LUT University

LAND OF THE CURIOUS

LUT University

TABLE OF CONTENTS

CT60A4304 - BASICS OF DATABASE SYSTEMS

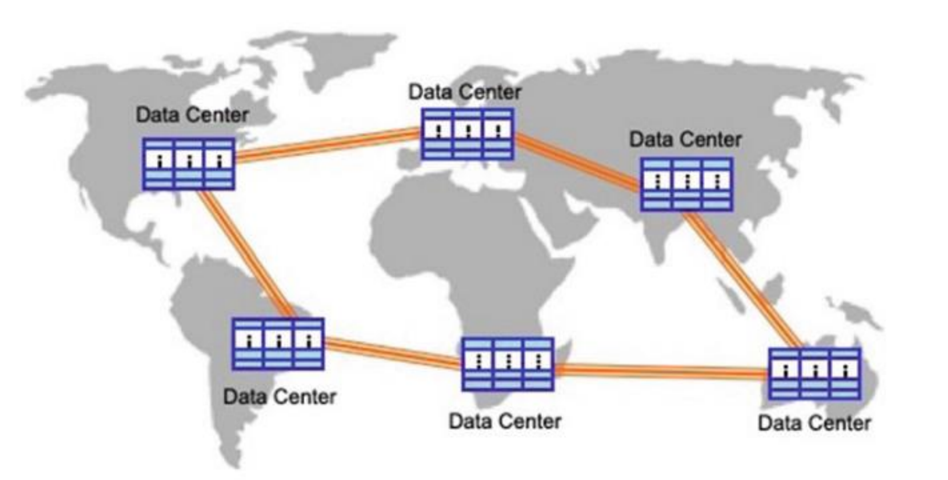
DISTRIBUTED DATABASE SYSTEMS

Lecture

Jiri Musto, D.Sc.

DISTRIBUTED DATABASE SYSTEM

- A collection of multiple, logically related databases that are managed by one software and transparent to users



Source: Özsy, M. T., Valduriez, P., Principles of Distributed Database Systems, 978-3-030-26252-5

BENEFITS AND DRAWBACKS

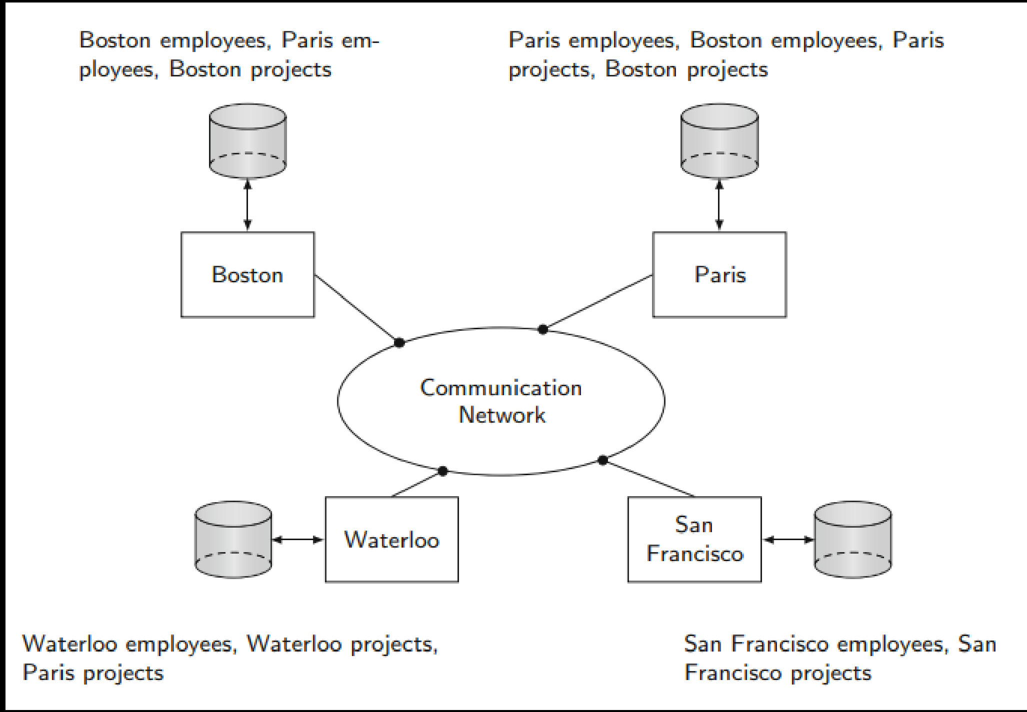
- Benefits
  - Scalability
  - Reliability
  - Improved performance
  - Transparent management
- Drawbacks
  - Data control
  - Query processing
  - Concurrency control
  - Reliability

