Institut für Informationssysteme
Technische Universität Braunschweig

Data Warehousing & Mining Techniques

Wolf-Tilo Balke
Muhammad Usman
Institut für Informationssysteme
Technische Universität Braunschweig
http://www.ifis.cs.tu-bs.de

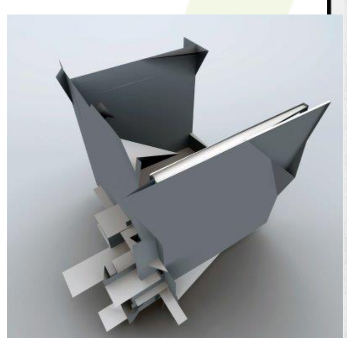
2. Summary

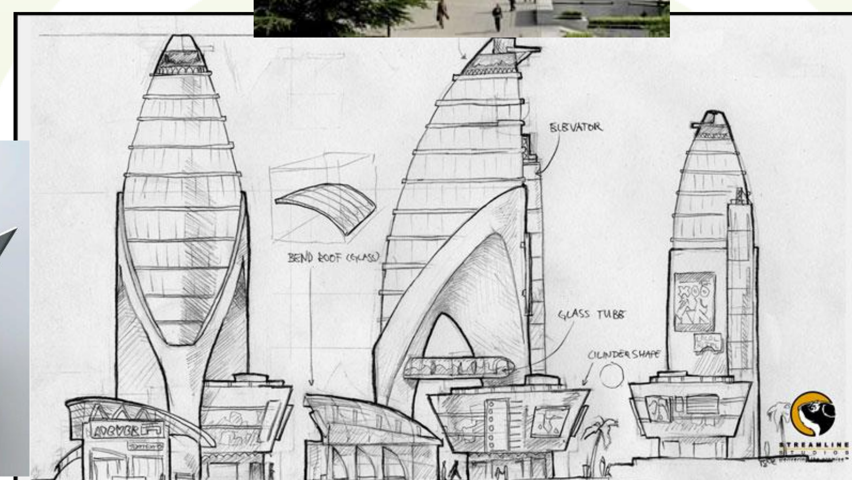
- Last week:
  - What is a Data Warehouse
  - Applications and users
  - Lifecycle and phases

- Architecture and Data model
  - This lecture

2. Architecture

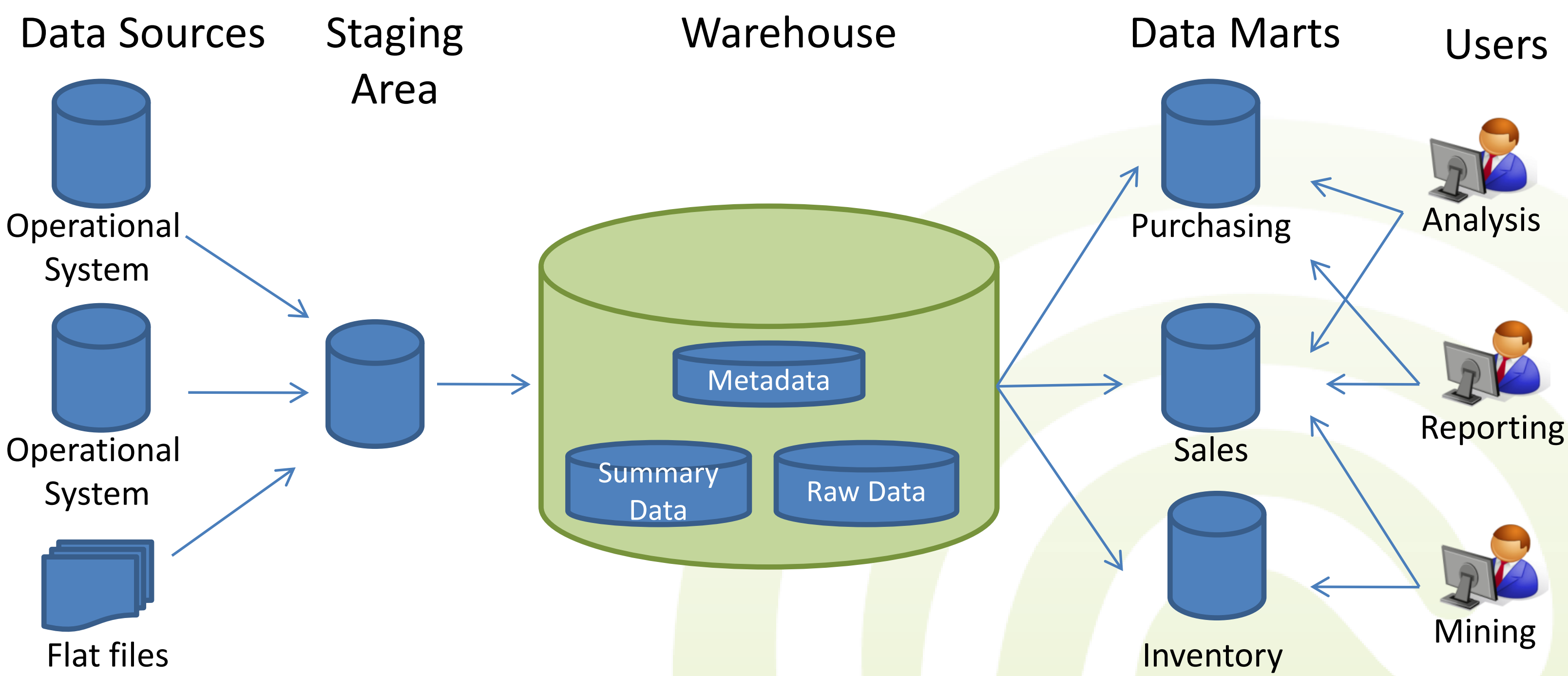
2. Architecture
   - 2.1 Basic Architecture
   - 2.2 Architectures in Practice
   - 2.3 DW Storage Structures
   - 2.4 DW Data Modeling





2.1 Basic Architecture

- Full DW architecture:



2.1 Operational Data Store

- Databases that serve daily operations of the enterprise e.g. production, sales (cash register), accounting
  - Usually rely on relational database technology (see RDB1)
  - Optimized for small queries like: simple product lookups, inserts, updates and deletes

2.1 Staging Area

- Contains a separate copy of the data which will be loaded from ODS to the DW
  - In the staging area the copied data is prepared (integrated, cleaned, etc.)
- Customers aren't invited to visit the kitchen…
  - Similar to a restaurant's kitchen, the data staging area should be accessible only to **skilled DW professionals,** neither ODS admins. nor analysts

2.1 Data Warehouse

- The DW persistently stores
  - Cleaned raw data
  - Derived (aggregated) data
    - Usual aggregates of the raw data e.g. quarter sales per regions
    - Performance reasons: avoid computing (the same) aggregates times and again at query time
  - Metadata
    - Describe the meaning, properties and origins of the data in the DW (e.g. provenance & lineage)

2.1 Presentation Area

- The presentation area comprises
  - **Data Marts** where data is organized according to the focus of one department
    - Similar to DB views, but usually stored (materialized view)
  - Reporting as well as **analytical processing** tools
- This area **is** the Warehouse as far as the business community is concerned

2.1 Building a complete DW

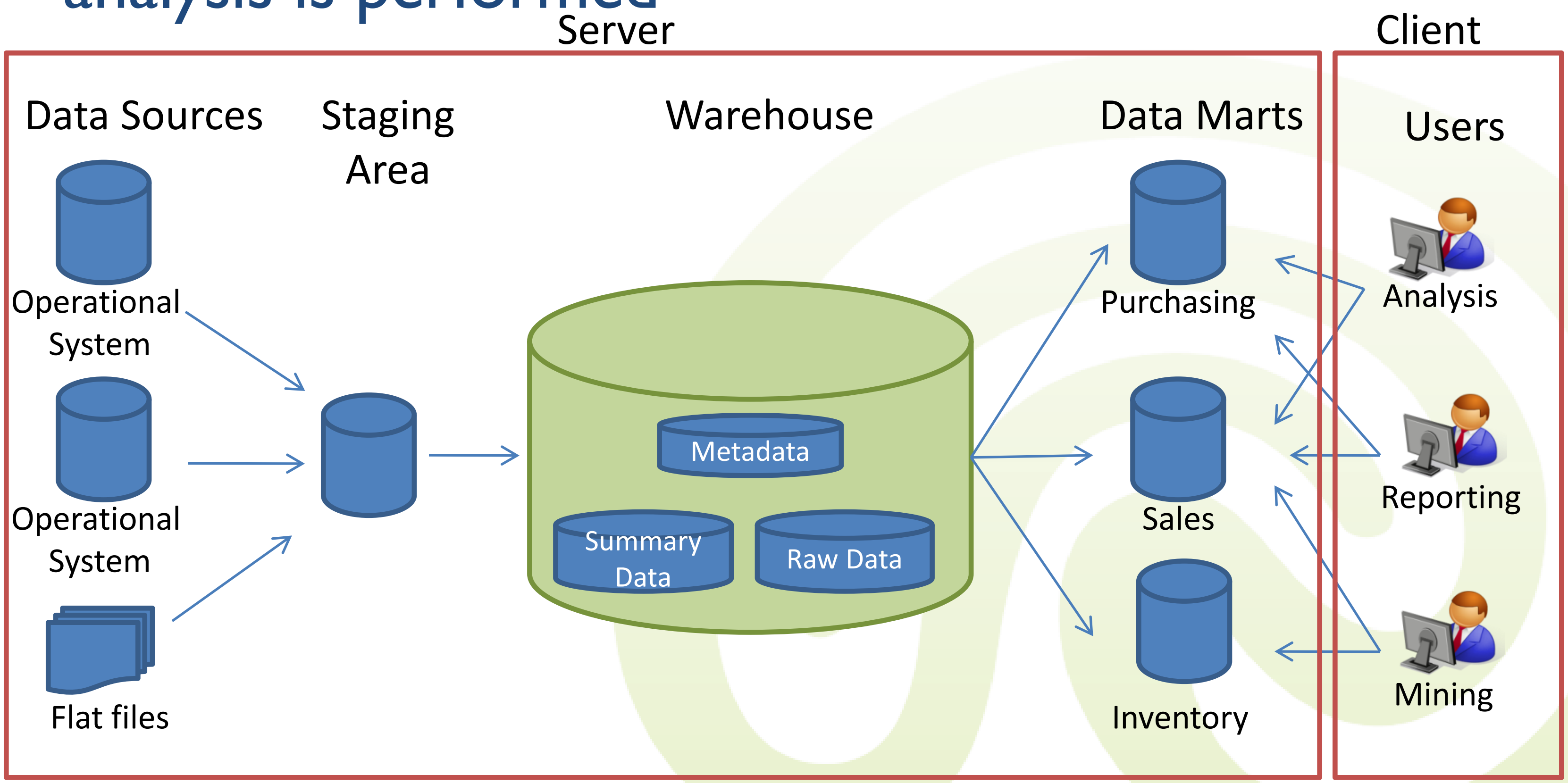
- Hardware and data flow architecture
  – Complete data flow from ODS up to the presentation
  – Most important step is the Extract – Transform – Load (ETL) process
- Storage structure
  – The used model for storing data in the DW
- Data modeling
  – Conceptual, logical and physical models for the DW storage structure

