Lecture 05

Data Warehouse Architecture

Conceptual Model

Summary – last week

- Last week:
  - DW Architecture
    - Storage Architecture
    - Tier Architecture

- This week:
  - Distributed DW
  - DW Data Modeling
    - Conceptual Model

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Distributed DW

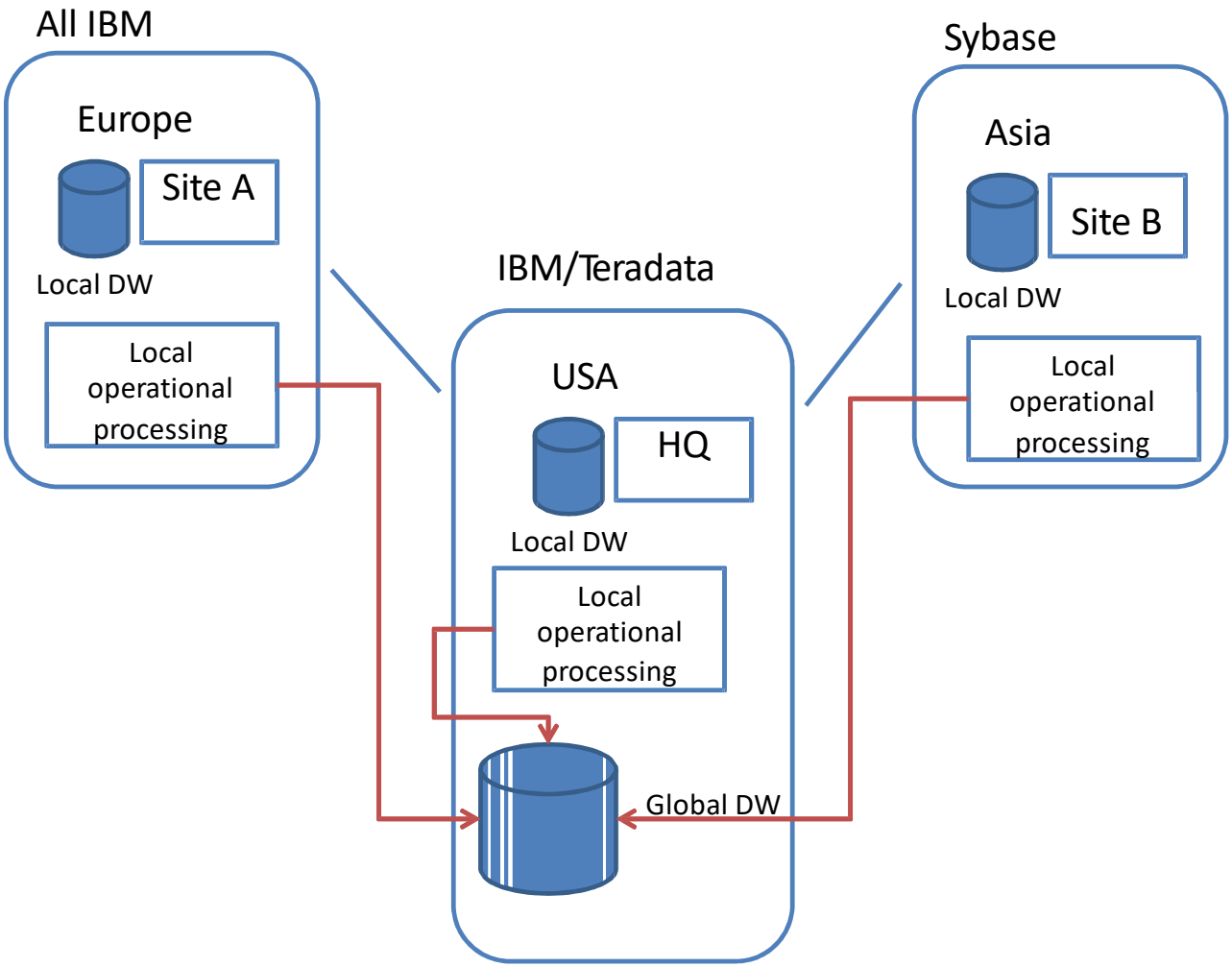
- In most cases the economics and technology greatly favor a single **centralized DW**
- But in some cases, distributed DW make sense
- Types of **distributed DW**
  - Geographically distributed
    - Local DW instances of global DW
  - Technologically distributed DW
    - Logically one DW but data is stored on multiple stores

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Distributed DW (cont'd.)

- Geographically distributed
  - In the case of corporations spread around the world
    - Information is needed both locally and globally
  - A distributed DW makes sense
    - When much processing occurs at the local level
    - Even though local branches report to the same balance sheet, the local organizations are their own companies

Distributed DW (cont'd.)



