ifis

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

0. Why should you be here?

- Bad decisions can lead to disaster
  - Data Warehousing is at the base of **decision support systems**

0. Why should you be here?

- Data Warehousing & Data Mining is important – discover information **hidden** within the organization's data
  - See data from different angles: **product, client, time, area**
  - Get adequate **statistics** to get your point of argumentation across
  - **Get a glimpse of the future…**

0. Why should you be here?

- Because you **love databases…**

Sr. Information Architect - Data Technology Lead

S&P Global ★★★★☆ 1,348 reviews

New York, NY 10041

Apply On Company Site | Save this job

Job | Company

Job details

Salary

$100,800 - $230,200 a year

Excerpts from the job description:

[…]

Lead the implementation of strategy roadmap for the enterprise data; including **data modeling, implementation and data management for our enterprise data, warehouse** and advanced data analytics systems

[…]

Create and maintain the enterprise **data model at the conceptual, logical and physical Level**

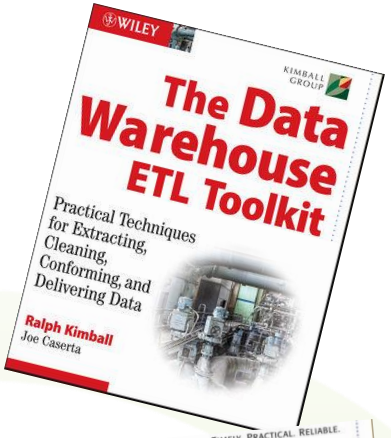
Work closely with strategic projects and coaches development teams on defined standards and methods for data usage and propagation.

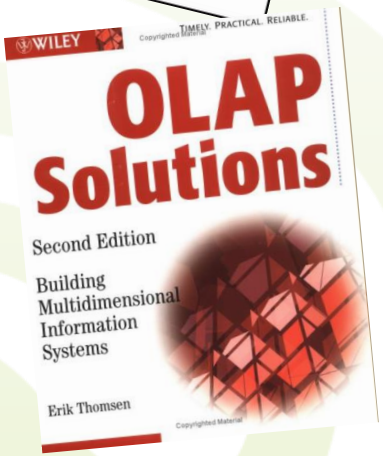
0. Recommended Literature

- Building the Data Warehouse
  - William H. Inmon
  - Wiley, ISBN 0-7645-9944-5
- The Data Warehouse Toolkit
  - Ralph Kimball & Margy Ross
  - Wiley, ISBN 0-471-20024-7
- Data Warehouse-Systeme
  - Andreas Bauer & Holger Günzel
  - dpunkt.verlag, ISBN 978-3898647854

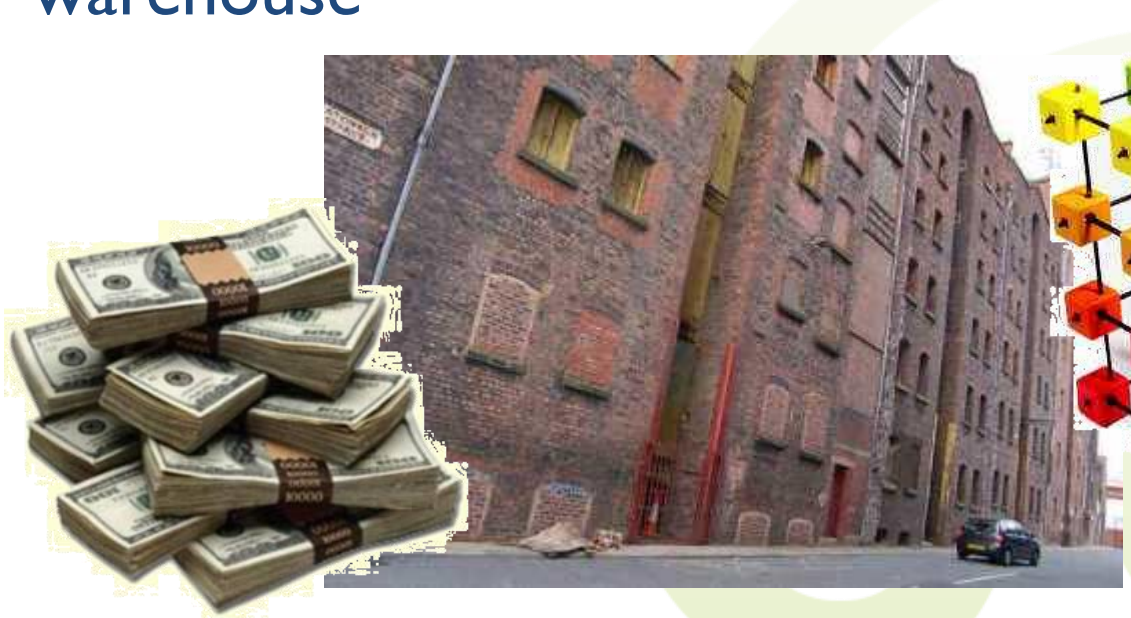
0. Recommended Literature

- The Data Warehouse ETL Toolkit
  - Ralph Kimball & Joe Caserta
  - Wiley, ISBN 0-7645-6757-8
- OLAP Solutions
  - Erik Thomsen
  - Wiley, ISBN 0-471-40030-0
- Data Warehouses and OLAP
  - Robert Wrembel & Christian Koncilia
  - IRM Press, ISBN 978-1599043654





1. Introduction

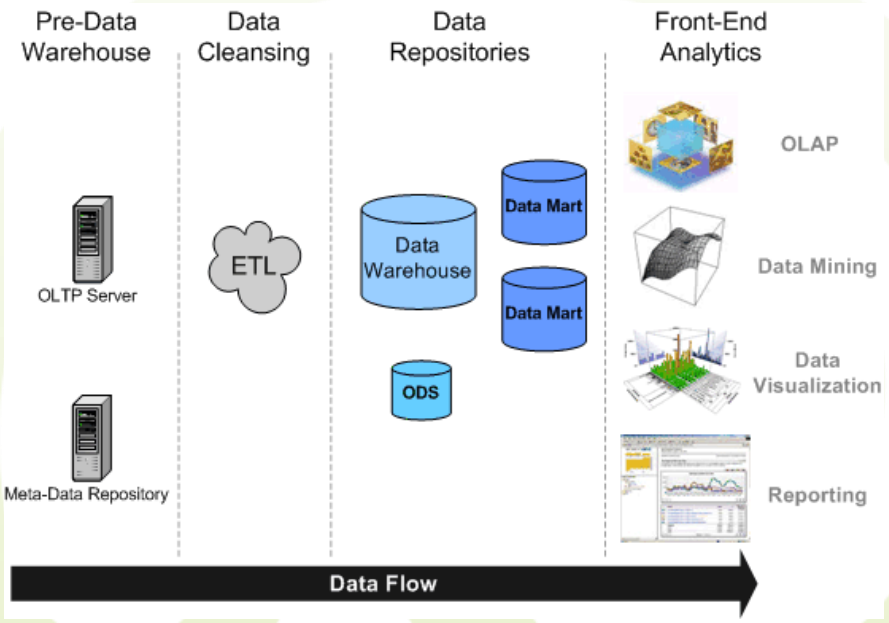


I.I What is a data warehouse?

