Dimension Modeling

- Overview
  - Requirements to design
  - Dimension modeling and E-R modeling
  - The STAR schema
  - Inside Fact and Dimensional tables
  - Advantages of STAR schema
  - Dimension modeling and OLAP

From Requirements to Data Design

- Data design consists of putting together the data structures.
  - Group of data elements form a data structure
- Information packages from requirements form the basis for the logical design.
  - Data design process results in a dimensional data model

Design decisions

- Choosing the process
  - Selecting the subject from information package for the first set of logical structures to be designed
  - Each subject will become a data mart
  - What are my sales over time?
- Choosing the grain
  - Level of detail
  - Each individual transaction or a summary?
- Identifying and conforming the dimensions
  - Must look the same for all data marts to link all data later (I.e time dimension)

Design decisions (cont..)

- Choosing the facts
  - Selecting the metrics or units of measurement
  - Rand sales, sale units
- Choosing the duration
  - How far back in time you should go for historical data

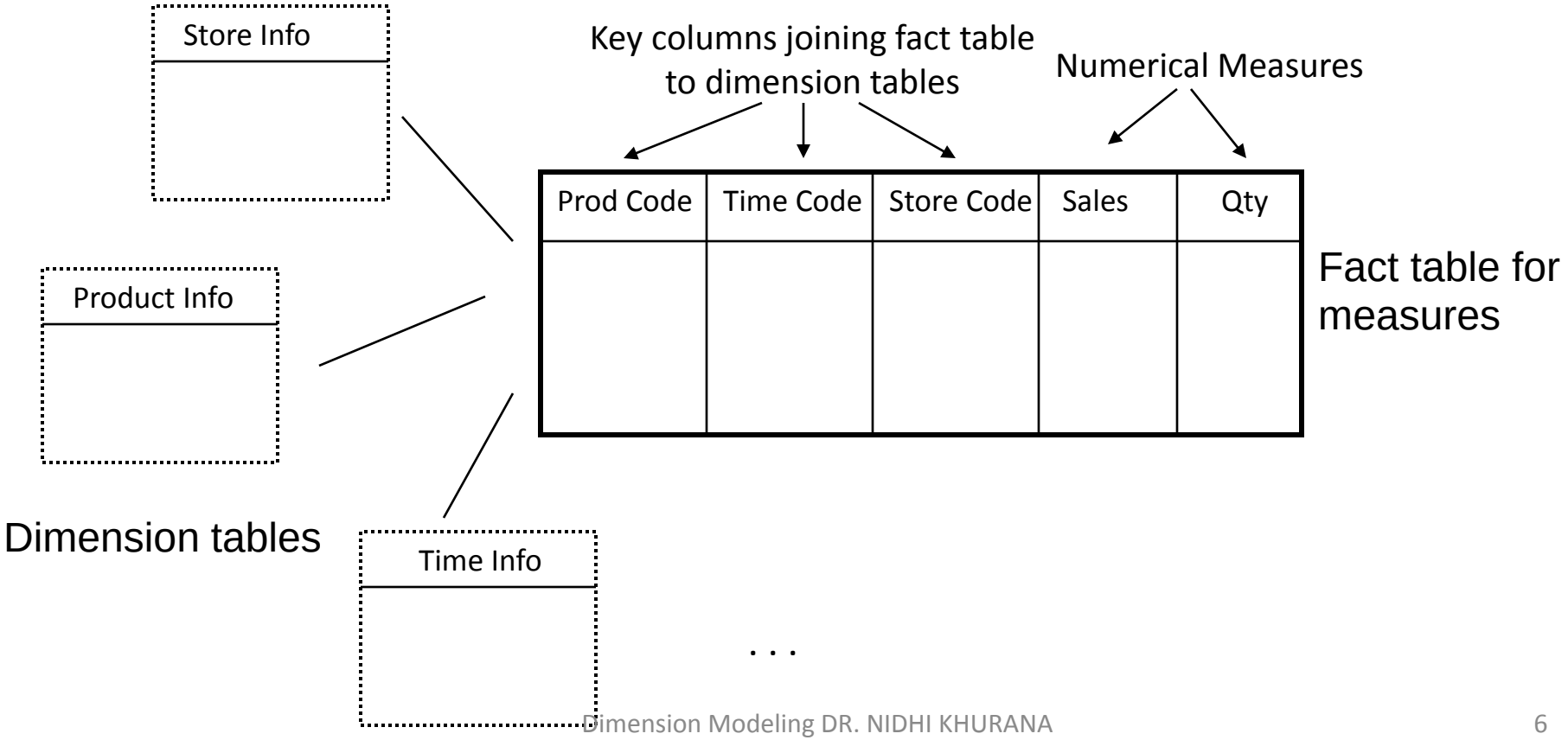
Dimension Modeling Basics

- Measures - numerical data being tracked
- Dimensions - business parameters that define a transaction
- Example: Analyst may want to view ***sales*** data (measure) by *geography*, by *time*, and by *product* (dimensions)
- Dimensional modeling is a technique for structuring data around the business concepts
- ER models describe “entities” and “relationships”
- Dimensional models describe “measures” and “dimensions”

The Multi-Dimensional Model

"Sales by product line over the past six months"

