FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

188.429 Business Intelligence

(4.0 VU / 6.0 ECTS)

2016W

Elmar Kiesling
elmar.kiesling@tuwien.ac.at

Andreas Rauber
rauber@ifs.tuwien.ac.at

A Min Tjoa
amin@ifs.tuwien.ac.at

Institute of Software Technology and Interactive Systems
Information and Software Engineering Group
Vienna University of Technology

Favoritenstraße 9-11/188
1040 Vienna, Austria

Phone: +43 (1) 58801 – 18801

FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

188.429 Business Intelligence

(4.0 VU / 6.0 ECTS)

Data Warehousing (Part 1)

Elmar Kiesling
elmar.kiesling@tuwien.ac.at

Institute of Software Technology and Interactive Systems
Information and Software Engineering Group
Vienna University of Technology

Favoritenstraße 9-11/188
1040 Vienna, Austria

Phone: +43 (1) 58801 – 18801

Prerequisites

You should have a solid grasp on:

1. Conceptual database design (e.g. ER)
2. Relational database model
3. Normalization
4. DBMSs
5. SQL

Agenda today (DWH 1)

1. **Introduction:**
   motivation; historical, technical and business context; definitions etc.

2. **Data Warehouse Architecture:**
   - Characterization of Data Warehouses (DWH)
   - Distinction between a DWH and an Operational DB
   - Basic DWH "building blocks"

Reading assignments for this lecture:

- The Economist: "*A different game*"
- Codd and Salley (1993): *"Providing OLAP to User-Analysts"*
- Breslin (2004): *"Data Warehousing Battle of the Giants"*

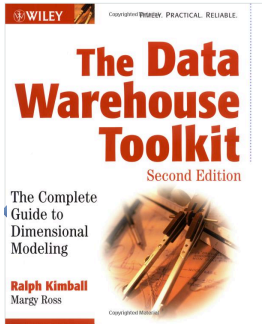
Preview

1. **Data Marts and Dimensional Modeling**
   - Snowflake Schema
   - OLAP Operations
   - Physical implementation, ROLAP Optimizations
2. **Data Warehouse:** time dimension, ETL..
3. **DWH Development Process**
4. **Recent Developments in DWHing**

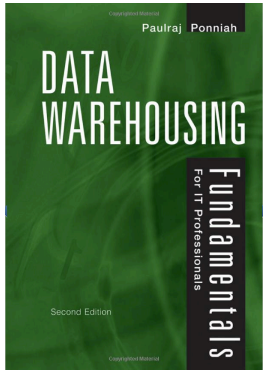
Reading assignments for this lecture:

- Kimball, Ross - The Data Warehouse Toolkit (2 ed.), Chapter 1
- Gray et al. (1997): Data Cube - A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals

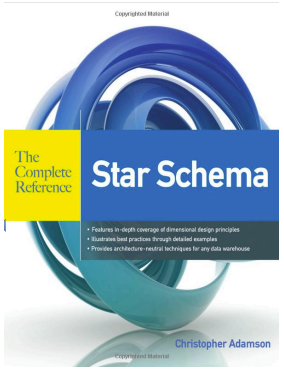
Recommended Literature (DWH part)



Kimball, R. and Ross, M.:
The Data Warehouse Toolkit:
The Complete Guide to Dimensional Modeling,
2nd Edition. John Wiley, 2002.



Ponniah, P.:
Data Warehousing Fundamentals:
A Comprehensive Guide for IT Professionals,
2nd Edition. John Wiley, 2010.



Adamson, C.:
Star Schema: The Complete Reference.
McGraw Hill, 2010.

…further references will be provided throughout the course

1. INTRODUCTION

Buzzwords

Industrial revolution of data

Digital data flood

Streaming Data

Information explosion

Data Science

Big Data

Data-driven discovery

Agile BI

Real-time BI

Mobile BI

Predictive Analytics

Cloud BI

Data-driven decision-making

Self-Service BI

The Age of (Big) Data

- Amount of digital information x10 every 5 years [1]
- Enterprises stored more than 7 EB of new data in 2010 [2]
- Scientific data is doubling every year [3]

Major drivers:

- More devices and inexpensive high-bandwidth sensors (mobile phones, IoT etc.)
- Decreasing cost of storage (Kryder's law)
- More people interact with information
- Social media growth
- Interest to store data in greater granularity and frequency

Sources:
[1] "Data, data everywhere," The Economist, 25-Feb-2010.
[2] McKinsey & Company, "Big data: The next frontier for innovation, competition, and productivity," 2011.
[3] Szalay: Data driven discovery in Science http://cra.org/ccc/docs/nitrdsymposium/pdfs/szalay.pdf

“Big” economic expectations..

“Data are becoming the new raw material of business:
an economic input almost on a par with capital and labour.”

The Economist [1]

*“..data can create **significant value for the world economy**, enhancing the productivity and competitiveness of companies and the public sector and creating substantial economic surplus for consumers.“*

McKinsey&Company [2]

Sources:
[1] “Data, data everywhere,” The Economist, 25-Feb-2010.
[2] McKinsey & Company, “Big data: The next frontier for innovation, competition, and productivity,” 2011.

… and “Information Crisis”

Amount of data increases faster than
- the ability of networks to carry it
- the ability of systems to store and analyze it
- **our ability make sense of it**

Enterprises collect more data than ever before **but..**

.. spread it across heterogeneous structures and systems

.. put it into systems that just produce even more data

.. do not trust in the collected data

.. struggle to get value out of the collected data

.. do not use it for strategic decision making

"We are drowning in information and starving for knowledge.”

John Naisbitt, 1982 (!) [1]

Then again.. “unreasonable effectiveness of data” [2]

→ **more data may actually make the job easier rather than harder**

[1] Naisbitt, J. and Aburdene, P.: Megatrends 2000: New Directions for Tomorrow, Avon Publishing, 1982

[2] Alon Halevy, Peter Norvig, Fernando Pereira, "The Unreasonable Effectiveness of Data", doi:10.1109/MIS.2009.36 -abs.html

Historical Context

1. **Age of Transactions** (ca. 1970 -)
   - Goal: reliability - make sure no data is lost
   - 1960s: hierarchical data model – e.g., IMS
   - 1970s - 1980s: relational data model – e.g., DB2, Oracle, Sybase,..

2. **Age of Business Intelligence** *(ca. 1995 -)*
   - Goal: analyze the data → make business decisions (*Decision support systems, dashboards, OLAP*)
   - Aggregate data for decision makers
   - Tolerate imprecision (e.g., slightly out-of-date information)
   - ROLAP (relational OLAP): SAP BW/Business Objects, IBM Cognos, …
   - MOLAP (Multi-dimensional OLAP): Oracle Hyperion, Essbase, …

3. **"Big Data", Predictive and Prescriptive Analytics..** (ca. 2010 -)
   - Goal: optimization, predictive forecasting, real-time decisions, …
   - Efficient technologies for processing big polystructured data (e.g., Hadoop)

Adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)
D. Kossmann & D. Binning 2009

Historic roots of Decision-Support Systems (DSSs)

1. **Ad-hoc reports**: IT wrote special programs on an ad-hoc basis
2. **Special extract programs:** suite of programs that were run periodically
3. **Small applications**: users could stipulate the parameters for each report
4. **Executive Information Systems** (starting in the 1960s):
   - Only static screens and reports from small, aggregated extracts of data
   - Mainframe-based
5. **Management Information Systems** (starting in the 1980s)
   - Still mainly static report generators, but introduction of hierarchies (roll-up, drill-down)
   - Client-Server-Architectures, GUIs
6. **Decision-Support Systems**:
   - More sophisticated systems intended to provide strategic information
   - Menu-driven, online information, ability to prepare specialized reports

Source: Adapted from
Ponniah, P.: Data Warehousing Fundamentals, 2ed. Wiley (2010)
Mansmann/Weiler, Uni Konstanz: Data Warehousing und OLAP Winter 2011/12

History of BI Concepts

1989: H. Dresner introduces the term “Business Intelligence”
- “*concepts and methods to improve business decision making by using fact-based support systems*” [1]
- Earliest use of the term already in 1958 [2]
- Broad umbrella term, mainly marketing at first
- Term not broadly adopted until the late 1990s

1992: W.H. Inmon defines the “Data Warehouse” concept
- Redundant data storage in one central location
- Independent from operational systems
- Collect data and optimize its structure for analysis purposes

1993: E.F. Codd coins the term “OLAP” [cf. reading assignment]
- “On-line analytical processing”
- Dynamic, multi-dimensional analysis

[1] D. J. Power (10 March 2007). "A Brief History of Decision Support Systems”. DSSResources.COM.
[2] H P Luhn (1958). “A Business Intelligence System”. IBM Journal 2 (4): 314. doi:10.1147/rd.24.0314.

Business Intelligence – A first characterization

Motivation

- Organizations usually have multiple operational systems
- Operational systems are not good at reporting
- ERPs do not produce good strategic reports
- Need “single version of the truth”

BI is strategic in nature

- Aims at better, fact-based decision-making
- More about strategic issues rather than operational efficiency (BPM)
- Target roles: executives, middle management (although benefits for other groups too)

Industries with relatively high penetration:

- Financial services (e.g., credit rating, risk, fraud detection, portfolios..)
- Retail and wholesale (e.g., SCM, CRM..)
- Airlines and hospitality (e.g., yield management)
- Healthcare
- Government

Partly adapted from “Business Intelligence Part 1 - Ray Rubenstein”
https://www.youtube.com/watch?v=zASyUyzlXQg&feature=related

Illustrative examples of opportunities created through BI

- found that 7% of its customers accounted for 43% of its sales [1]
- reorganized its stores to concentrate on those customers' needs

- 35% of item purchases from recommendation engine [3]

- handles more than 200m customer transactions/week [2]
- More than 2.5 PB of DBs (167x Library of Congress) in 2010 [1]
- Each transaction in DWH ready for analysis within 7mins[4]
- SCM: put stock management in the hands of suppliers (able to see the exact number of products on every shelf of every store)
- BI integrated into operational processes
- BI to extract strategic insights

- Data-driven businesses built fundamentally on data and data mining
- Strategic BI, but also integrated into operational business processes

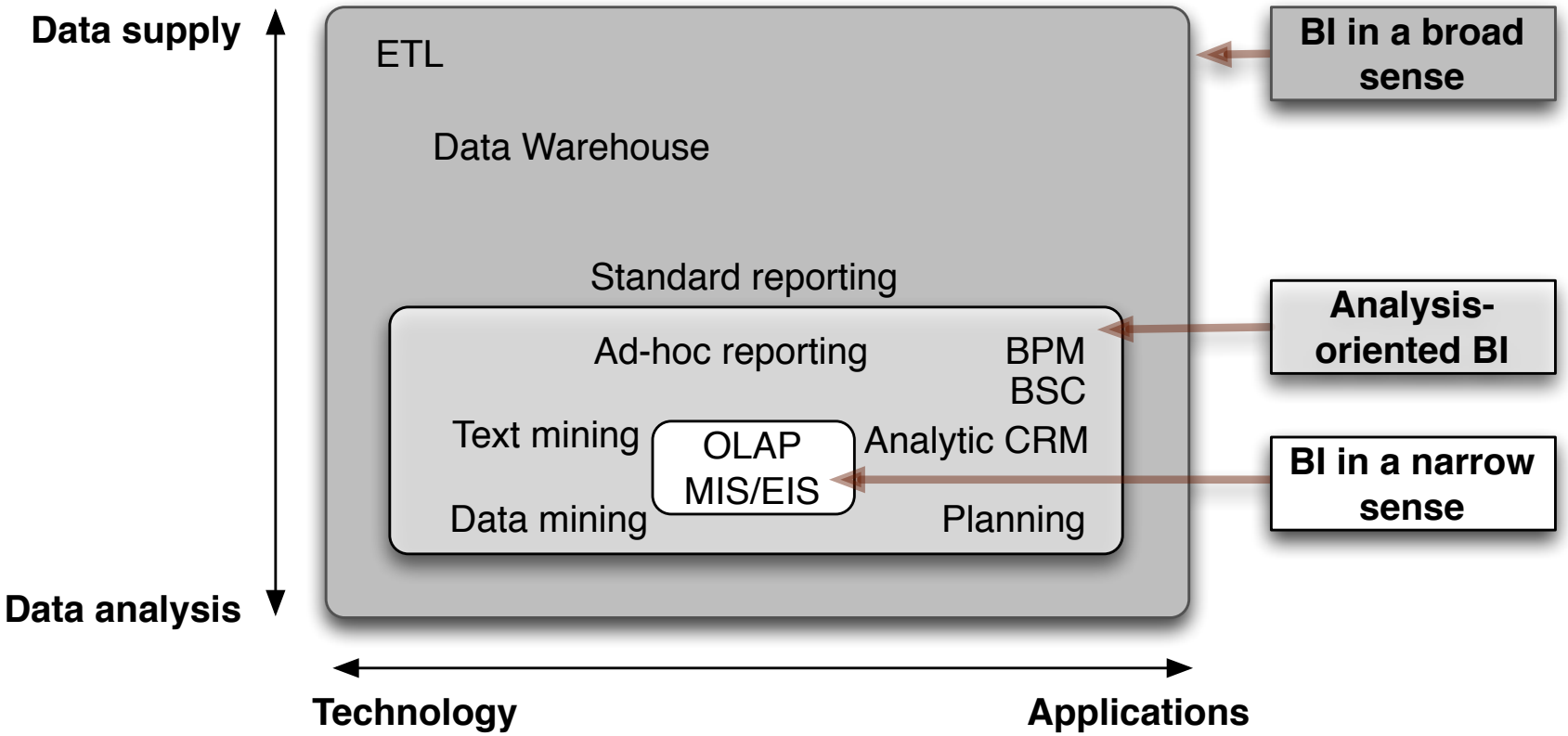
Sources:
[1] "A different game," The Economist, 25-Feb-2010.
[2] "Data, data everywhere," The Economist, 25-Feb-2010.
[3] http://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers, Nov 2016
[4] https://www.healthcatalyst.com/?p=1948

What is “Business Intelligence”?

Data Warehousing Institute (TDWI 2002) working definition:

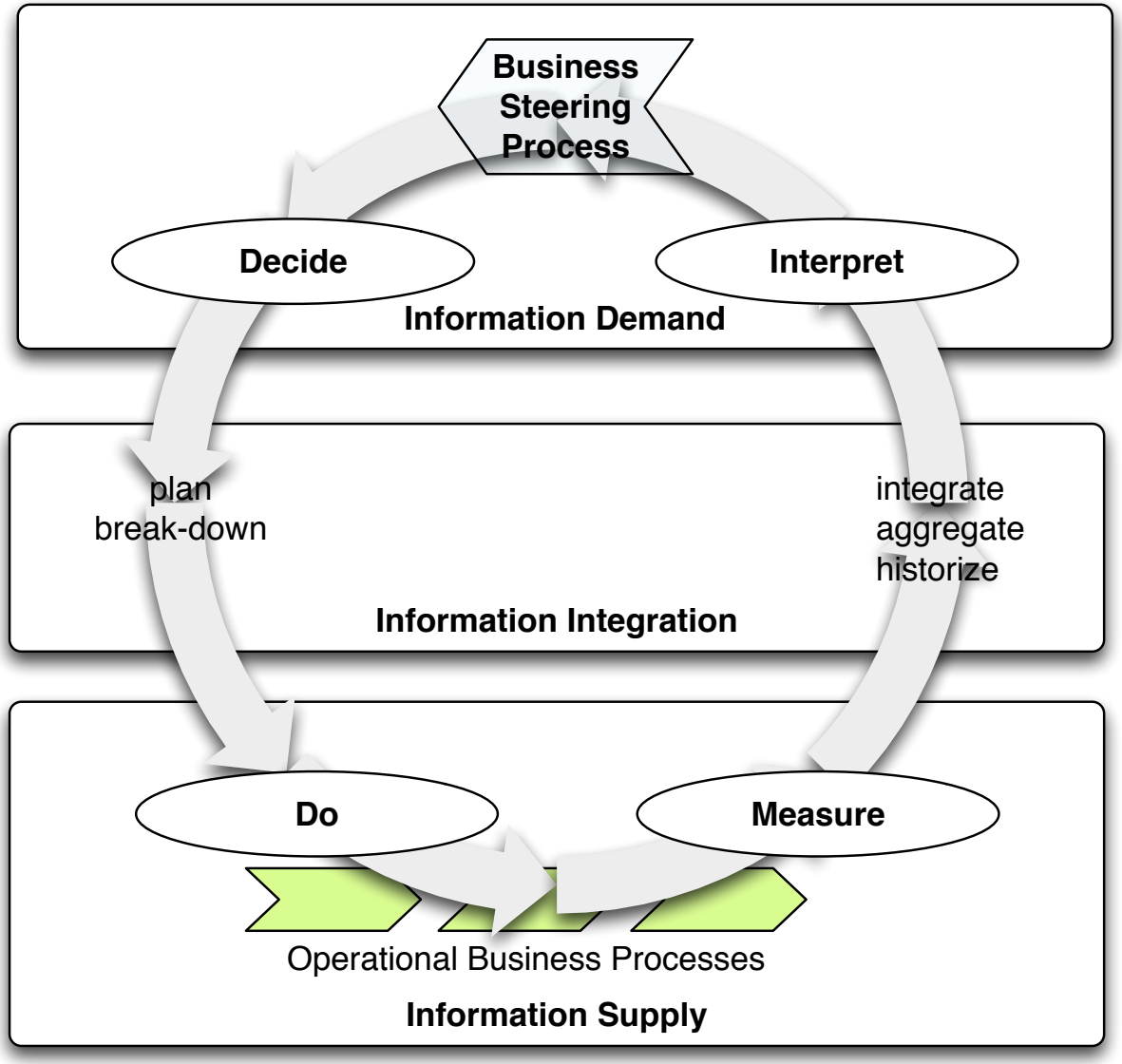
> *“The processes, technologies, and tools needed to turn data into information, information into knowledge, and knowledge into plans that drive profitable business action. Business intelligence encompasses data warehousing, business analytic tools and content/knowledge management”*

What is “Business Intelligence”?