TERMINOLOGY

- Fragmentation / partitioning
  - Both terms are used, fragmentation also has another meaning in database terminology
  - In distributed databases, the terms refer to splitting tables into smaller pieces and storing them separately
- Sharding
  - Similar to partitioning with the difference that sharding explicitly implies that the data is stored in different locations
- Replication
  - Same data is duplicated to multiple databases

DISTRIBUTED DATABASE SYSTEM ARCHITECTURE

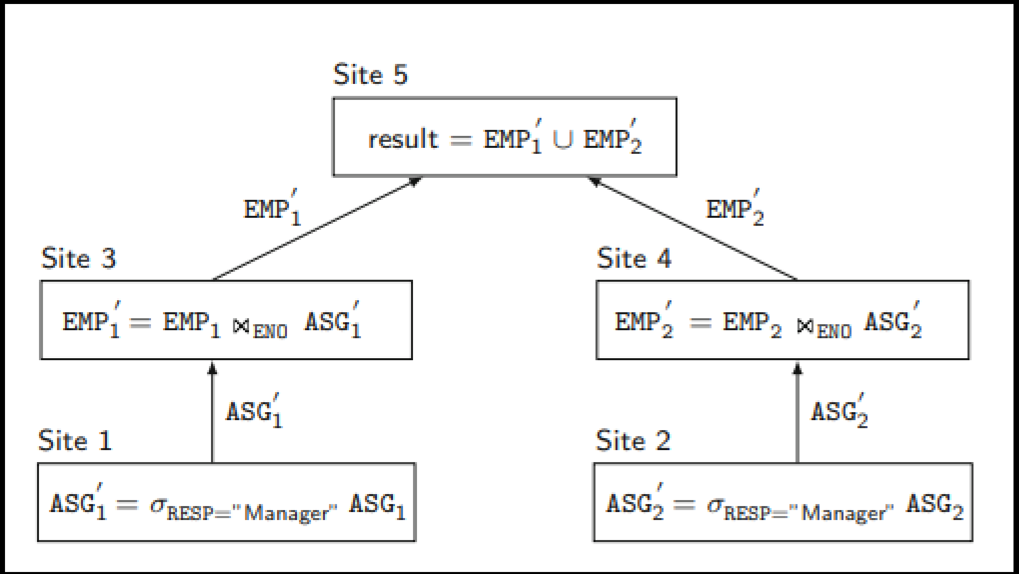
- Example of partitioning and replication

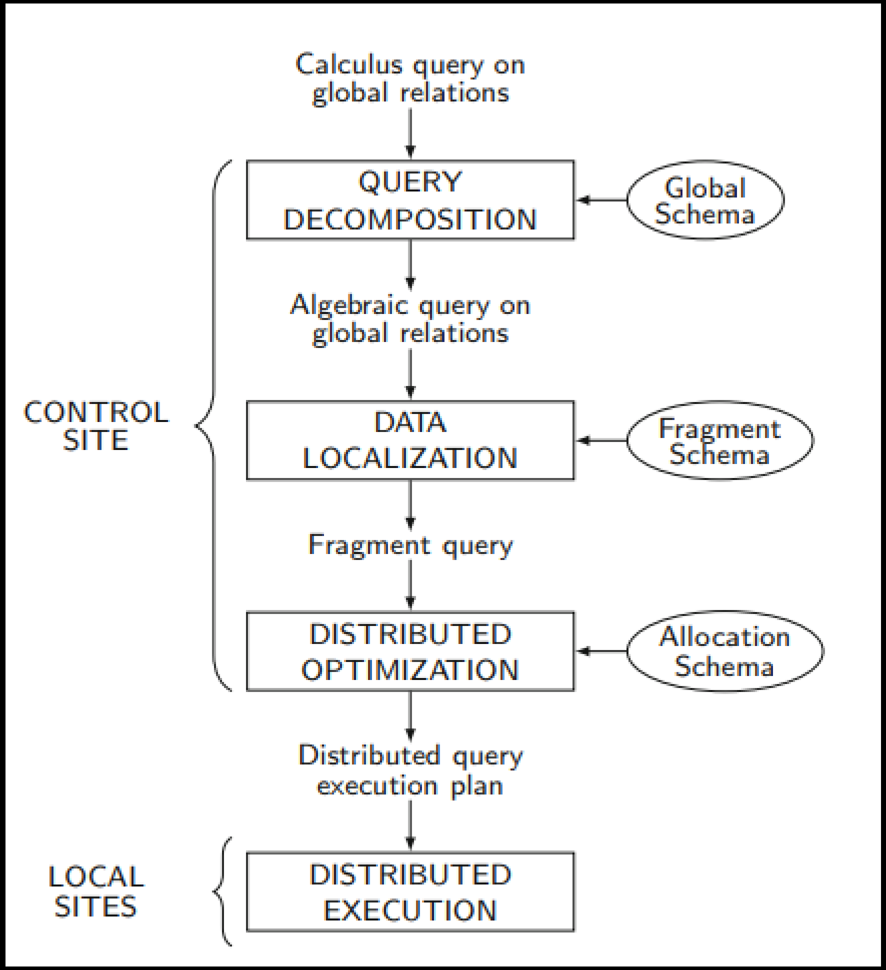


Source: Özsy, M. T., Valduriez, P., Principles of Distributed Database Systems, 978-3-030-26252-5