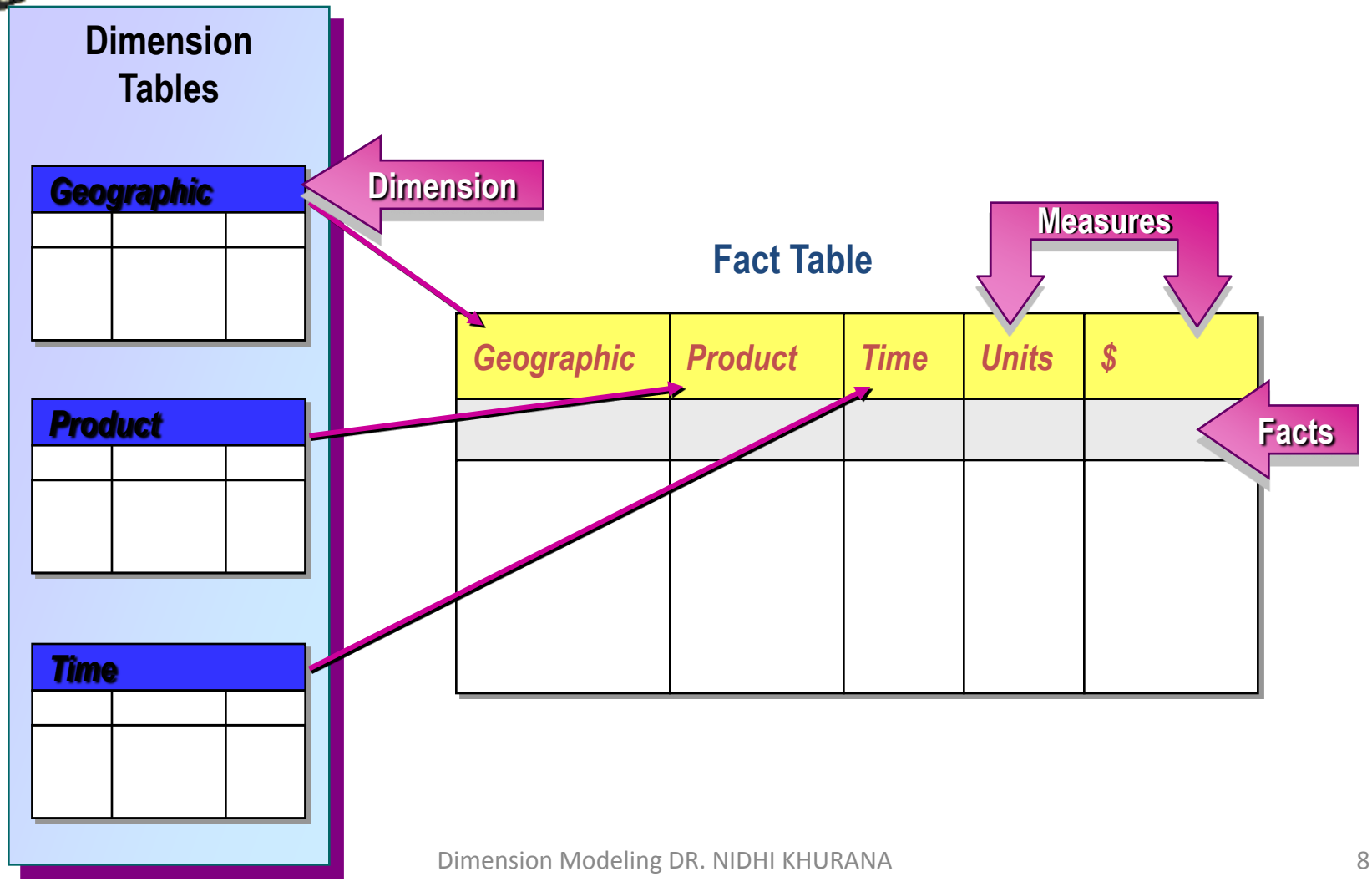
"Sales by store between 1990 and 1995"



Dimension Modeling Criteria

- The model should provide the best data access
- Model must be query-centric
- Must be optimized for queries and analyses
- Should show that the dimension tables interact with the fact table
- Should be structured that every dimension can interact equally with the fact table
- Should allow drilling down or rolling up along dimension hierarchies
- Example: STAR schema of practical

Dimension Modeling Components



ER Modeling

- Entity representing a class objects which are things in the real world that can be observed and classified by their property.
  - A person, place, event, …
  - An entity is represented by a rectangle
- Relationship
  - It depicts the structural interaction and association among the entities in a model

The objective of dimensional modelling is to represent a set of business measurements in a standard framework that is easily understandable by end users. A Dimensional model contains the same information as an ER model but packages the data in a symmetric format whose design goals are

- oUser understand ability
- oQuery Performance
- oResilience to Change

The main components of a Dimensional Model are Fact Tables and Dimension Tables.

- A fact table is the primary table in each dimensional model that is meant to contain measurements of the business.
- The most useful facts are numeric and additive.
- Every fact table represents a many to many relationship and every fact table contains a set of two or more foreign keys that join to their respective dimension tables

Designing a Dimensional Model: Steps Involved

Step 1 - Select the Business Process

Step 2 - Declare the Grain

Step 3 – Choose the Dimensions

Step 4 – Identify the Facts

Dimensional Modeling is the only viable technique for delivering data to the end users in a data warehouse.