Translated and adapted from Gluchowski, P. (2001),
Business Intelligence, in: HMD – Praxis der Wirtschaftsinformatik, 38 (222)
As cited in Kemper, Mehanna, Baars: Business Intelligence, Vieweg 2010

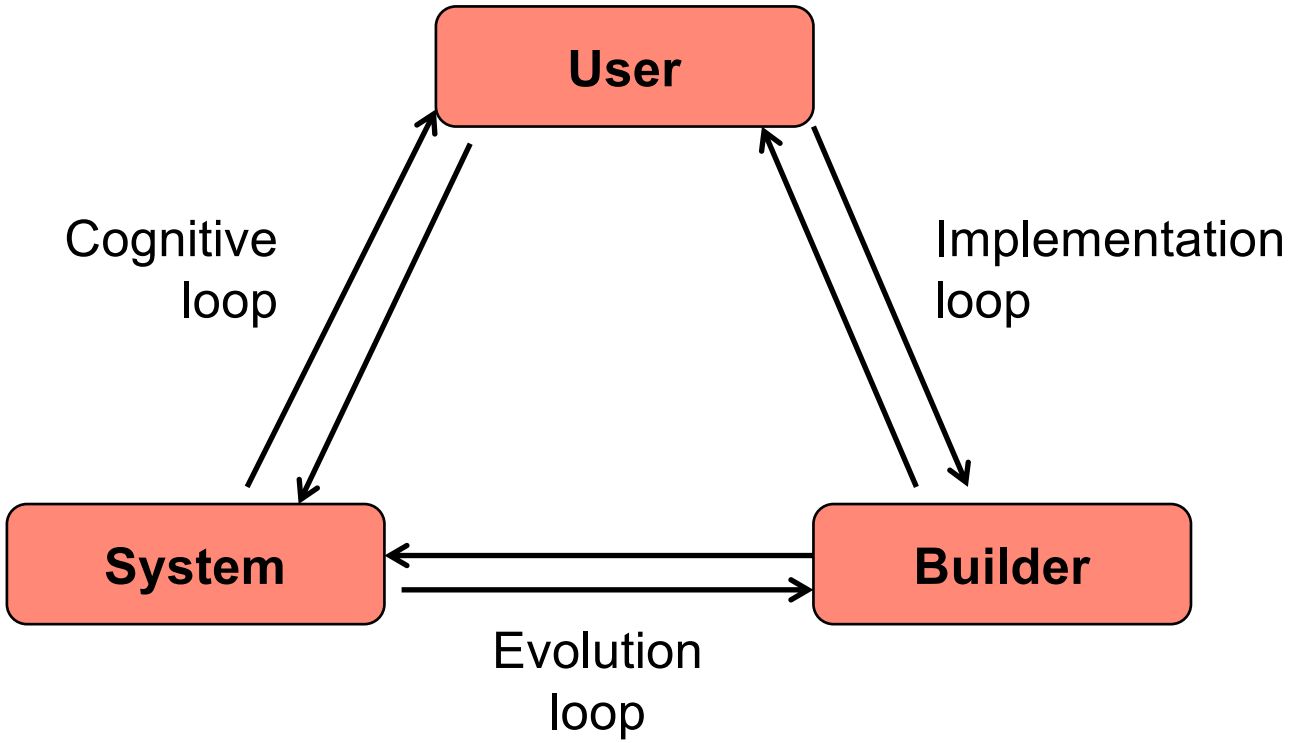
Goal: “Closing the Loop” and Business Performance Management



Source:
Adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

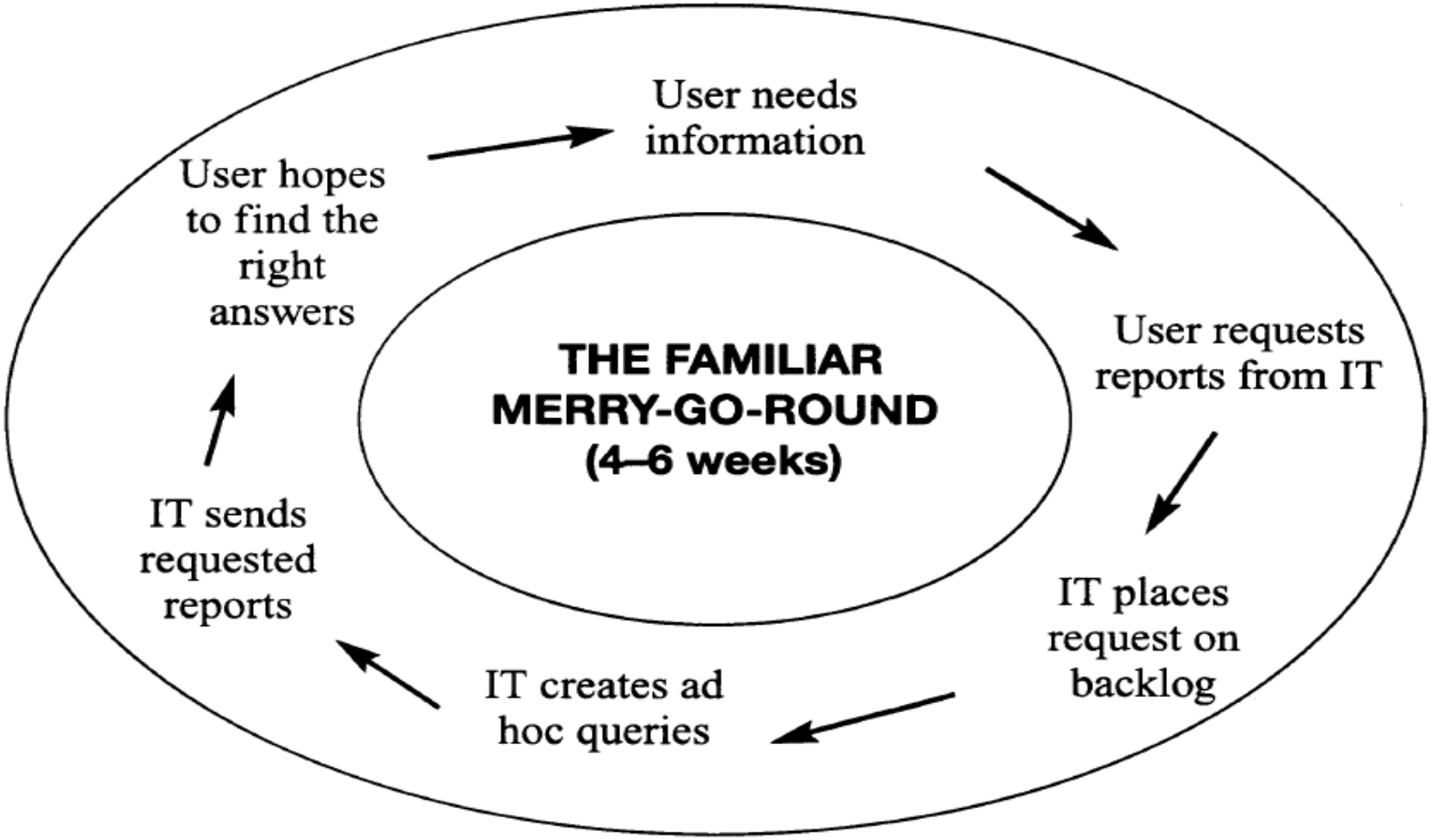
DSS implementation dynamics

Keen's Adaptive Design Framework for Decision Support Systems:



Keen (1980) as cited in D. Arnott, "A framework for understanding decision support systems evolution," in *9th Australasian Conference on Information Systems, Sydney, Australia: University of New South Wales*, 1998.

Typical failure pattern



Current Development in BI

Traditional approach: driven by information requirements

- Business users determine what questions to ask
- IT structures the data to answer that question

  e.g. sales reports, profitability analysis etc.

More recently: data-driven discovery

- IT delivers an information platform for creative discovery
- Business explores what questions could be asked

  e.g. brand sentiment, trends in consumer behavior etc.
- Integration of semi-structured and unstructured data
- Use of “big data” technologies (Map reduce, NoSQL etc.)
- Do it all in (near) real-time (not just offline reporting and analytics)
- Integrate BI into data-driven business processes

[BUT: so far, many unrealized promises]

Image of the "Analytics-based Organization"

- Analytics as a necessity for success and survival in the marketplace
- Data-driven business decisions rather than intuition
- Entire business driven by analytics (think Amazon.com), e.g.
  - Demand forecasting
  - Pricing
  - Dynamic display of product recommendations
  - Customer segmentation analysis (or mass customization - segment of one)
  - Campaign management
  - Customer lifetime value analysis
  - A/B tests
  - etc.
- *"Top-performing companies are more likely to use analytics rather than intuition"* [1]

Next level: "advanced analytics" - not only *descriptive*, but also *predictive* and *prescriptive*

Source:
Watson, "The Requirements for Being an Analytics-Based Organization," Business Intelligence Journal, June 2012, pp. 4-6.
[1] LaValle et al (2010): "Analytics: The New Path to Value," *MIT Sloan Management Review*, October.

3 Types of Analytics

1. **Descriptive analytics**
   Display the data that is relevant to the decisions you're looking for, and discard the rest.

2. **Predictive analytics**
   Discover insights in data that are then presented to the decision maker.
   Look for *actionable insights*, and discard the rest

3. **Prescriptive analytics**
   Compute best decisions and present them as recommendations to the decision maker.
   In this case, make sure you are computing the decisions that matter to the decision maker.

Image source: http://www.rosebt.com/uploads/8/1/8/1/8181762/6779091_orig.jpg
https://www.ibm.com/developerworks/community/blogs/jfp/entry/analytics_is_a_mean_to_an_end10?lang=en

“Analytics-based Organization”: Requirements

1. A clear business need
2. Strong, committed sponsorship
3. Alignment between the business and IT strategy
4. A fact-based decision-making culture
5. A strong data infrastructure **[DWH and big data part]**
6. The right analytical tools **[Data mining part]**
7. Personnel with advanced analytical skills

   “Data scientists”, not only IT, but strong background in
   - Mathematics, statistics
   - Operations research/management science
   - Artificial intelligence
   - Econometrics

Source: Watson, "The Requirements for Being an Analytics-Based Organization," Business Intelligence Journal, June 2012, pp. 4-6.

2. DATA WAREHOUSE ARCHITECTURE

Outline

1. **Information Integration**
2. What is a Data Warehouse?
3. Data Warehouse Reference Architecture
4. OLTP vs OLAP
5. Architectural Options

Information Integration: Motivation

In most organizations, data is dispersed geographically and logically among many databases

→ simple aggregate queries are difficult

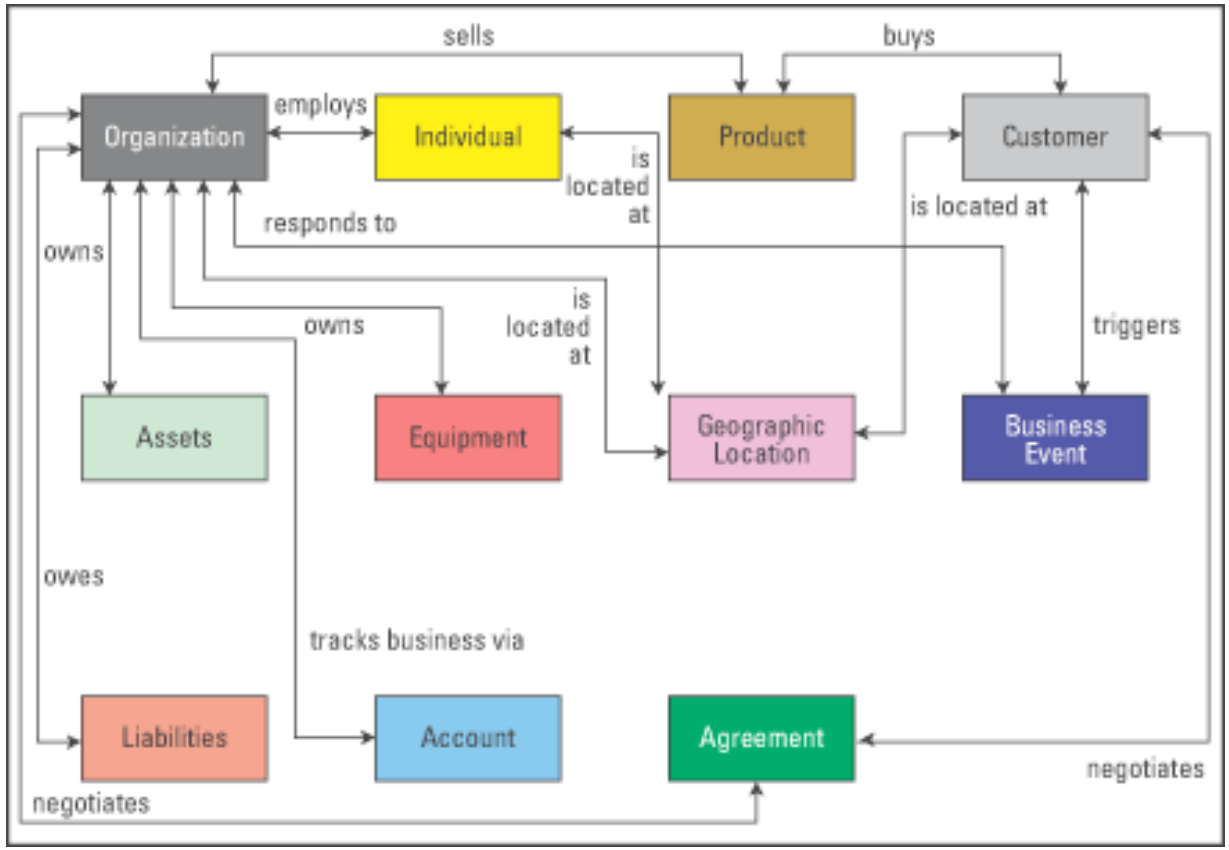
Potential solutions:

- Distributed DB:
  - Distribute queries across databases, create view with union
  - Slow distributed execution

- Centralized DB:
  - Have only one centralized DB used by everyone
  - Slow in operative use
  - Neglects heterogeneous requirements

Both approaches require a single enterprise-wide data model

Theory: Enterprise Data Model (high-level example)

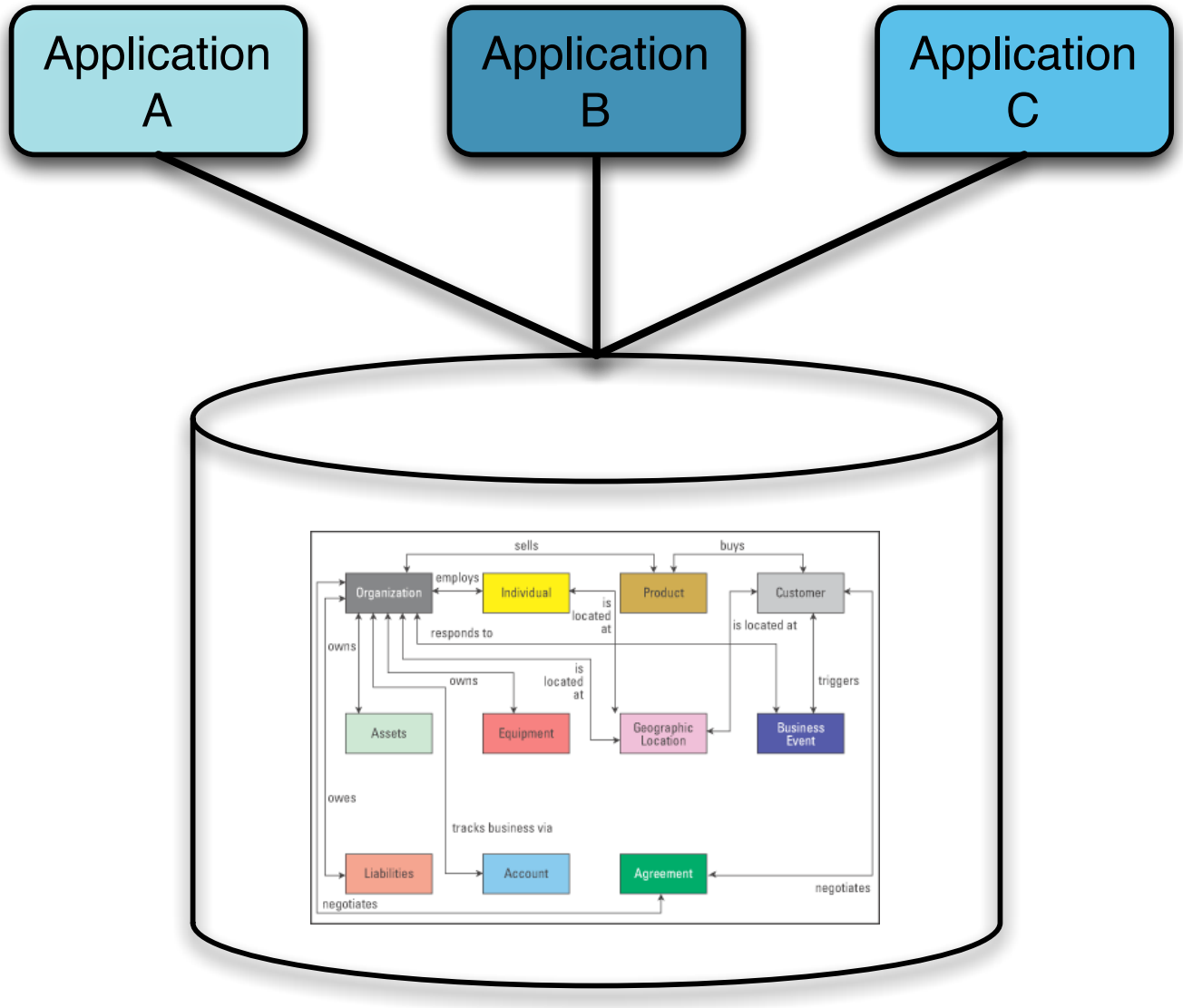


Source: [1]

Adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

[1] http://www.information-management.com/issues/20040101/7910-1.html; 2012-10-19

Theory: One integrated enterprise-wide database



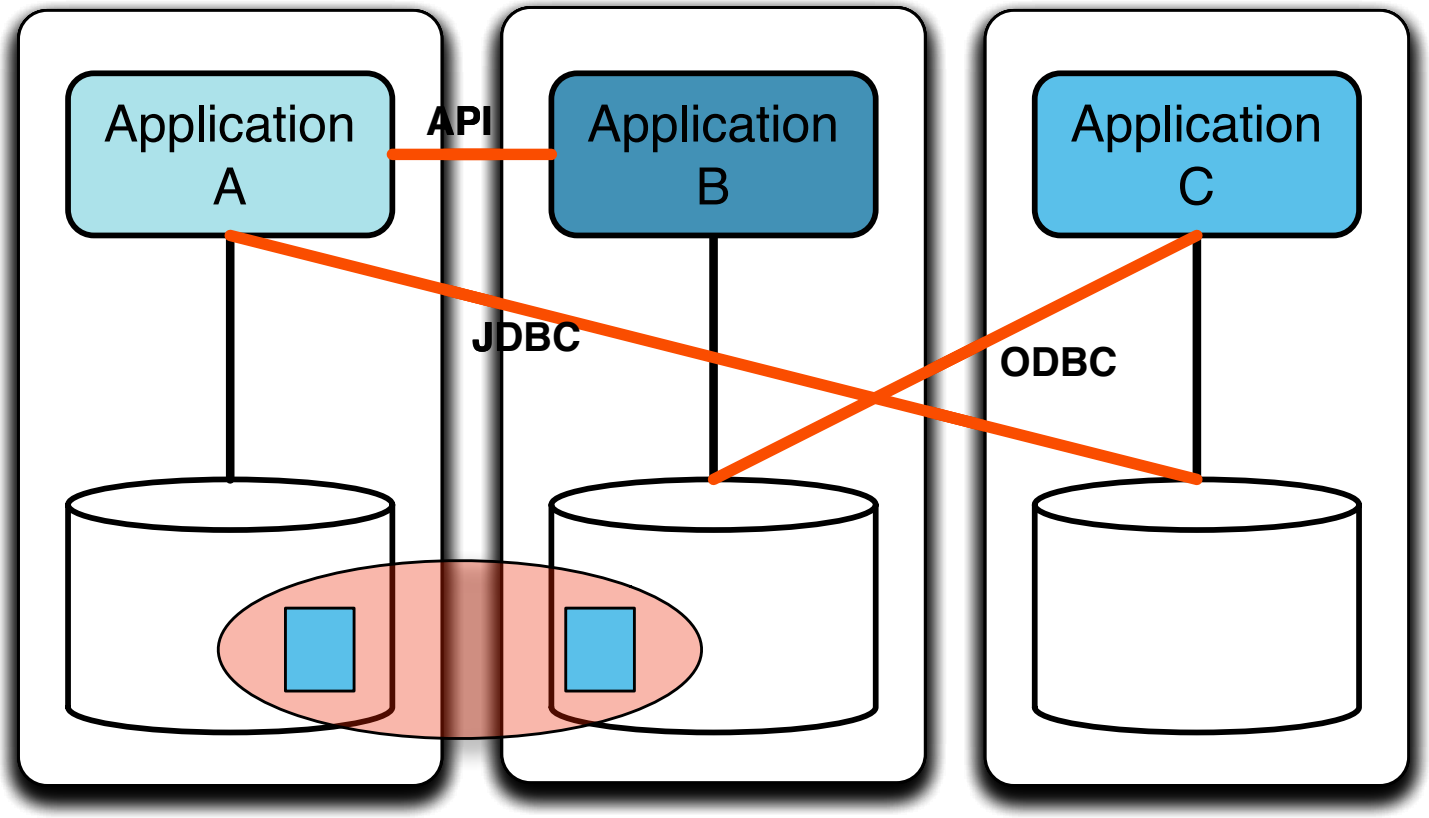
Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Practice: Drawbacks of an EDM approach

- Management is usually not willing to fund this approach:
  - development of EDM is time-consuming and expensive
  - mapping “legacy” data stores to the EDM is even more expensive
  - data migration is again typically orders of magnitude more expensive
- Sections of the EDM developed at the beginning of the project may become obsolete before the entire EDM is “complete”
- Central “control” of an EDM represents a bottleneck (and hurt feelings of “subject matter experts” and “data owners”)
- Interdependencies between existing (“legacy”) applications are often unknown (and undocumented).