QUERY PROCESSING

- Queries are distributed amongst the partitions / shards / fragments





Source: Özsy, M. T., Valduriez, P., Principles of Distributed Database Systems, 978-3-030-26252-5

COMMUNICATION

- Communication between databases is necessary and there are different communication protocols
- One point always acts as the ***coordinator*** that makes sure all involved databases are ready and the query can be processed
- Depending on the communication, there are different failure/termination protocols as well

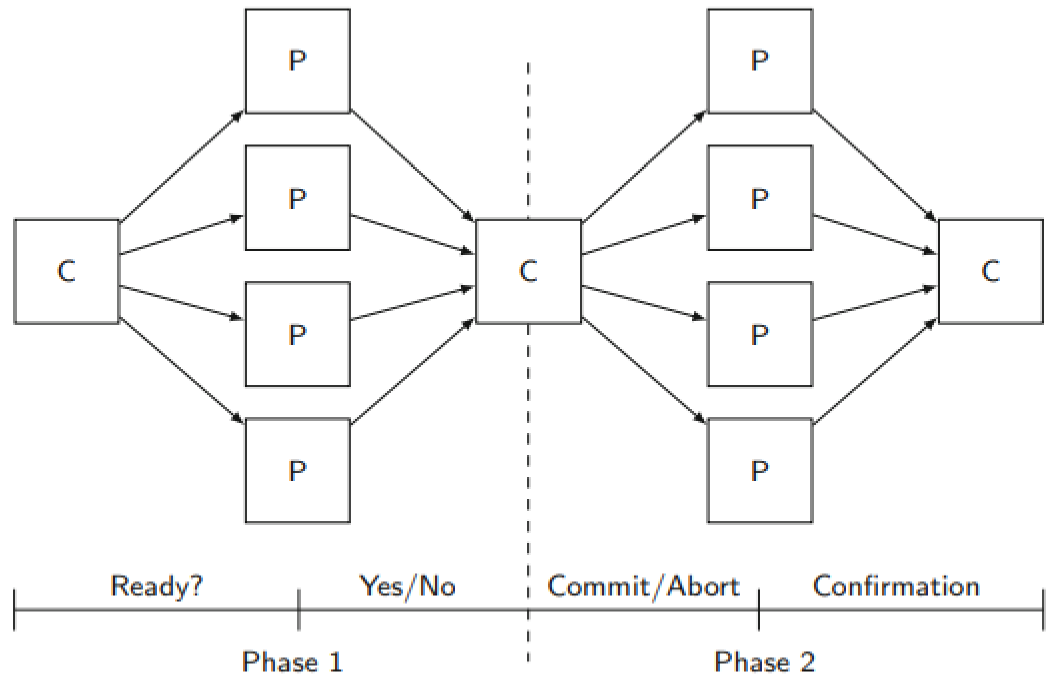


Fig. 5.12 Centralized 2PC communication structure. C: Coordinator, P: Participant

Source: Özsy, M. T., Valduriez, P., Principles of Distributed Database Systems, 978-3-030-26252-5

DEADLOCKS

- Deadlock can happen in normal transactions
- In one database, the deadlocks are less common
- In distributed databases, amount of deadlocks increase because of delays between the databases