2.2 Architectures in Practice

- Popular DW architectures in practice
  - Vertical tiers
    - Generic Two-Tier Architecture
    - Three-Tier Architecture

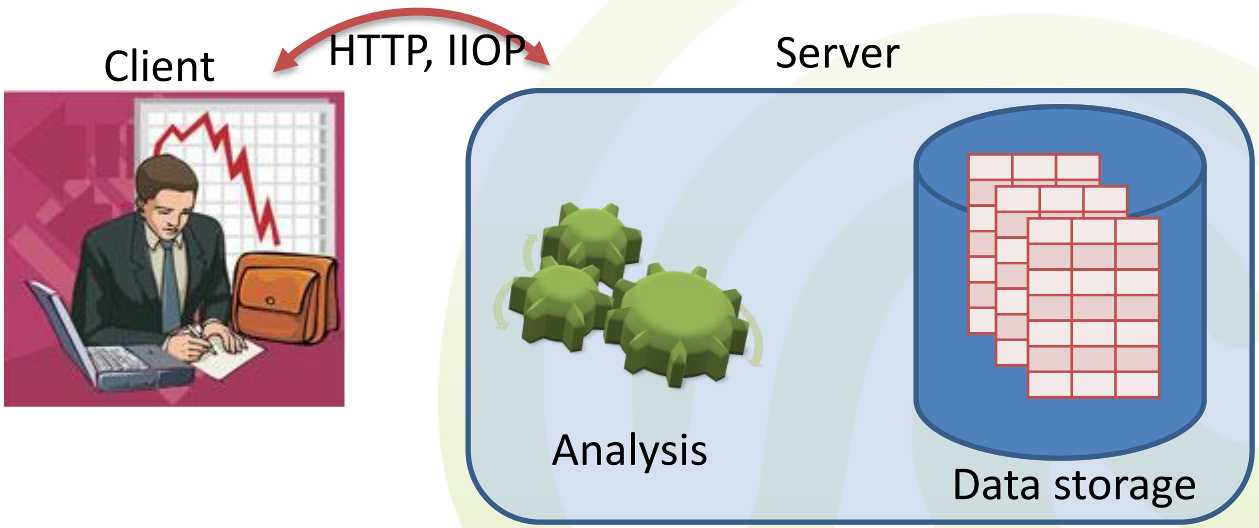
2.2 Two-Tier Architecture

- Generic client-server architecture
  - Fat or thin client depending on where the data analysis is performed



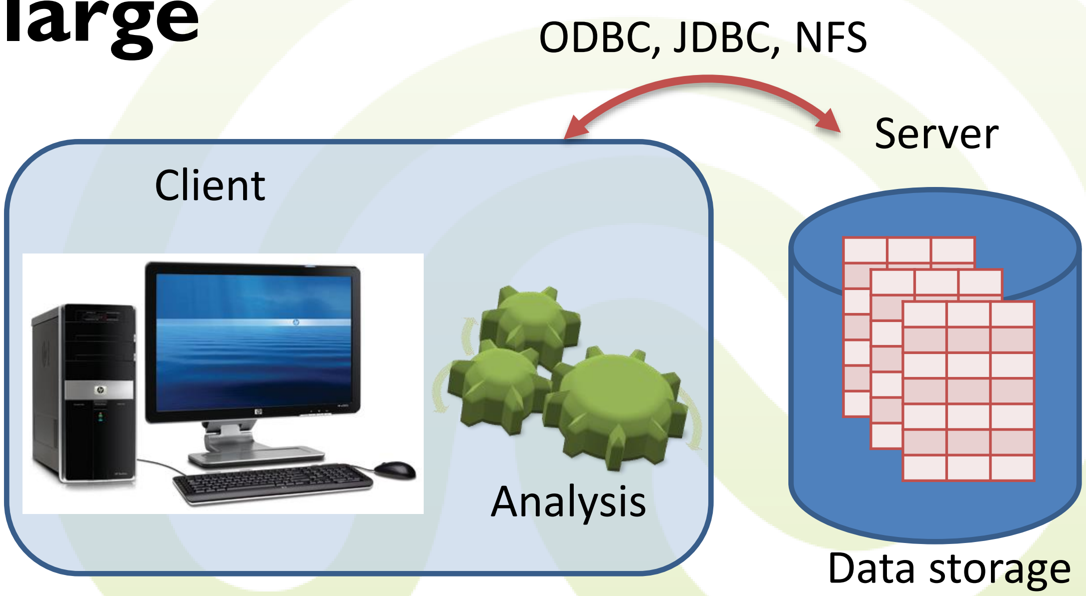
2.2 Thin Client

- Operations are executed on the server
- The client is just used to display the results
- This architecture fits well for Internet DW access



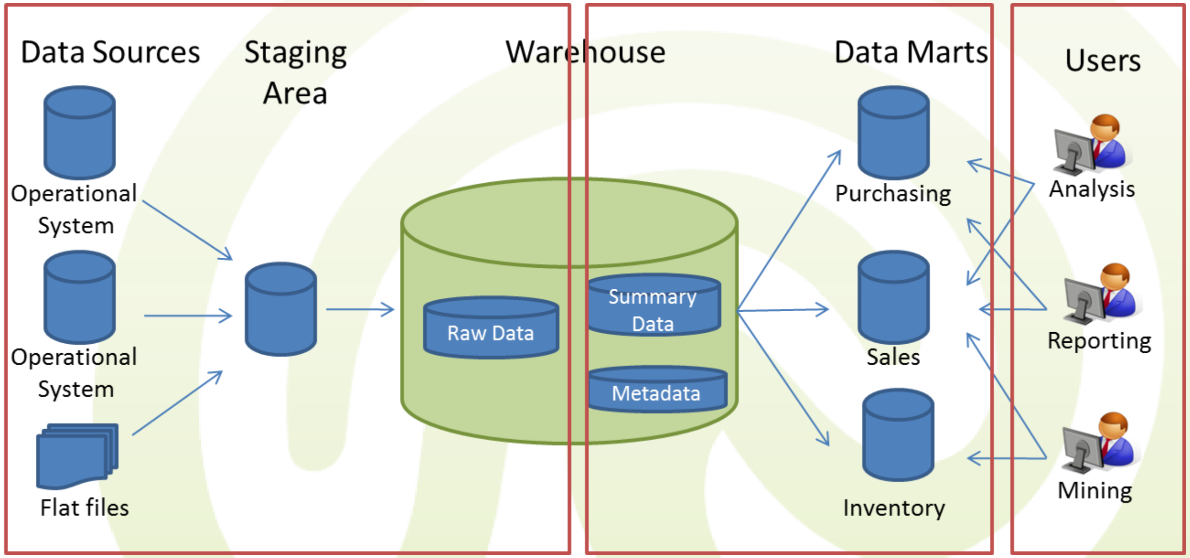
2.2 Fat Client

- The server just delivers the data e.g. the corresponding data mart
- Operations are executed on the client
- Communication between client and server must be able to sustain **large data transfers**



2.2 Three-Tier Architecture

- Tier 1: raw and detailed data intended to be the single source for all decision support
- Tier 2: derived data that had been aggregated for DSS support
- Tier 3: reporting and analysis



2.2 Other Architectures

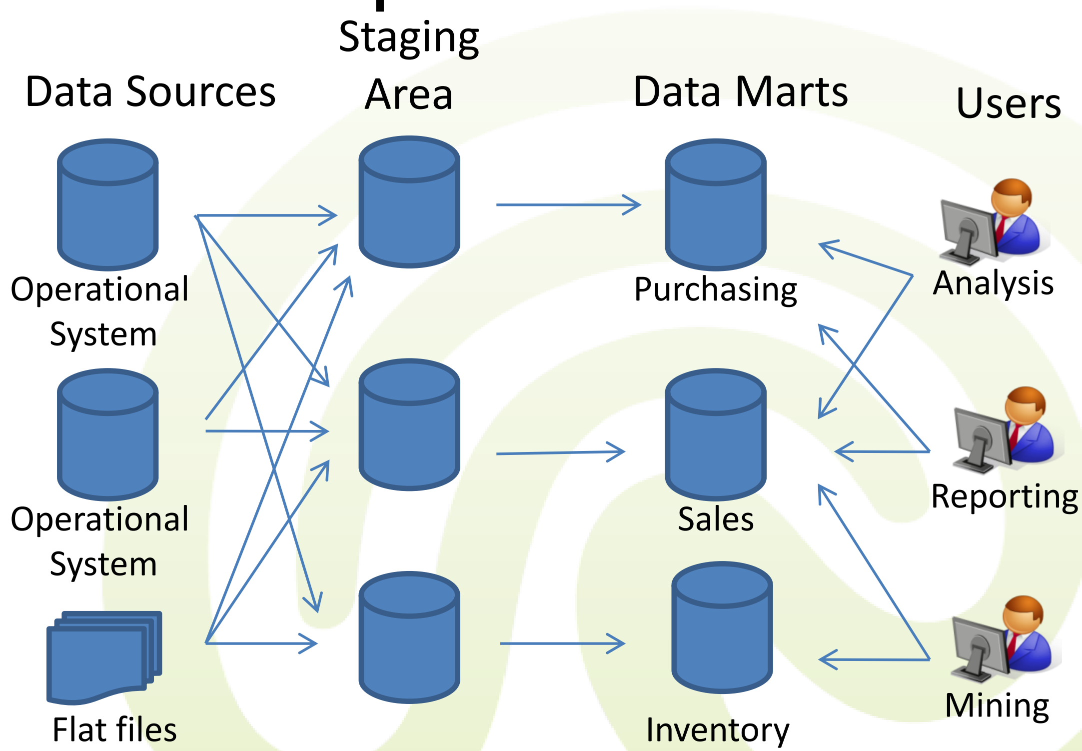
- N-Tier Architecture
  - Higher tier architecture is also possible but the complexity grows with the number of tier-interfaces
- Web-based Architectures
  - Advantage: Usage of existing software, reduction of costs, platform independence
  - Disadvantage: Security overhead e.g. data encryption, user access and identification