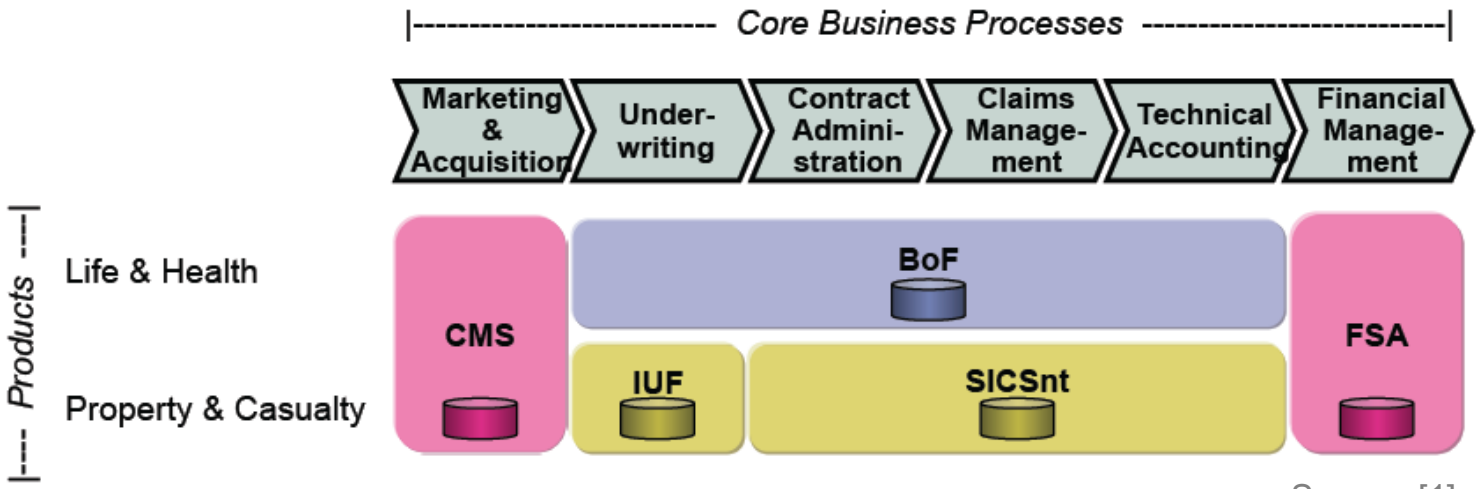
Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Practice: Data Silos



Source: Adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Application Landscape (simplified example)



Source: [1]

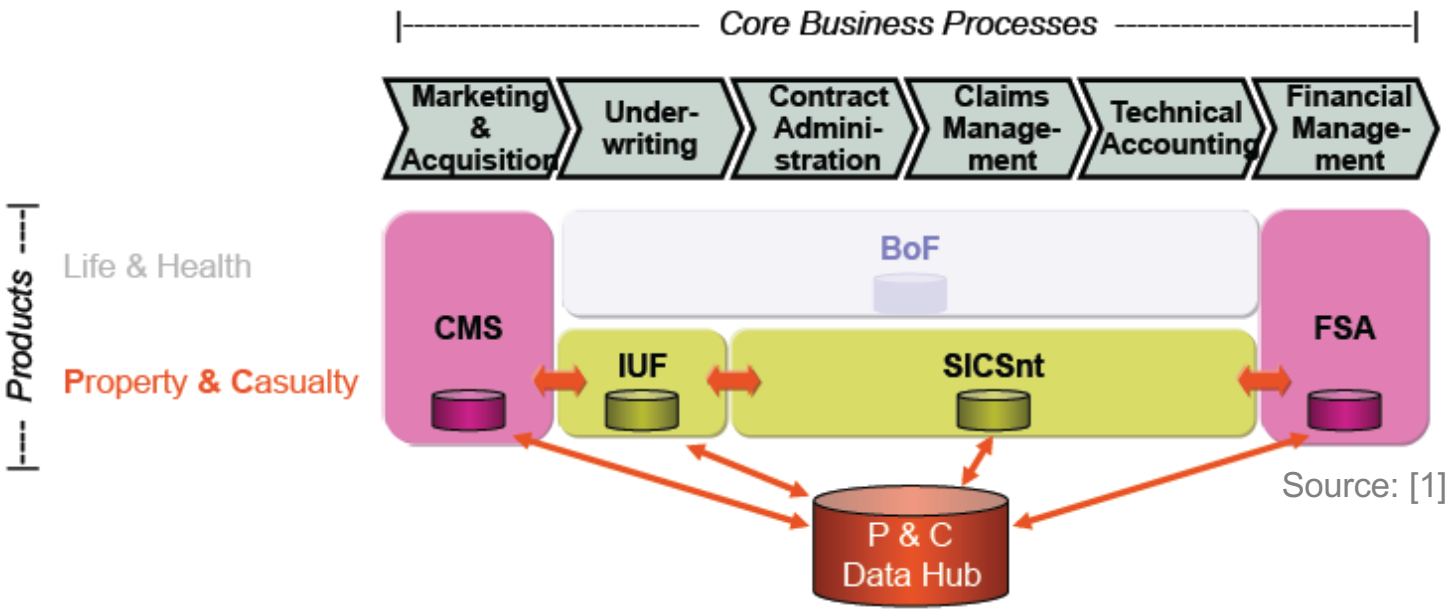
Applications typically only support a subset of org units, locations, products, and/or business processes

→ interdependencies and redundancies in data and processes

→ need for integration and consolidation of data

Source: [1] R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Information Integration Along Business Processes

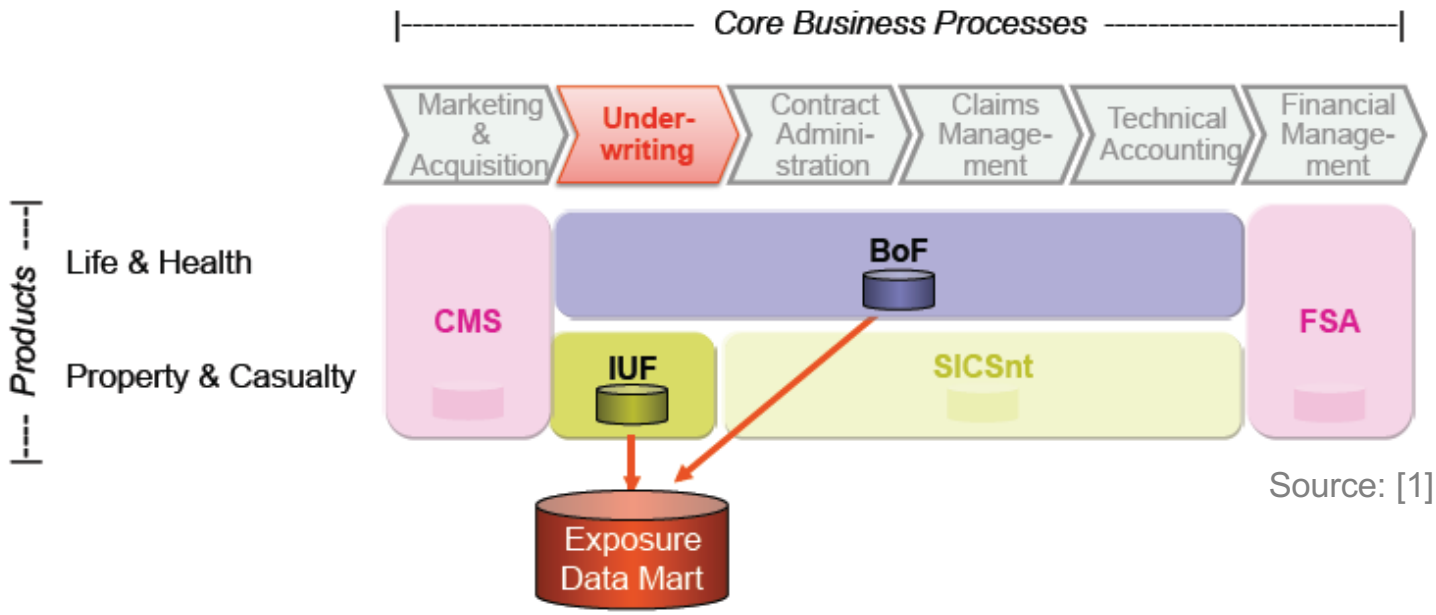


Source: [1]

- Data collected in one business process must be read and updated in another business processes to avoid double entry
- Data collected in different business processes must be combined to get a complete picture across the value chain for decision support & reporting

Source: [1] R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Integrating Across Organizations, Locations, Products



Source: [1]

Data collected in different org units / locations must be combined to get a picture across a larger org unit for decision support & reporting

Source: [1] R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Heterogeneity challenges

1. **Technical heterogeneity**
   - different systems and communication protocols
   - File-based applications (including Excel spreadsheets)
   - “Legacy” mainframe applications using e.g. IMS and CICS
   - 2-tier client/server applications using relational DBMSs
   - Multi-tier applications using e.g. CORBA, EJB, Web application servers etc.
   - Enterprise service bus
   - etc.

2. **Structural and Syntactic heterogeneity**
   - “home-grown” codes vs. ISO encodings for countries and currencies
   - colors specified as enumerations vs. RGB values
   - different units, e.g. *km/h* vs *mph*, *€ vs. $*, etc.
   - synonyms, homonyms; implicit contexts

3. **Semantic heterogeneity:** interpretation?!

Approaches for Information Integration

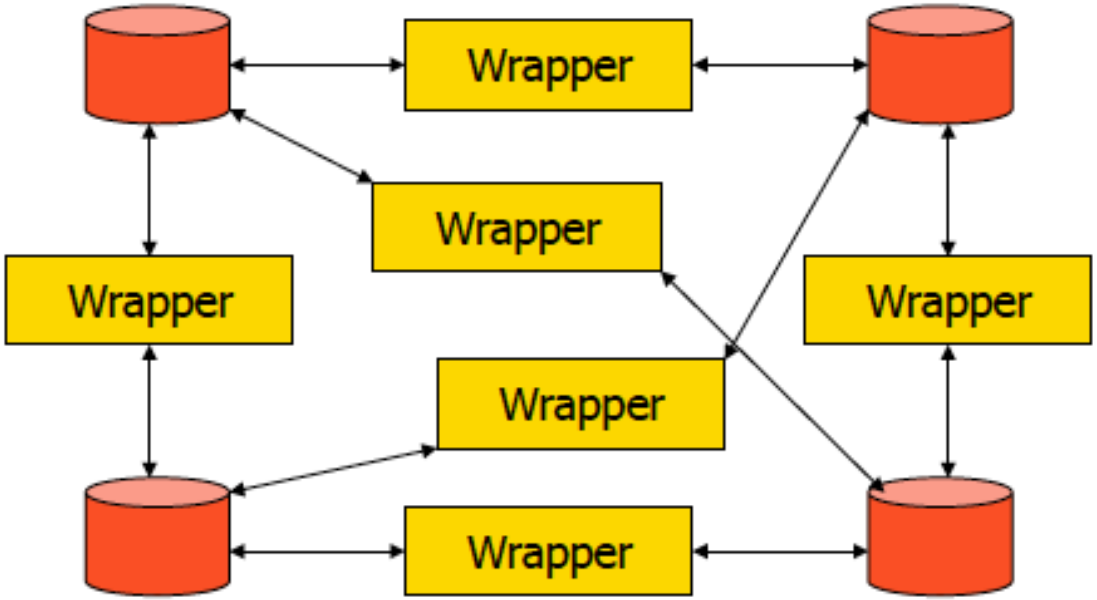
1. **Federation**
   Everybody talks directly to everyone else

2. **Warehouse**
   Sources are translated from their local schema to a global schema and copied to a central DB

3. **Mediator**
   Virtual warehouse – turns a user query into a sequence of source queries and assembles the results of this queries into an “aggregate” result

Source: [1] R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Integration Approach 1: Federation



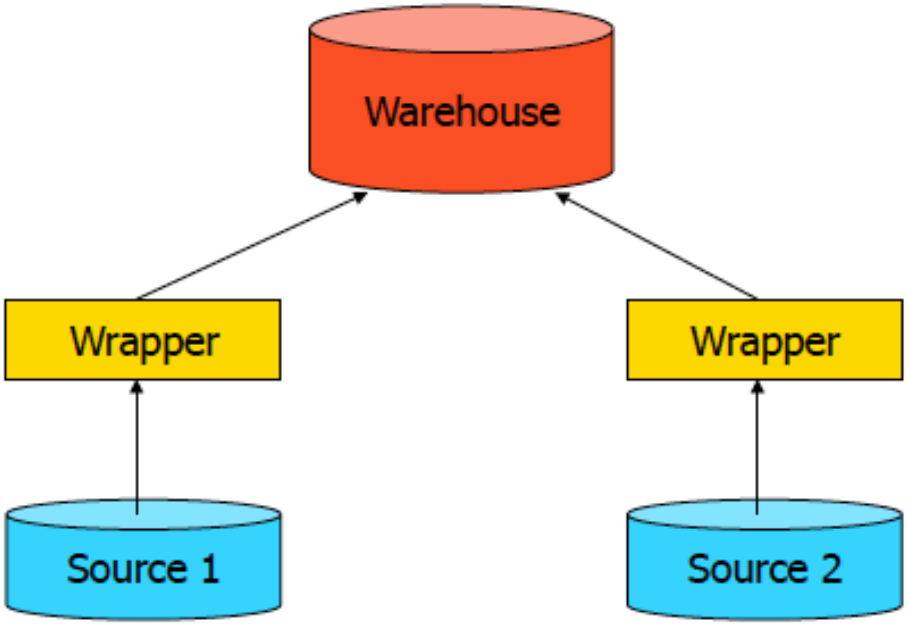
Source: [1]

Issue:

n applications / data stores → up to n^2 connections

Source: [1] R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Integration Approach 2: Warehouse



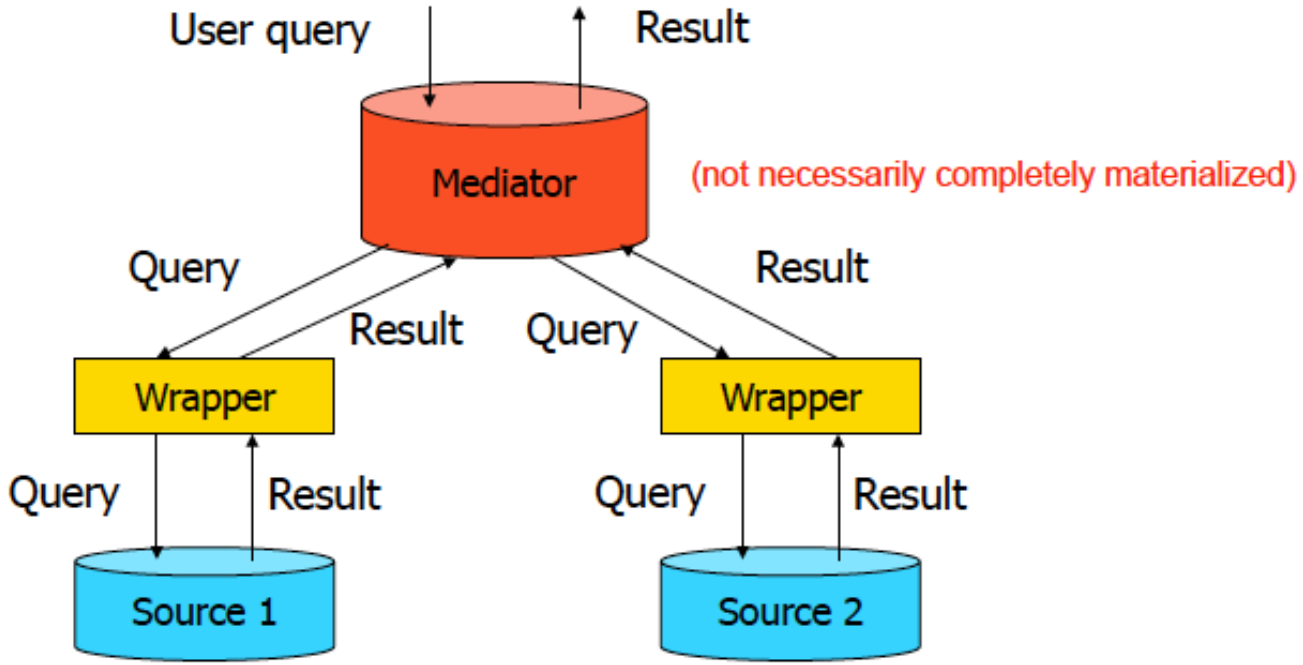
Source: [1]

Issue:

usually only unidirectional data flows supported

Source: [1] R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Integration Approach 3: Mediator



Source: [1]

Issue:
complex architecture, potentially slow, difficult to maintain

Source: [1] R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Outline

1. Information Integration
2. **What is a Data Warehouse?**
3. Data Warehouse Reference Architecture
4. OLTP vs OLAP
5. Architectural Options

What is a “Data Warehouse”? - Definitions

„A data warehouse is a copy of transaction data specifically structured for query and analysis.“

Kimball (1996)

Source:
Kimball, R. (1996): The Date Warehouse Toolkit, 1ed, Wiley

What is a “Data Warehouse”? - Definitions

„A data warehouse is a

- ***subject-oriented,***
- ***integrated,***
- ***time-variant,***
- ***nonvolatile*** *collection of data*

in support of management’s decision-making process.“

Inmon (1996)

Source:
Inmon, W.H., (1996): Building the Data Warehouse. John Wiley & Sons. P. 36

What is a “Data Warehouse”? - Definitions

“A storage architecture designed to hold data extracted from transaction systems, operational data stores and external sources.

The warehouse then combines that data in an aggregate, summary form suitable for enterprisewide data analysis and reporting for predefined business needs.”