- Basically a very large **database**…
  - Not all very large databases are DW, but all data warehouses are pretty large databases
  - Nowadays a warehouse is considered to start at around a TB and goes up to several PB
  - It spans over several servers and needs an impressive amount of computing power

I.I What is a data warehouse?

- More specific, a **collective data repository**
  - Containing snapshots of the operational data (history)
  - Obtained through data cleansing (Extract-Transform-Load process)
  - Useful for **analytics**



1.1 What is a data warehouse?

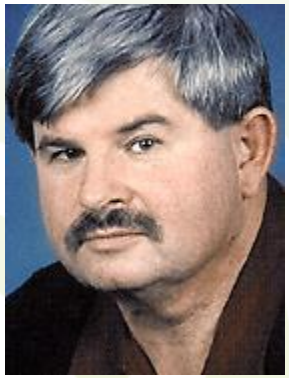
- Compared to other solutions it…
  - Is suitable for **tactical/strategic focus**
  - Implies a **small number of transactions**
  - Implies **large transactions** spanning over a long period of time

	OLTP	ODS	OLAP	DM / DW
Business Focus	Operational	Operational / Tactical	Tactical	Tactical / Strategic
End User Tools	Client/Server or Web	Client/Server or Web	Client/Server	Client/Server or Web
DB Technology	Relational	Relational	Cubic	Relational
Transaction Count	Large	Medium	Small	Small
Transaction Size	Small	Medium	Medium	Large
Transaction Time	Short	Medium	Medium	Long
DB Size in GB	10–400	100–800	100–800	800—80,000
Data Modeling	Traditional ERD	Traditional ERD	N/A	Dimensional
Normalization	3–5 NF[1]	3 NF	N/A	0 NF

I.I Some Definitions

- Experts say…
  - **Ralph Kimball:** "a copy of transaction data specifically structured for query and analysis"
  - **Bill Inmon:** "A data warehouse is a:
    - Subject oriented
    - Integrated
    - Non-volatile
    - Time variant

    collection of data in support of management's decisions."

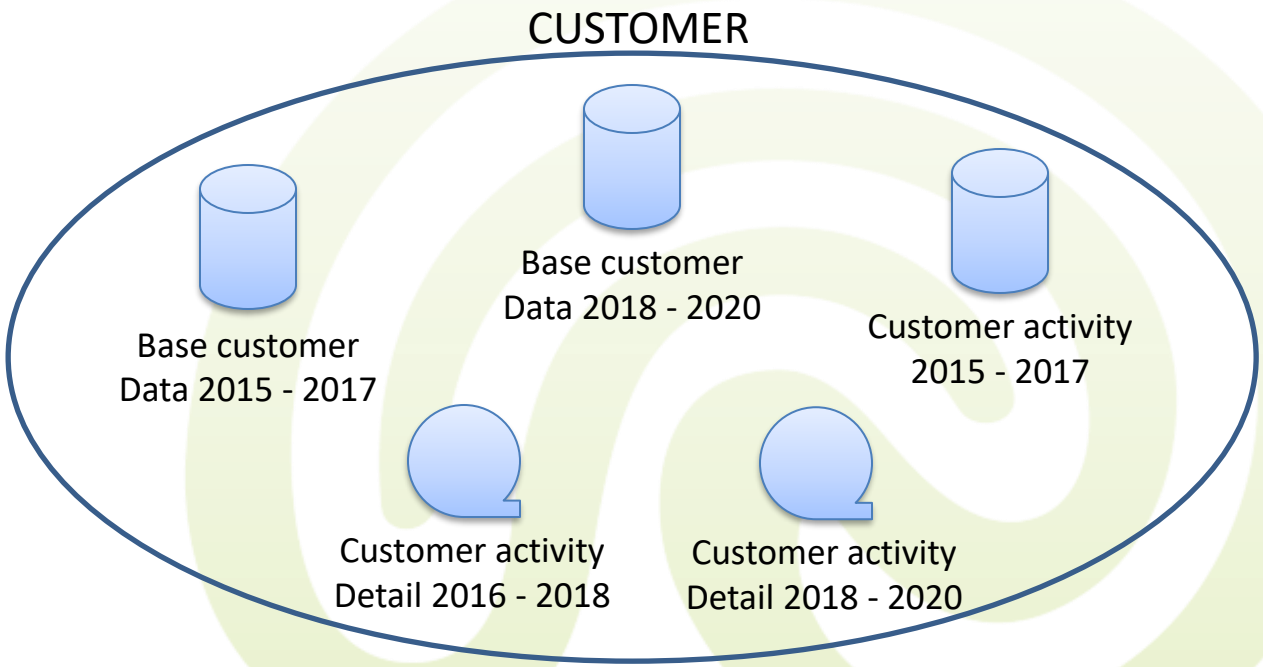


I.I Inmon Definition

- **Subject oriented**
  - The data in the DW is organized in such a way that all the data elements relating to the same real-world event or object are **linked together**
    - Typical subject areas in DWs are Customer, Product, Order, Claim, Account,…

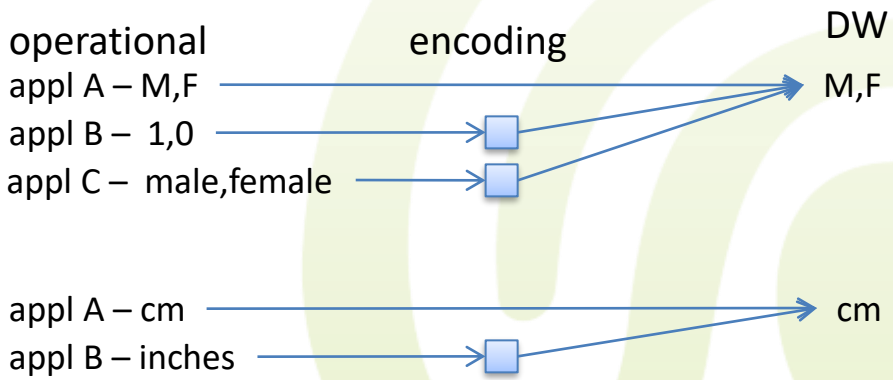
I.I Inmon Definition

- **Subject oriented**
  - Example: customer as **central subject** in some DW
    - The complete DW is **organized** by customer
    - It may consist of hundreds or more physical tables that are related