Distributed DW (cont'd.)

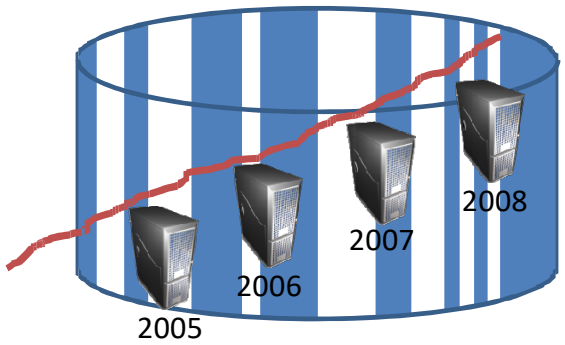
- **Technologically** distributed DW
  - Placing the DW on the distributed technology of a vendor
  - Advantages
    - The entry cost is cheap – large centralized hardware is expensive
    - No theoretical limit to how much data can be placed in the DW – we can add new servers to the network

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Distributed DW (cont'd.)

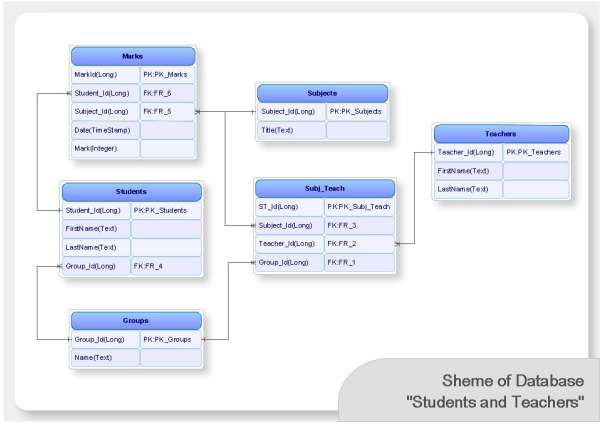
– As the DW starts to expand network data **communication** starts playing an important role

- Example: Let's simplify and consider we have 4 nodes holding each data regarding the last 4 years
- Now let's consider we have a query which needs to access the data from the last 4 years: such a query arises the issue of transporting large amount of data between processors



Data Modeling

- Data Modeling / DB Design - Basics
  - Is the process of creating a data model by analyzing the requirements needed to support the business processes of an organization
    - It is sometimes called database modeling/design because a data model is eventually implemented in a database



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

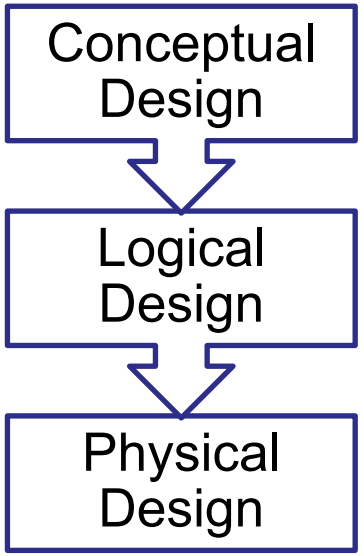
Data Modeling (cont'd.)

- **Data models**
  - Provide the **definition** and **format** of data
  - Graphical representations of the data within a specific area of interest
    - Enterprise Data Model: represents the integrated data requirements of a complete business organization
    - Subject Area Data Model: Represents the data requirements of a single business area or application