Gartner IT Glossary

Source: http://www.gartner.com/technology/it-glossary/#3_0 (accessed November 2, 2011)

Data Warehousing – Typical use cases

Retail
- Customer loyalty
- Market planning

Manufacturing
- Cost reduction
- Logistics management

Finance
- Risk management
- Fraud detection

Utilities
- Asset management
- Resource management

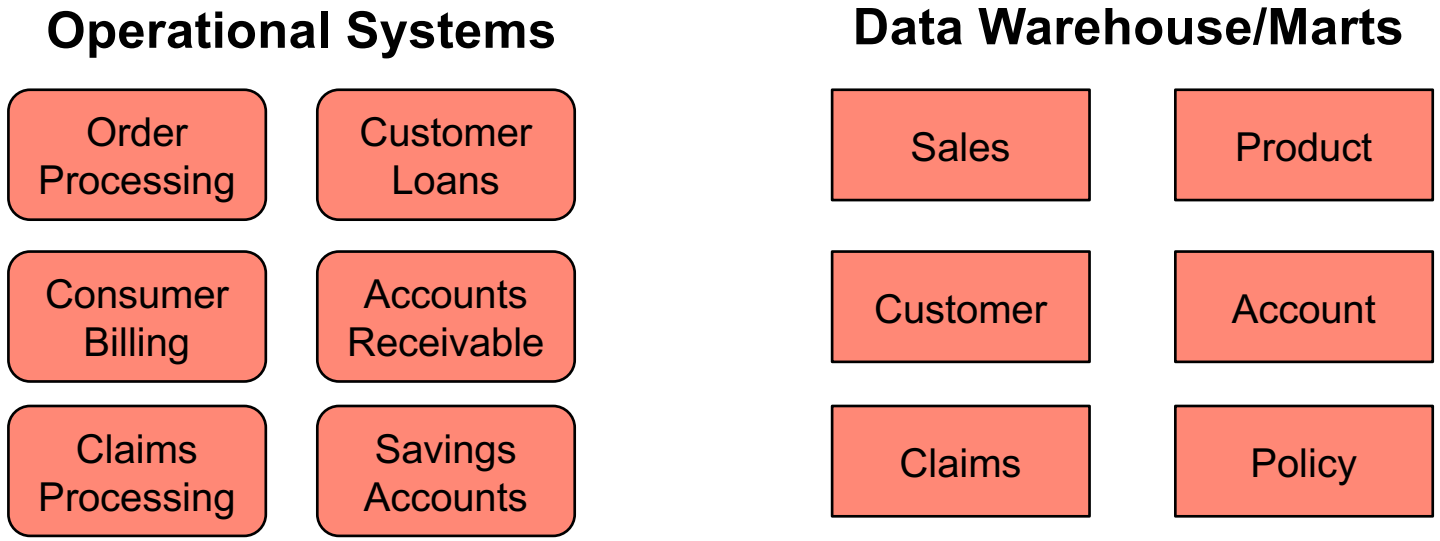
Airlines
- Route profitability
- Yield management

Government
- Manpower planning
- Cost control

Source:
Ponniah, P.: Data Warehousing Fundamentals, 2ed. Wiley (2010)

Key Data Warehouse Characteristics: Subject-orientation

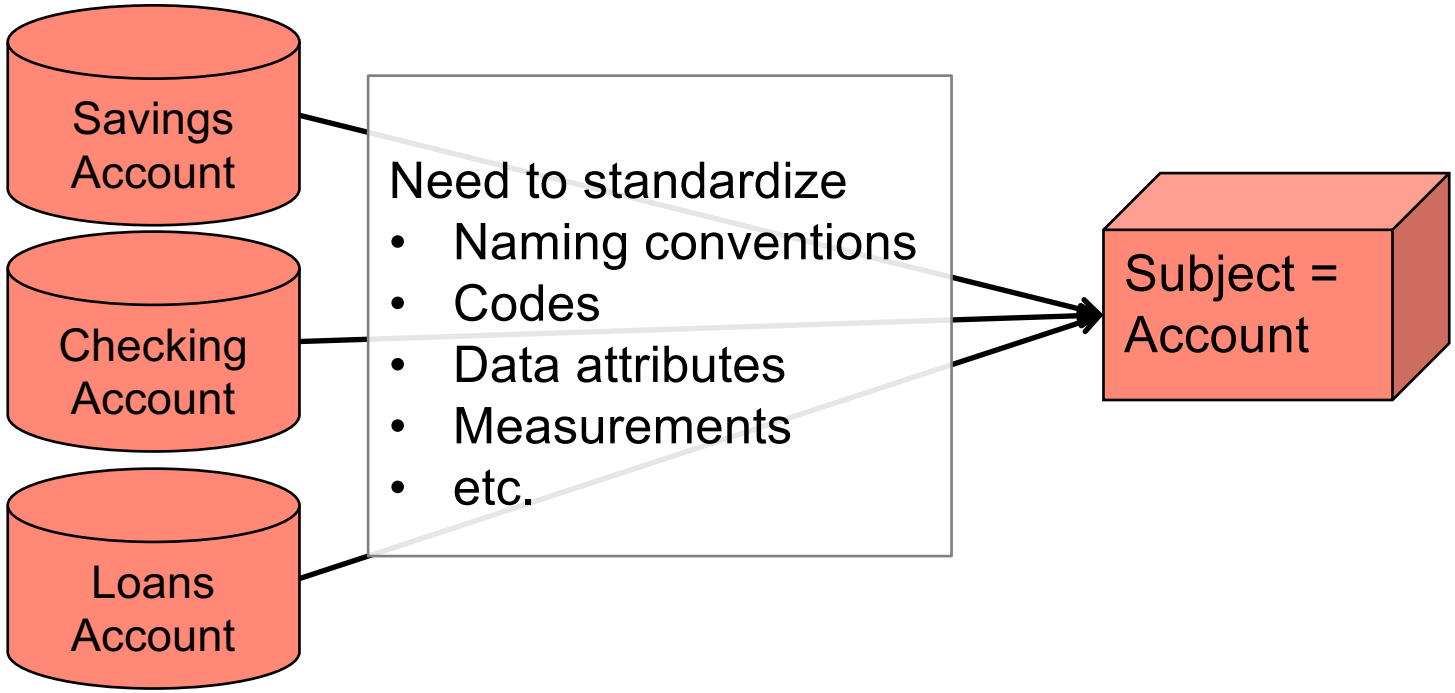
- In the DW, data is not stored by operational applications, but by business subjects
- Data is grouped around subjects, and its structure is designed to make querying the data simple, especially for business analysts.



Source:
Ponniah, P.: Data Warehousing Fundamentals, 2ed. Wiley (2010)

Key Data Warehouse Characteristics: Integration

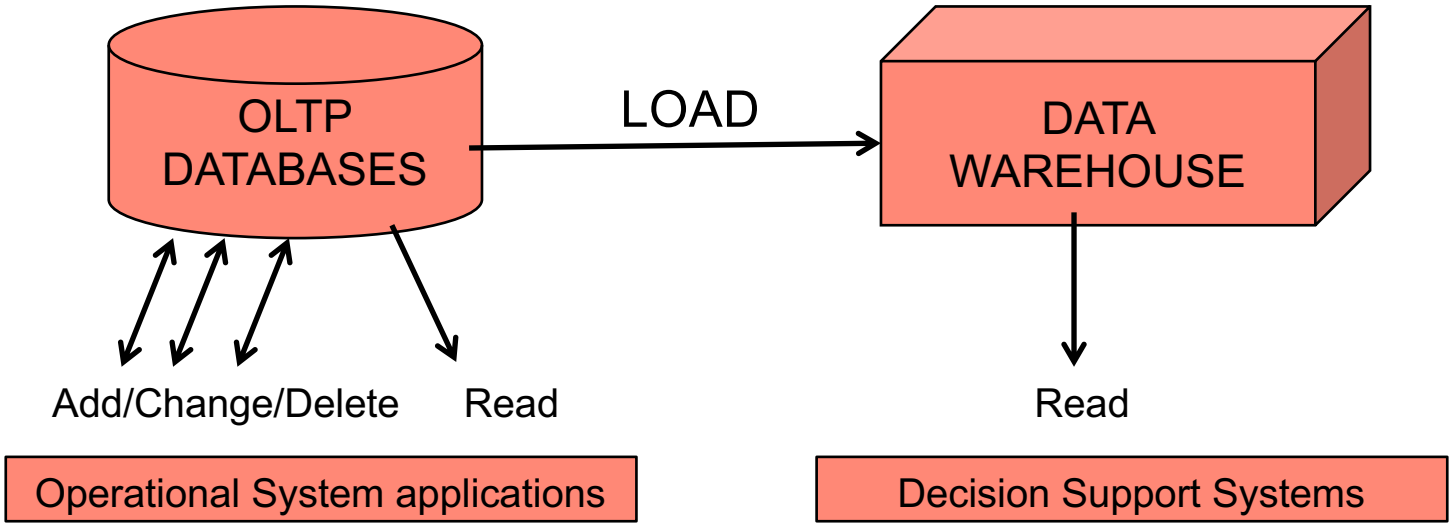
- DW contains consolidated data from several applications/their databases
- Usually the data in the DW is not updated or deleted
- Data inconsistencies are removed; data from diverse operational applications is integrated.



Source:
Ponniah, P.: Data Warehousing Fundamentals, 2ed. Wiley (2010)

Key Data Warehouse Characteristics: Non-volatile and time-variant

- The data stored in operational systems contains the **current** values.
- DW: When new current data becomes available, the “old” data is not overwritten.
- A data warehouse has to contain historical data to
  - Allow for analysis of the past
  - Relate information to the present
  - Enable forecasts for the future

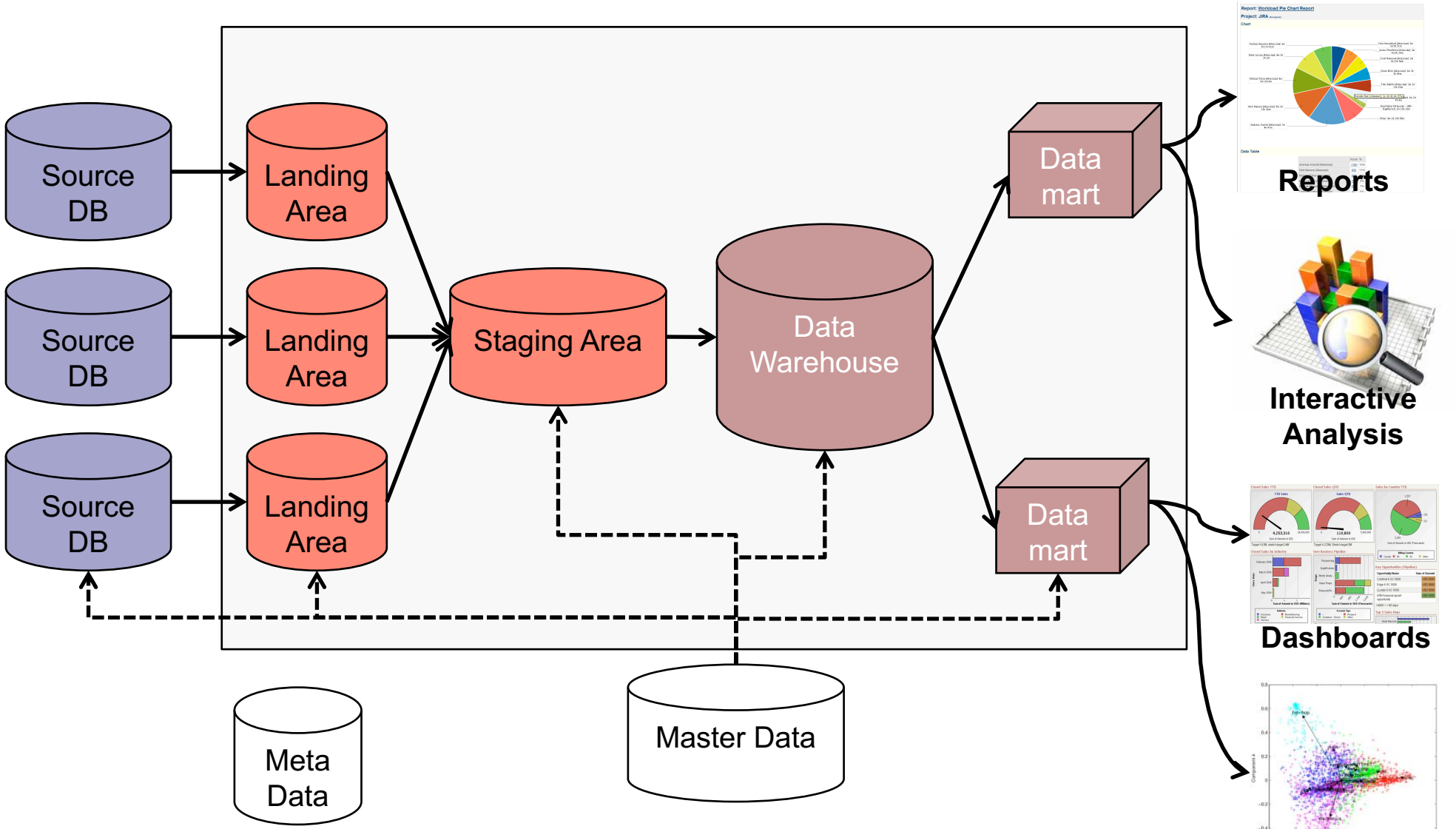


Source:
Ponniah, P.: Data Warehousing Fundamentals, 2ed. Wiley (2010)

Outline

1. Information Integration
2. What is a Data Warehouse?
3. **Data Warehouse Reference Architecture**
4. OLTP vs OLAP
5. Architectural Options

Data Warehouse Reference Architecture



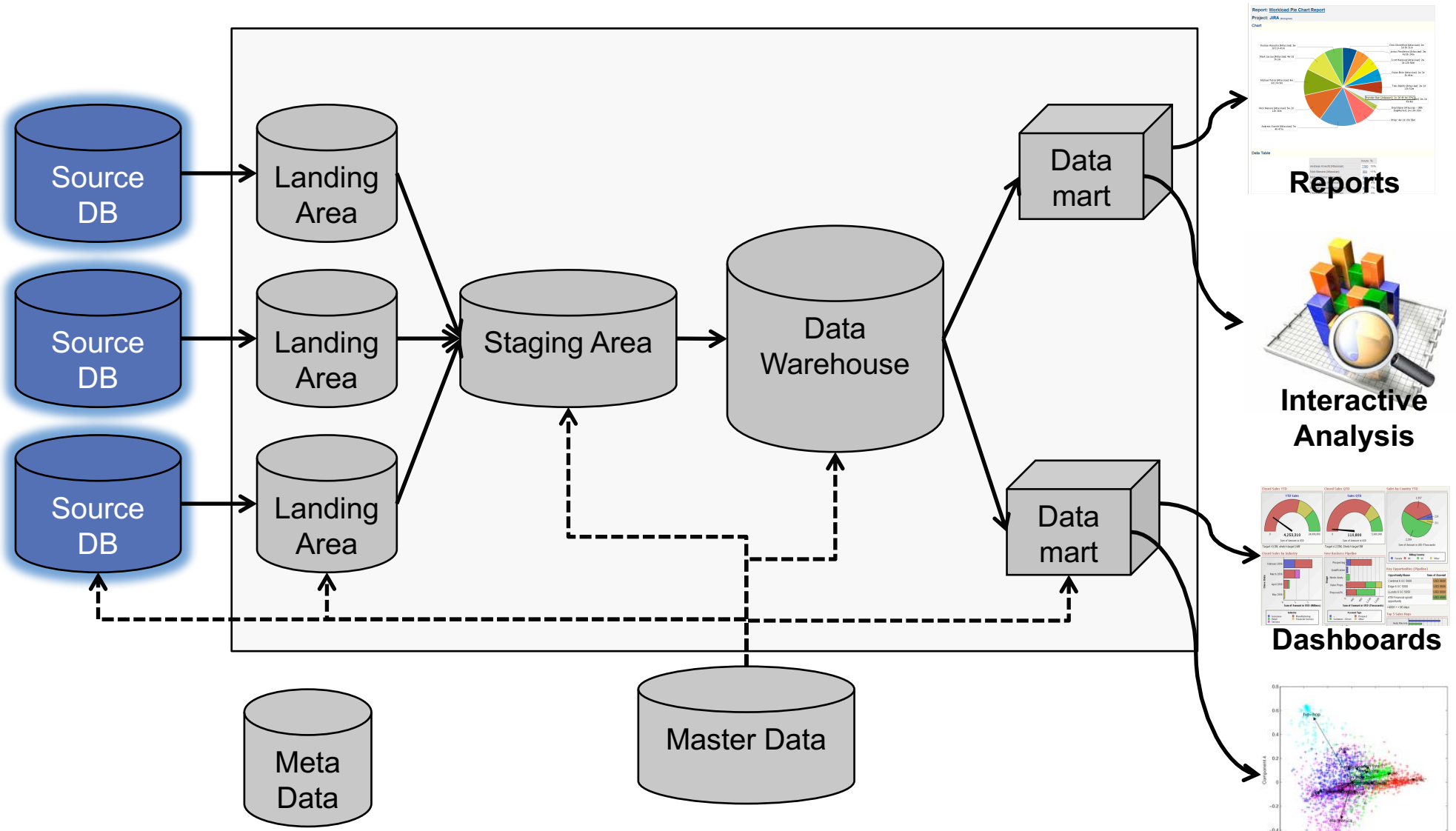
Source: R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Terminology

- **Source DB** – also: Operational Application, OLTP Application
  DB of an application which supports one or more types of business transactions.
- **Landing Area (LA)**
  DB that is able to store a single data extract of a subset of one Source DB. Its schema basically corresponds 1:1 with the schema of the subset of the Source DB.
- **Staging Area (SA)**
  DB that is able to store matching data extracts from various Landing Areas in an integrated format, waiting for the upload to the DW once data from all Landing Areas are available. Its schema basically corresponds 1:1 with the DW schema.
- **Data Warehouse (DW)**
  DB containing the history of all complete Staging Areas. Its integrated schema is frequently still more or less in Third Normal Form (3NF).
  Note: 3NF is a slight contradiction to the criterion “subject-orientation”
- **Data Mart (DM) – also: OLAP Application**
  DB – on disk or in main memory – containing data describing the (present and past) performance of one or more types of business transactions, taken form the DW. The schema of a Data Mart often has the form of one or more denormalized “stars”.

Source: R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Data Warehouse Reference Architecture



Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

OLTP = Online Transaction Processing

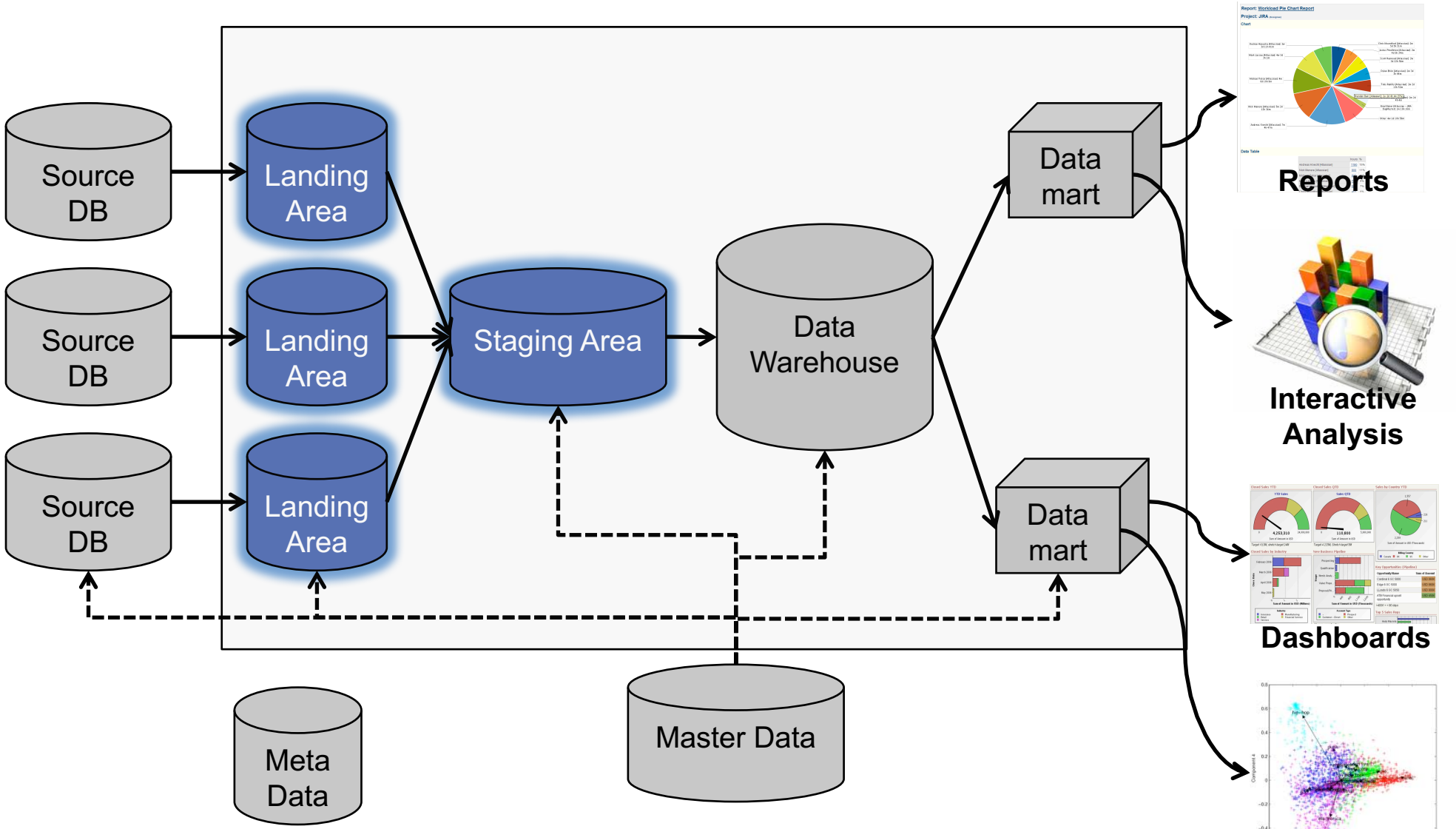
Examples:

- Take an order
- Process a claim
- Make a shipment
- Generate an invoice
- Receive cash
- Reserve an airline seat

Requirements:

- Optimize for many short and "small" transactions: point queries, single-row **update**s and/or **insert**s
- Access to up-to-date, consistent DB
- Avoid (uncontrolled) redundancies → use normalized schemas

Data Warehouse Reference Architecture



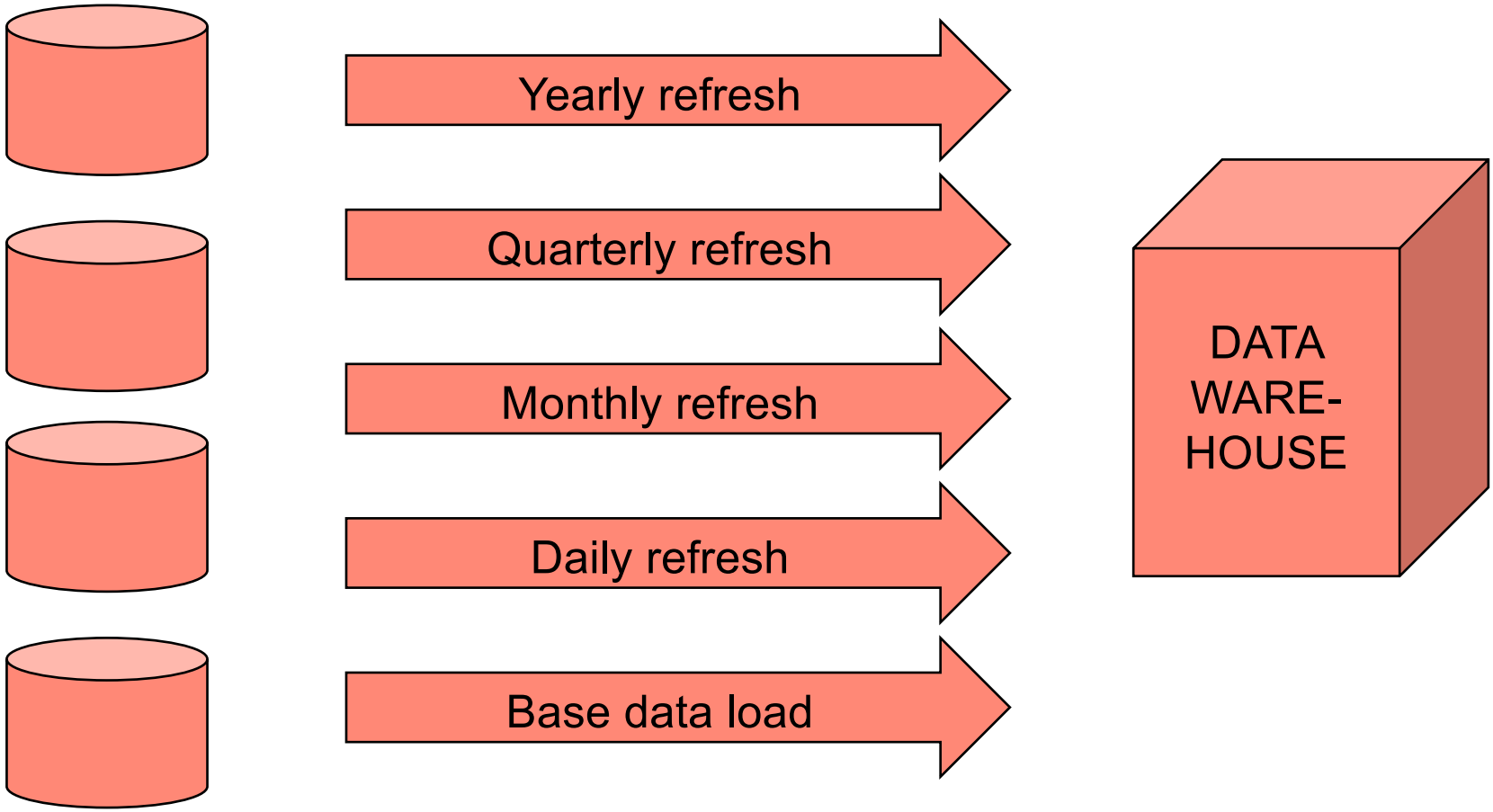
Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Data Staging/ETL

- **E**xtraction – extract data from numerous data sources
- **T**ransformation – perform a number of individual tasks:
  - Cleaning
  - Standardization
  - Combining pieces of data from different sources
- **L**oading – two distinct groups of tasks
  - The initial load moves large volumes of data into the data warehouse storage
  - Feed the incremental data revisions on an ongoing basis.

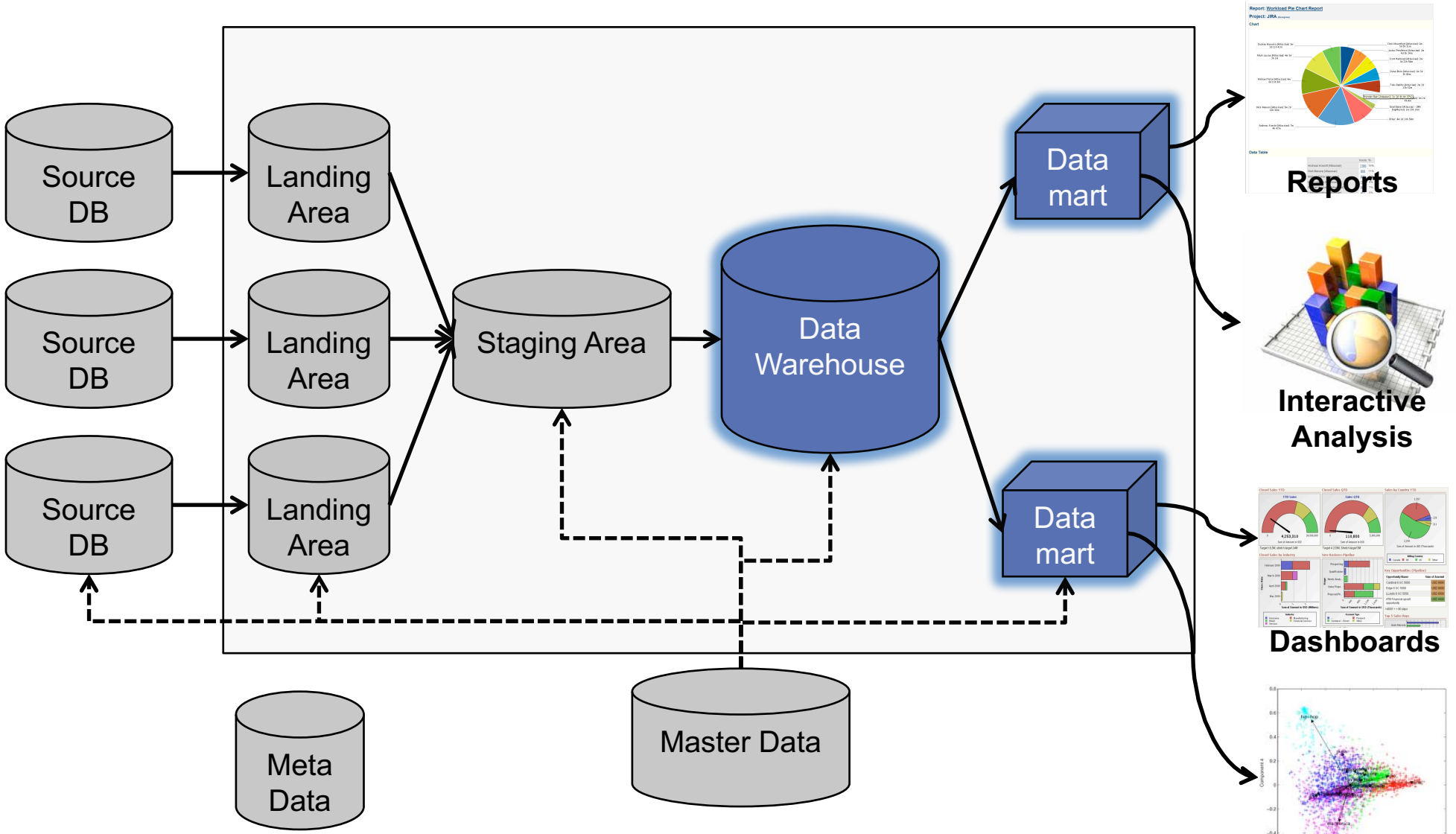
Source: Ponniah, P.: Data Warehousing Fundamentals, 2ed. Wiley (2010)

Data Movements to the DW



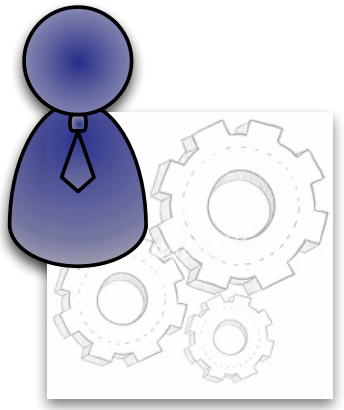
Source: Ponniah, P.: Data Warehousing Fundamentals, 2ed. Wiley (2010)

Data Warehouse Reference Architecture



Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

OLAP Applications:
"Watching the wheels of business turn"



- OLAP = Online Analytical Processing
- Analyzing the data describing business transactions
- Objective:
  turn raw (transactional) data into strategic/tactical/operative information

Examples:

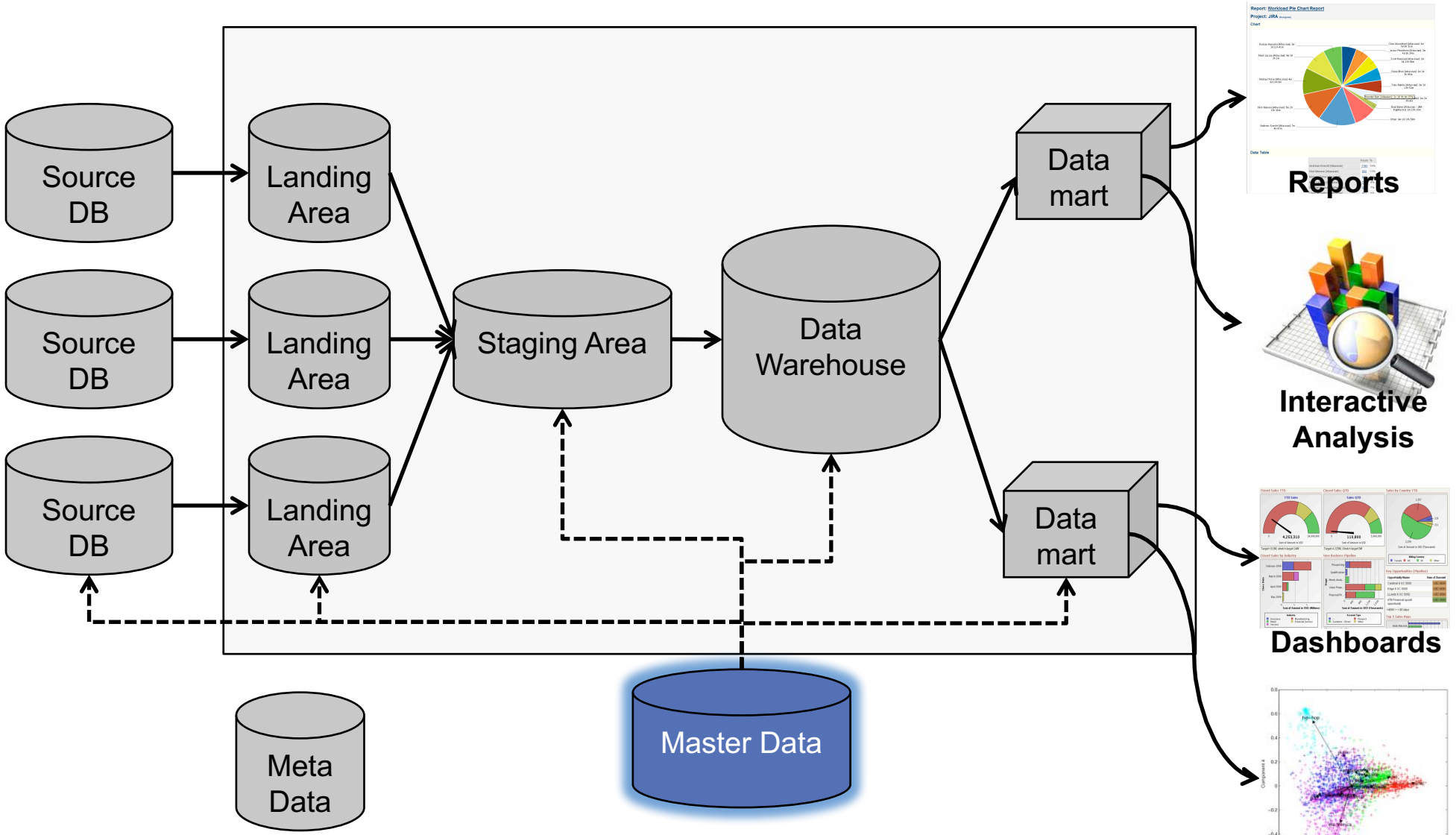
- Management Information Systems (MIS)
- Decision Support Systems (DSS)
- Statistical Databases

Requirements:

- Queries with large result sets (all the data, joins)
- No immediate **update**s and/or **insert**s,
  but large periodic (daily, weekly) batch **insert**s
- (Controlled) redundancy a necessity for performance reasons:
  denormalized schemas, materialized views; indexes
- Goal: Response Time of seconds / a few minutes

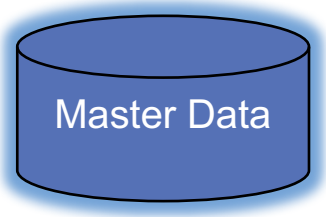
Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Data Warehouse Reference Architecture



Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Master Data Stores



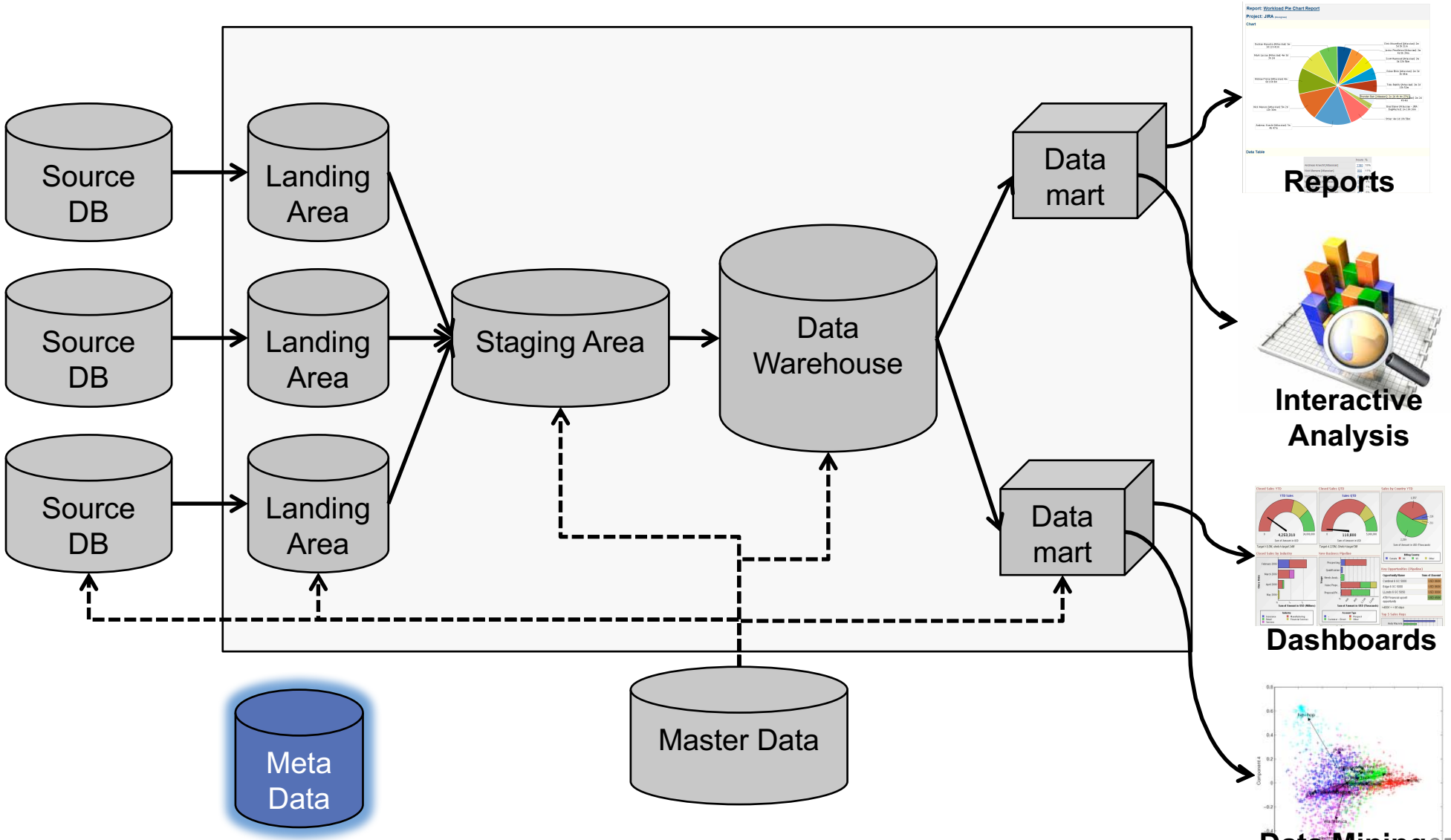
„Factored out“ information required by many applications (OLTP and OLAP)

- serves to establish the context of data collected in business transactions
- typically ends up in the dimensional tables of star schema
- may be „owned“ by someone external to the enterprise (countries, currencies etc.)
- may be „owned“ internally (e.g., profit centers, categorizations of clients, etc.)