Dimensional Modeling is the name of a logical design technique often used for data warehouses. It is different from entity-relationship modeling. ER modeling is very useful for transaction capture in OLTP systems

Comparison between ER and Dimensional Modelling

The characteristics of ER Model are well understood; its ability to support operational processes is its underlying characteristic. The conventional ER models are constituted to

a. Remove redundancy in the data model
b. Facilitate retrieval of individual records having certain critical identifiers and
c. Therefore, optimize online transaction processing (OLTP) performance

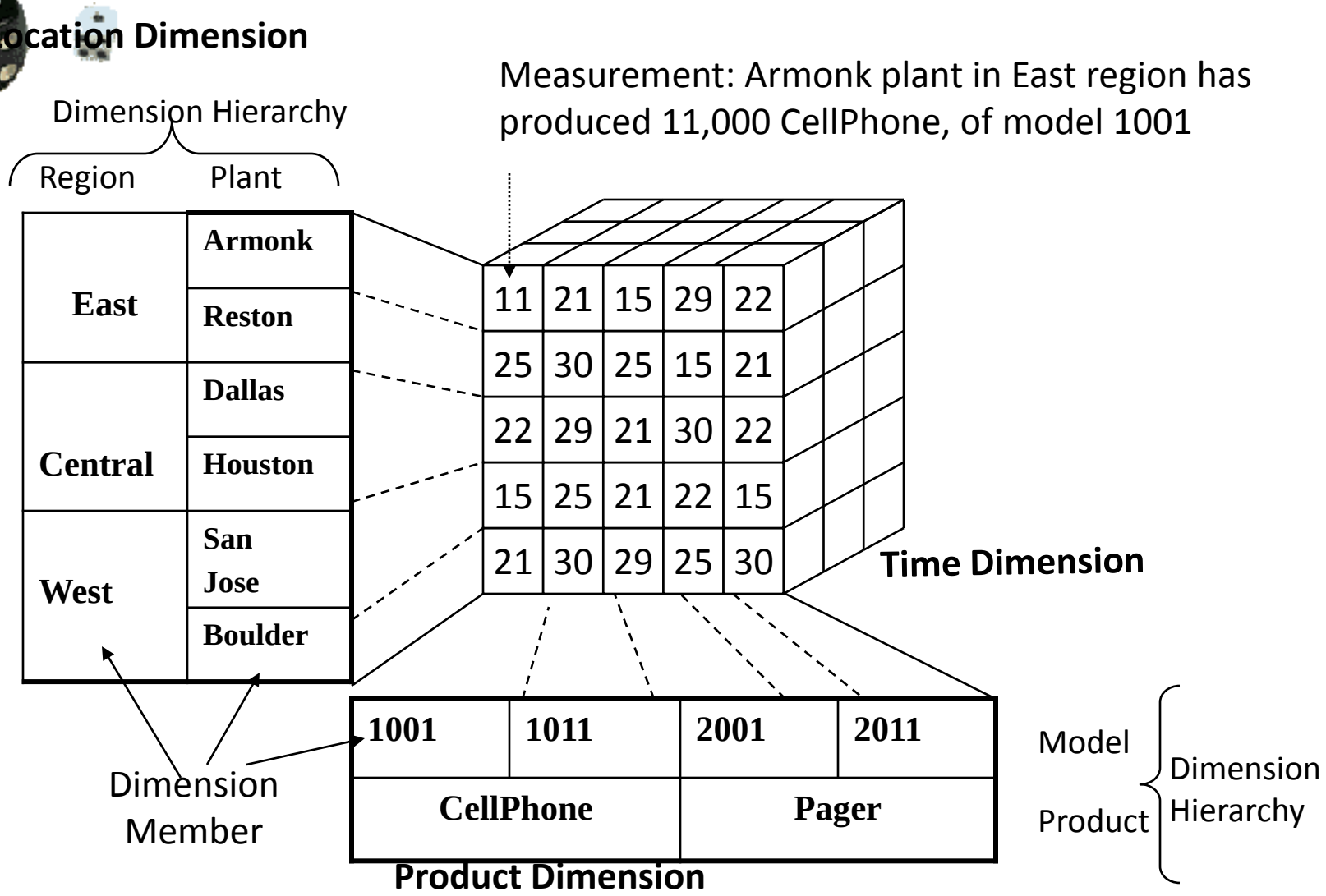
- Why ER is not suitable for Data Warehouses?
  - oEnd user cannot understand or remember an ER Model. End User cannot navigate an ER Model. There is no graphical user interface or GUI that takes a general ER diagram and makes it usable by end users.
  - oER modeling is not optimized for complex, ad-hoc queries. They are optimized for repetitive narrow queries

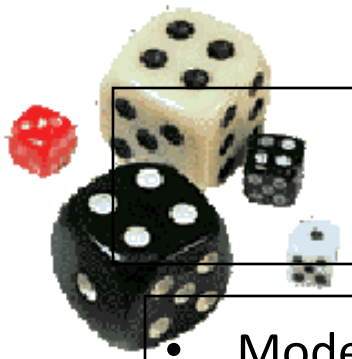
Use of ER modeling technique defeats this basic allure of data warehousing, namely intuitive and high performance retrieval of data because it leads to highly normalized relational tables.

E-R v. Dimension Modeling

- OLTP captures detail of events or transactions.
- E-R model removes data redundancy.
- Normalized.
- E-R model illuminates microscopic relationships.
- E-R model leads to numerous tables and difficult to understand diagrams.
- Query optimizers have difficulty with performance.
- E-R leads to efficient data storage