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

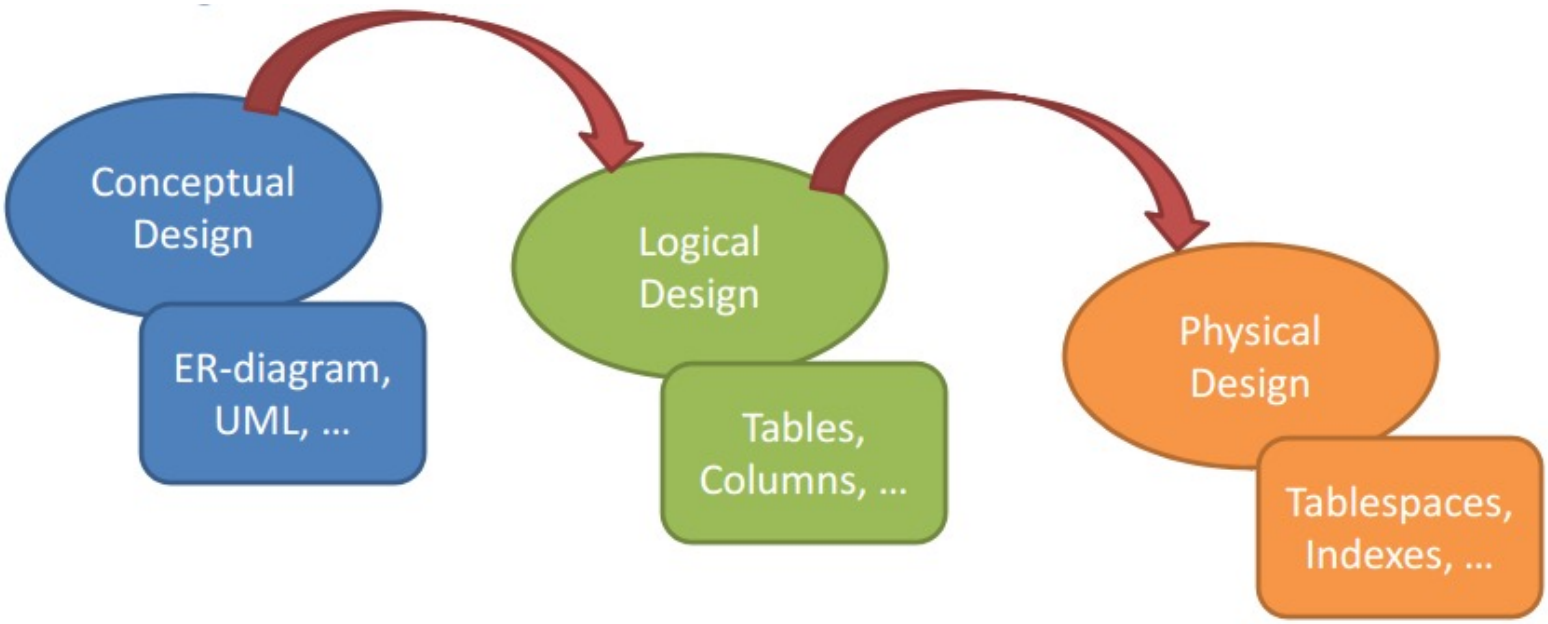
Data Modeling (cont'd.)

- Conceptual Design
  - Transforms data requirements to conceptual model
  - Conceptual model describes data entities, relationships, constraints, etc. on high-level
    - Does not contain any implementation details
    - Independent of used software and hardware
- Logical Design (next lecture)
  - Maps the conceptual data model to the logical data model used by the DBMS
    - e.g. relational model, dimensional model, …
    - Technology independent conceptual model is adapted to the used DBMS software
- Physical Design (2nd next lecture)
  - Creates internal structures needed to efficiently store/manage data
    - Table spaces, indexes, access paths, …
    - Depends on used hardware and DBMS software



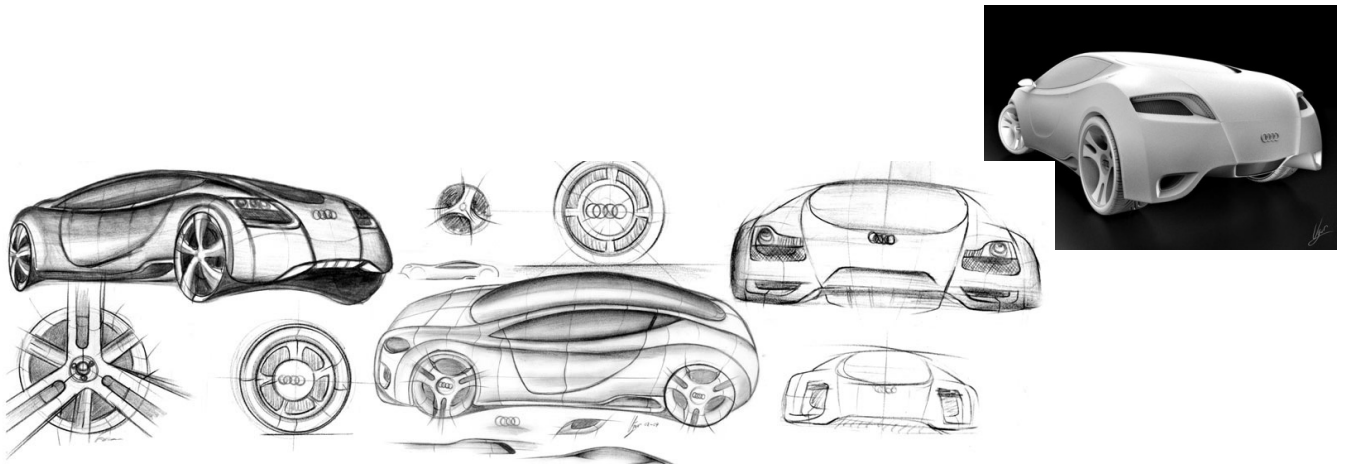
Data Modeling (cont'd.)