„Structured“ master data:
customers, vendors, maybe products, categorization attributes / codes

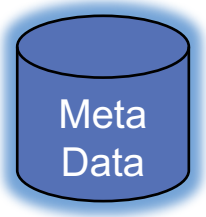
Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Data Warehouse Reference Architecture



Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Metadata Component



- Similar to the data dictionary or data catalog in DBMS
- Data about data in the DWH
- Necessary for..
  - **Using**
    users need to know about the available data in the DWH,
  - **Building**
    source systems, source-to-target mappings, transformation rules etc.
  - **Administering**
    complexity, size → impossible to administer without metadata

.. a DWH

Three types:

1. Operational metadata
2. Extraction and transformation metadata
3. End-user metadata

Source:
Ponniah, P.: Data Warehousing Fundamentals, 2ed. Wiley (2010)

Outline

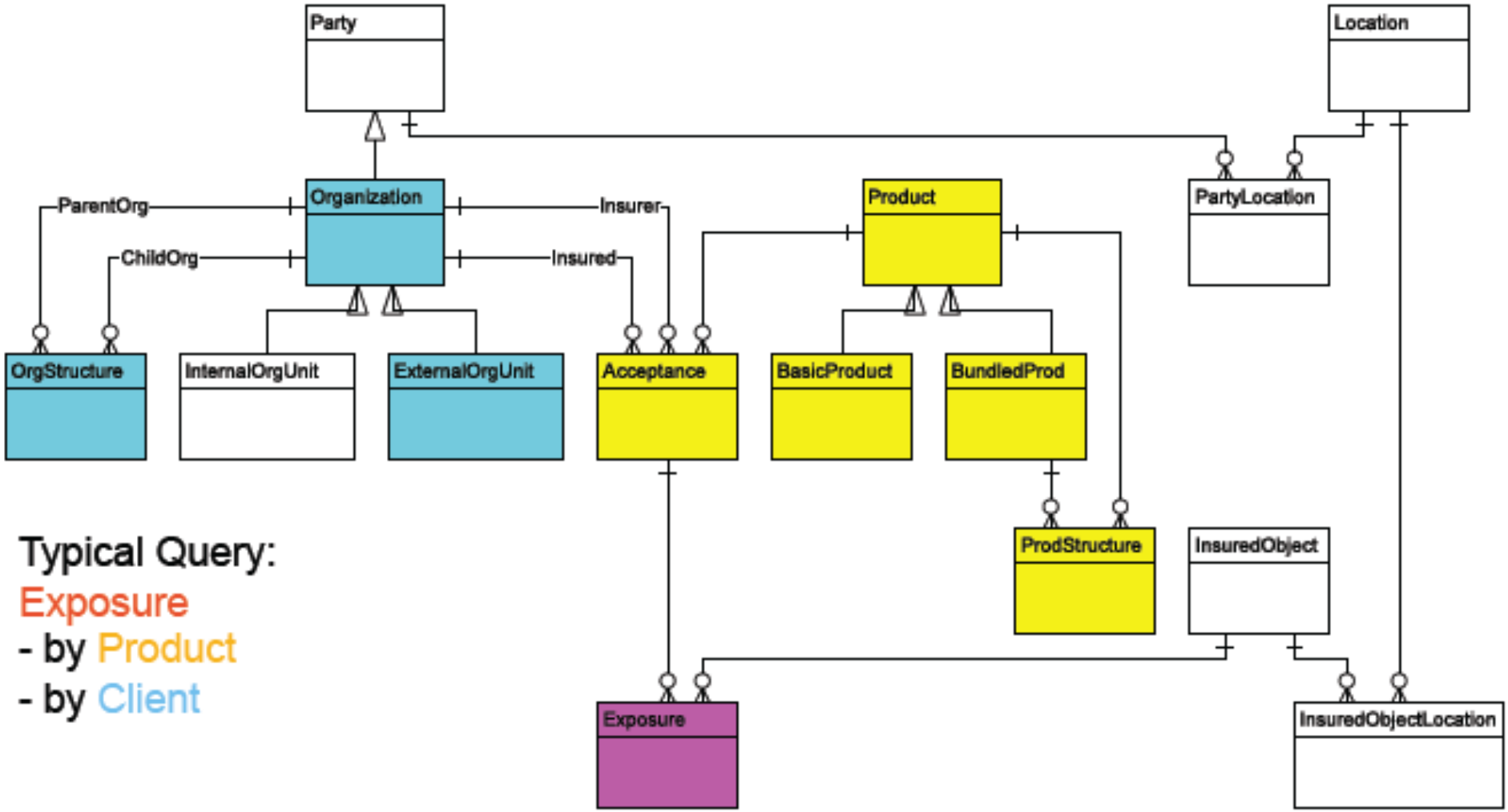
1. Information Integration
2. What is a Data Warehouse?
3. Data Warehouse Reference Architecture
4. **OLTP vs OLAP**
5. Top-down vs bottom-up approach

OLTP vs OLAP – How are they different?

	Operational	Informational
Data Content	Current values	Archived, derived, summarized
Data Structure	Optimized for transactions	Optimized for complex queries
Access Frequency	High	Medium to low
Access Type	Read, update, delete	Read
Usage	Predictable, repetitive	Ad-hoc, random, heuristic
Response Time	Sub-seconds	Several seconds to minutes
Users	Large number	Relatively small number

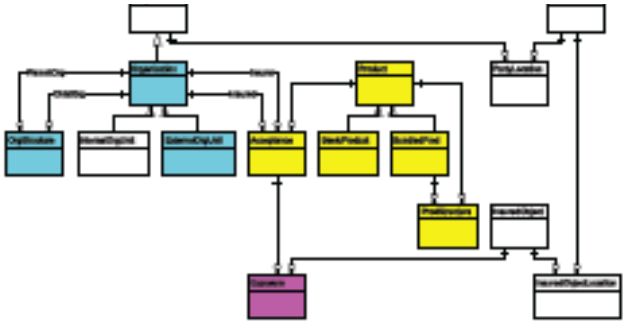
Source:
Ponniah, P.: Data Warehousing Fundamentals, 2ed. Wiley (2010)

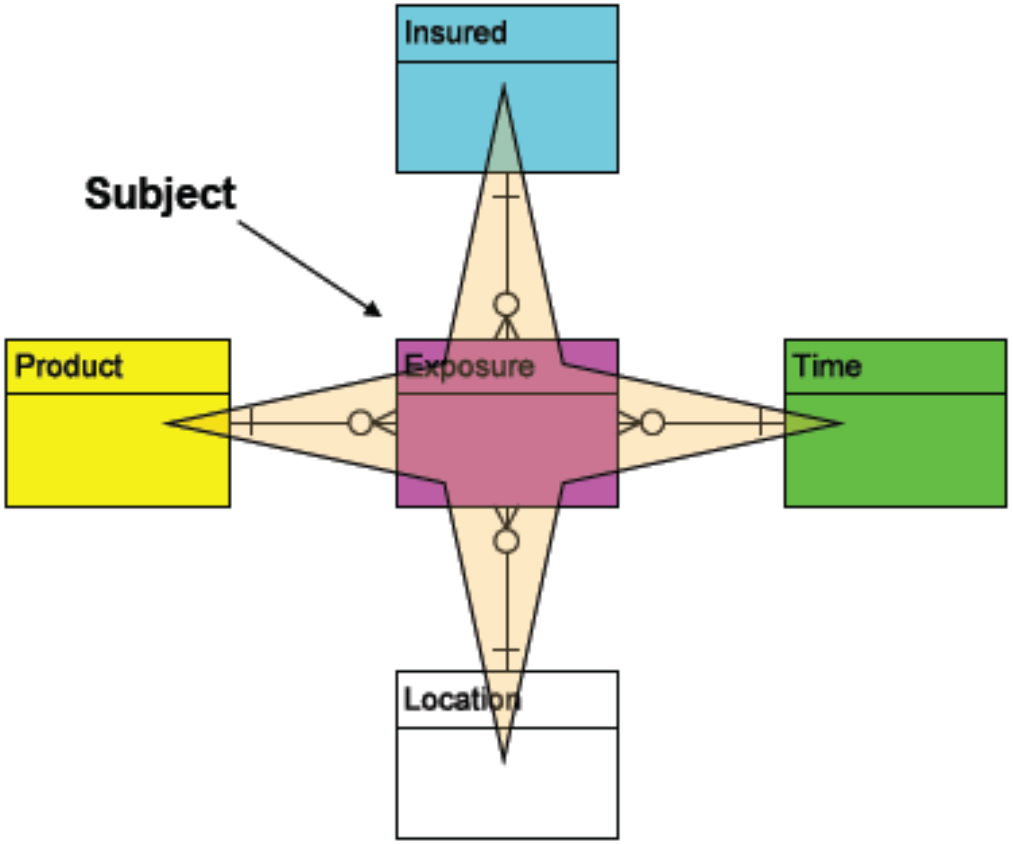
Normalized Schema of Operational DB (Example)



Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Subject-orientation: Star Schema used in Data Marts





Typical Query:
Exposure
- by Product
- by Client

- by Time

Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

OLTP vs. OLAP: Water and Oil

- Lock Conflicts
  - long-running OLAP reads may block OLTP writes
- Freshness of data
  - OLTP: up-to-date data → serializability
  - OLAP: reproducibility of analyses → historization
- Precision
  - OLTP: (usually) exact
  - OLAP: sampling, statistical summaries, confidence intervals

Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

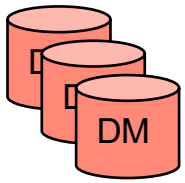
Outline

1. Information Integration
2. What is a Data Warehouse?
3. Data Warehouse Reference Architecture
4. OLTP vs OLAP
5. **Architectural Options**

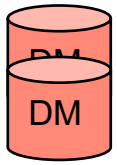
Typical Architectural Options

Single stand-alone data mart
OK if there is no need for integration

- separation of OLAP solution from a single integrated OLTP system
- local reporting / OLAP solution for (non-integrated) OLTP system

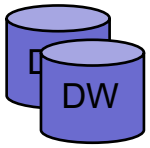


Several independent stand-alone data marts
Usually started as „quick and dirty“ solution, soon to be replaced by a more integrated solution.



A conformed constellation of data marts
(conforming = sharing common dimensions)

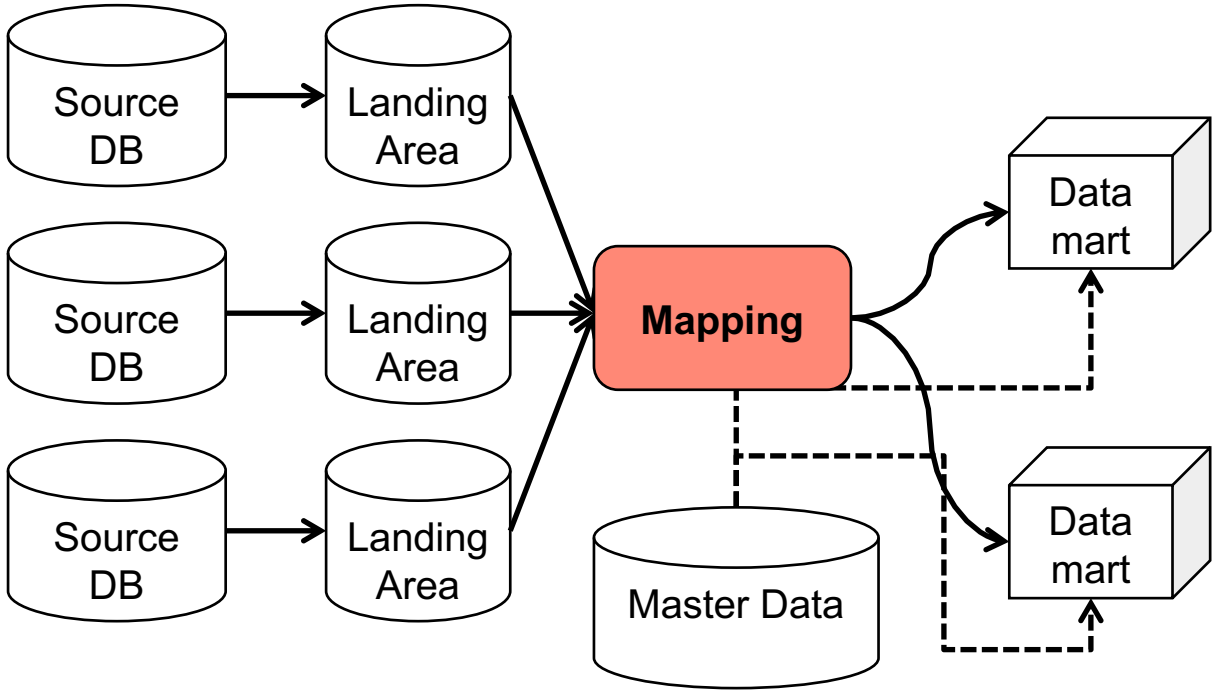
A single Integrated Data Warehouse
(Corporate Information Factory) feeding all data marts



Multiple data warehouses for different functions
Highly problematic, but not uncommon in large corporations

Source:R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Conformed Constellation of Data Marts (Bus Architecture)

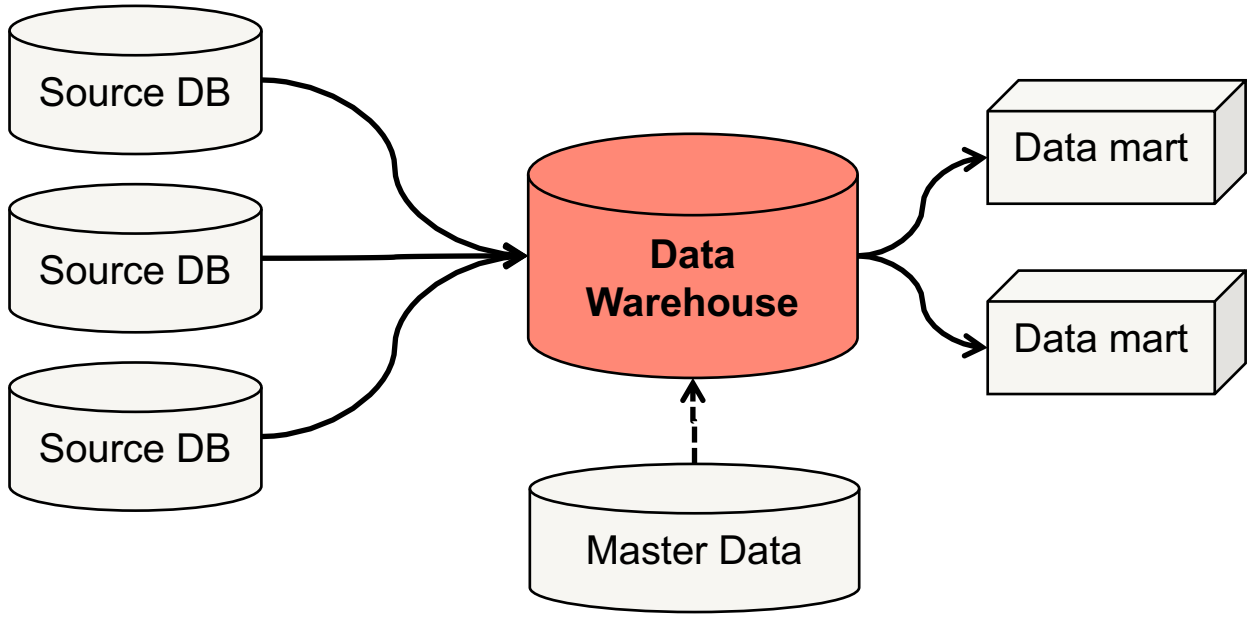


Characteristics

- No “explicit” DW, no common schema
- DW bus: conformed facts & dimensions
- Common mapping to dimensions through Master Data store(s)

Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Integrated Data Warehouse Architecture

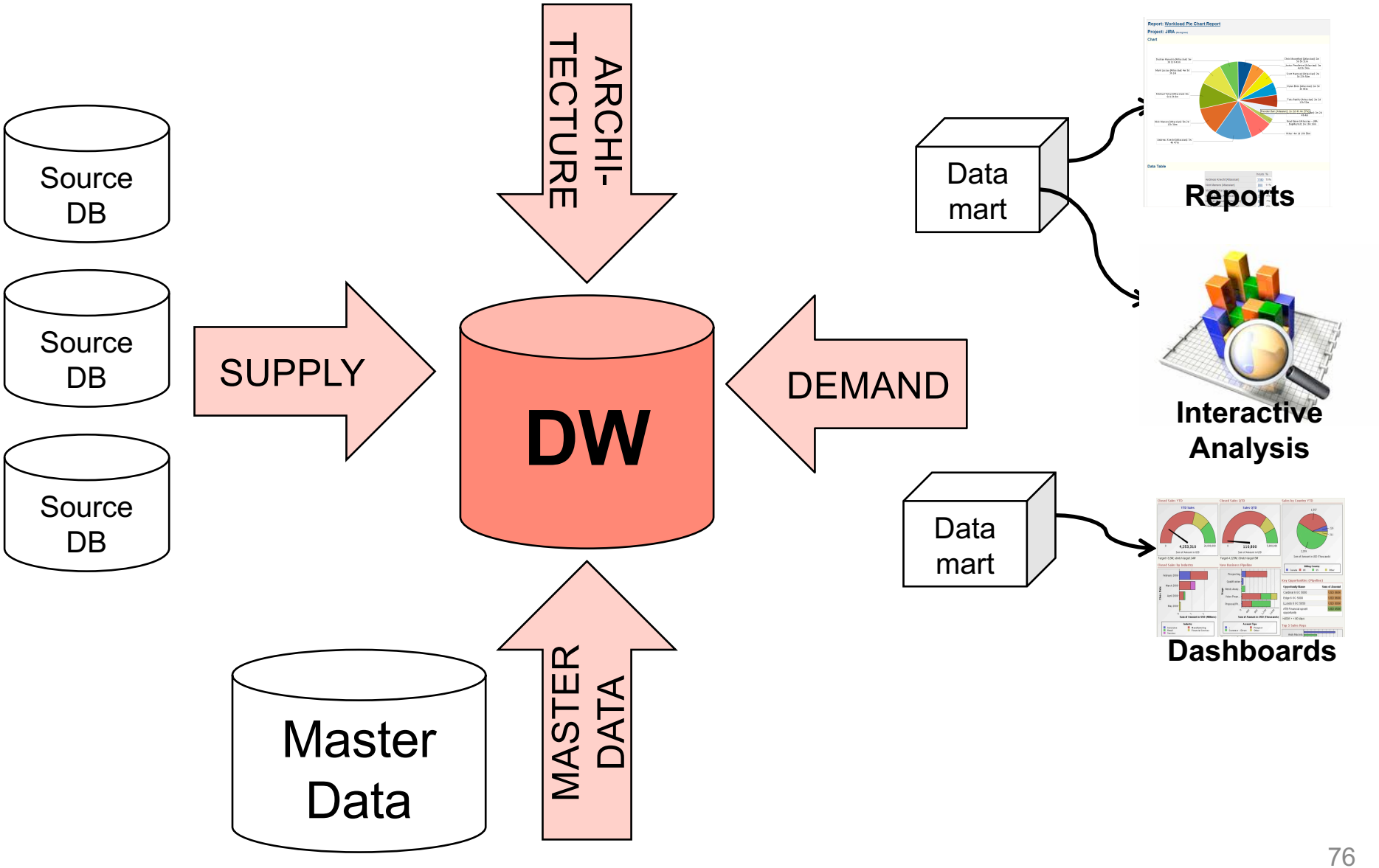


Characteristics:

- Common, integrated schema
- All Data Marts fed from the integrated Data Warehouse
- Master Data can be viewed as “just another Source DB”.
- Loading: typically one landing area per Source DB + integrated Staging Area.

Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

DW Schema Design - Forces



Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Top-down vs. bottom-up approach – Kimball vs Inmon models

Approach determines both architecture and development methodology

Top-down approach – Inmon model

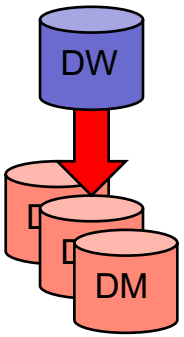
- Deductive
- Generic → specific
- Build centralized DW ("Corporate Information Factory") first, then departmental DMs to accommodate the requirements of various user groups

Bottom-up approach – Kimball model

- Inductive
- Specific → generic
- One DM per major business process rather than building a single centralized DW first
- Enterprisewide cohesion through data bus

Breslin, M.: Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models
Business Intelligence Journal, Winter 2004

Data Warehouses – Top-down Approach (deductive)

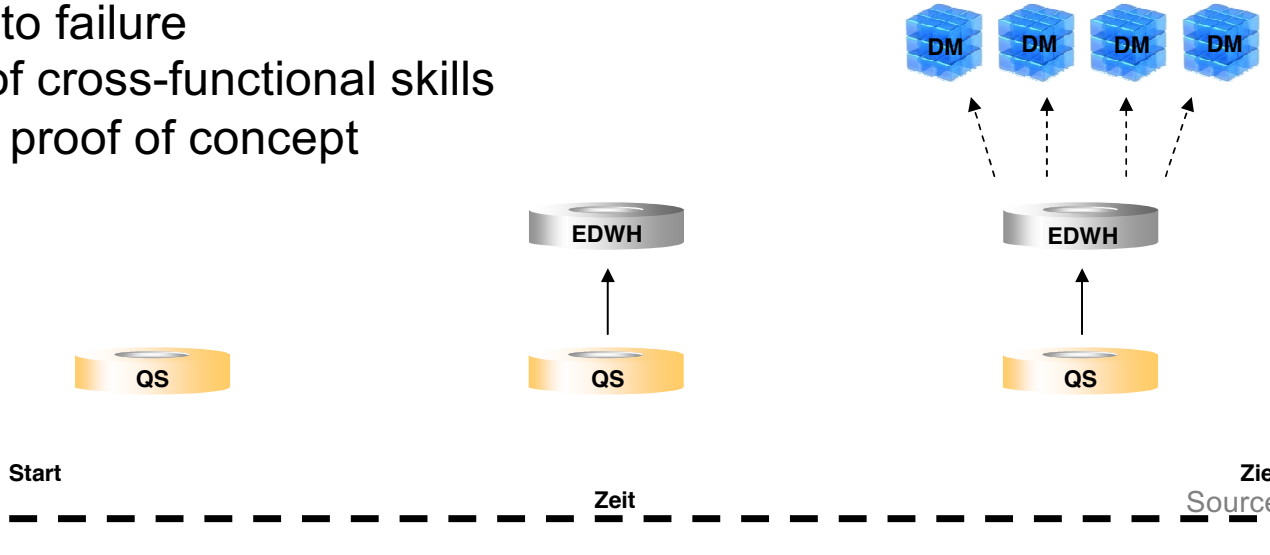


Advantages

+ Truly corporate effort, enterprise view of data
+ Inherently architectonic – not a union of disparate data marts
+ Single, central storage of data about the content
+ Centralized rules and control
+ May see quick results if implemented in iterations

Disadvantages

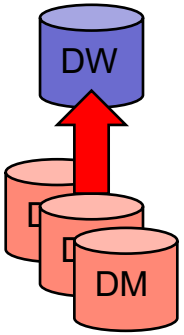
– Takes longer to build even with an iterative method
– High exposure/risk to failure
– Needs high levels of cross-functional skills
– High outlay without proof of concept



Source: [1]

Sources:
Ponniah, P.: Data Warehousing Fundamentals, 2ed. Wiley (2010)
[1] Lusti: Data Warehousing and Data Mining as cited by Melchard 2008: Lecture slides “Data Warehousing and Data Mining”; WU Vienna

Data Marts – Bottom-Up Approach (inductive)

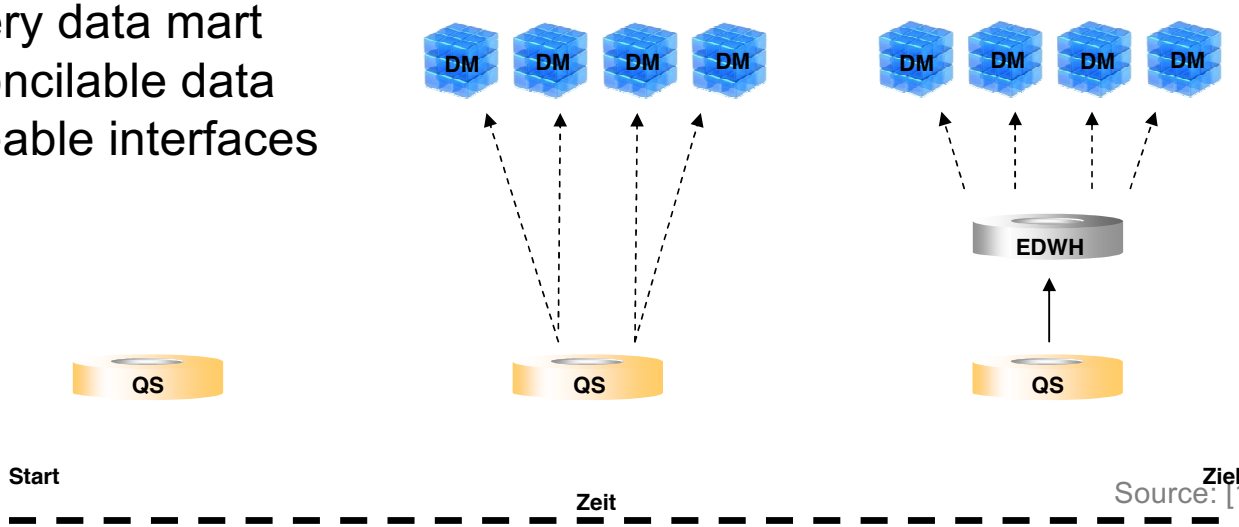


Advantages

+ Faster and easier implementation of manageable pieces
+ Favorable return on investment and proof of concept
+ Lower risk of failure
+ Inherently incremental; can schedule important data marts first
+ Allows project team to learn and grow

Disadvantages

– Each data mart has its own narrow view of data
– Redundant data in every data mart
– Inconsistent and irreconcilable data
– Proliferates unmanageable interfaces

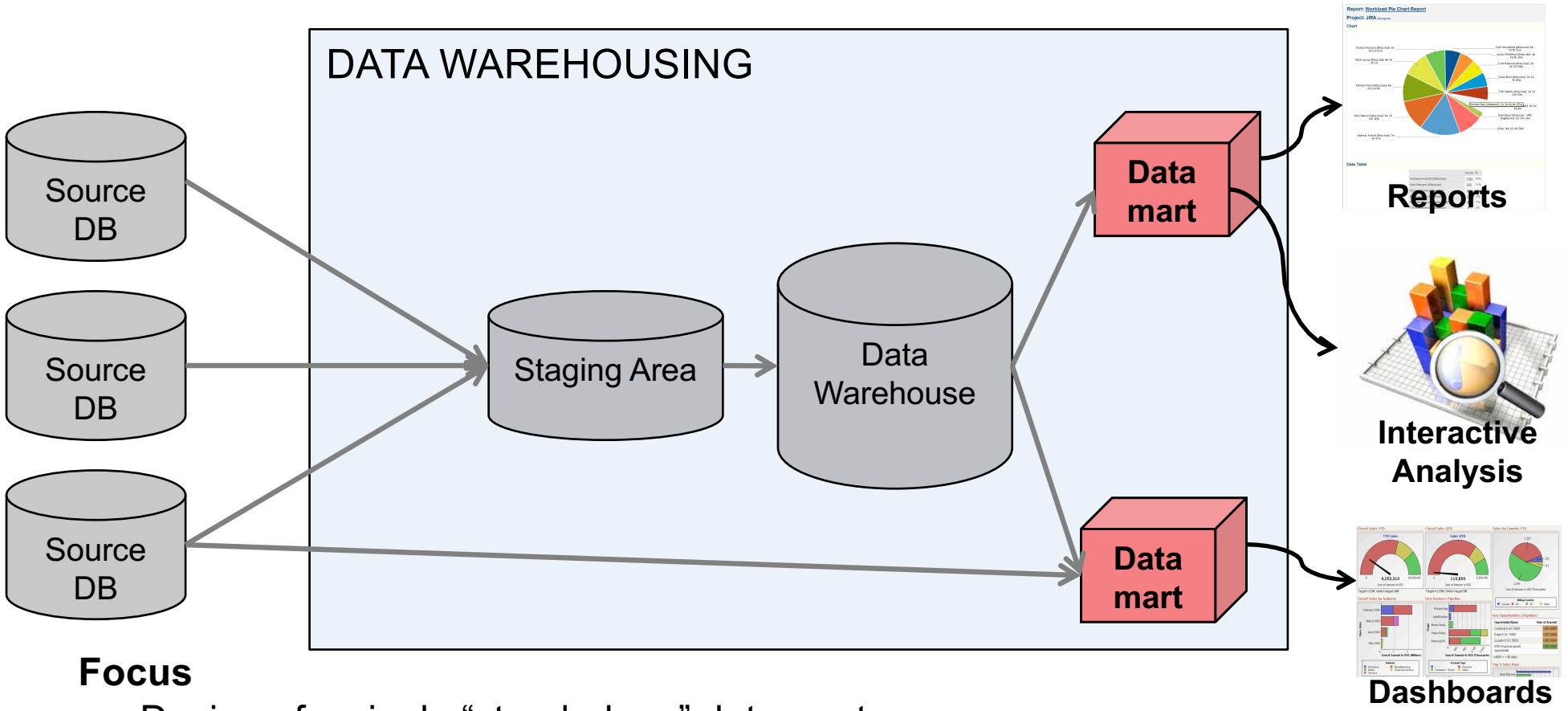


Source: [1]

Sources:
Ponniah, P.: Data Warehousing Fundamentals, 2ed. Wiley (2010)
[1] Lusti: Data Warehousing and Data Mining as cited by Melchard 2008: Lecture slides “Data Warehousing and Data Mining”; WU Vienna

DATA MARTS AND DIMENSIONAL MODELING

Data Marts in the DW Reference Architecture



Focus

- Design of a single “stand-alone” data mart
- Objective: analyze the performance of a single business process
- Data marts have less complex and easier to understand data models
- Optimize access paths and special physical data structures for typical queries

Source: adapted from Systems@ETH zürich R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Outline

1. **Recap: Operational Database Design**
2. Roles of attributes in Data Mart Design
3. Dimensional modeling
   - STAR Schema
   - OLAP Cube
   - Snowflake Schema

Recap: Designing Operational Databases

Goals

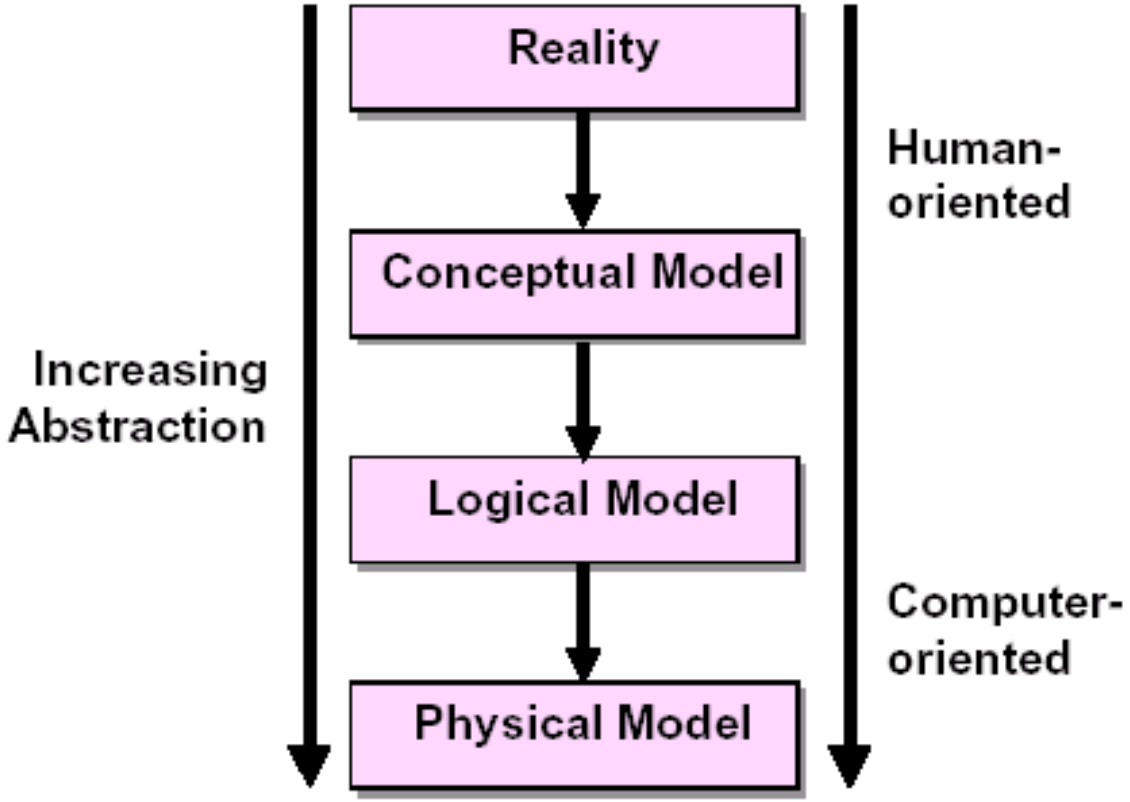
- Support execution of business processes
- Avoid (uncontrolled) redundancies (→ normalized schemas)
- Requirements profile: many short and small transactions
  - point and range queries
  - single-row inserts and/or updates

Typical design process

1. Use a variant of a “semantic” data model (e.g., ER model)
2. Transform the result into (normalized) tables
3. Tune using indexes and other physical design options
4. Selectively denormalize tables to improve overall performance

Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

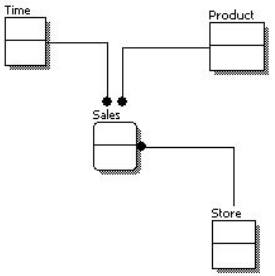
Data Modeling Recap: 3 layers of Abstraction



Source: [1]

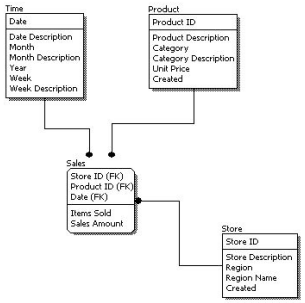
[1] http://www.gisdevelopment.net/proceedings/gita/2001/system/sa002.asp

Data Modeling Recap: 3 layers of Abstraction



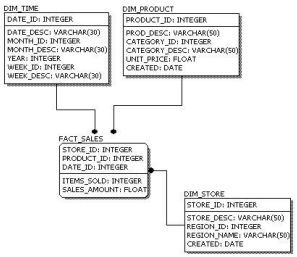
Conceptual model (e.g. E-R):

- Database independent
- Users' view



Logical model (e.g. Star Schema)

- DB-dependent (relational, object, graph, key-value..)
- Structure of the data



Source: [2]

Physical model (e.g., storage media, DB etc.)

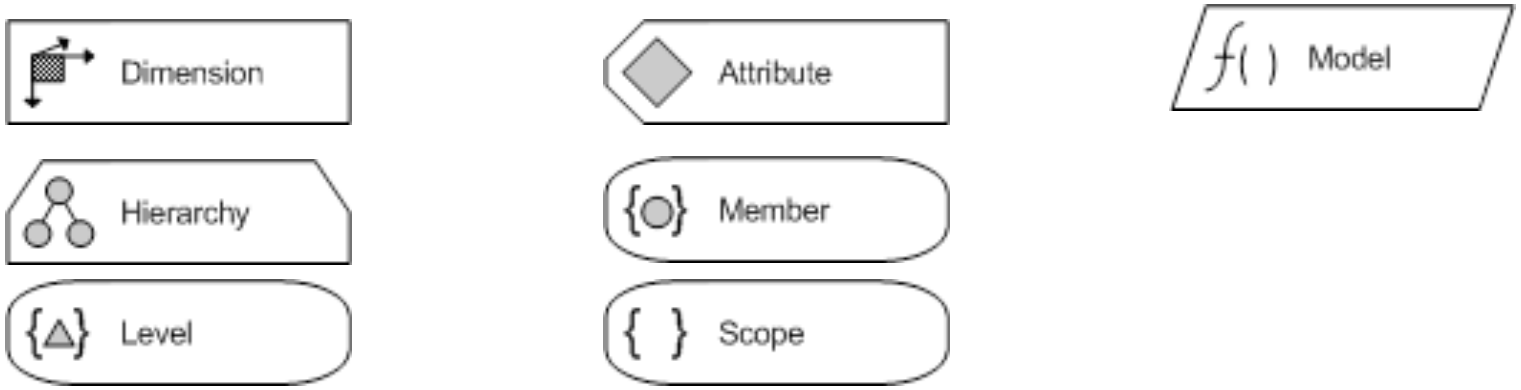
- Product dependent (e.g. row vs. column-oriented)
- Describes the physical implementation for a specific DBMS

Images Source:
[2] http://www.1keydata.com/datawarehousing/data-modeling-levels.html

ADAPT notation for *conceptual* DW-design

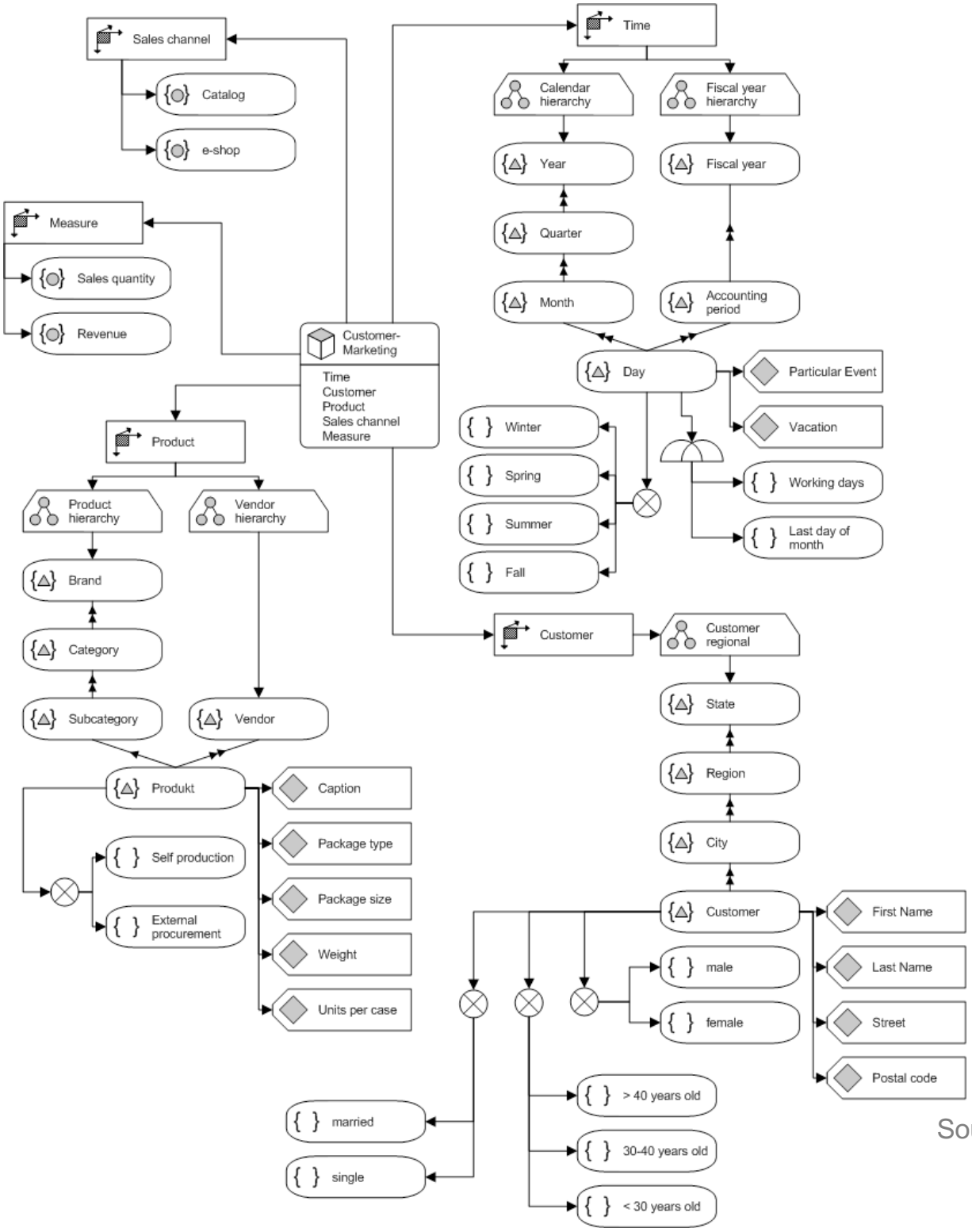
- Star-Schema = Logical level
- On a conceptual level, ER modeling is not particularly well suited for expressing user-requirements w.r.t multi-dimensional data
- ADAPT is one of several approaches for modeling at a level of abstraction above the logical level

Elements:



Source: http://www.t-adapt.com/tadapt_conceptual.html

T-ADAPT Example



Source: http://www.t-adapt.com/tadapt_conceptual.html

Recap: Logical Modeling
Normalization in a Nutshell…

1NF: "atomic" data values, no internal structure, no "repeating fields"

2NF,3NF: Functional dependencies between key and nonkey attributes

2NF: Nonkey attributes must not functionally depend on a part of the key

3NF: Nonkey attributes must not functionally depend on nonkey attributes

A nonkey attribute must provide information about the key, the whole key, and nothing but the key.