Visualization of a Dimensional Model

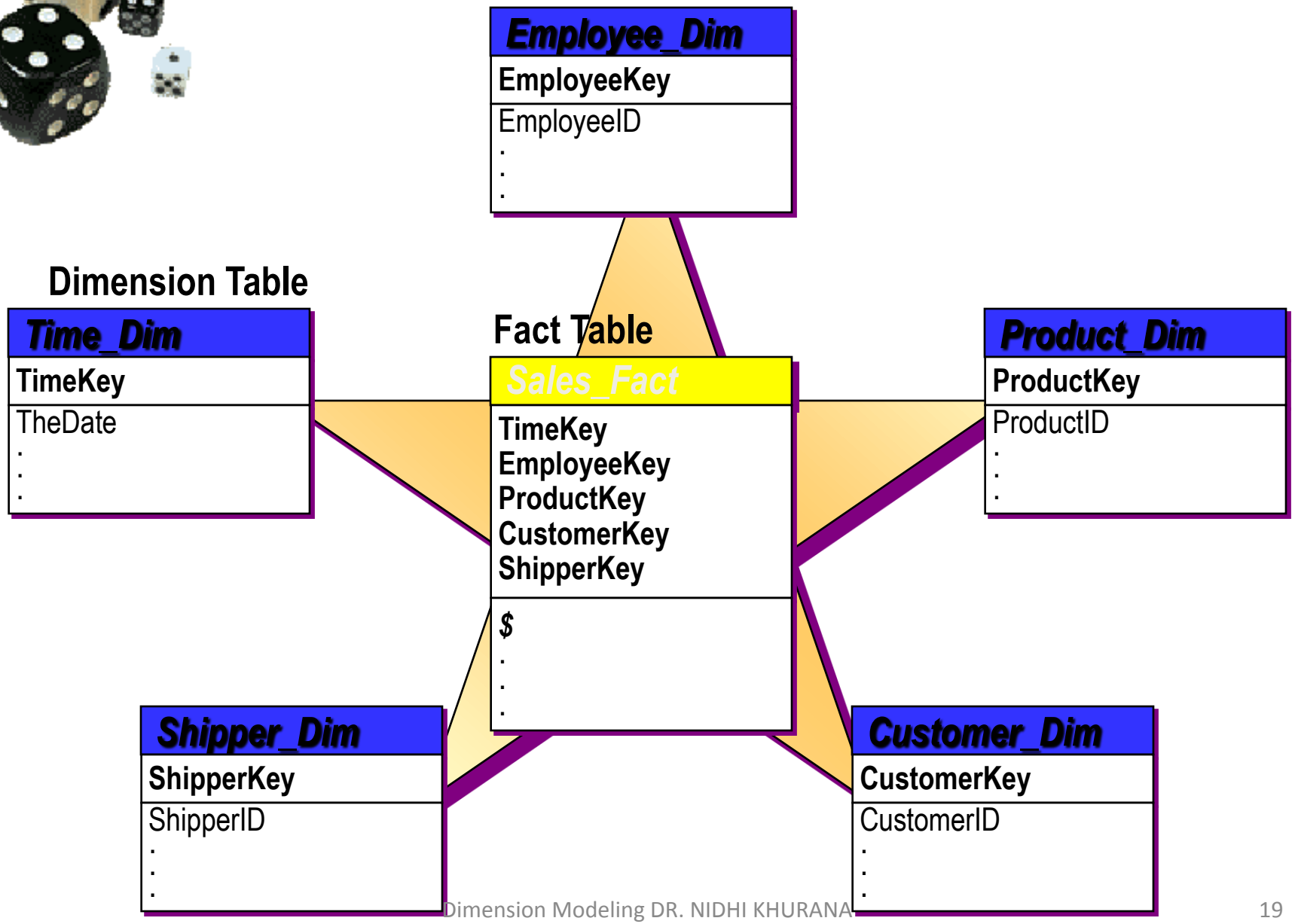




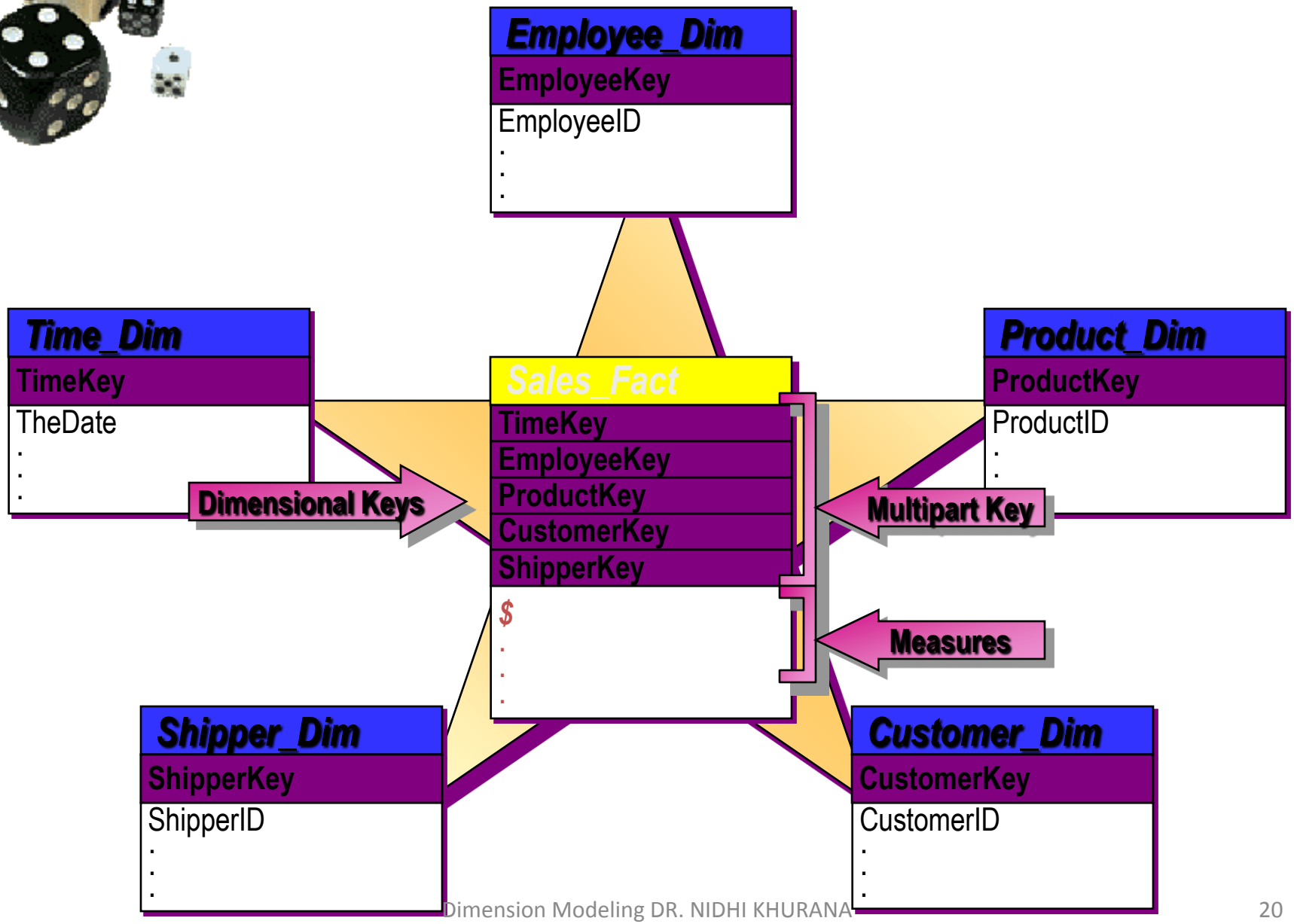
Conceptual Modeling of Data Warehouses

- Modeling data warehouses: dimensions & measures
  - Star schema: A fact table in the middle connected to a set of dimension tables
  - Snowflake schema: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  - Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