- Going from one phase to the next:
  - The phase must be complete
    - The result serves as input for the next phase
  - Often automatic transition is possible with additional designer feedback

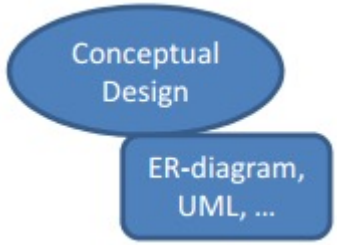


Conceptual Model

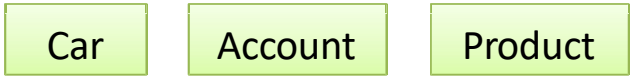
- Highest conceptual grouping of ideas
  - Data tends to naturally cluster with data from the same or similar categories relevant to the organization
- The major relationships between subjects have been defined
  - Least amount of detail



Conceptual Model (cont'd.)

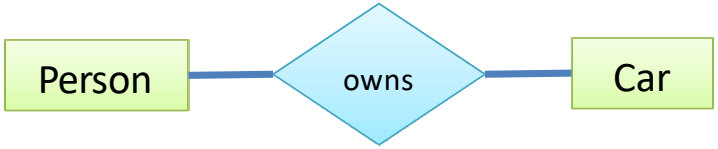


- Conceptual design
  - Entity-Relationship (ER) Modeling
    - Entities - "things" in the real world
      - E.g. Car, Account, Product



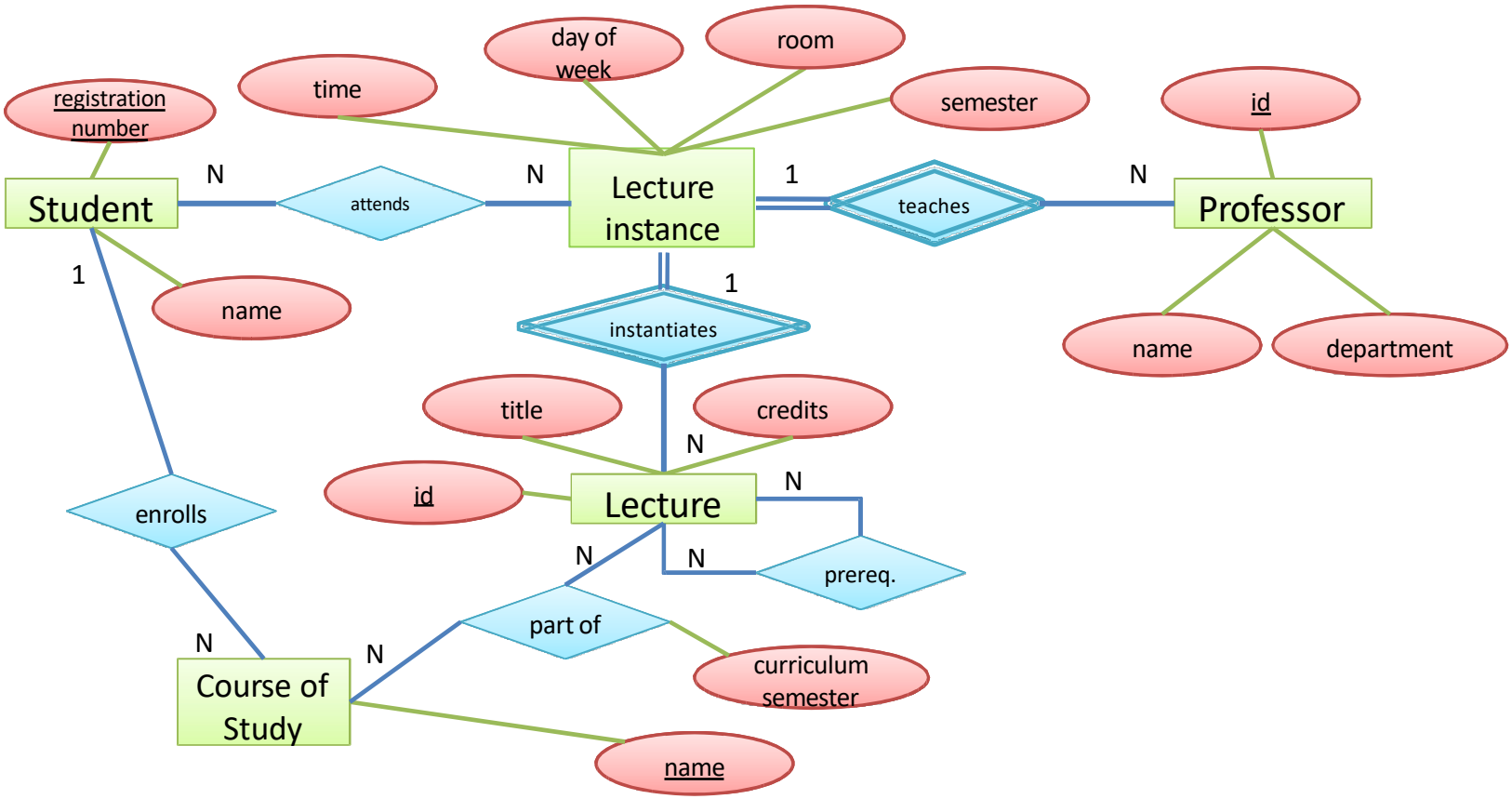
    - Attributes – property of an entity, entity type, or relationship type