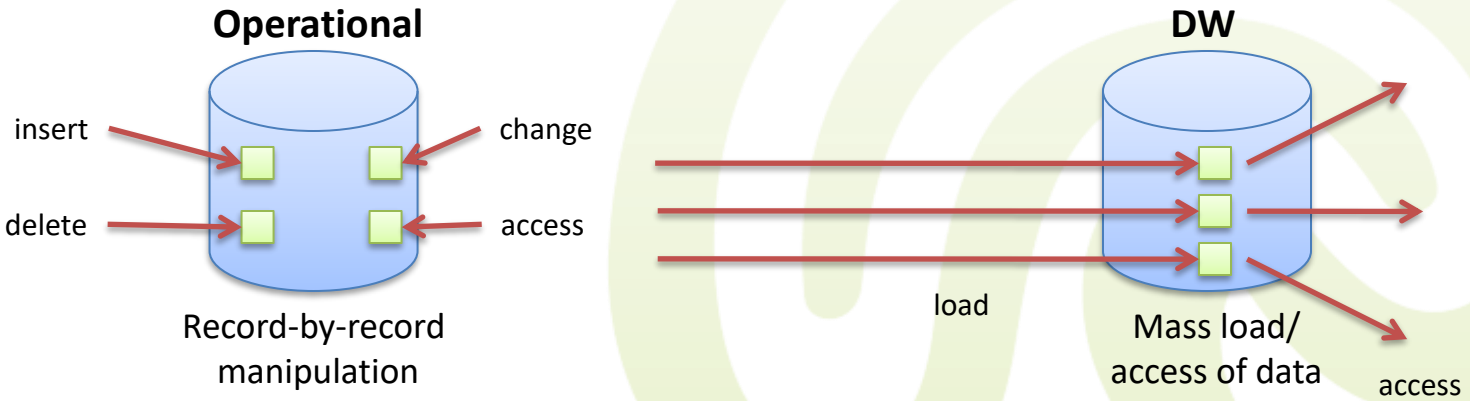
I.I Inmon Definition

- **Integrated**
  - The DW contains data from **most or all** the organization's operational systems and this data is made **consistent**
  - E.g. gender, measurement, conflicting keys, consistency,…



I.I Inmon Definition

- **Non-volatile**
  - Data in the DW is **never over-written** or **deleted** - once committed, the data is static, read-only, and retained for future reporting
  - Data is loaded, but not updated
  - When subsequent changes occur, a new version or snapshot record is written

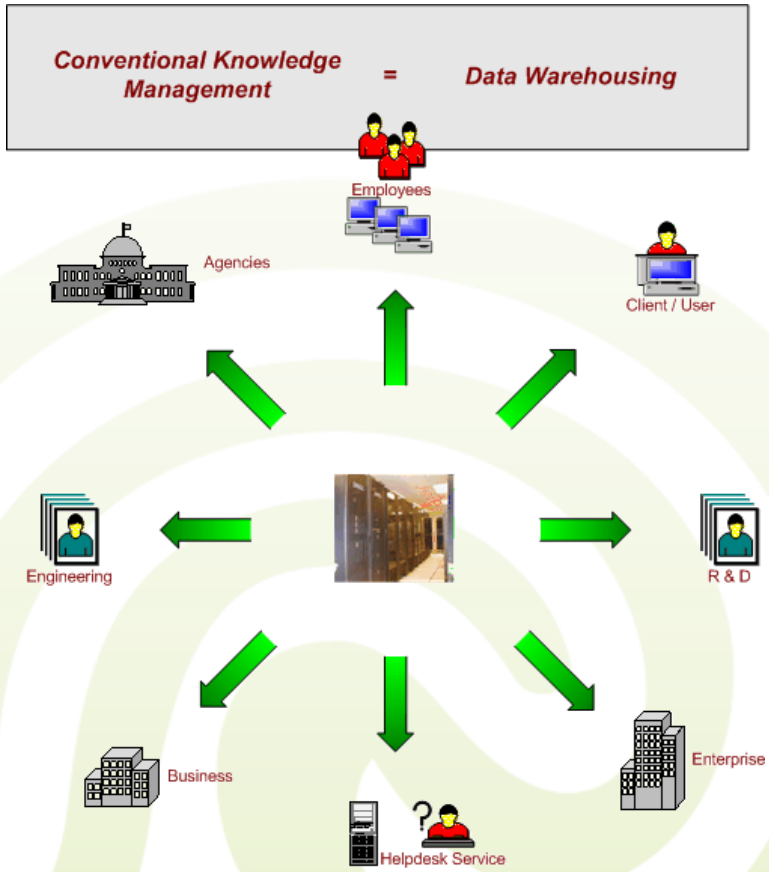


I.I Inmon Definition

- **Time-varying**
  - The changes to the data in the DW are tracked and recorded so that reports show **changes over time**
  - Different environments have different **time horizons** associated
    - While for operational systems a 60-to-90 **day** time horizon is normal, DWs have a 5-to-10 **year** horizon

I.I General Definition

- More general, a DW is…
  - **A large repository of some organization's electronically stored data**
  - **Specifically designed to facilitate reporting and analysis**



I.I Typical Features

- DW typically…
  - Reside on computers **dedicated to this function**
  - Run on **enterprise scale DBMS** such as Oracle, IBM DB2, Teradata, or Microsoft SQL Server
  - Retain data for **long periods of time**
  - **Consolidate data** obtained from a variety of sources
  - Are built around their own **carefully designed data model**

1.1 Use Case

Detour

- DW stands for big data volume, so lets take an example of **2 big companies**, Walmart and a RDBMS vendor, Teradata (in 1990):
  - Walmart CIO: *I want to keep track of sales in all my stores simultaneously*
  - Teradata consultant: *You need our wonderful RDBMS software. You can stuff data in as sales are rung up at cash registers and simultaneously query data right in your office*
  - So Walmart buys a $1 million Sun E10000 multi-CPU server, a $500 000 Teradata license, a book "Database Design for Smarties", and builds a normalized SQL data model

1.1 Use Case

Detour

- After a few months of stuffing data into the table a Walmart executive asks…
  - *I have noticed that there was a **Colgate promotion** recently, directed to people who live in small towns. How much toothpaste did we sell in those towns yesterday?*
  - Translation to a query:
    - 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

1.1 Use Case

Detour

- The tables contain large volumes of data and the query implies a **6 way join** so it will take some time to execute
- The tables are at the **same time also updated** by new sales
- Soon after executive start their quest for marketing information, the store employees notice that there are times during the day when it is **impossible to process a sale**

→ Any attempt to **update** the database results in freezing the cash registers for 20 minutes