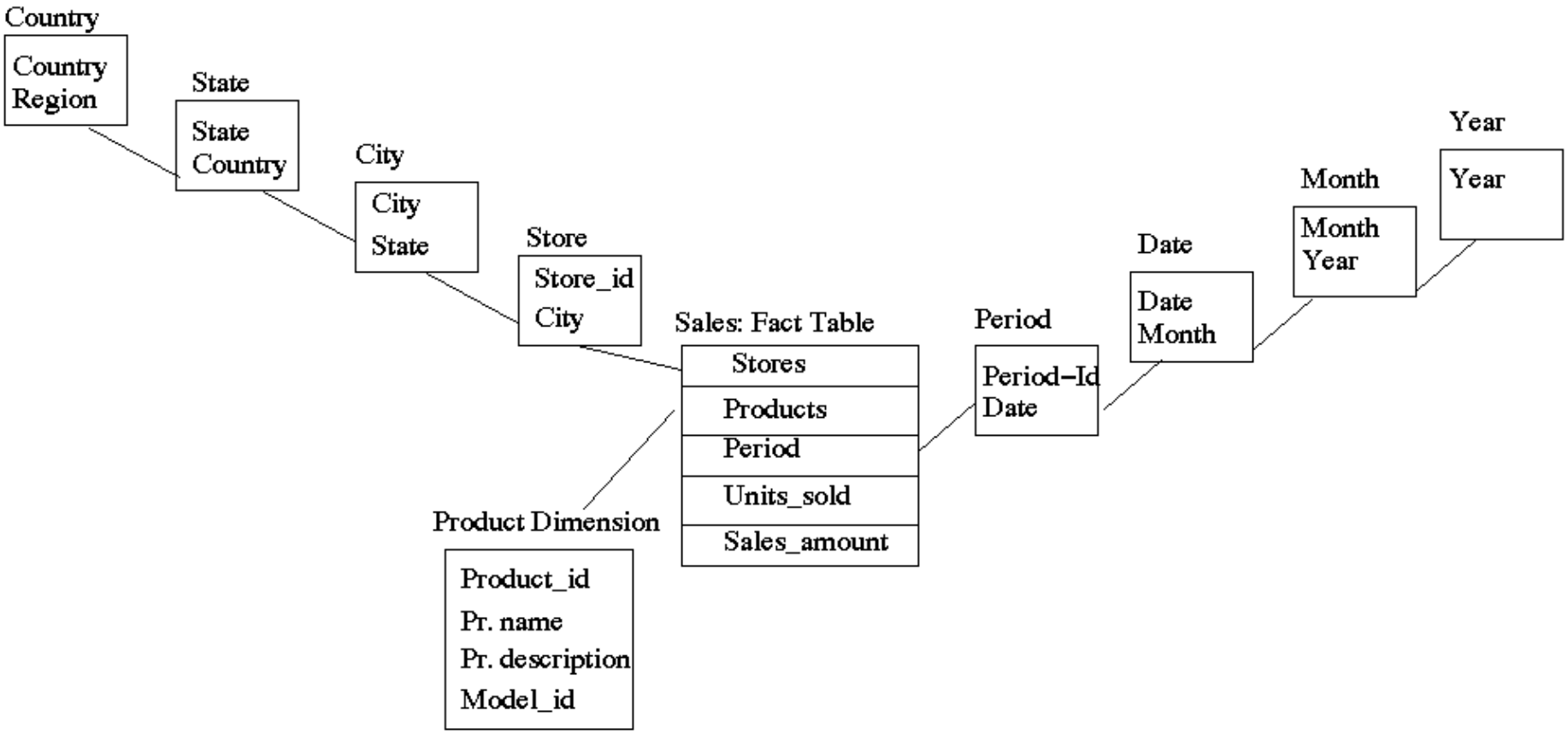
Using a Star Schema



Components of a Star Schema



Example of the Snowflake Schema



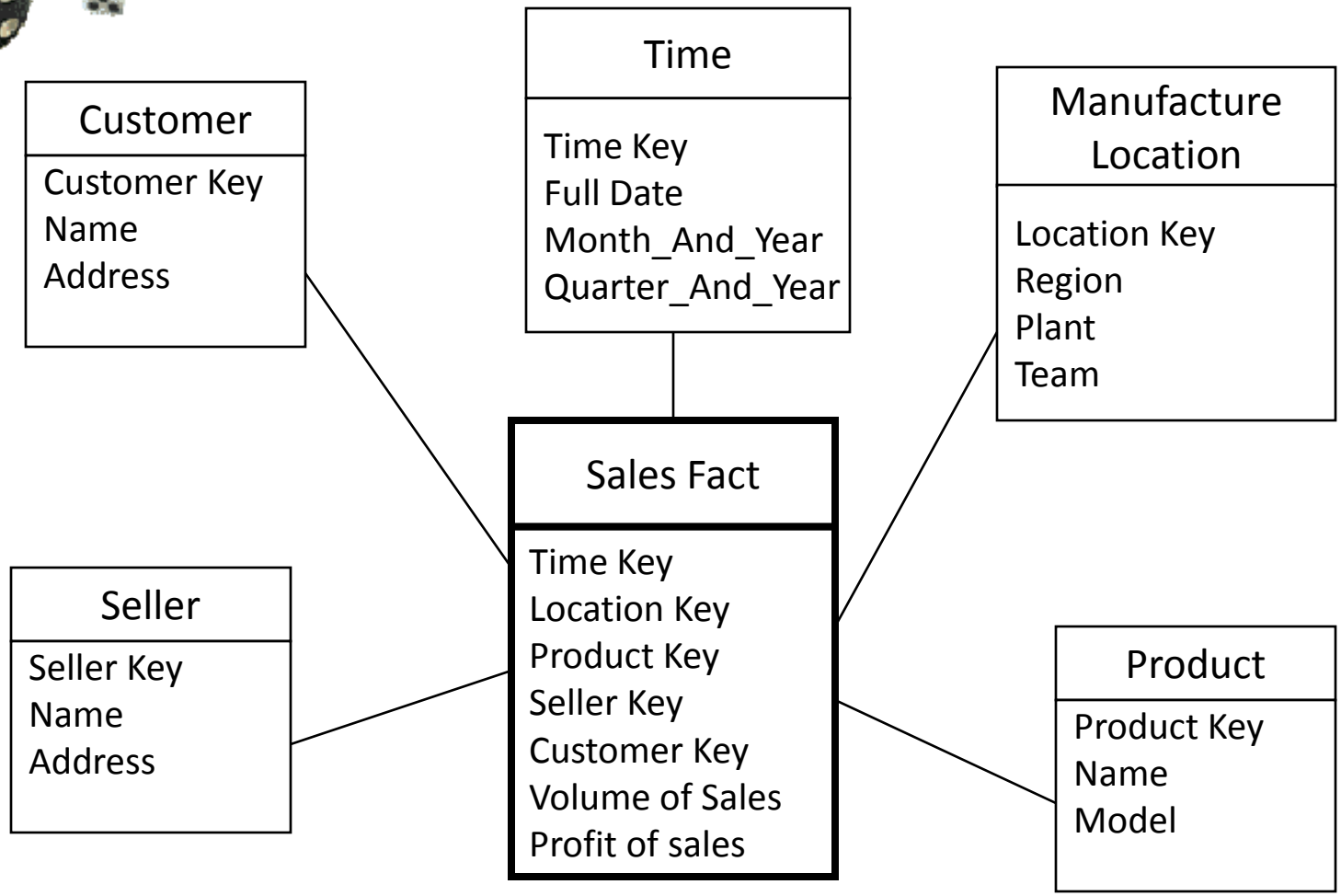
Star Schema

- **Star Schema** makes heavy use of renormalization to optimize for speed, at a potential cost of storage space.
- The star schema is a data-modeling technique used to map multidimensional decision support into a relational database.
- Star schemas yield an easily implemented model for multidimensional data analysis while still preserving the relational structure of the operational database.
- Four Components:
  - **Facts**
  - **Dimensions**
  - **Attributes**
  - **Attribute hierarchies**

Star Model

- It is a specific dimensional model
- Always, it connotes a dimensional model
- By this method, dimensions and facts are modeled as tables, dimension—fact relationships are implemented by ‘foreign keys’

Star Model (cont.)



Star Schema

- Facts
  - **Facts are numeric measurements (values) that represent a specific business aspect or activity.**
  - **The fact table contains facts that are linked through their dimensions.**
  - **Facts can be computed or derived at run-time (metrics).**
- Dimensions
  - **Dimensions are qualifying characteristics that provide additional perspectives to a given fact.**
  - **Dimensions are stored in dimension tables.**