2.2 Architectures in Practice

- Popular DW architectures in Practice
  - Horizontal tiers
    - Independent Data Mart
    - Dependent Data Mart
    - Logical Data Mart

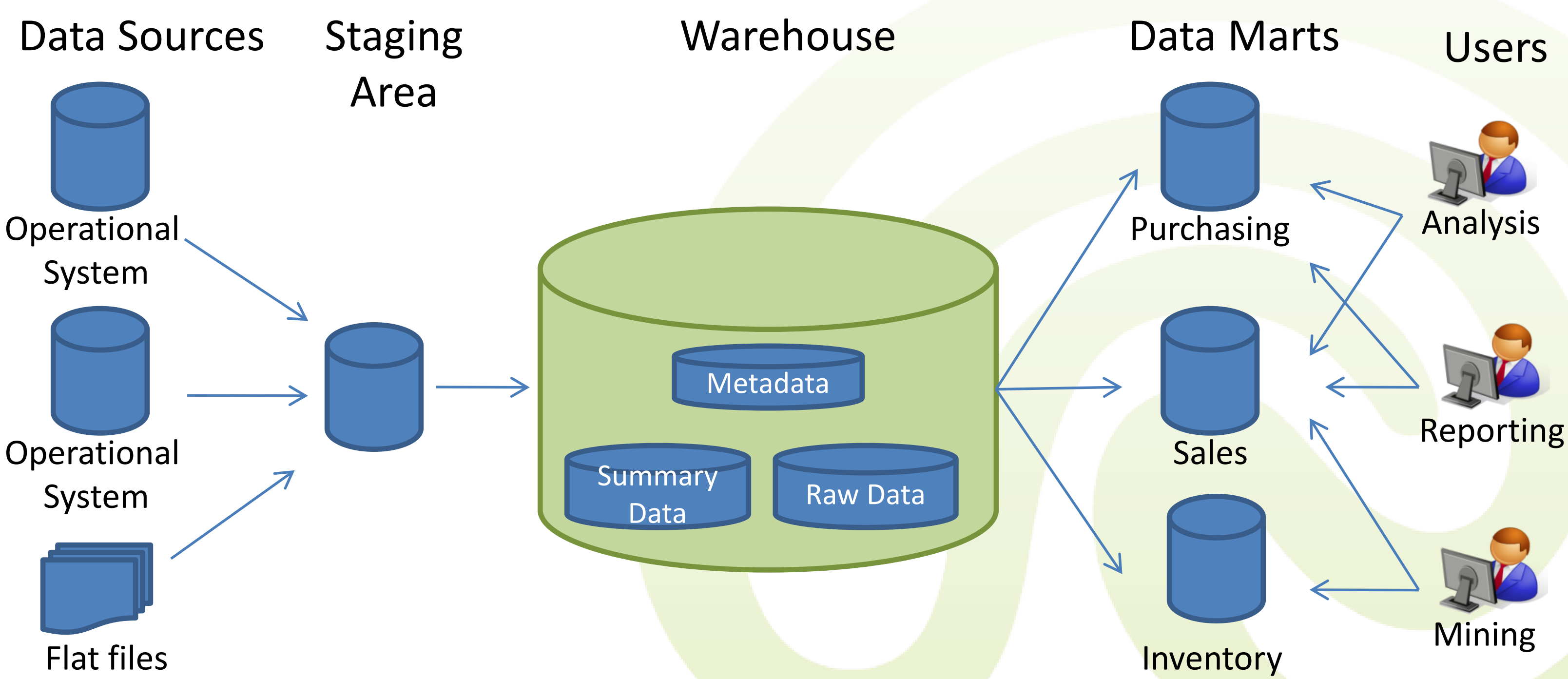
2.2 Independent Data Marts

- Mini warehouses – limited in scope
  - Faster and cheaper to build than DWs
- Separate ETL for each independent Data Mart
  - Redundant processing for each mart



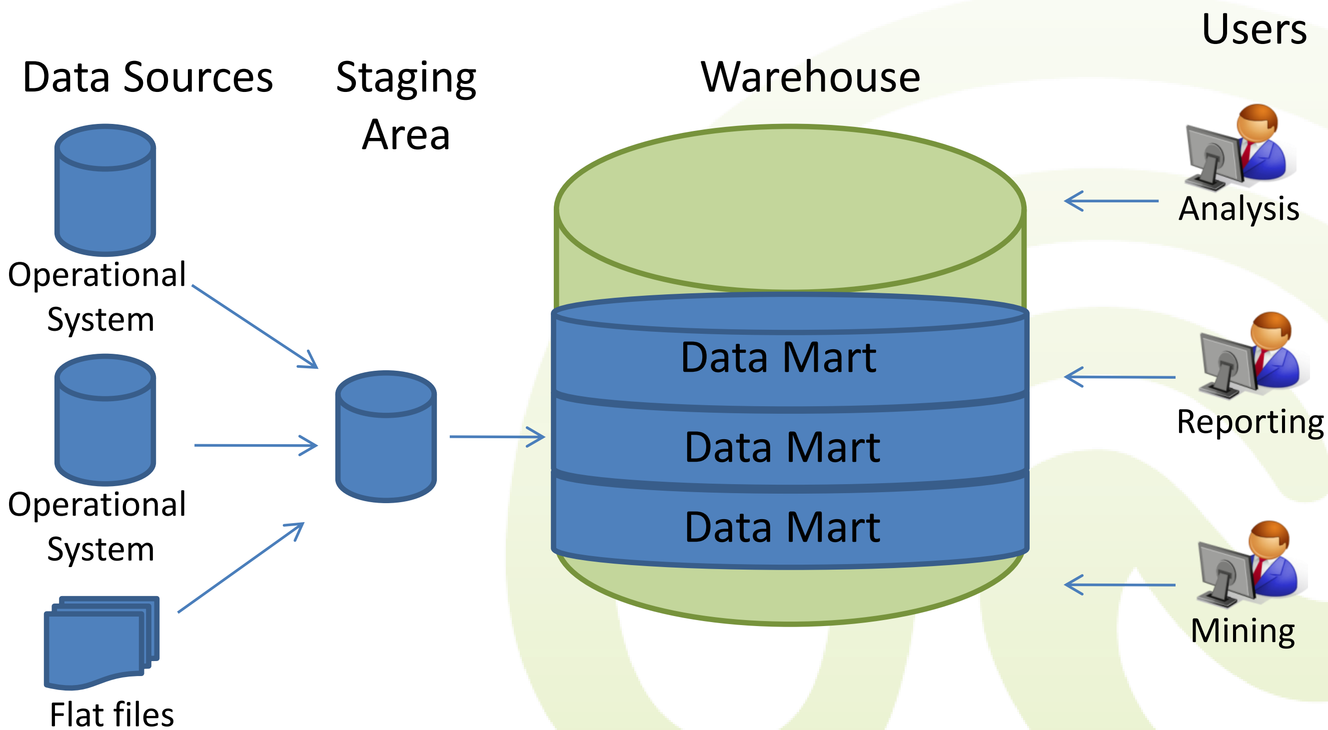
2.2 Dependent Data Mart

- Single ETL for the DW
  - No redundancy in the ETL process
- Data Marts are loaded from the DW



2.2 Logical Data Mart

- Data Marts are *not* separate databases, but logical views of the DW
  - Integrated view of the enterprise



2.2 DW vs. Data Marts

Scope	
DW	**Data Marts**
Application independent	Specific DSS application
Centralized,	Decentralized by user area
Planned	Organic, possibly not planned

Data	
DW	**Data Marts**
Historical, detailed, summarized	Some history, detailed, summarized
Lightly denormalized	Highly denormalized

Subjects	
DW	**Data Marts**
Multiple subjects	One central subject

Sources	
DW	**Data Marts**
Many internal and external sources	Few internal and external sources

Other characteristics	
DW	**Data Marts**
Flexible	Restrictive
Data-oriented	Project oriented
Long life	Short life
Large	Start small, becomes large
Single complex structure	Multiple, semi-complex structure, together complex

2.2 Centralized vs. Distributed

- DW may be **centralized** or **distributed**
- Centralized DW (e.g. Volkswagen)
  - Analytical queries are run only at the main enterprise location - no need to transport data via network
  - High costs for large dedicated hardware
- Distributed DW (e.g. WalMart)
  - More natural form due to corporations being active all over the world and having different types of hardware and software
  - Higher overhead but lower cost

2.2 Distributed DW

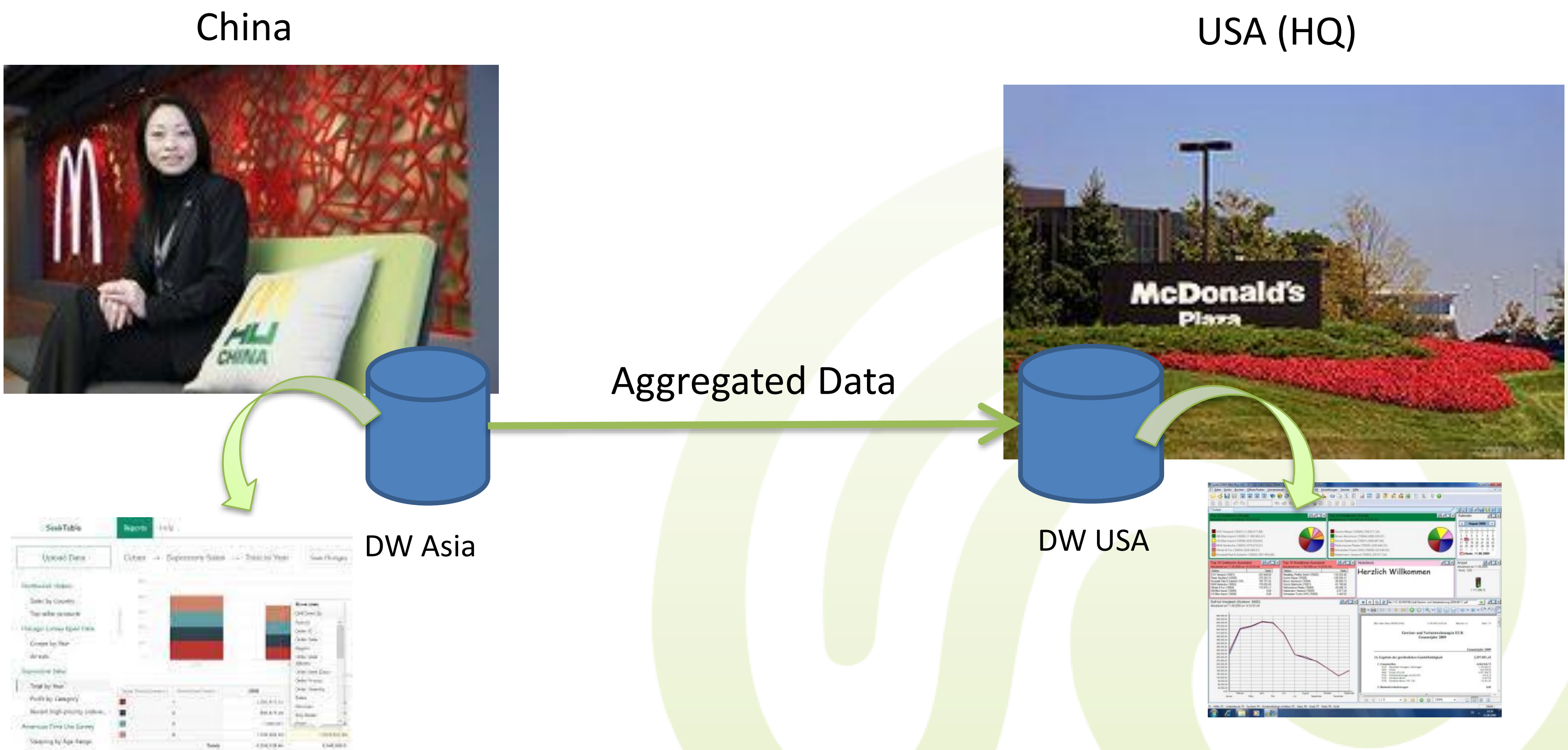
- Types of **distributed DW**
  - **Geographically** distributed
    - Local DW/global DW
  - **Technologically** distributed DW
    - Logically one DW, physically more DW
  - **Independently evolving** distributed DW
    - Uncontrolled growth