1.1 Use Case

Detour

- Minutes later… the Walmart CIO calls Teradata tech support

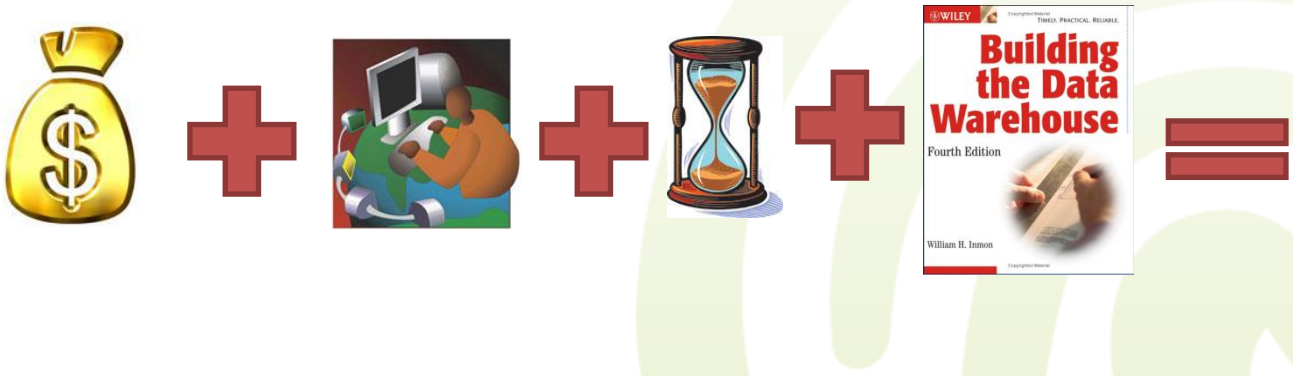
  - Walmart CIO: *WE TYPE IN THE TOOTHPASTE QUERY AND OUR SYSTEM HANGS!!!*
  - Teradata support: *Of course it does! You built an **on-line transaction processing (OLTP)** system. You can't feed it a **decision support system (DSS)** query and expect things to work!*
  - Walmart CIO: *!@%$#. I thought this was the whole point of SQL and your RDBMS…to query and insert simultaneously!!*
  - Teradata support: *Uh, not exactly. If you're **reading** from the database, nobody can **write** to the database. If you're **writing** to the database, nobody can **read** from the database. So if you've got a query that takes 20 minutes to run and don't specify **special locking instructions**, nobody can update those tables for 20 minutes.*

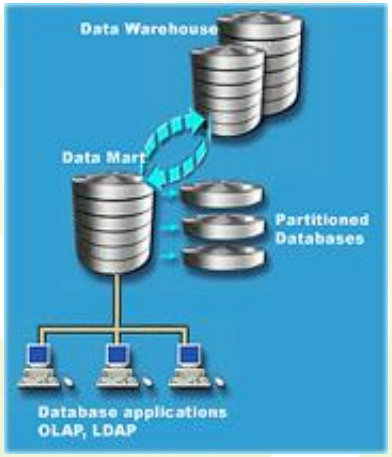
1.1 Use Case

Detour

- Walmart CIO: *It sounds like a bug.*
- Teradata support: *Actually it is a feature. We call it **pessimistic locking**.*
- Walmart CIO: *Can you fix your system so that it doesn't lock up???*
- Teradata support: *No. But we made this great loader tool so that you can copy everything from your OLTP system into a separate Data Warehouse system at 100 GB/hour*

- After a while…





1.1 OLTP

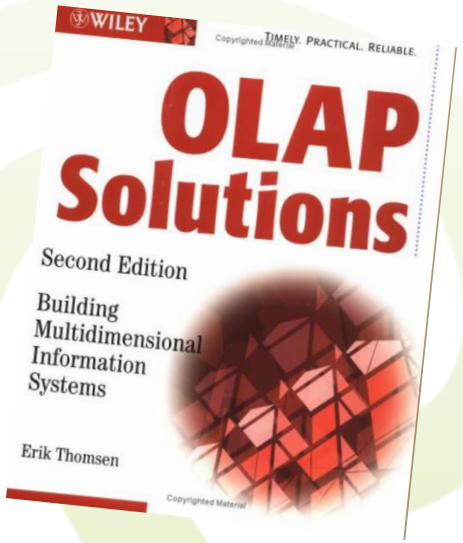
Detour

- OLTP (**O**n**L**ine **T**ransaction **P**rocessing)
  - Typically for **data entry / retrieval** and transaction processing
  - Works on the **operational data stores (ODS)** and represents day-to-day operational business activities
    - Purchasing, sales, production distribution, …
  - Reflects only the **current state** of the data

I.I OLAP

Detour

- OLAP (**O**n**L**ine **A**nalytical **P**rocessing)
  - Provides information for activities like
    - Enterprise resource planning, capital budgeting, marketing initiatives,…
  - Represents **front-end analytics** based on a DW repository
  - Is used for **reporting** and **decision oriented**



I.I ODS vs. DW

Detour

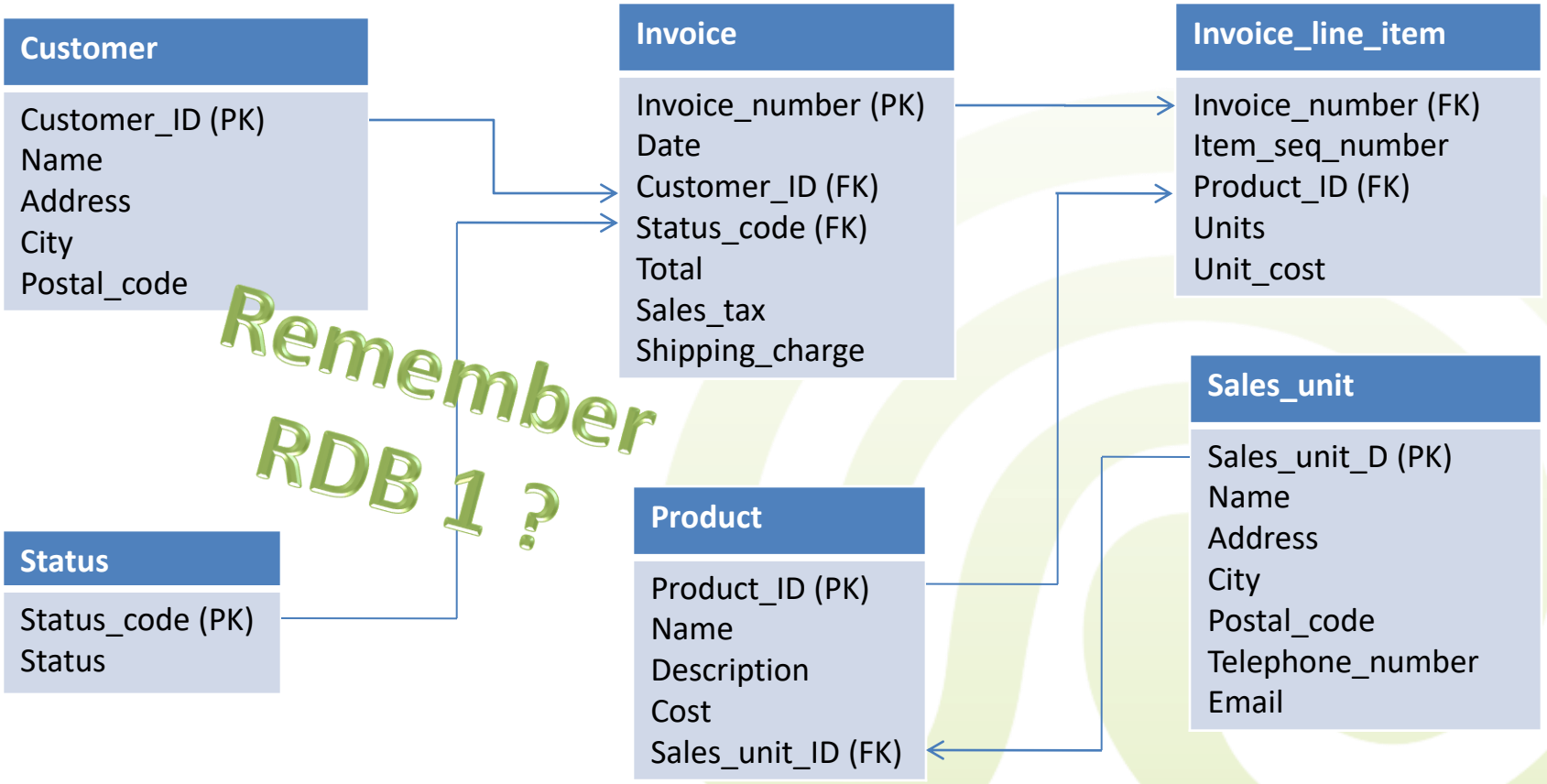
- Properties of Operational Data Stores and DWs