Star Schema

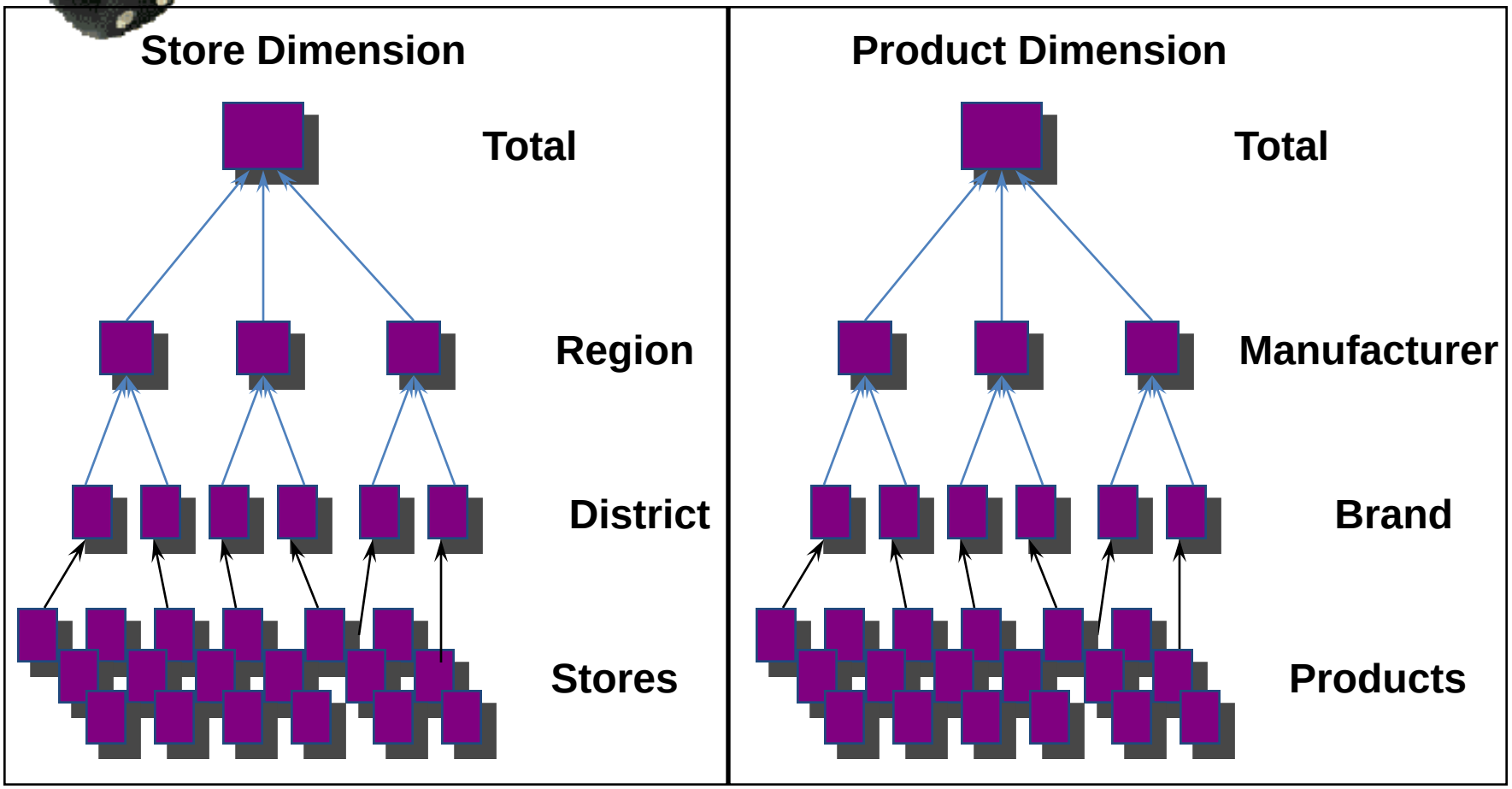
- Attributes
  - **Each dimension table contains attributes. Attributes are often used to search, filter, or classify facts.**
  - **Dimensions provide descriptive characteristics about the facts through their attributes.**