2.2 Distributed DW

- **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 somewhat autonomous

2.2 Distributed DW

- Typical example is franchising e.g. McDonald's

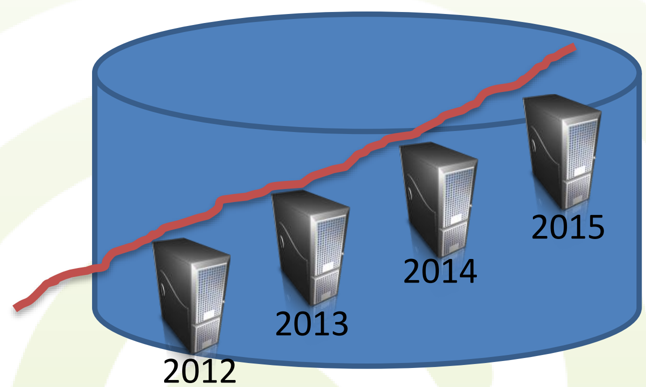


2.2 Distributed DW

- **Technologically** distributed DW
  - Placing the DW on the distributed technology of some vendor
  - Advantages
    - Entry costs are cheap – large centralized hardware is expensive
    - No theoretical limit on how much data can be placed in the DW –new servers can be added to the network on demand

2.2 Distributed DW

– As the DW starts to expand network **communication** starts playing an important role
  - Example: Let's simplify and consider we have 4 nodes each holding data regarding a specific year
  - Now let's consider a query which needs to access data from the last 4 years
  - Large amount of data has to be shipped to processing units



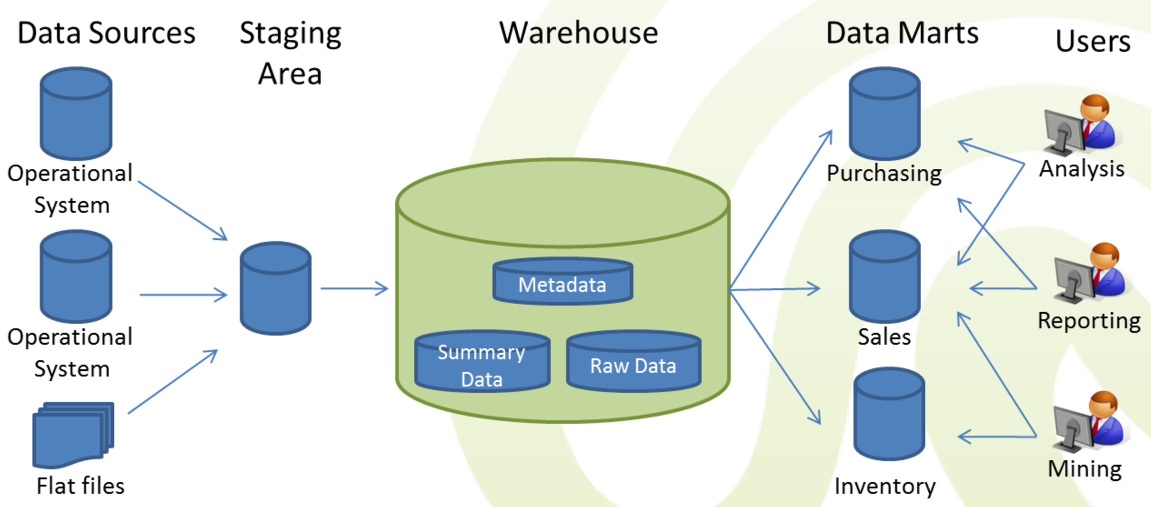
2.2 Distributed DW

- **Independently evolving** distributed DW
  - In practice there are many cases in which independent DW are developed concurrently in the same organization
    - The first step in many corporations is to build a DW for financial or marketing
    - Once this is successfully set up, other parts of the organization follow independently

2.3 DW Data Storage

Detour

- Goal of data storage :
  - Store data in a form that assists data mining, analytics, reporting and ultimately the users
- The last architecture layer dictates the way storage is performed!



2.3 DW Data Storage

Detour

- DW users look at the data from different **perspectives** e.g., time, location, product, etc.
  - Perspectives are called **dimensions** and the resulting data structure is **multidimensional**
  - Example: The sales department of a car manufacturer takes a closer look at the sales volumes
    - **View historical sales** volume figures from multiple perspectives:
      Sales volume by **model**,
      by **color**, by **dealer**, over **time**.

2.2 Multidim. Structure

Detour

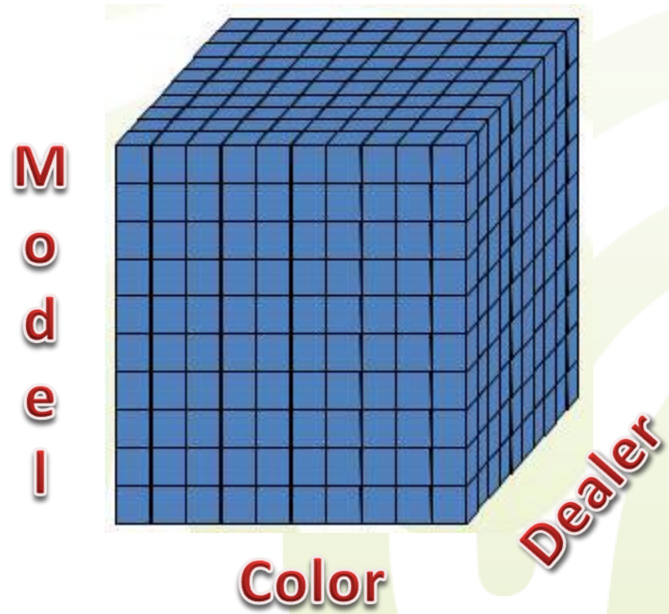
Model \ Color	Black	Blue	Red	*
*	289	451	40	1560
Mini VAN	113	324	18	455
Sedan	160	115	6	281
Coupe	16	12	16	44

Measure

Dimensions

2.3 DW Data Storage

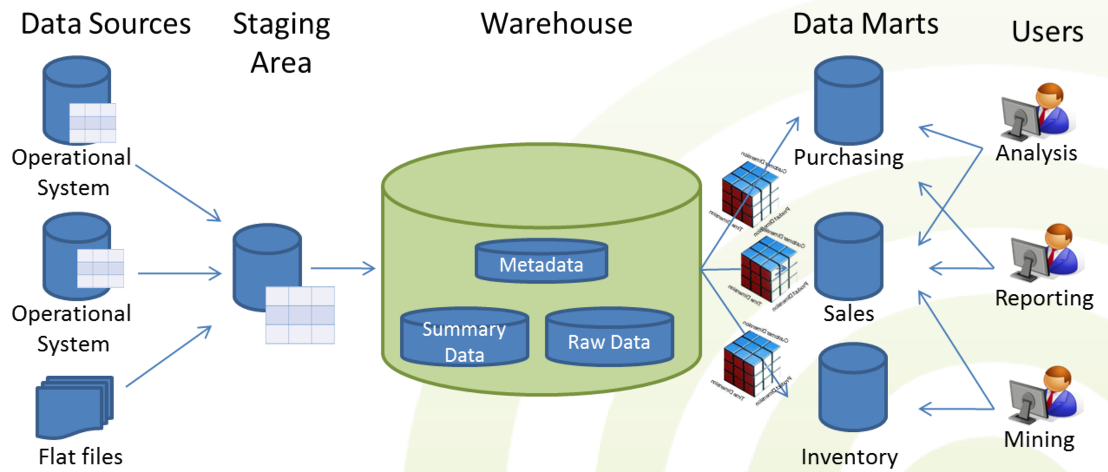
- The **complexity** grows quickly with the number of dimensions and the number of positions
  - E.g. 3 dimensions with 10 values each



2.3 DW Data Storage

Detour

- Visualization is **multidimensional**
- At the same time operational data is stored in **relational model**



- Data in the DW can be stored either according to the **relational** or **multidimensional** model

- Any database manipulation is possible with **both technologies**
- The multidimensional model however offers some advantages

  in the context of DW:
  - Ease of data **presentation**
  - Ease of **maintenance**
  - **Performance**

2.3 Ease of Presentation

Detour

- Multidimensional model
  - The presentation is the natural output of the multidim. model
- Relational model
  - Obtaining the same presentation in the relational model requires a complex query - think about the WalMart example:

select sum(sales.quantity_sold) from sales, products, product_categories, manufacturers, stores, cities where manufacturer_name = 'Colgate'
and product_category_name = 'toothpaste'
and cities.population < 40 000
and trunc(sales.date_time_of_sale) = trunc(sysdate-1)
and sales.product_id = products.product_id
and sales.store_id = stores.store_id
and products.product_category_id = product_categories.product_category_id
and products.manufacturer_id = manufacturers.manufacturer_id
and stores.city_id = cities.city_id