ODS	DW
Mostly updates	Mostly reads
Many small transactions	Few, but complex queries
GB-PB of data	TB-EB of data
Raw data	Summarized data
Clerks	Decision makers
Up-to-date data	May be slightly outdated

I.I ODS vs. DW

Detour

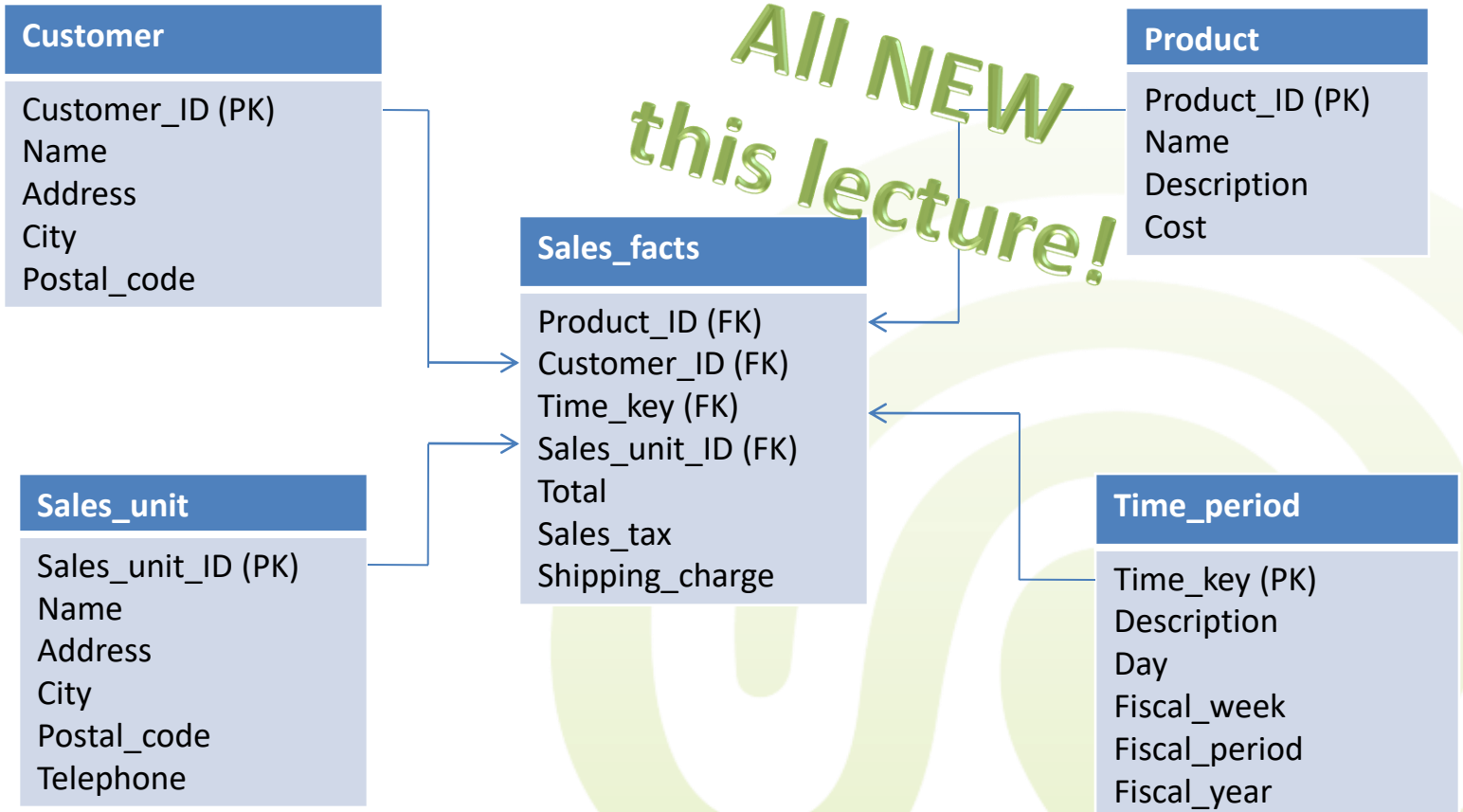
- Consider a **normalized database** for a store
  - The schema would look somewhat like this…



I.I ODS vs. DW

Detour

- If we were to set up a **DW** for that store, we would start by building the following ***schema***



I.I ODS vs. DW

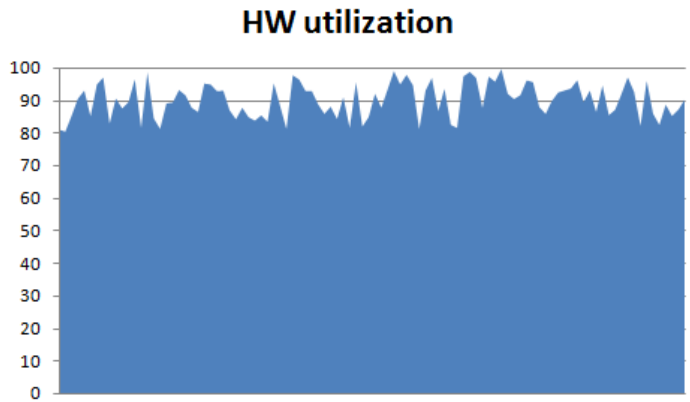
Detour

- **Basic insights** from comparing ODS and DWs
  - A DW is a **separate (DBMS) installation** that contains copies of data from **operational** systems
    - Physically separate hardware may not be absolutely necessary if you have lots of **extra computing power**, but it is recommended
  - With an **optimistic locking** DBMS you might even be able to get away for a while with keeping just one copy of your data

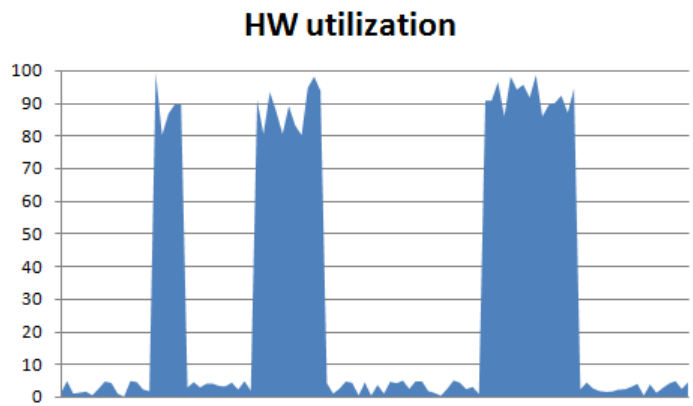
I.I ODS vs. DW

Detour

- There is an essentially different pattern of **hardware utilization** between transactional and analytical processing