      - E.g. color of a car, balance of an account, price of a product
    - Relationships – between entities there can be relationships, which also can have attributes
      - E.g. Person owns Car



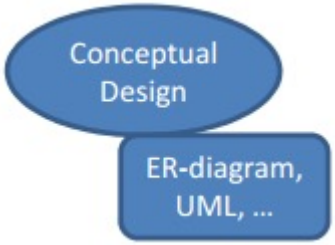
Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Conceptual Model (cont'd.)



Conceptual Model (cont'd.)

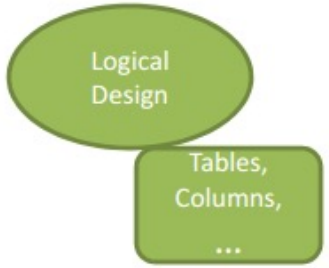
- Conceptual design in usually done using the Unified Modeling Language (UML)
  - Class Diagram, Component Diagram, Object Diagram, Package Diagram…
  - For Data Modeling only Class Diagrams are used
    - Entity type becomes class
    - Relationships become associations
    - There are special types of associations like: aggregation, composition, or generalization



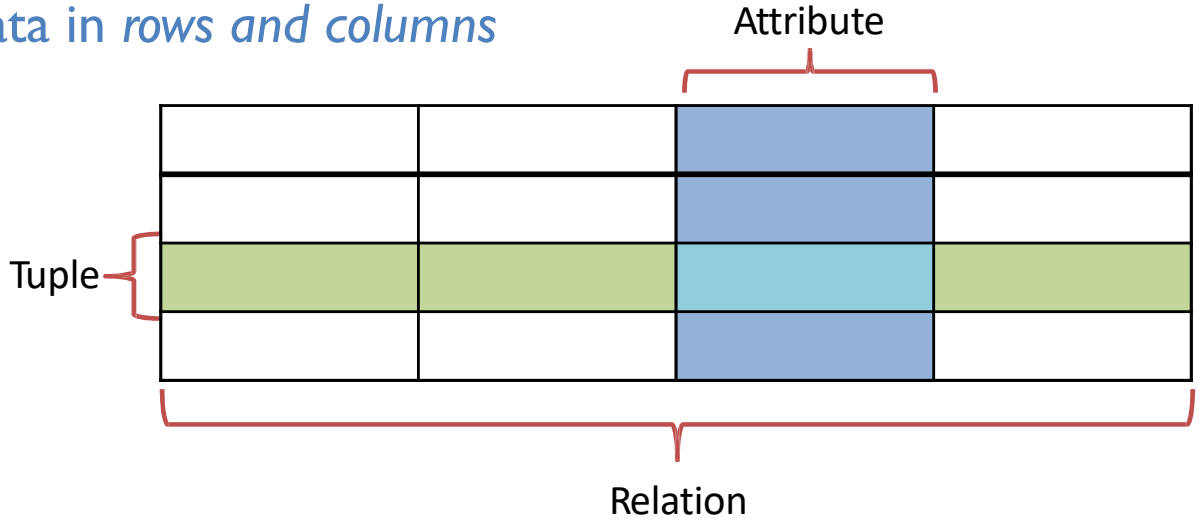
CLASS NAME
attribute 1 : domain … attribute n : domain
operation 1 … operation m