TABLE 13.9 POSSIBLE ATTRIBUTES FOR SALES DIMENSIONS

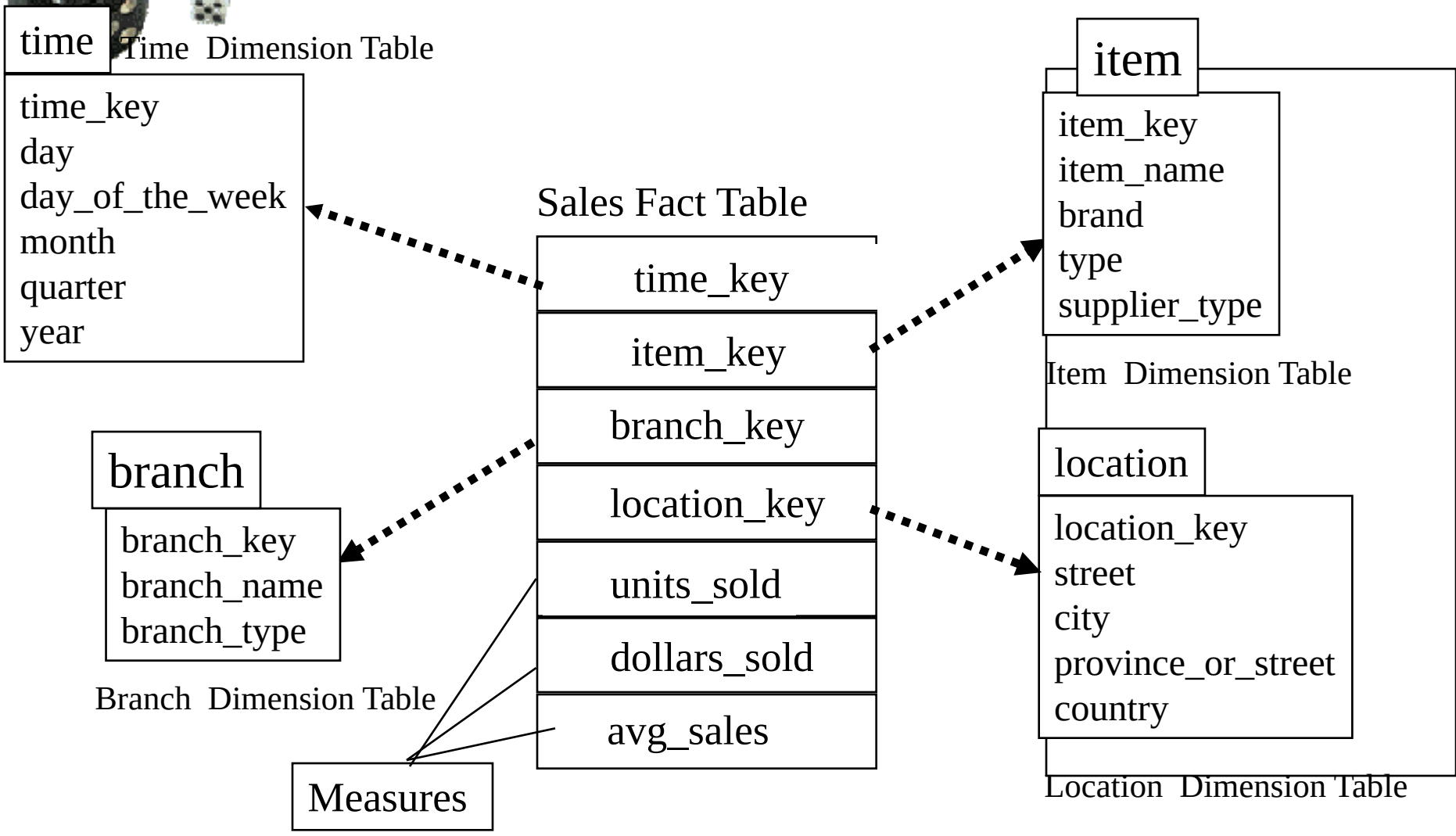
DIMENSION NAME	DESCRIPTION	POSSIBLE ATTRIBUTES
Location	Anything that provides a description of the location. Example: Nashville, Store 101, South Region, TN, etc.	Region, state, city, store, etc.
Product	Anything that provides a description of the product sold. For example, hair care product, shampoo, Natural Essence brand, 5.5 oz. bottle, blue liquid, etc.	Product type, product ID, brand, package, presentation, color, size, etc.
Time	Anything that provides a time frame for the sales fact. For example, the year of 1999, the month of July, the date 07/29/1999, the time 4:46 p.m., etc.	Year, quarter, month, week, day, time of day, etc.

Possible Attributes For Sales Dimensions

Dimension Hierarchies



Example of Star Schema



ER Model vs. Star Model (cont.)

- Advantage of star Model for a data warehouse
  - It has superior performance because it is designed and tuned for a specific set of parameters.
  - It works well with OLAP tools.
  - It can be used as a data mart
  - It can be used for the main warehouse for a limited scope of data.
  - It is flexible in user queries within the defined dimensions

ER Model vs. Star Model (cont.)

- Disadvantages of ER model
  - Slower performance because of large amounts of data
  - Large number of joins for a user's query
  - Not tuned for a limited set of parameters (for business analysis)
  - May need large space to store data, some of which may not used frequently
  - Not easy to use with OLAP tools.
  - Have many tables, therefore be hard for user to use.

ER Model vs. Star Model (cont.)

- **Disadvantages of Star model**
  - To add new data element to support a new requirement may cause heavy data redundancy.
  - If the dimensions are independent of each other, the total record number is bigger than that in a ER Model
  - It is tuned for a particular set of queries by dimensions. If new queries view data out of the pattern, they will not perform well.
  - When the view of the data changes, a new extract must be created to populate the new fact table.
  - In a complex star schema, there can be hundreds of fact and dimension tables

STAR Schema
Inside a Dimension Table

- Dimension table key (primary key)
- Table is wide (many columns or attributes)
- Textual attributes (gender, race, regions)
- Attributes not directly related
- Not normalized (optimized for queries)
- Drilling down/Rolling up (hierarchy of attributes)
- Multiple hierarchies
- Fewer records (good idea to cache a dimension)

STAR Schema
Inside the Fact Table

- Concatenated key (combination of PK's)
- Data grain (level of detail for a metric – every order details are stored)
- Fully additive measures ("*extended_costs*")
- Semi-additive measures (calculated members)
- Table deep, not wide (large record numbers)
- Sparse data (null values, null measures = gaps)
- Degenerate dimensions - Not a metric or a measure ("*order_number*")
- Factless fact tables – no measures associated with these

STAR Schema Keys

- Primary keys: Each row in dimention table is identified by a unique value of an attribute designated as primary key of the dimension.
- Foreign keys: The foreign key identifies a column or a set of columns in one (referencing) table that refers to a column or set of columns in another (referenced) table.
- Surrogate keys : a surrogate represents an *entity* in the outside world. The surrogate is internally generated by the system but is nevertheless visible by the user or application

A *surrogate* may also be called a ::-- surrogate key, entity identifier, system-generated key, database sequence number, synthetic key, technical key, or arbitrary unique identifier. Eg Oracle SEQUENCE

Star Schema - Summary

- A single fact table and a single table for each dimension
- Every fact points to one tuple in each of the dimensions and has additional attributes
- Does not capture hierarchies directly
- Generated keys are used for performance and maintenance reasons
- Fact constellation: Multiple Fact tables that share many dimension tables
  - Example: Projected expense and the actual expense may share dimensional tables

Advantages of Star Schema

- Easy for users to understand
- Optimizes Navigation
  - Uses the fact table with dimension tables
- Most suitable for Query processing
  - Mostly just joins
- STARJoin
  - High-speed, single-pass, parallelizable multiable join in a single operation
- STARIndex
  - Specialized index on one or more foreign keys