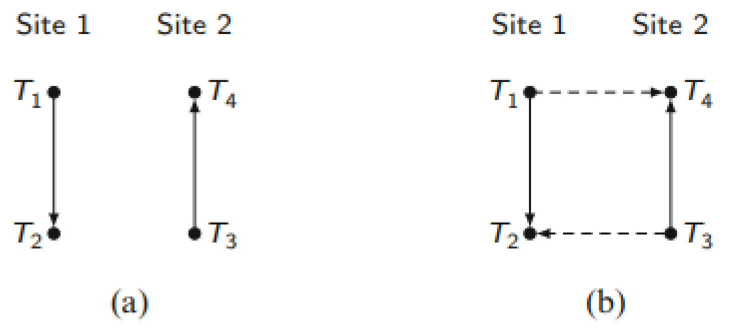
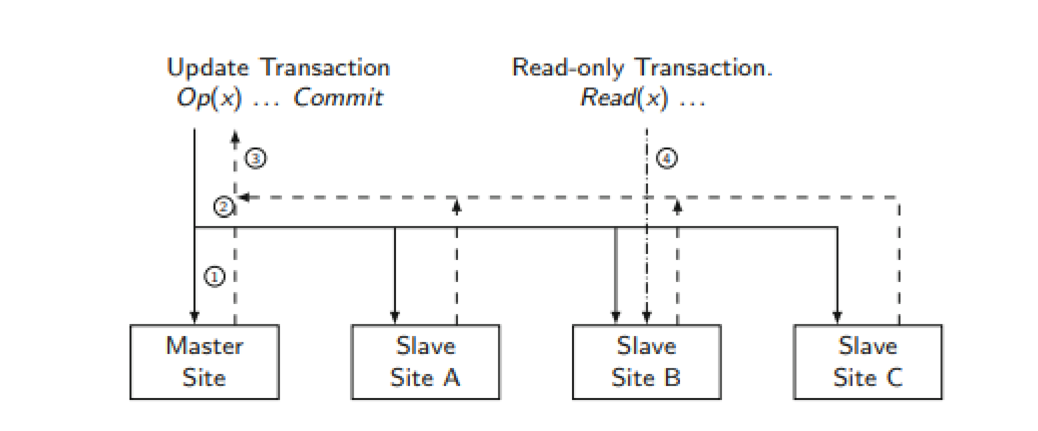


Fig. 5.5 Difference between LWFG and GWFG

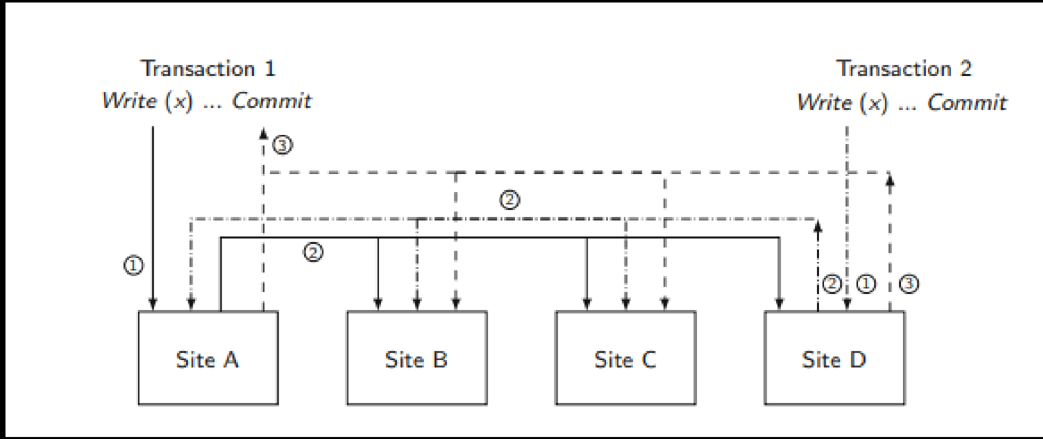
Source: Özsy, M. T., Valduriez, P., Principles of Distributed Database Systems, 978-3-030-26252-5

DATA REPLICATION

- When data is replicated, all changes to data has to be propagated to all other databases
- Generally, there is at least one “master” replica and others are “slave” replicas
  - Different variations exists