Operational



Data warehouse

I.I ODS vs. DW

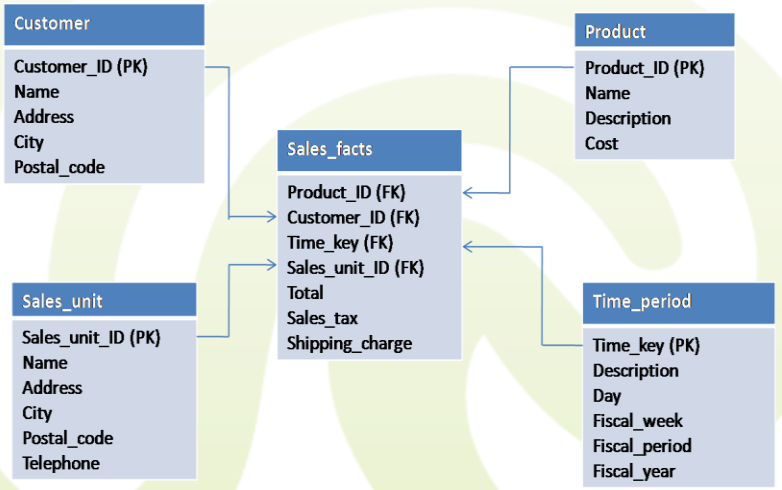
Detour

- **Typical questions** to be be answered by OLAP
  - How much did sales unit A earn in January?
  - How much did sales unit B earn in February?
  - What was their combined sales amount for the first quarter?
- Answering these questions with **SQL-queries** is difficult
  - Complex query formulation necessary
  - Processing will be **slow** due to complex joins and multiple scans

I.I ODS vs. DW

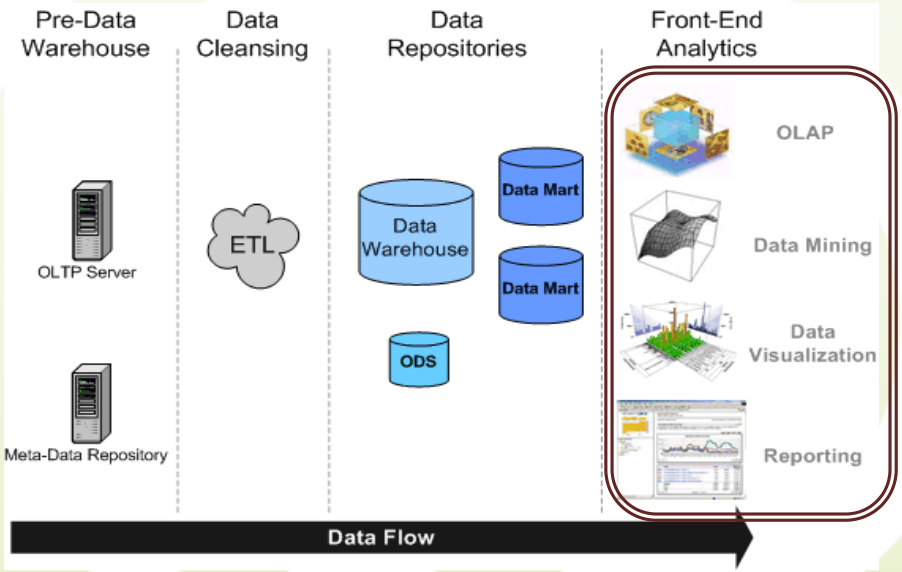
Detour

- Why can such questions be answered faster with a DW?
  - Because in a DW data is **rearranged** in tables and **pre-aggregated**
  - The table arrangement is **subject oriented**, usually some star schema



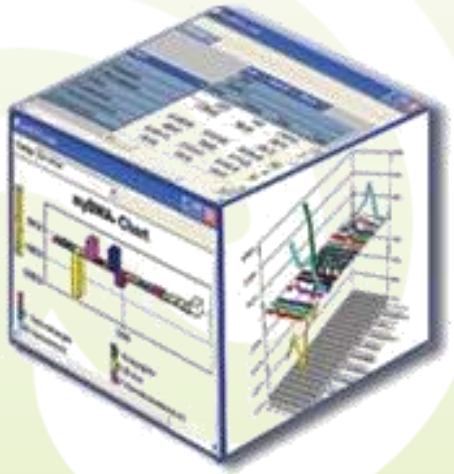
I.2 Typical Application

- A DW is the base repository for **front-end analytics** (or business cockpits)
  - **OLAP**
  - **Knowledge discovery in databases (KDD) and data mining**
- Results are used for
  - Data visualization
  - Reporting



I.2 Typical Application

- As a form of information processing **OLAP** needs to provide **timely**, **accurate** and **understandable** information
  - 'Timely' is however a relative term…
    - In OLTP we expect a query/update to go through in a matter of **seconds**
    - In OLAP the time to answer a query can take **minutes, hours or even longer**



I.2 Typical Application

- **KDD & Data Mining**
  - **Constructs models** of the data in question
    - Models can be seen as high level summaries of the underlying data
    - E.g. "customers older than 35 having at least 1 child and driving a minivan usually spend more than €100 for grocery shopping"

ID	Name	Age	Income	Children	Car	Spent
12	Peter	45	€ 65,000	2	Mini Van	€ 210.00
15	Gabriel	28	€ 53,000	0	Coupe	€ 30.00
…	…	…	…	…	…	…
122	Claire	40	€ 52,000	1	Mini Van	€ 250.00

I.2 Who are the users?

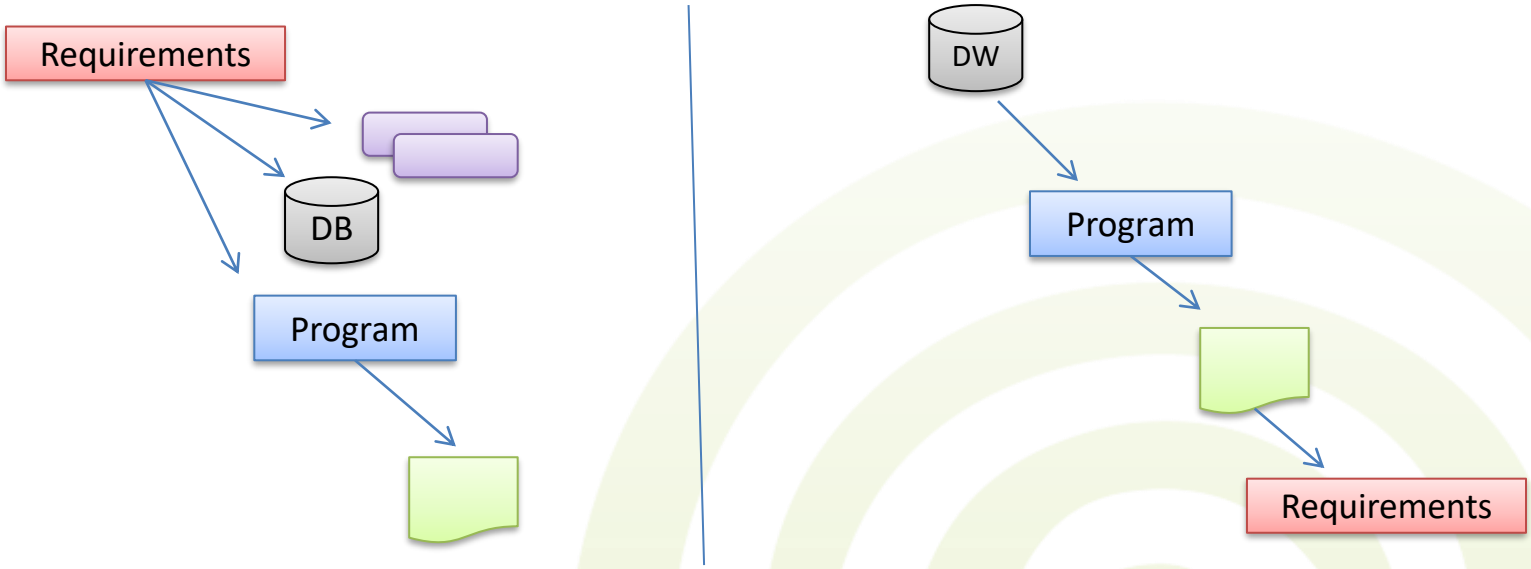
- **Users of DW** are called decision support system (DSS) **analysts** and usually have a business background
  - Their primary job is to **define** and **discover** information used in corporate **decision-making**
  - The way they think
    - "Show me what I say I want… and then I can tell you what I really want"
    - They work in an **explorative manner**