2.3 Ease of Maintenance

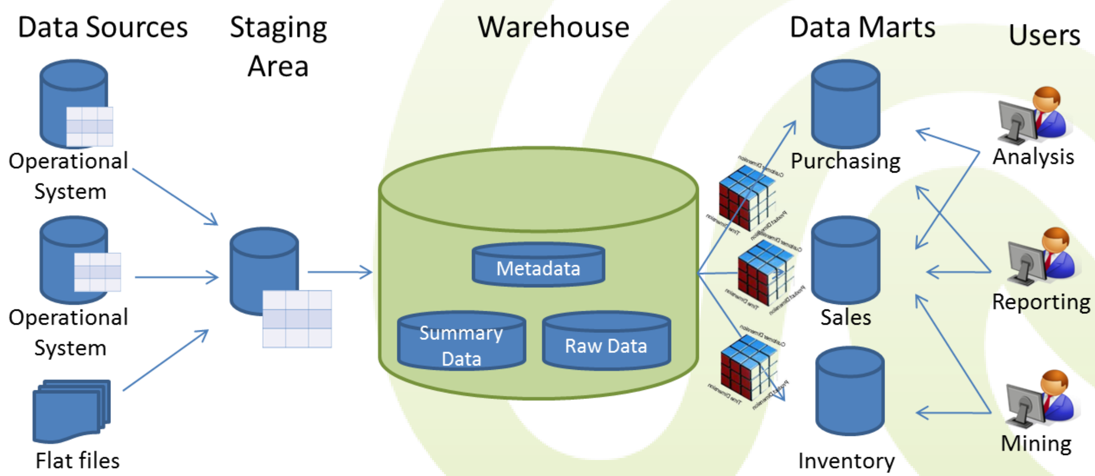
Detour

- Multidimensional model
  - When new data is added to the DW, **aggregates** need to be maintained in the case of the multidimensional model
- Relational model
  - The relational model use **indexes** and **sophisticated joins** which require **significant maintenance** and storage to provide same intuitiveness

2.3 Performance

Detour

- Consider storing the data in DW according to the relational model
  - **For each query**, the data has to be transformed from relational to multidim. representation
  - Storing the data in DW in a multidim. model, the transformation is performed only **on each load**



2.3 Performance

Detour

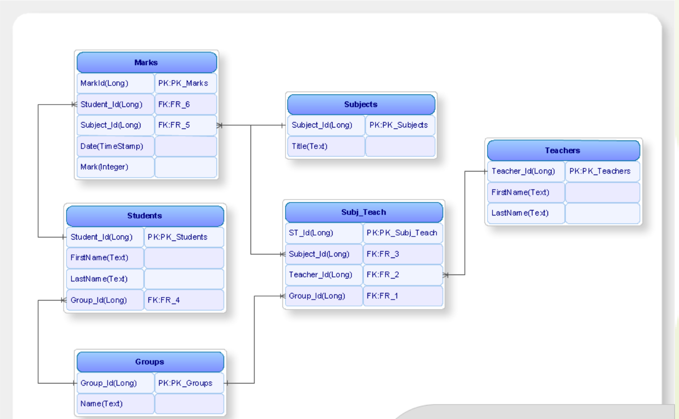
- For DW, relational model can reach similar performance as the multidim. model through database tuning
  - Not possible to **tune** the DW for **all** possible ad-hoc queries
- Conclusion: both models can be used, but the **multidimensional model** is the practical choice!
- How do we **model** the multidimensional representation?

2.4 Data Modeling

Detour

- **Data modeling - 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

Remember RDB 1 ?



Sheme of Database "Students and Teachers"

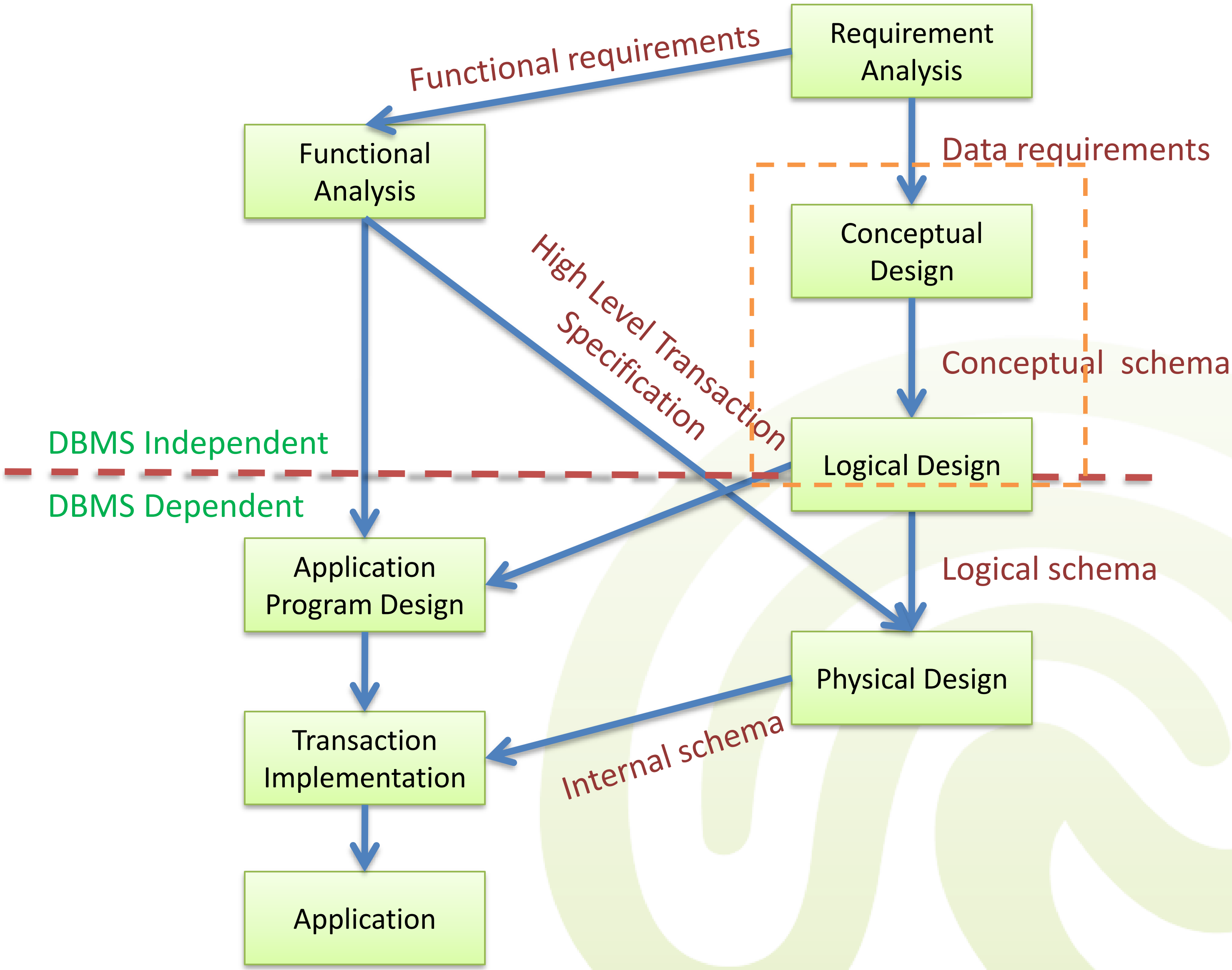
2.4 Data Modeling

Detour

- **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

2.4 Phases

Detour



2.4 Phases

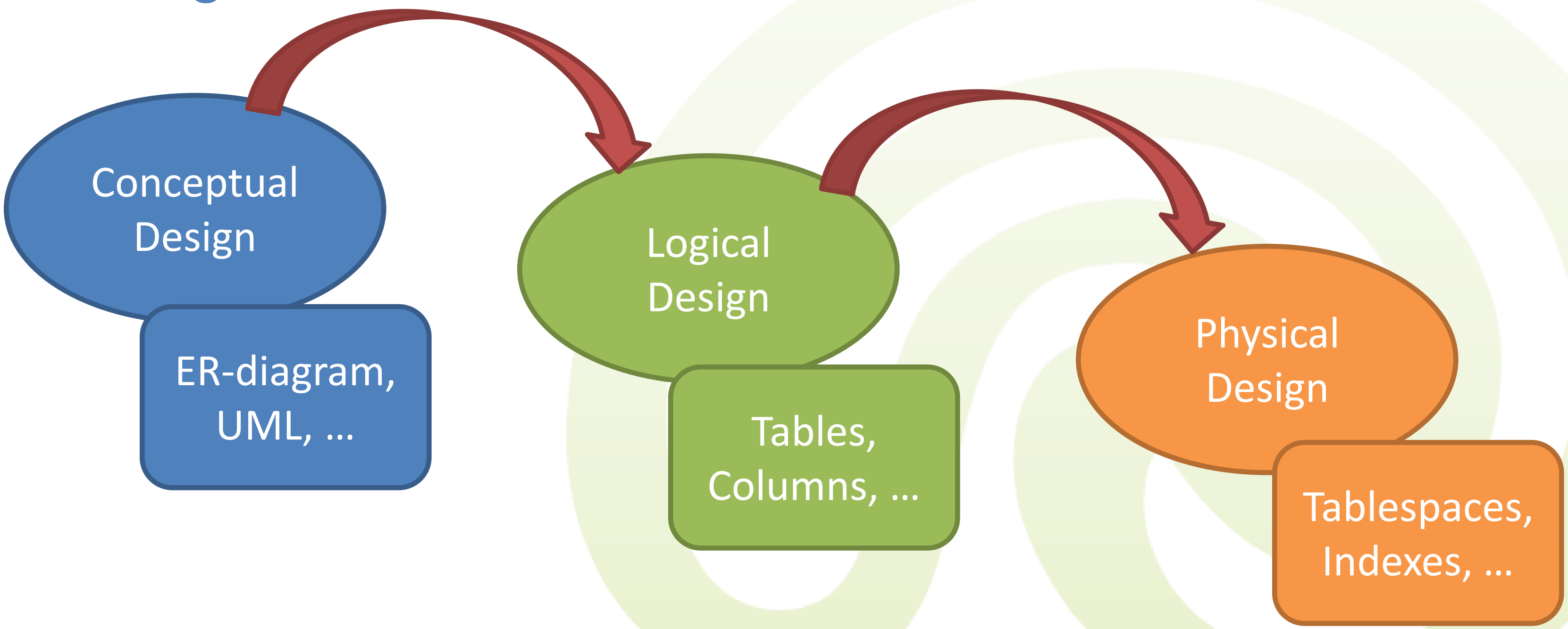
Detour

- **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, Multidimensional model, …
    - Technology independent conceptual model is **adapted to the used DBMS software**
- **Physical Design (next lecture)**
  - Creates internal structures needed to efficiently store/manage data
    - Table spaces, indexes, access paths, …
    - Depends on used hardware and DBMS software