Eager single master replication protocol

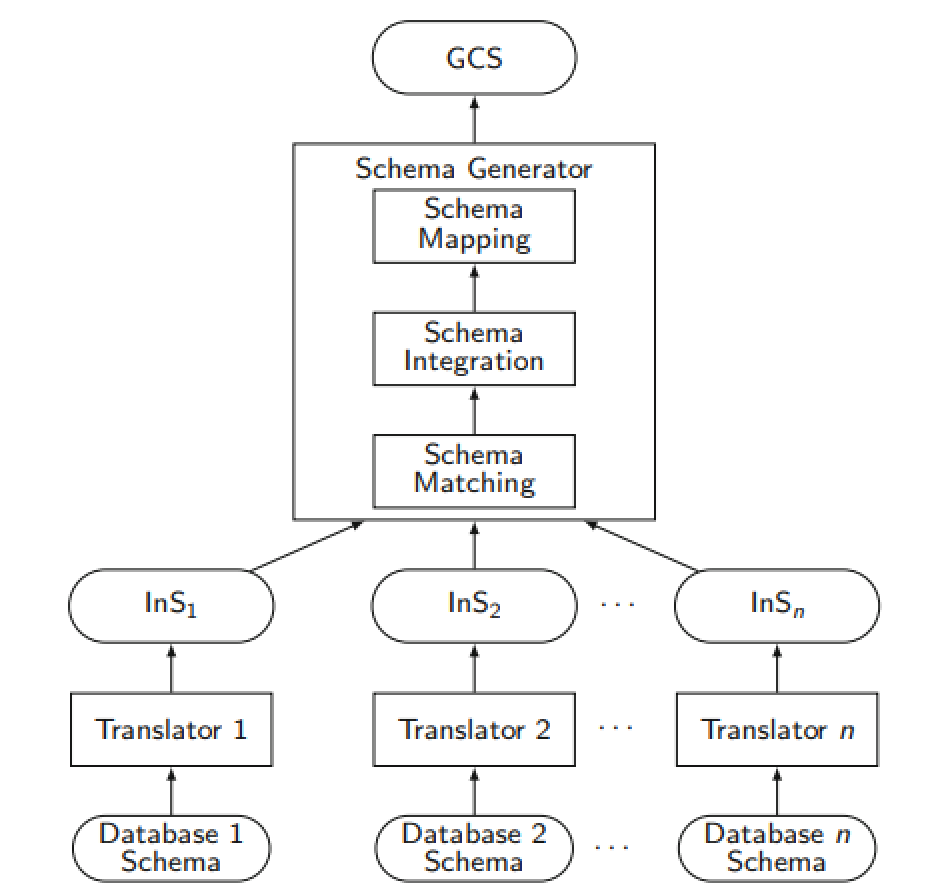


Eager distributed replication protocol

Source: Özsy, M. T., Valduriez, P., Principles of Distributed Database Systems, 978-3-030-26252-5

DATABASE INTEGRATION

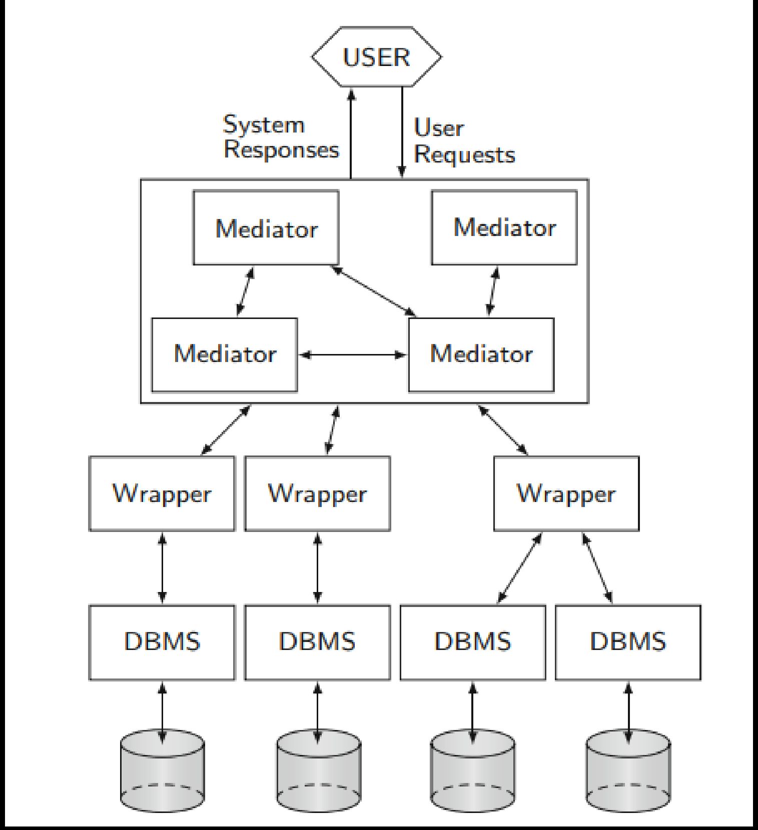
- Integrating different databases together requires that the database schemas can be matched together
- Two ways: Bottom-up or top-down
- Bottom-up
  - Each local database schema is translated to a intermediate schema which are then matched an combined into a global schema
- Top-down follows the normal distributed database design



Source: Özsy, M. T., Valduriez, P., Principles of Distributed Database Systems, 978-3-030-26252-5