I.2 Who are the users?

- Typical **explorative line** of work
  - "When I see what the possibilities are, I can tell what I really need to see. But until I see what the possibilities are, I cannot describe exactly what I want…"
- This usage has profound effect on **the way a DW is developed**
  - The classical **system development life cycle** assumes that the requirements are known at the start of design
  - The DSS analyst starts with existing requirements, but factoring in **new requirements** is almost impossible

I.3 Lifecycle of DWs

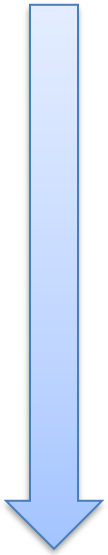
- **S**ystem **D**evelopment **L**ife **C**ycle (SDLC)
  - Classical SDLC vs. DW SDLC



  - DW SDLC is almost the **opposite** of classical SDLC, since requirements are not known from the beginning

I.3 Lifecycle of DWs

- **Classical SDLC** vs. **DW SDLC**



Classical SDLC	DW SDLC
Requirements gathering	Implement warehouse
Analysis	Integrate data
Design	Test for bias
Programming	Program against data
Testing	Design DSS system
Integration	Analyze results
Implementation	Understand requirements

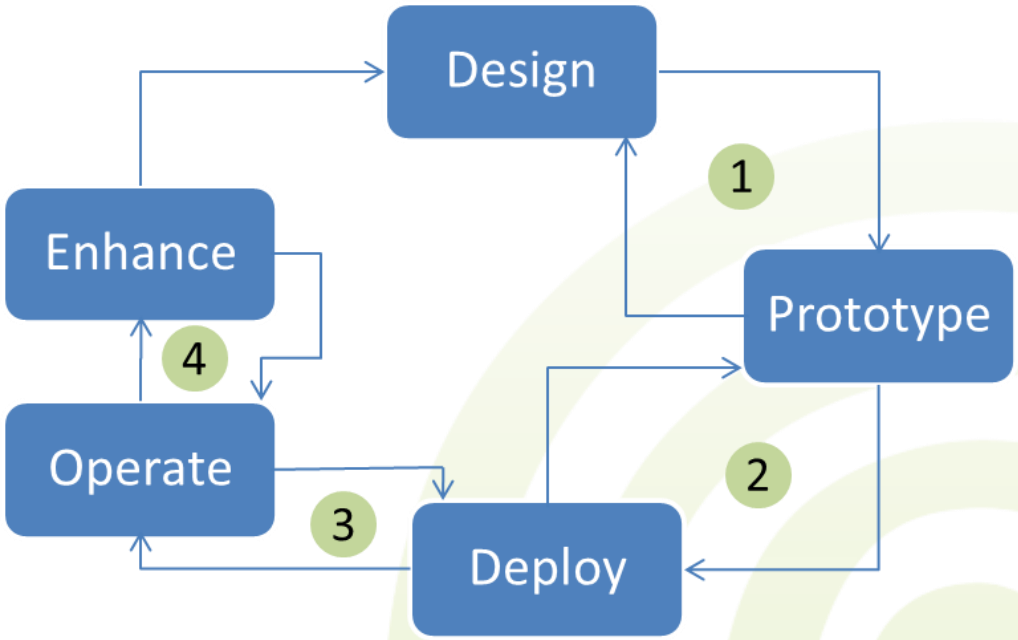
– Because it is the opposite of SDLC, DW SDLC is also called CLDS

I.3 Lifecycle of DW

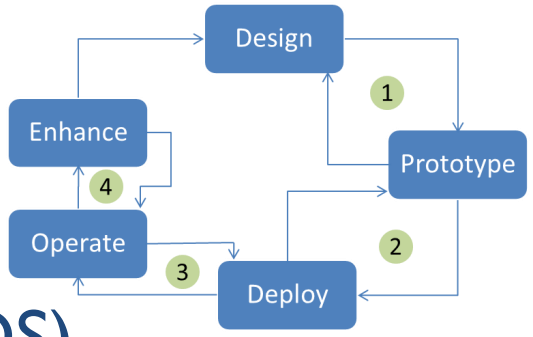
- CLDS is a **data driven** development life cycle
  - It starts with data
    - Once data is at hand it is integrated and tested against bias
    - Programs are written against the data and the results are analyzed and finally the requirements of the system are understood
    - Once requirements are understood, adjustments are made to the design and the cycle starts all over
  - **"spiral development methodology"**

I.3 Lifecycle of DWs

- Lifecycle phases

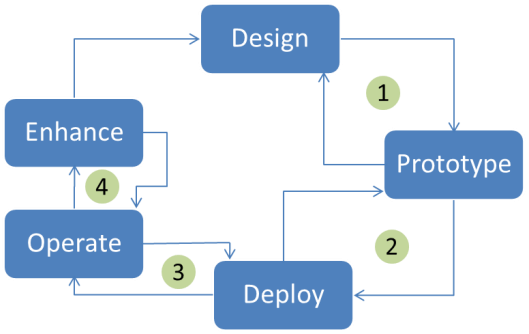


I.3 Lifecycle of DWs



- **Design**
  - Interviewing the end-users in cycles
  - Analyzing the data source system (ODS)
  - Defining the key performance indicators
  - Mapping the decision-making processes to the underlying information needs
  - Logical and physical schema design
- **Prototype**
  - Objective is to **constrain** and in some cases **reframe** end-user requirements