2.4 Phases

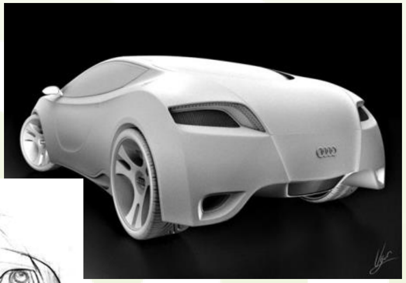
Detour

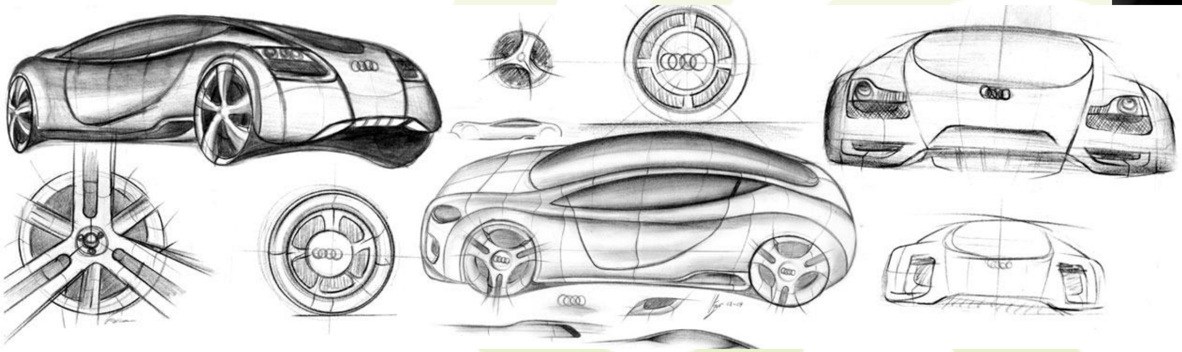
- Going from one phase to the next:
  - The phase must be complete
    - The result serves as input for the next phase
  - Often automatic transition is possible with additional designer feedback



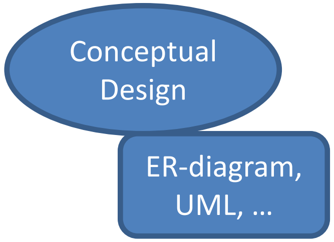
2.4 Conceptual Model Detour

- 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



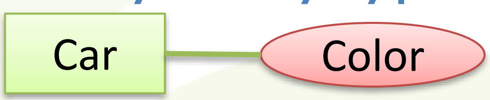


2.4 Conceptual Model Detour

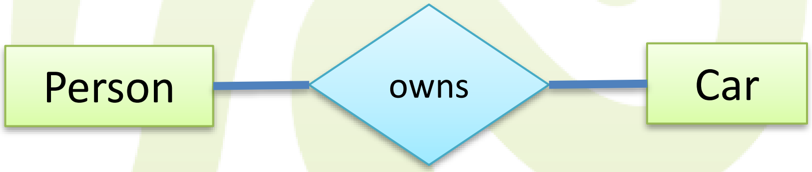


- Conceptual design
  - **Entity-Relationship (ER) Modeling**
    - Entities - "things" in the real world
      - E.g. Car, Account, Product

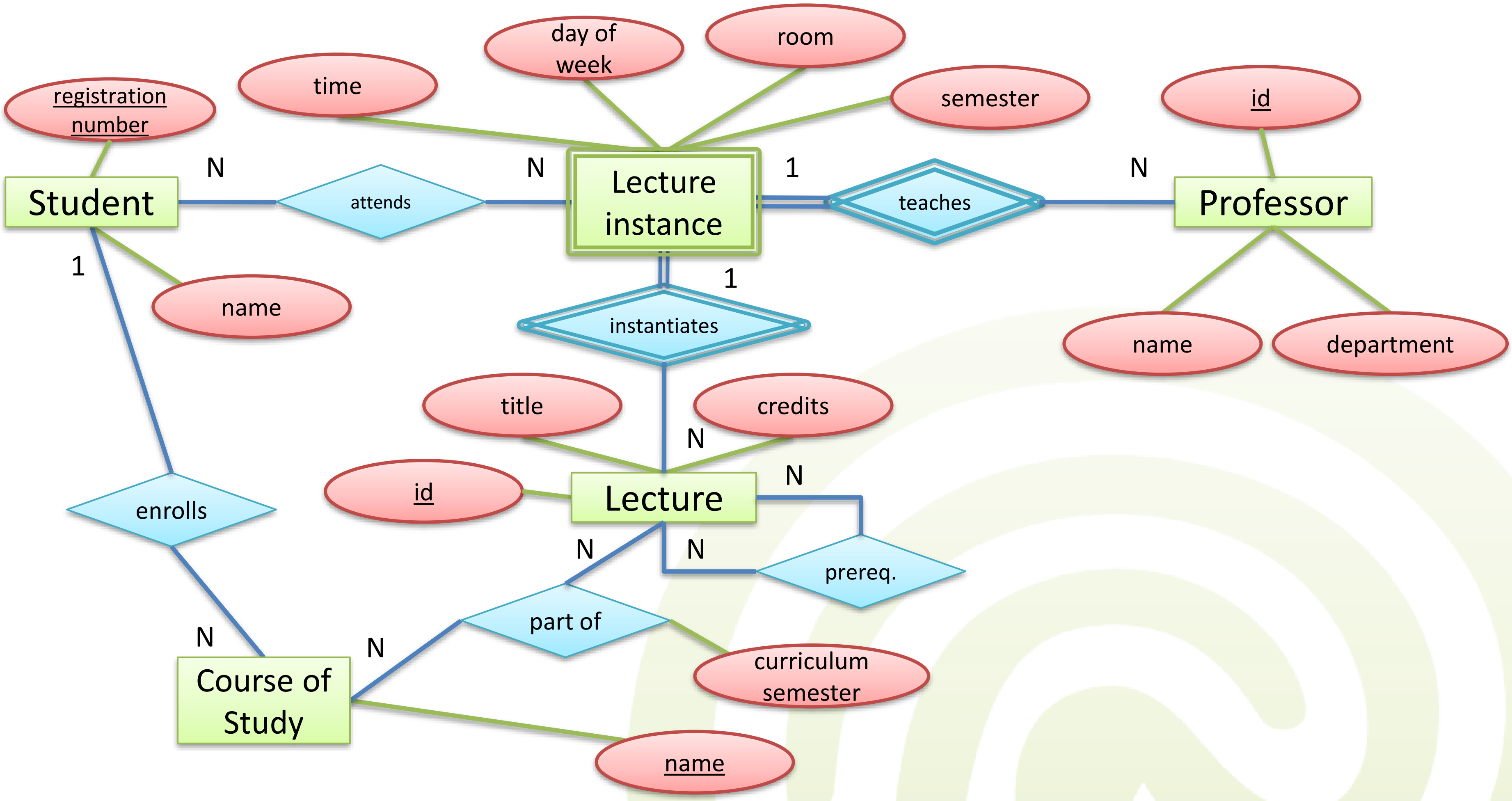
    - Attributes – property of an entity, entity type, or relationship type



      - 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

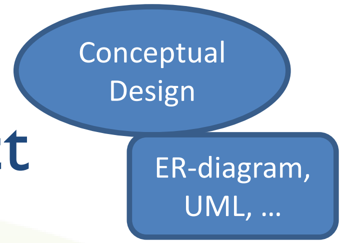


2.4 Conceptual Model Detour

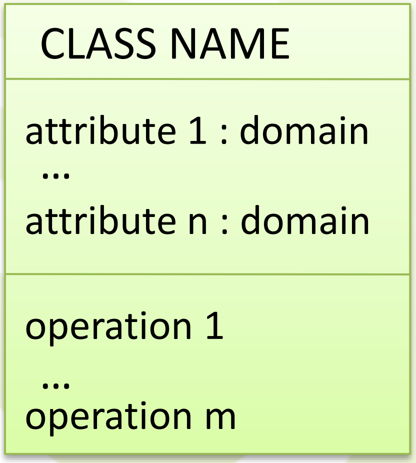


2.4 Conceptual Model Detour

- Conceptual design in usually done using the **Unified Modeling Language (UML)**

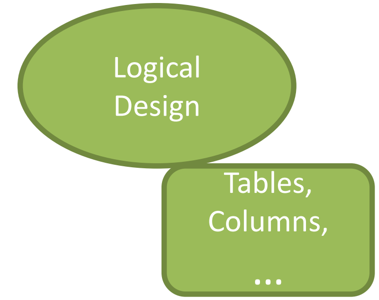


  - Class Diagram, Component Diagram, Object Diagram, Package Diagram…
  - For Data Modeling only **Class Diagrams** are used
    - Entity type becomes **class**
    - Relationships become **associations**