MULTIDATABASE SYSTEMS

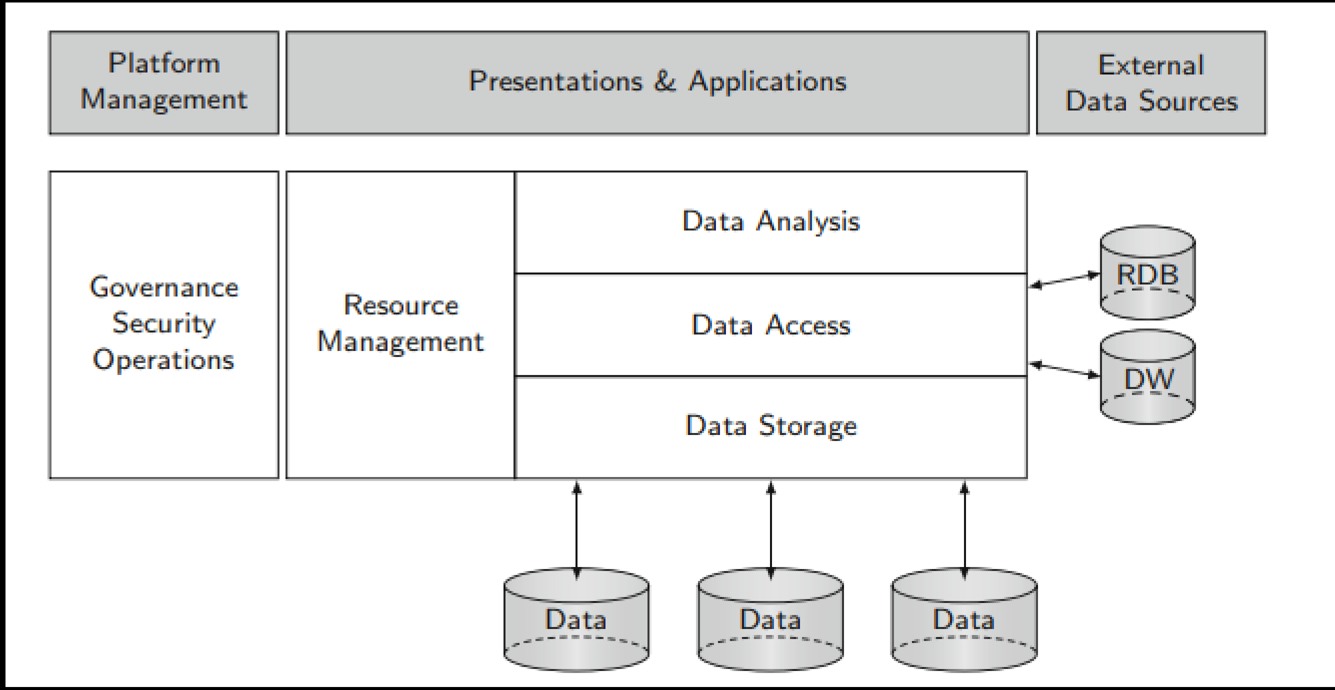
- Multiple databases that may have different data
- Wrappers make sure data is in specific format
- Mediators handle queries and distribute them into specific databases



Source: Özsy, M. T., Valduriez, P., Principles of Distributed Database Systems, 978-3-030-26252-5

DATA LAKES

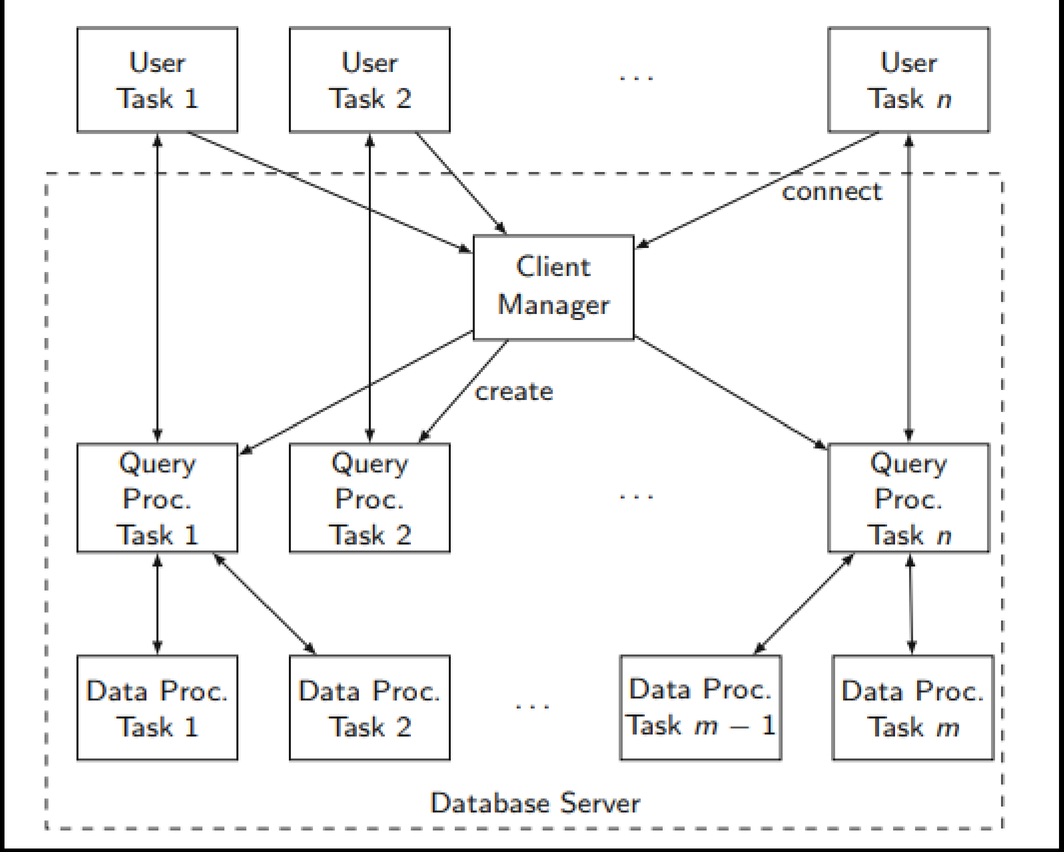
- Data lakes are used in big data applications
- Similar to data warehouse but the data is stored in its "natural" format
  - Data parsing done during query
- Lacks consistency and quality