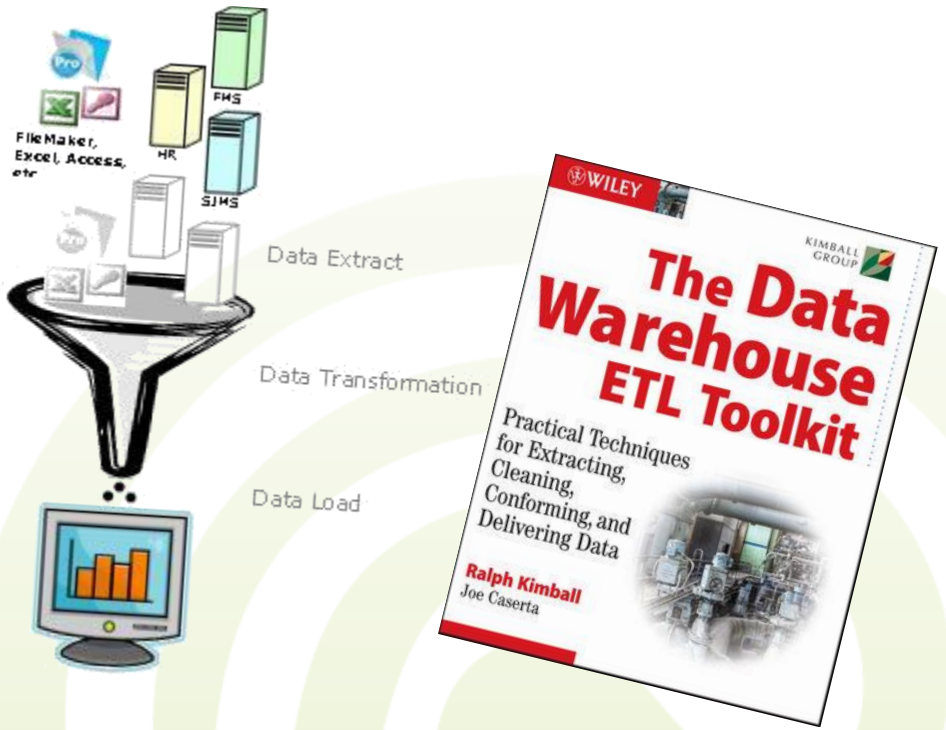
I.3 Lifecycle of DWs



- **Deployment**
  - Development of documentation
  - Personal training
  - Operations and management processes
- **Operation**
  - Day-to-day maintenance of the DW needs a good management of ongoing **E**xtraction, **T**ransformation and **L**oading (ETL) process
- **Enhancement** requires the modification of
  - HW - physical components
  - Operations and management processes
  - Logical schema designs

I.3 Operating a DW

- When **operating** a DW the following phases can be identified
  - Monitoring
  - Extraction
  - Transforming
  - Loading
  - Analyzing



I.3 Monitoring

- **Monitoring**
  - Surveillance of the operational data sources
  - Identification of data modification which is relevant to the DW

- Monitoring has an important role over the whole process deciding on which data to load, and when to load it into the DW

I.3 Monitoring

- **Monitoring techniques**
  - **Active** mechanisms - Event Condition Action (ECA) rules:

EVENT	Payment
CONDITION	Account sum > 10 000 €
ACTION	Transfer to economy account

  - **Replication** mechanisms:
    - Oracle 9i – Snapshots are local copies of data (similar to a view): a snapshot is replaced completely on change
    - IBM DB2 – Data replication maintains and replicates data in destination tables through a data propagation processes (data is incrementally updated)

I.3 Monitoring

– **Protocol** based mechanisms:
  - Since DBMSs write protocol data for transaction management, the protocol can also be used for monitoring
  - Problematic since protocol formats are proprietary and subject to change
– **Application** managed mechanisms:
  - Hard to implement for legacy systems
  - Based on *time stamping* or *data comparison*

I.3 Extraction

- **Extraction**
  - Reads the data selected during the monitoring phase and inserts it in the intermediate data structures of the workplace ("staging area")
  - Due to large data volume, compression can be used

I.3 Extraction

- The time-point for performing extraction can be
  - **Periodical**: weather or stock market information can be actualized more times in a day, while product specification can be actualized in a longer period of time
  - **On request:** e.g. when a new item is added to a product group
  - **Event driven:** event driven e.g. number of modifications over passing a specified threshold triggers the extraction
  - **Immediate:** in some special cases like the stock market it can be necessary that the changes propagate immediately to the warehouse
- Extraction largely depends on **hardware** and the **software** used for the DW and the data source

I.3 Transforming

- Transforming
  - Implies **adapting data, schema** as well as **data quality** to the application requirements
  - Data integration:
    - Transformation in de-normalized data structures
    - Handling of key attributes
    - Adaptation of different types of the same data
    - Conversion of encoding: "Buy", "Sell" → 1,2 vs. B, S → 1,2
    - Date handling: "MM-DD-YYYY" → "MM.DD.YYYY"

I.3 Transforming

- String normalization
  - "Michael Schumacher" → "Michael, Schumacher" vs. "Schumacher Michael" → "Michael, Schumacher"
- Measurement units and scaling
  - 10 inch → 25,4 cm
  - 30 mph → 48,279 km/h
- Save calculated values
  - Price including tax = Price without tax * 1.19
- Aggregation
  - Daily sums can be added into weekly ones
  - Different levels of granularity can be used

I.3 Transforming

- **Data cleansing (or data cleaning)**
  - Consistency check: Delivery Date < Order Date
  - Completeness: management of missing values as well as NULL values
  - Dictionary approaches for city or person names
  - Regular expressions for phone numbers or email addresses
  - Duplicate detection for redundancy elimination
  - Outlier detection as a warning system for possible errors

I.3 Loading

- Loading
  - Loading usually takes place during weekends or nights when the system is not under user stress
  - Split between initial load to initialize the DW and the periodical load to keep the DW updated
  - **Initial loading**
    - Implies big volumes of data and for this reason a **bulk loader** is used
  - Usually optimized by means of **parallelization** and **incremental actualization**

I.3 Analyzing

- Analysis phase
  - Data access - useful for extracting goal oriented information
    - How many iPhones 3G were sold in the Braunschweig stores of T-Mobile in the last 3 calendar weeks of 2010?
    - Although it's a common OLTP query, it might be to complex for the operational environment to handle
  - OLAP - the class of analytical operations running on the DW
    - In which district does a product group register the highest profit? And how did the profit change in comparison to the previous month?

I.3 Analyzing

- Data mining
  - Useful for identifying hidden patterns, e.g. customers buying wine also buy cheese
  - Useful for answering questions like: How does the typical iPad buyer look like? (for a targeted marketing campaign)
  - Methods and procedures for data mining: association rule mining, sequence pattern mining, classification, clustering, etc.

Summary

- Data Warehousing overview
  - Simplified, a data warehouse is a collective data repository built for **analytical tasks**
  - Data is extracted from the operational environment, it is transformed (and cleaned) and finally loaded into the DW
  - Typical usage scenarios of DW are **budgeting**, **resource planning**, **marketing**, etc.
  - Users of the DW are **DSS analysts** and they work **explorative**
  - Since requirements are not known at the beginning, the lifecycle of the data warehouse is almost the reverse of classical software development projects

Next Lecture

- Data Warehouse Architecture
  - Basic architectures
  - Storage models
  - Layers
  - Middleware