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Logical Model

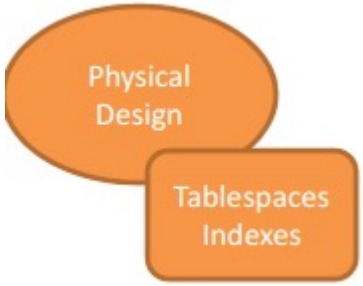


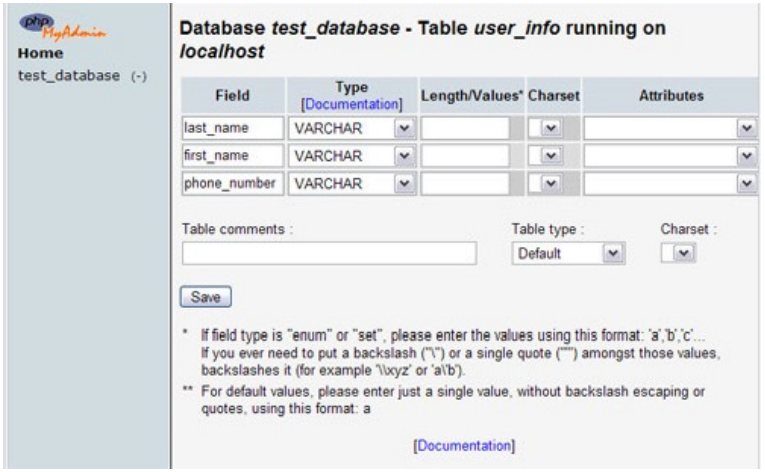
- Logical design arranges data into a logical structure
  - Which can be mapped into the storage objects supported by DBMS
    - In the case of RDB, the storage objects are *tables* which store data in *rows and columns*



Physical Model

- Physical design specifies the physical configuration of the database on the storage media
  - Detailed specification of: data elements, data types, indexing options, and other parameters residing in the DBMS data dictionary





Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Data Model in DW

- Managing Complex Data Relationships
  - Helps keep track of the complex environment that is a DW
    - Many complex relationships exist, with the ability to change over time
  - Transformations and integration from various systems of record need to be worked out and maintained
  - Provides the means of supplying users with a **roadmap** through the data and relationships

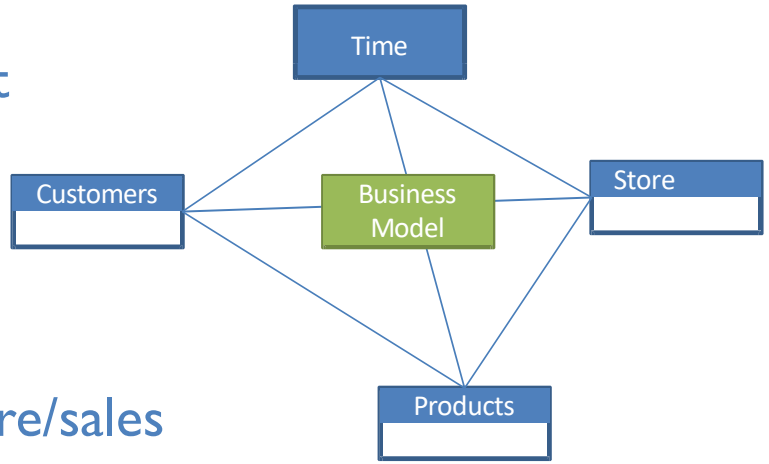
Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Conceptual Model in DW

- Modeling business queries
  - Goal
    - Define the purpose, and decide on the subject(s) for the data warehouse

      Identify questions of interest
  - Subject
    - Who bought the products?

      (customers structure)
    - Who sold the product? (store/sales organization structure)
    - What was sold? (product structure)
    - When was it sold? (time structure)



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Conceptual Model in DW (cont'd.)

- For **Conceptual design** in DW conventional techniques like E/R or UML are not appropriate
  - Lack of necessary semantics for modeling the multidimensional data model
  - E/R are constituted to
    - Remove redundancy in the data model
    - Facilitate retrieval of individual records
  - Therefore optimize OLTP
  - In the case of DW, however redundancy and MaterializedViews help speed up Analytical queries

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Conceptual Model in DW (cont'd.)

- Components
  - **Facts:** a fact is a focus of interest for decision-making, e.g., sales, shipments..
  - **Measures:** attributes that describe facts from different points of view, e.g., each sale is measured by its revenue
  - **Dimensions:** discrete attributes which determine the granularity adopted to represent facts, e.g., product, store, date
  - **Hierarchies:** are made up of dimension attributes
    - Determine how facts may be aggregated and selected, e.g., day – month – quarter - year

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Conceptual Model in DW (cont'd.)

- Conceptual design models for DW
  - Multidimensional Entity Relationship (ME/R) Model
  - Multidimensional UML (mUML)
  - Other methods e.g., Dimension Fact Model, Totok approach, etc.

Multidimensional ER Model

- ME/R Model
  - Its purpose is to create an **intuitive representation** of the multidimensional data that is optimized for high-performance access
  - It represents a specialization and evolution of the E/R to allow specification of **multidimensional semantics**

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Multidimensional ER Model (cont'd.)