Source: Özsy, M. T., Valduriez, P., Principles of Distributed Database Systems, 978-3-030-26252-5

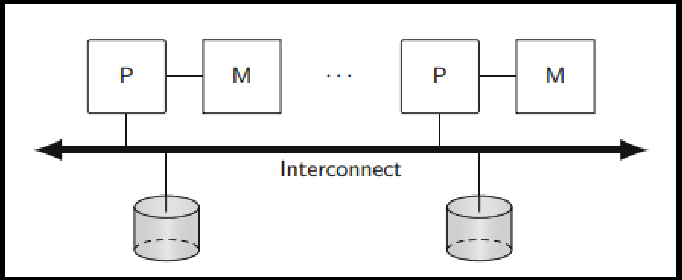
PARALLEL DATABASE SYSTEM

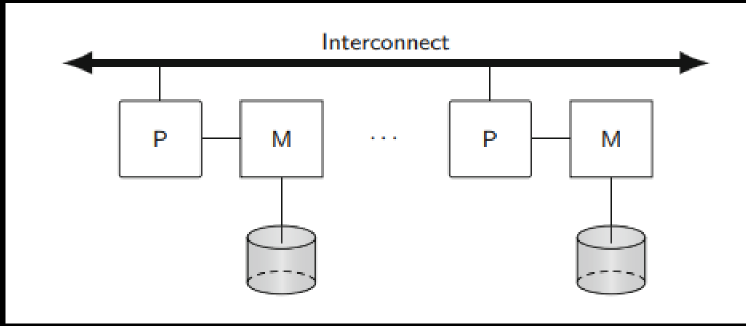
- A distributed database system on parallel computers
- Main purpose is to improve performance
  - I/O bottleneck
- Useful in:
  - Online transaction processing (OLTP)
  - Decision support systems (DSS)
  - Parallel query processing

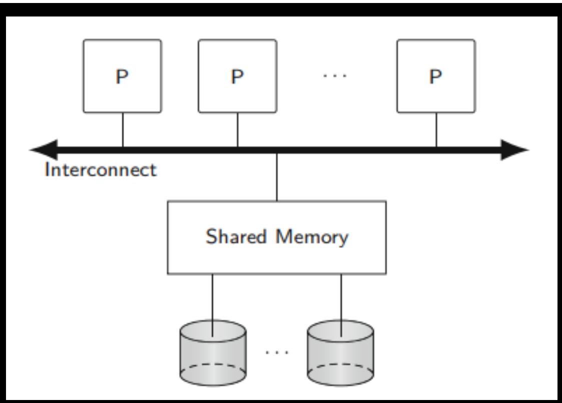


PARALLEL ARCHITECTURE

- Shared disk
  - Processors have access to a shared disk unit but memory modules are independent
  - Requires a lock manager for global cache consistency
- Shared nothing
  - Each processor has its main memory and disk. Similar to a distributed system
  - Best cost/performance ratio
- Shared memory
  - Processors have access to any memory module or disk unit
  - Main advantage is simplicity







CT60A4304 - BASICS OF DATABASE SYSTEMS

DATA WAREHOUSES

Lecture

Jiri Musto, D.Sc.