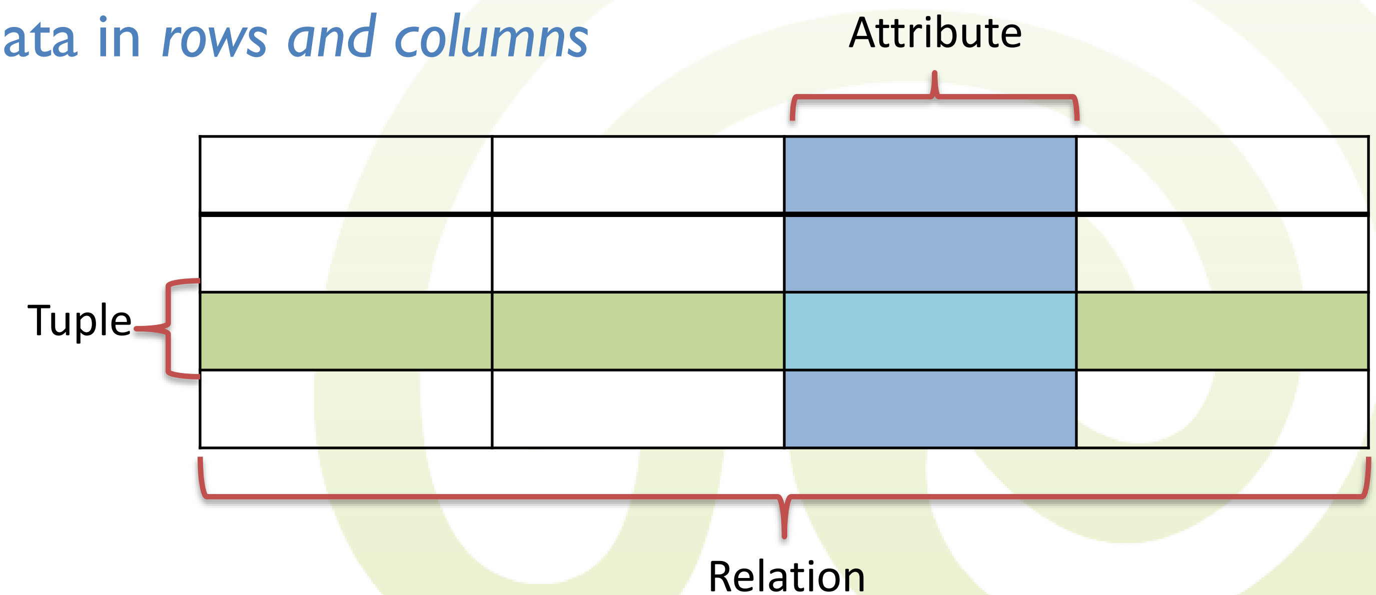


2.4 Logical Model

Detour

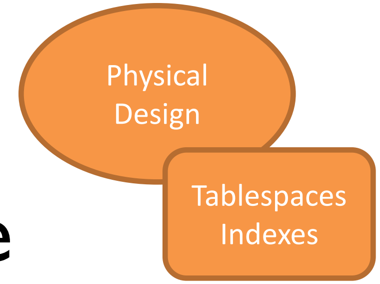


- Logical design **arranges data** into a logical structure
  - Which can be mapped into the **storage objects** supported by DBMS
    - In the case of RDB, the storage objects are *tables* which store data in *rows and columns*

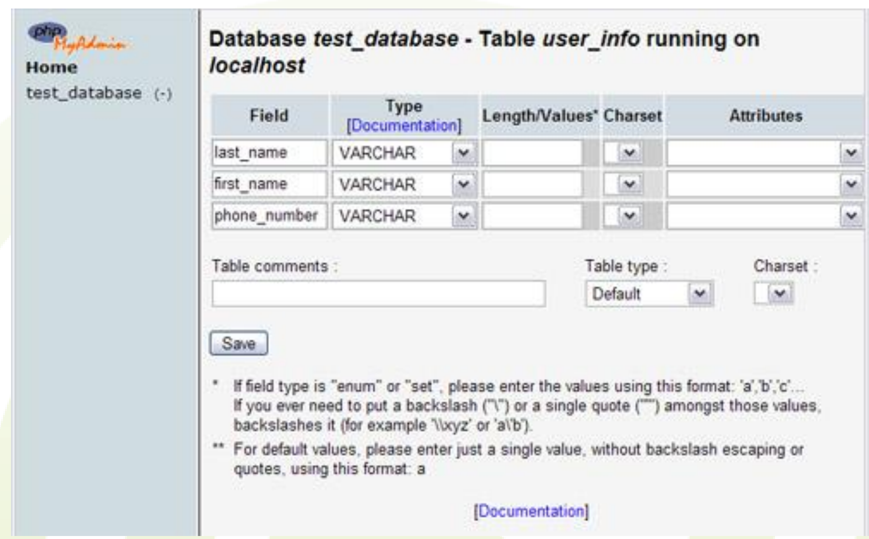


2.4 Physical Model

Detour



- 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**

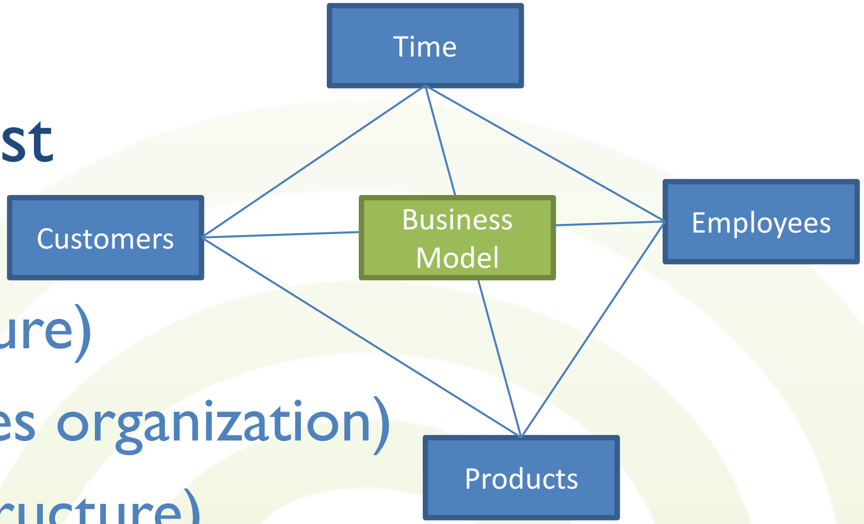


2.4 Data Modeling for DW

- For DW the models have to offer support for **multidimensional** data
- In the relational model the classical goal is to
  - **Remove redundancy**
  - Allow efficient retrieval of **individual records**
- In the case of DW
  - **Redundancy** is necessary to speed up queries
  - OLAP queries usually involve **multiple records** (range queries) and aggregates

2.4 Multidim. Conceptual Model

- Modeling business queries
  - Define the purpose of the DW and decide on the subject(s)
  - Identify questions of interest
    - Who bought the products? (customers and their structure)
    - Who sold the product? (sales organization)
    - What was sold? (product structure)
    - When was it sold? (time structure)



2.4 Multidim. Conceptual Model

- Components of conceptual design for DW
  - **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

2.4 Conceptual Design Models

- Multidimensional Entity Relationship **(ME/R)** Model
- Multidimensional UML **(mUML)**
- Other methods e.g., Dimension Fact Model, Totok, etc.

2.4 Multidim. E/R Model

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

2.4 Multidim. E/R Model

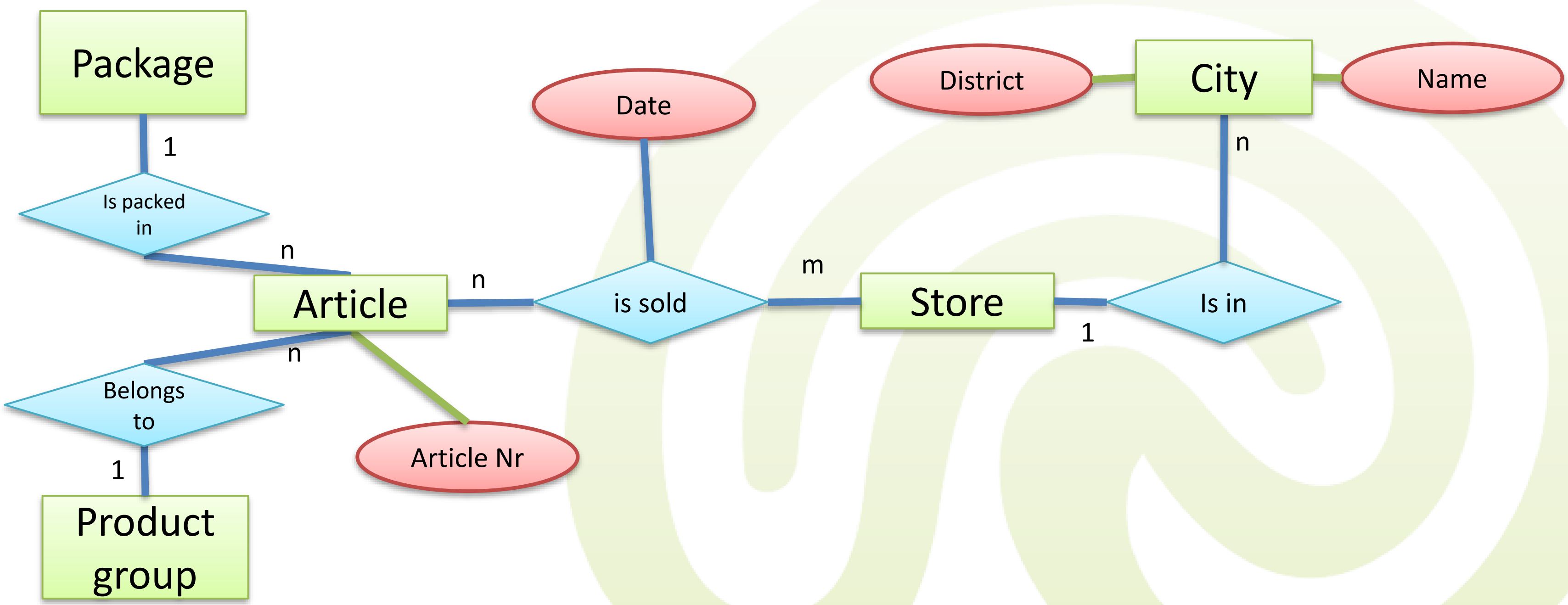
- ME/R notation was influenced by the following considerations
  - Specialization of the E/R model
    - All new elements of the ME/R have to be specializations of the E/R elements
    - In this way the flexibility and power of expression of the E/R models are not reduced
  - Minimal expansion of the 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**

2.4 Multidim. E/R Model

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

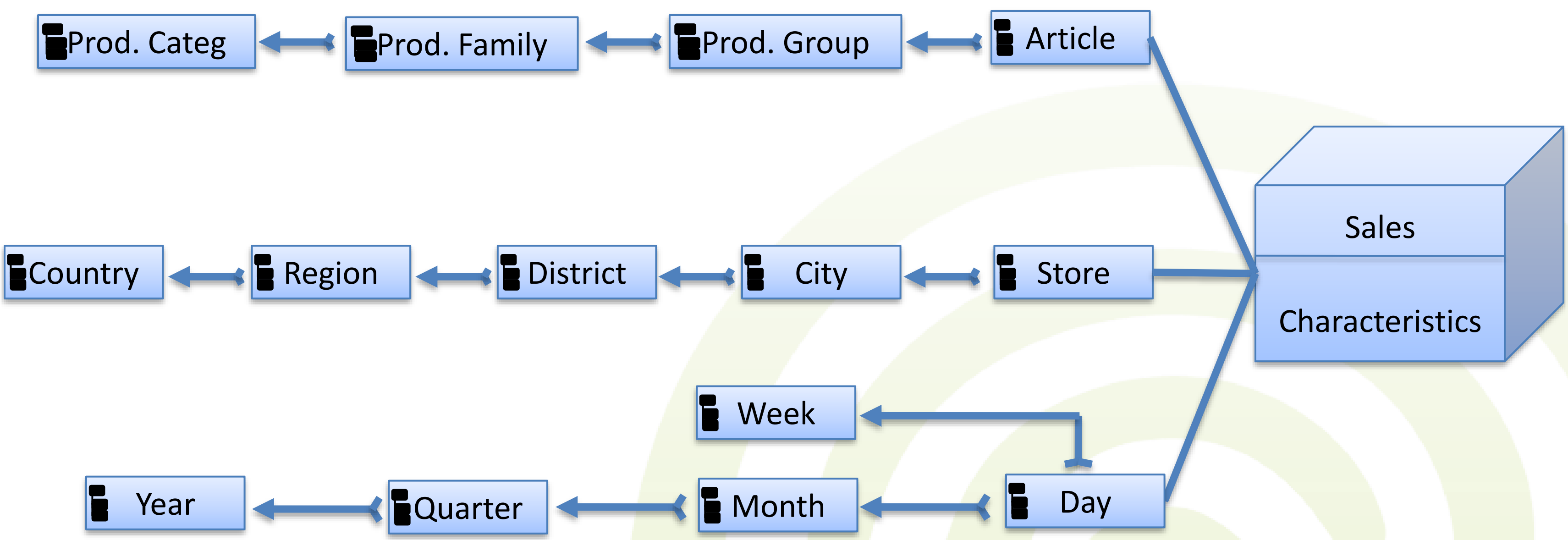
2.4 Multidim. E/R Model

- **Store scenario** designed in E/R
  - Entities bear little semantics
  - E/R is not suitable for representing classifications e.g. Store – City – Country, etc.



