included, the level of granularity is higher.
- Declaring the grain means saying exactly what a fact table record represents. For example, grain definitions can include the following items:
  - A line item on a grocery receipt
  - A monthly snapshot of a bank account statement
  - A single airline ticket purchased on a day
  - A line item on a bill received from a doctor
  - Details on a student transcript

# Identify the grain...

- Considering:
  - "A line item on a grocery receipt", the grain can be "item sale information by location by the day."
  - "A line item on a bill received from a doctor", the grain can be "bill amount per patient per doctor per day per diagnosis"
- If more than one grain definition exists for a single business process, then we must design separate fact tables. Fact tables are often defined by their grain.

# Guidelines for choosing the grain

Guidelines for choosing the grain definition include the following considerations:

- During the business requirements gathering phase, try to collect any documents, such as **invoice forms, order forms, and sales receipts**. Typically, these documents have transactional data associated with them, such as order number and invoice number.
- Documents can often point you to the important elements of the business, such as customer and the products. The documents often contain information at the lowest level that may be required by the business.
- Another important point to consider is the date. Understand what level of detail is associated with a customer, product, or supplier. Is the information in the source systems available at the day, month, or year level?

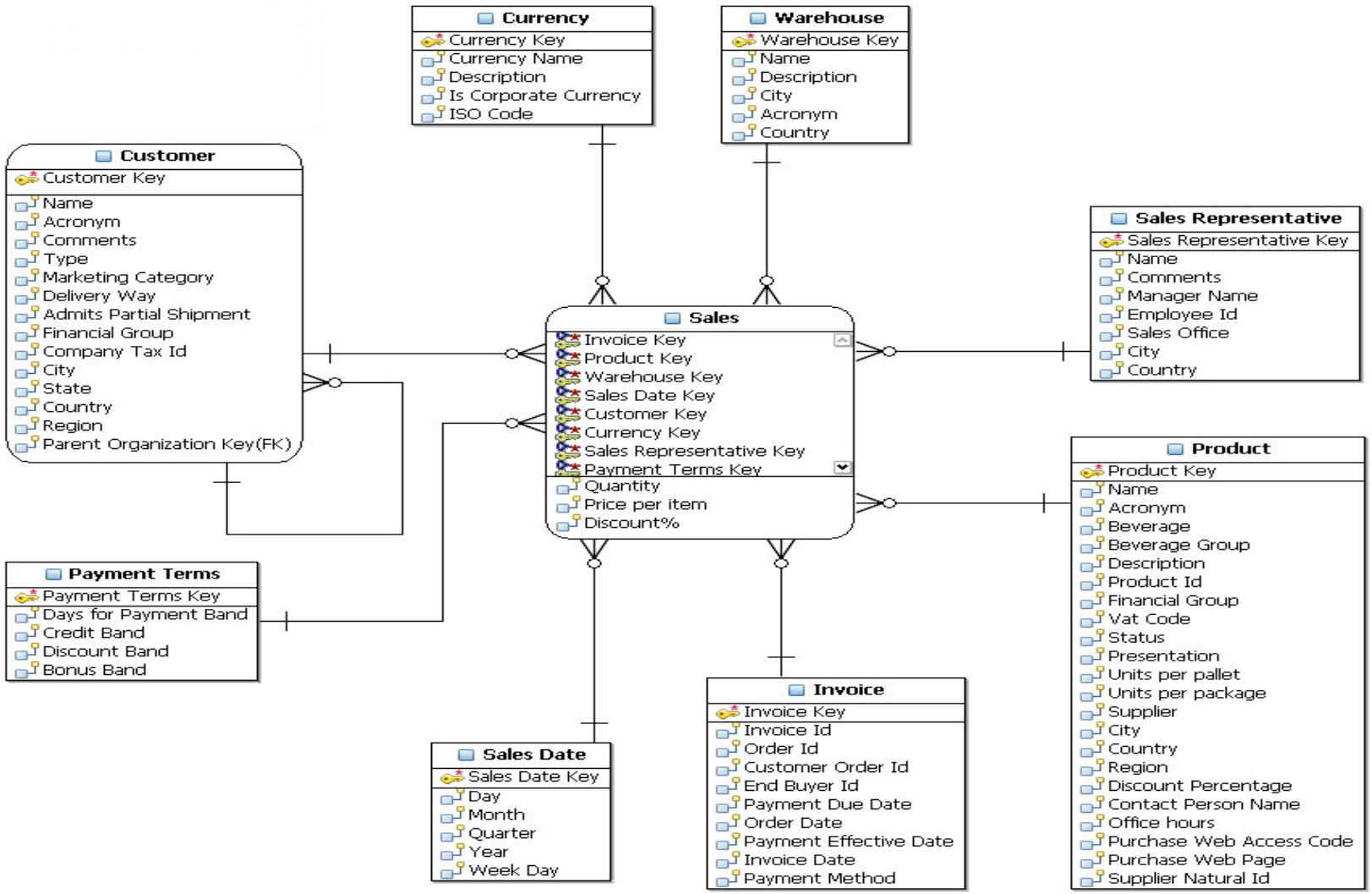
# Identify dimensions, facts & verify model

- Here we identify the dimensions that are valid for the grain chosen in the previous step.
- Now we identify the facts that are valid for the grain definition we chose.
- Before continuing, we must verify that the dimensional model can meet the business requirements. Sometimes it may be required to revisit, and perhaps change, the definition of the grain to assure we can meet the requirements.

# Identify dimensions.- Example

- Considering the grain - "bill amount per patient per doctor per day per diagnosis", sample dimensions can be: Billing, Patient, Doctor, Diagnosis and Date

# Sales Star Schema



# Physical design considerations

- Now that the model has been designed, we can **focus on other considerations, such as performance**. It may require tuning by taking actions such as **indexing**.

# Meta data management

- To **properly understand the model**, and be able **to confirm that it meets requirements**, you **must have access to the meta data** that describes the dimensional model in business terms that are easily understood.
- Both **non-technical** meta data and **technical** meta data should be documented.
- At the dimensional model level, a list should be provided of what is available in the data warehouse, such as
  - models, dimensions, facts, and measures
  - For **each model**, provide a **name, definition** (what is modeled), and **purpose** (what the model is used for).
  - The meta data for the model should also contain a list of **dimensions**, **facts**, and **measures** associated with it, as well as the name of a **contact person** so that users can get additional information when there are questions about the model.
  - A **name, definition**, and **aliases** must be provided for all dimensions, dimension attributes, facts, and measures.

# Meta data management....

- The attributes of a dimension are used to identify which facts to analyze.
- For attributes to be used effectively, meta data about them should include the data type, domain, and derivation rules.
- The domain of an attribute defines the set of valid values.
- For attributes with derived values, the rules for determining the value must be documented.
- Meta data about a fact should include, the derivation rules and dimensions associated with the fact, and the grain of time or date for the fact. For example,
  - the date dimension may be at the day level.
  - the time dimension may be at the hourly level.

# Questions

- Describe the differences between a dimensional model and E/R model.
- Consider ordering a book from Amazon.com
  - Draw an entity-relationship diagram for ordering a book from Amazon.com. (Include all processes). **NB:** You may visit www.amazon.com to familiarize yourself with ordering online, if you have not yet covered an-E-Commerce course.
  - How can you link the ER model in (a) to a database model?
  - Assuming that Amazon has many databases across the globe, describe how the transactions at Amazon can exploit the BI/DW concepts.
- What are the advantages of a dimensional model to data warehouses that the ER model lacks.