4NF: (multivalued dependencies)

BCNF: (functional dependencies in relations with key attributes only)

DKNF, 5NF, 6NF …

Designing Analytic Databases (i.e., Data Marts)

Goals:

- Support analysis of one or more business processes, i.e.,
  1. queries that may need to look at a **substantial amount of data** – even if they may return relatively small (aggregated) amount of data
  2. relatively infrequent but often **large periodic batch inserts**
- Maximize query read performance

  → caching mechanisms and **DENORMALIZATION**

Typical design process for data marts:

1. Separate analytic databases from operational databases
2. Determine how to measure the execution of a given business process and look at the context ("dimensions") of this business process
3. Design a corresponding Star or Snowflake Schema
4. Tune using indexes and other physical design options

Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Analysis of Business Processes

Typical Business Questions	Corresponding Business Process
Gross margin by product category in February 2011?	Sales
Average account balance by education level?	Account Management
Number of sick days by employees in marketing in 2010?	Time Management in Human Resources
Sum of outstanding payables by supplier?	Supplier Payment Processing
Product return rate by customer?	Client Returns Processing

Ingredients:

Measure context(category) time + aggregation, selection

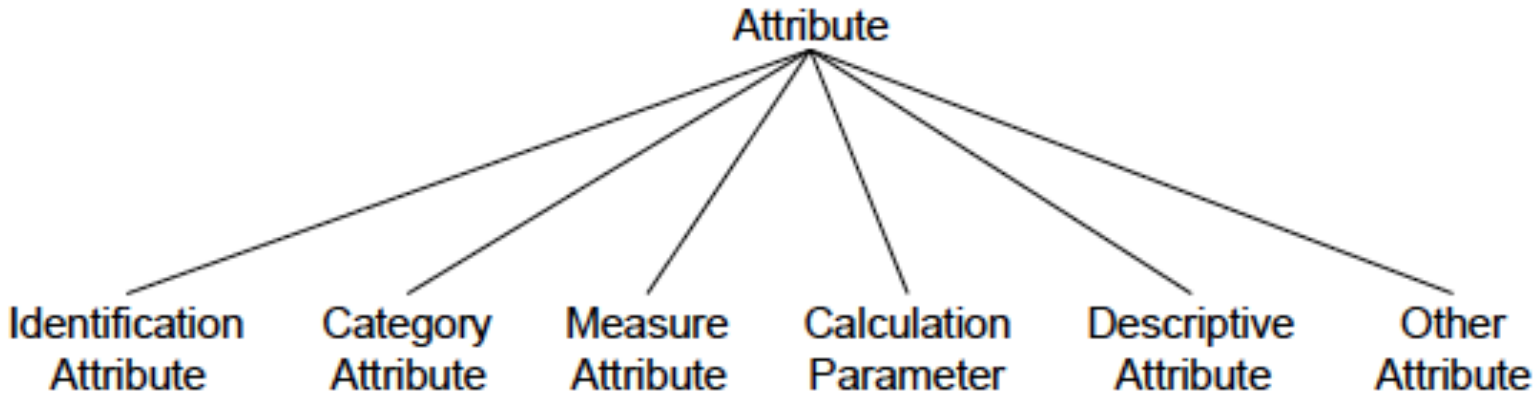
Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Outline

1. Recap: Operational Database Design
2. **Roles of attributes in Data Mart Design**
3. Dimensional modeling
   - STAR Schema
   - OLAP Cube
   - Snowflake Schema

Excursion: Different Purposes of Attributes

- Identification
- Categorization
- Quantification / measurement
- Calculation
- Textual description



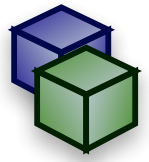
Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Attributes: Object Identification

- Attribute used to uniquely identify individual objects
- Value is a “meaningless” number which is not necessarily exposed to the user
- This number is also called a **surrogate** because it acts as a placeholder (proxy) of a real world object
- Principles:
  - must not change during the lifetime of the object which it identifies / represents, and
  - after the lifetime of this object, it must not be re-used for another (new) object.

Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Attributes: Category/Dimensional Attribute



- Objects assigned to one of a relatively small number of (discrete) categories, based on the value of an attribute
- A category attribute is an independent attribute that primarily serves to group or segment business data
- Categories are often arranged in taxonomic hierarchies

Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Attributes: Category Attribute with Taxonomy

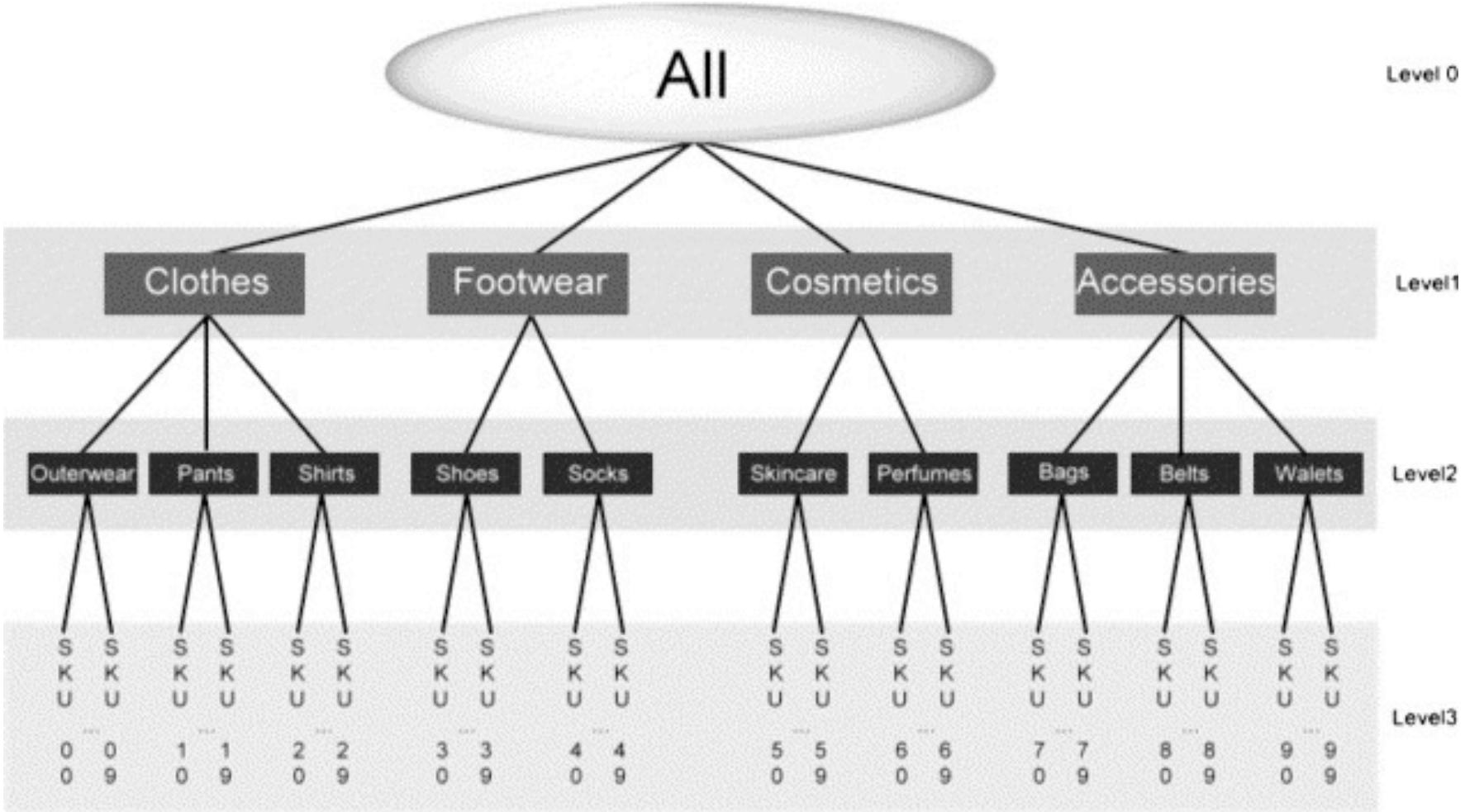


Image Source:
Lun-ping Hung, A personalized recommendation system based on product taxonomy for one-to-one marketing online, Expert Systems with Applications, Volume 29, Issue 2, August 2005

Source: Idea from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Attributes: Calculation parameter

$$a = 5b$$

A calculation parameter is a numeric (integer, decimal, fractional, floating point) attribute used to compute values of other attributes (typically measures, see below), e.g.

- an index
- a factor
- a rate

Its value is usually (but not always) determined by the values of one or more category attributes and a timestamp

Example:

- the exchange rate USD → CHF fixed on 5 May 2016
  < 'USD', 'CHF', '2016-05-05 04:34:53 GMT', 1.2313 >
- the inflation rate in Austria for 2007.
  < 'AUT', '2007', 0.023 >

Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Attributes: Measure

- Attribute used to express quantitative properties of objects (e.g., monetary amounts, magnitudes etc.)
- May have an associated unit (e.g., €, meters, tons, mph)
- May be a count, a ratio or a percentage without an associated unit.

Observed measures:
e.g., the annual base salary of an employee as recorded in a payroll system

Derived measures:
e.g., total annual compensation, computed from the observed measures annual base salary, family and child allowance, and annual bonus

Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Outline

1. Recap: Operational Database Design
2. Roles of attributes in Data Mart Design
3. **Dimensional modeling**
   - **STAR Schema**
   - OLAP Cube
   - Snowflake Schema

Facts vs. Dimensions

Facts:

- *usually verbs*
- Business performance indicators
- Usually numeric and continuous
- Can be aggregated
- Data volume: large

Dimensions:

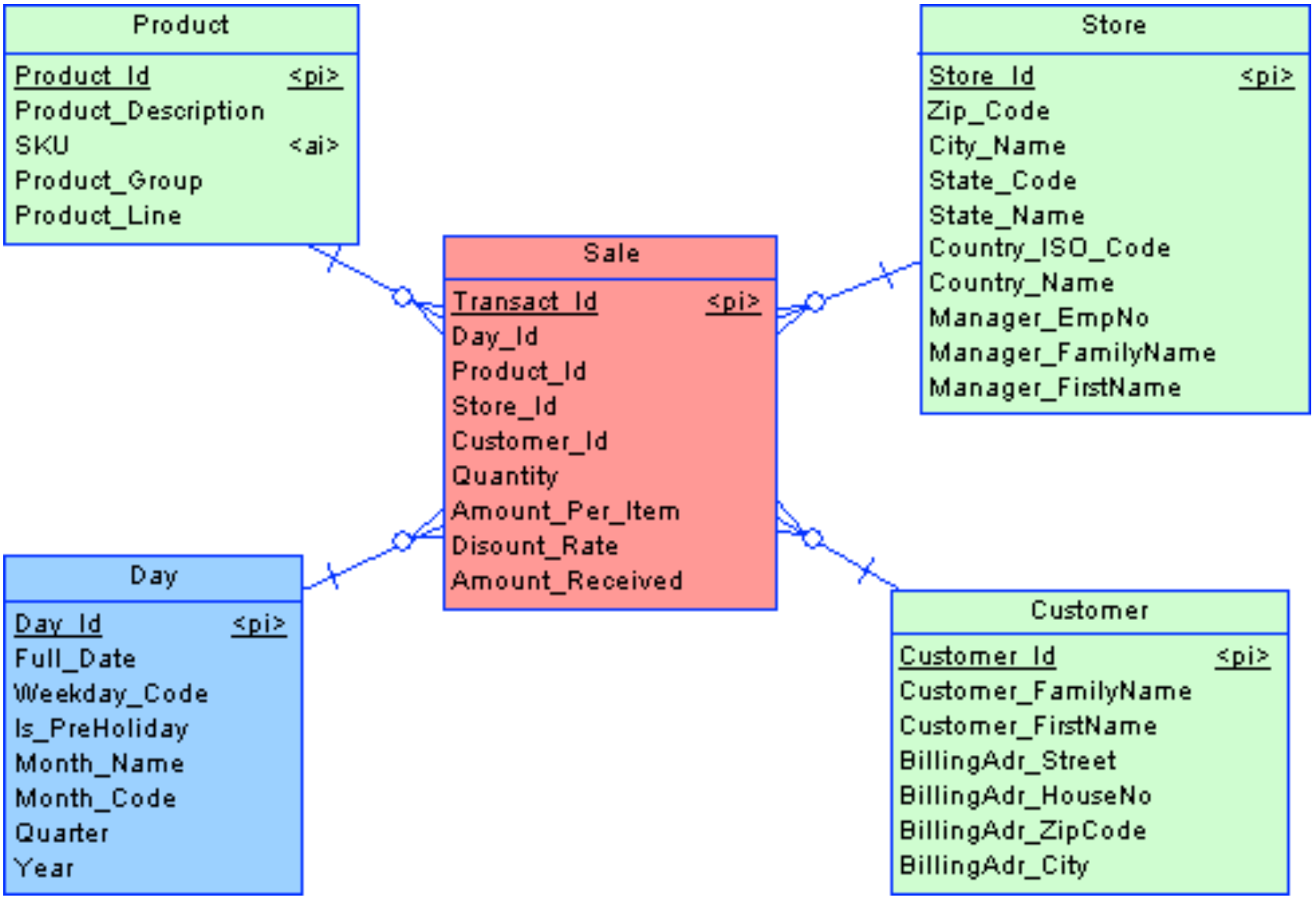
- *usually nouns*
- Represent criteria
- Usually symbolic and discrete
- Selection, aggregation and navigation of facts
- Data volume: low

Adapted from Melchard(2008): Lecture slides “Data Warehousing and Data Mining”; WU Vienna

Fact or Dimension?

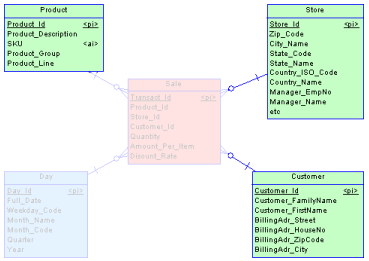
	Fact	Dimension
Time	☐	☒
POS transaction	☒	☐
Product	☐	☒
Profit	☒	☐
Branch	☐	☒

Star Schema Example



Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Star Schema: Dimension tables

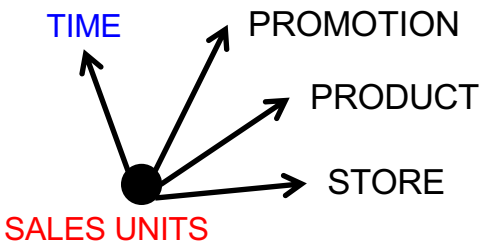


- provide “business context”
- used for categorization, e.g. Product, Store, Customer
- Attributes
  - surrogate (ending in *_Id* by convention)
  - more or less natural (often mnemonic) key (as used in operational DB)
  - many descriptive fields, often text
  - often hierarchical (e.g., Store -> City -> State -> Country)
- Flattened out, not normalized – i.e. may violate 2NF or 3NF (e.g., FD *Product_Group* → *Product_Line*, or FD *Manager_EmpNo* → *Manager_Name*)
- table is usually
  - relatively “short” (few rows)
  - but relatively “wide” (many colums)

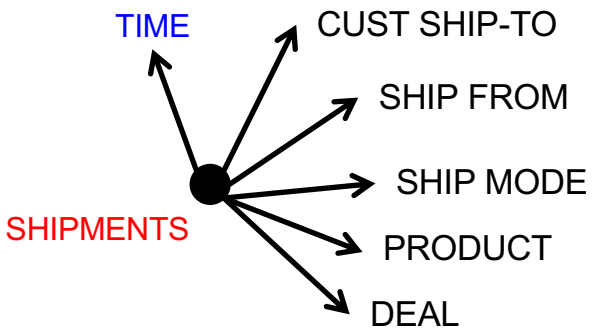
Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Examples of Business Dimensions

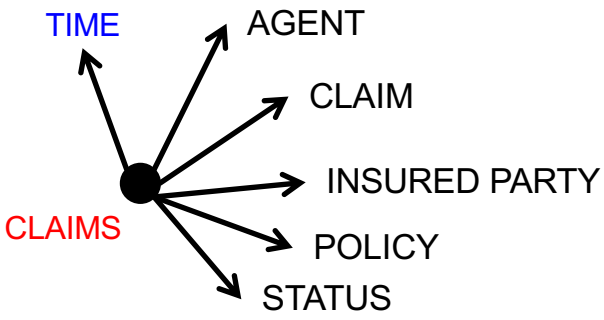
Supermarket Chain



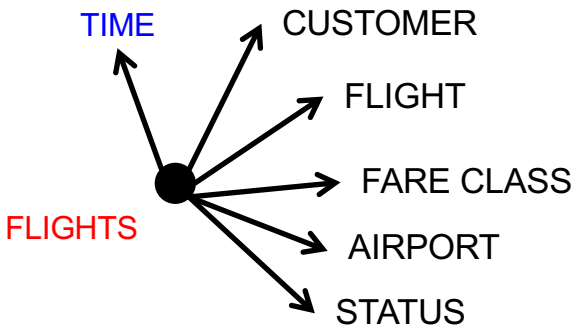
Manufacturing Company



Insurance Business

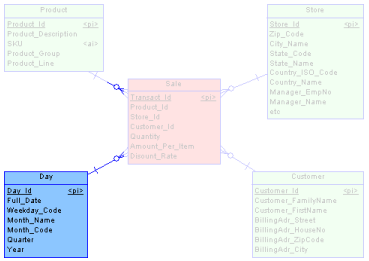


Airline



Sources:
Adapted from Ponniah, P.: Data Warehousing Fundamentals, 2ed. Wiley (2010)

Star Schema: Time Dimension



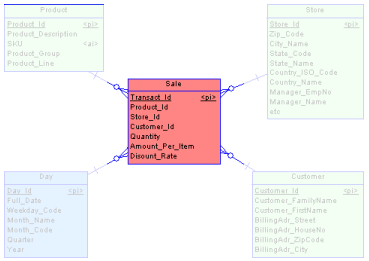
- Special dimension time (Day)
- Attributes
  - surrogate (ending in _*Id* by convention)
  - descriptive fields (also e.g. flags like “is holiday”, “is day before holiday”)

Note:

- Dates and timestamps are built-in types in SQL, with quite a few built-in functions to operate on them
- Treating a Day as an entity is a form of caching
- For example, selecting only Friday sales can be done using a simple filter predicate, without invoking any function!

Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)

Star Schema: Fact table



- Attributes
  - measures, either
    - basic measures, e.g., *Quantity*, *Amount_Per_Item*, or
    - (redundant) derived measures,
      e.g., *Amount_Received* := (*Quantity* * *Amount_Per_Item*) * (1 – *Discount_Rate*)
      foreign key attributes
  - surrogates (ending in *_Id* by convention), each of which serves as a foreign key to one dimension table, and which collectively establish the business context
  - a primary key, either a surrogate or a more or less mnemonic key
- Fact table is usually
  - relatively „long“ (many rows)
  - but not very „wide“
- A fact table has a grain defining the most atomic unit of data being captured, e.g.
  - transaction level: measures + their context are captured for each transaction
  - daily / weekly / monthly / quarterly summaries

Source: adapted from R. Marti – Data Warehousing (ETH Zürich, Spring 2011)