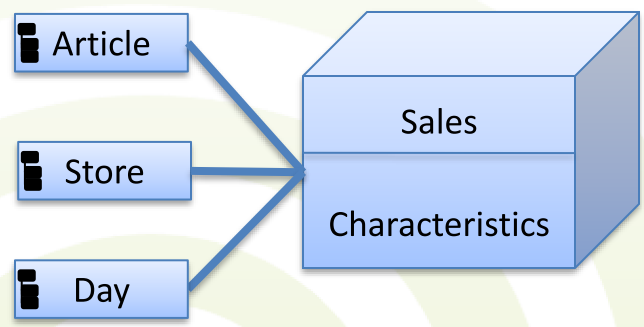
2.4 Multidim. E/R Model

- **ME/R** notation:

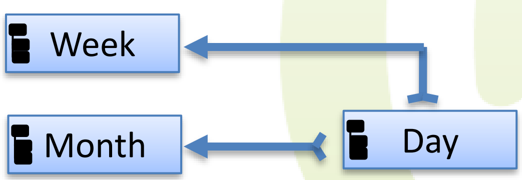


2.4 Multidim. E/R Model

- **ME/R** notation:
  - **Sales** was selected 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

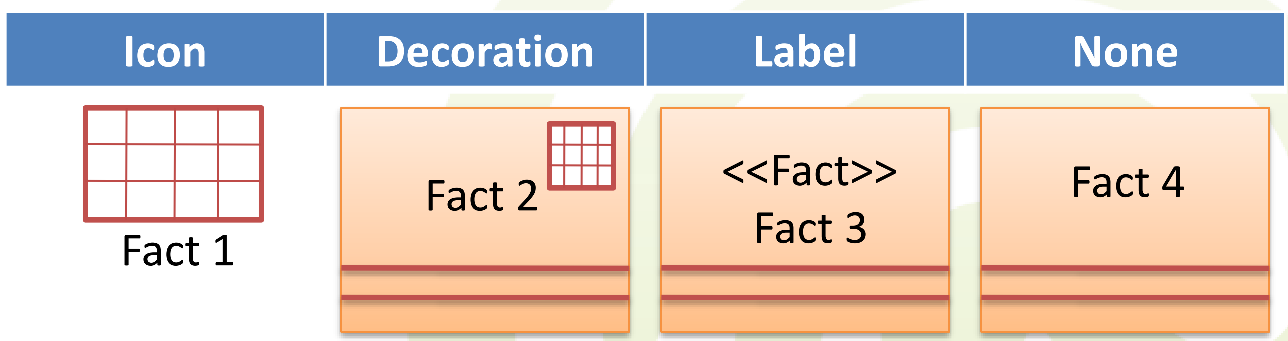


2.4 Unified Modeling Language

- **UML** is a general purpose modeling language
- It can be **tailored to specific domains by** using the following mechanisms
  - Stereotypes: building new elements
  - Tagged values: new properties
  - Constraints: new semantics

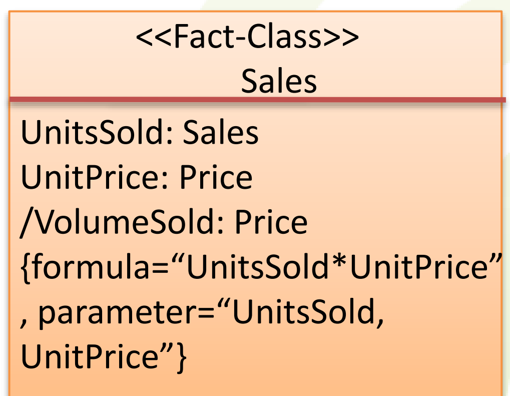
2.4 mUML

- **Stereotype**
  - Grants a **special semantics** to UML construct without modifying it
  - There are 4 possible representations of the stereotype in UML

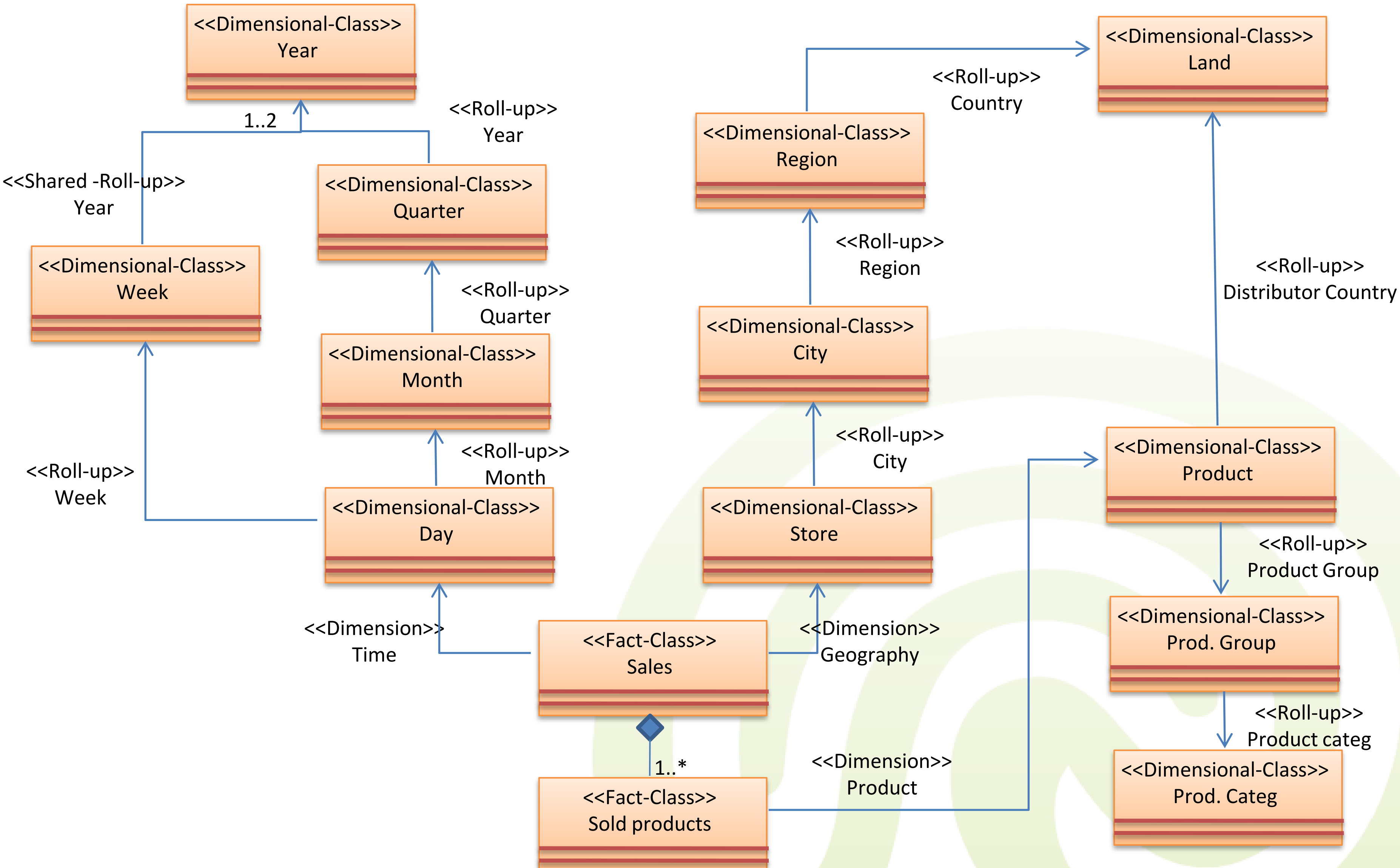


2.4 mUML

- **Tagged value**
  - Define properties by using a pair of **tag** and **data value**
    - Tag = Value
    - E.g. formula=“UnitsSold*UnitPrice”



2.4 mUML


<<Dimensional-Class>>
Year
1..2
<<Roll-up>>
Year
<<Shared -Roll-up>>
Year
<<Dimensional-Class>>
Quarter
<<Dimensional-Class>>
Week
<<Roll-up>>
Quarter
<<Dimensional-Class>>
Month
<<Roll-up>>
Month
<<Roll-up>>
Week
<<Dimensional-Class>>
Day
<<Dimension>>
Time
<<Fact-Class>>
Sales
1..*
<<Fact-Class>>
Sold products
<<Dimensional-Class>>
Land
<<Roll-up>>
Country
<<Dimensional-Class>>
Region
<<Roll-up>>
Region
<<Dimensional-Class>>
City
<<Roll-up>>
City
<<Dimensional-Class>>
Store
<<Dimension>>
Geography
<<Roll-up>>
Distributor Country
<<Dimensional-Class>>
Product
<<Roll-up>>
Product Group
<<Dimensional-Class>>
Prod. Group
<<Roll-up>>
Product categ
<<Dimension>>
Product
<<Dimensional-Class>>
Prod. Categ

Summary

- Architectures:
  - Basic architecture, vertical three-tier architecture, horizontal dependent/independent data mart architecture
  - DW may be centralized or geographically and technologically distributed
- Data Modeling: Data in the DW is represented in a multidimensional manner
  - Multidimensional conceptual model
    - Multidimensional Entity Relationship (ME/R) Model
    - Multidimensional UML (mUML)

Next lecture

- Data Modeling (continued)
  - Logical model
  - Physical model