- ME/R notation was influenced by the following considerations
  - Specialization of the E/R model
    - All new elements of the ME/R have to be specializations of the E/R elements
    - In this way the flexibility and power of expression of the E/R models are not reduced
  - Minimal expansion of the E/R model
    - Easy to understand/learn/use: the number of **additional elements** should be small
  - Representation of the multidimensional semantics
    - Although being minimal, it should be powerful enough to be able to represent multidimensional semantics

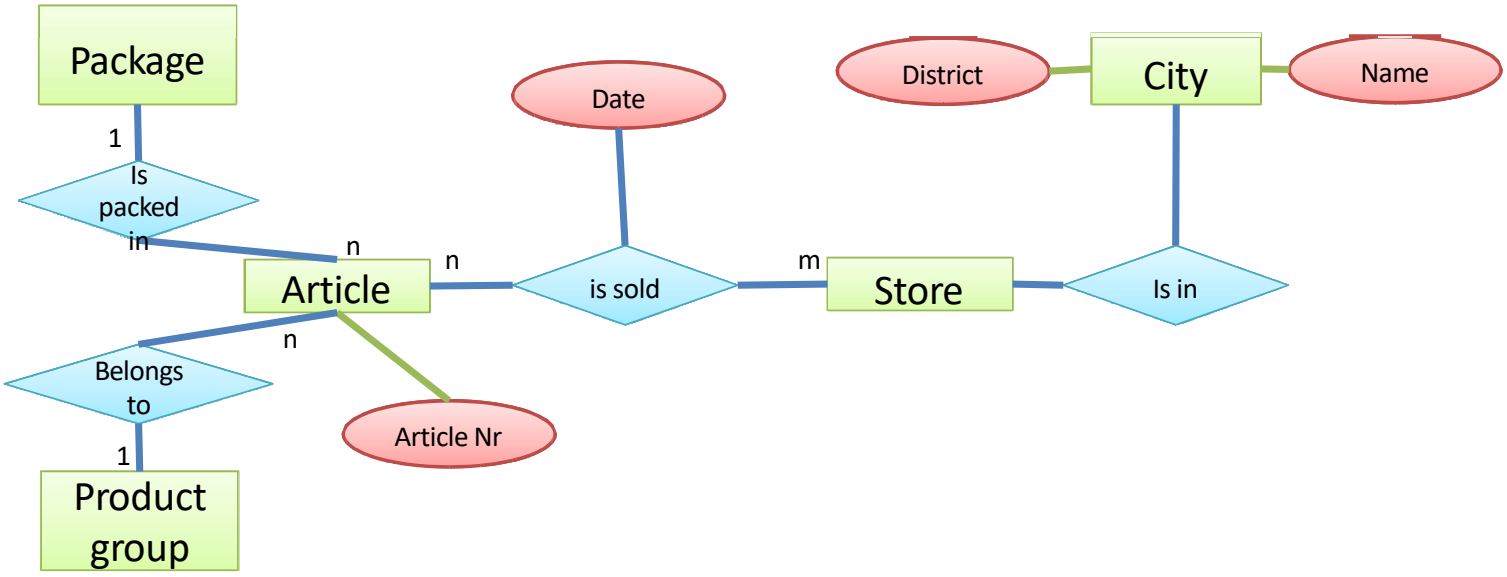
Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Multidimensional ER Model (cont'd.)

- There are 3 main ME/R constructs
  - The **fact node**
  - The **level node**
  - A special **binary classification edge**

Multidimensional ER Model (cont'd.)

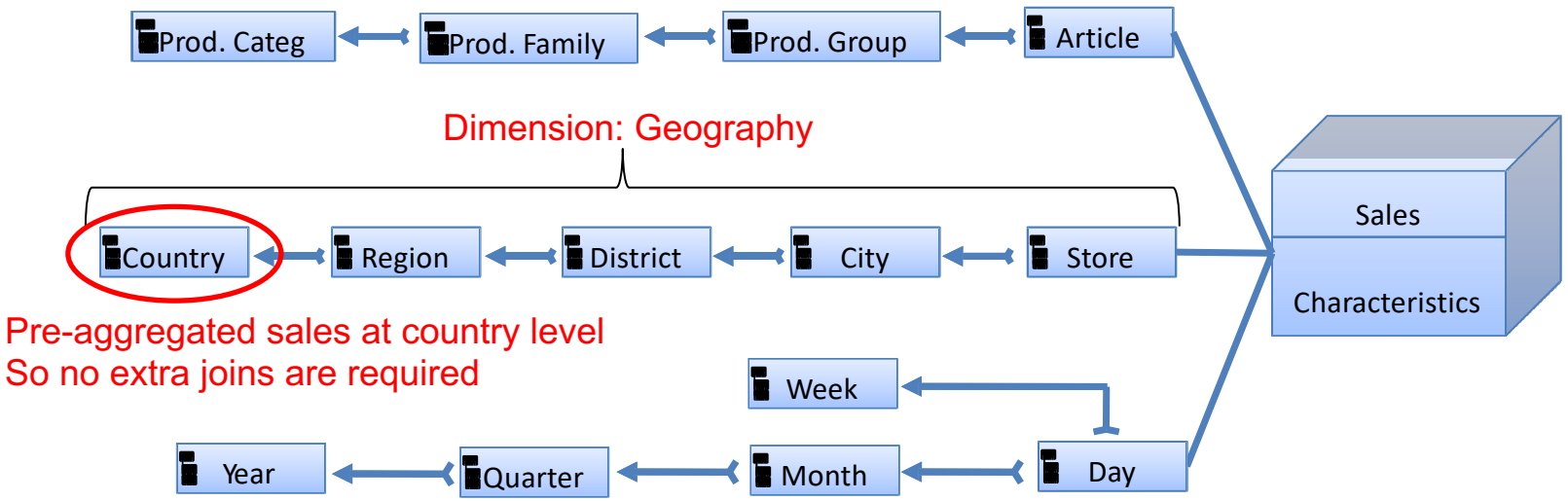
- Lets consider a **store scenario** designed in E/R
  - Entities bear little semantics
  - E/R doesn't support classification and aggregation levels



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

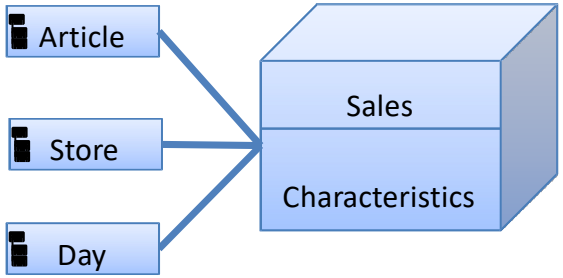
Multidimensional ER Model (cont'd.)

- **ME/R** notation:

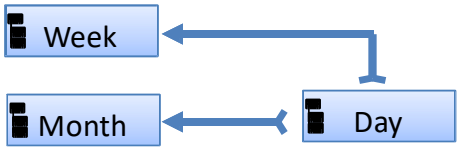


Multidimensional ER Model (cont'd.)

- ME/R notation:
  - Sales was elected as **fact node**
  - The dimensions are **product, geographical area and time**
  - The dimensions are represented through the so called **Basic Classification Level**



  - Alternative paths in the classification level are also possible



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Summary

- Tier Architecture
- Distributed DW
- DW Data Modeling

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

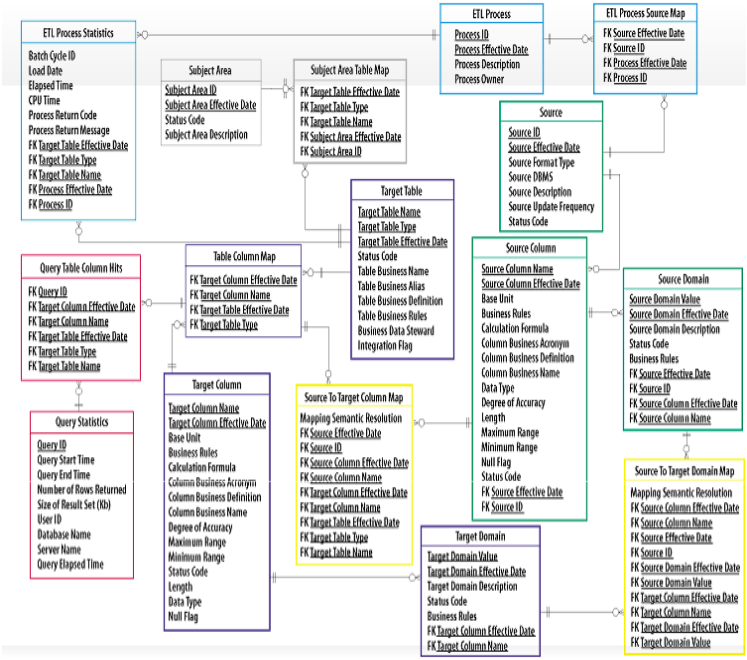
Summary (cont'd.)

- DW are usually distributed geographically and technologically
- Data Modeling – Conceptual Modeling
  - In conceptual modeling for DW, conventional techniques like E/R or UML are not appropriate
  - Appropriate methods are:
    - Multidimensional Entity Relationship (ME/R) Model
    - Multidimensional UML (mUML)

Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides

Next Lecture

- Data Modeling (continued)
  - Logical model
    - Cubes, Dimensions, Hierarchies, Classification Levels



Acknowledgment - Thanks to Wolf-Tilo Balke and Silviu Homoceanu - TU Braunschweig for the slides