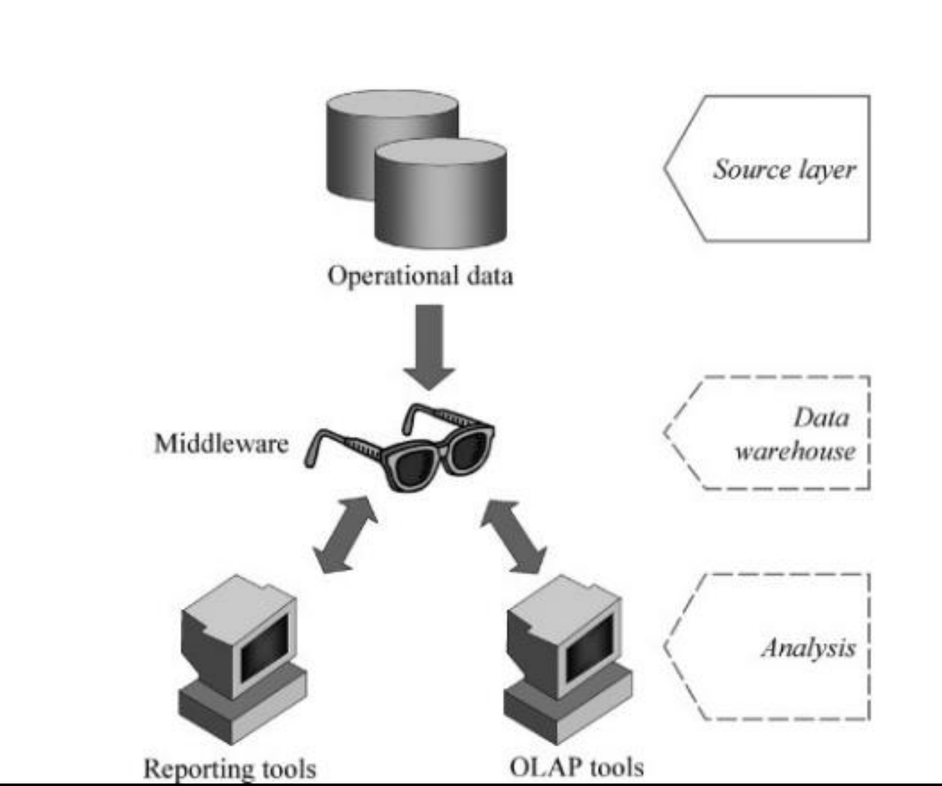
DATA WAREHOUSE

- Data warehouses are a type of distributed database system
- Used extensively in business analytics and business intelligence
- Data warehouses collect data from multiple sources and combines them into one pile
- Online analytical processing (OLAP), online transactional processing (OLTP) and decision support systems (DSS) are the main uses for data warehouses
- Data warehousing could be defined as "*A collection of methods, techniques, and tools to support knowledge workers and analysts to conduct data analyses that help performing decision making processes and improving information resources*"

Source: Golfarelli, M., Rizzi, S., Data Warehouse Design, Modern Principles and Methodologies, 978-0-07-161040-7

DATA WAREHOUSE ARCHITECTURE: SINGLE-LAYER

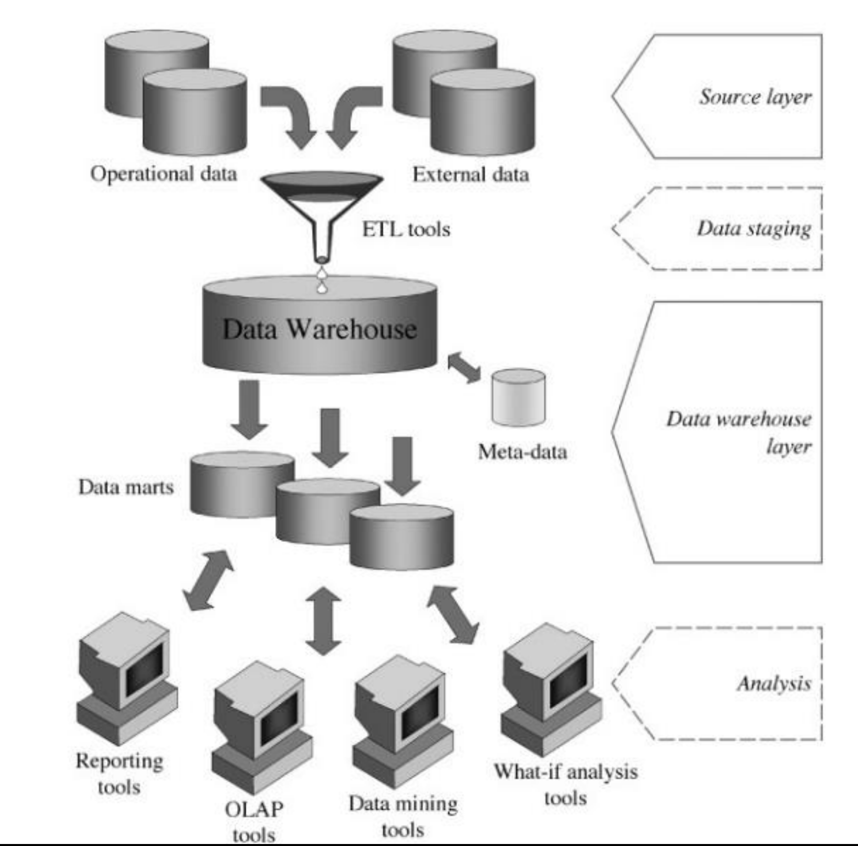
- Single-layer architecture is the simplest version of a data warehouse architecture
- Data source is (generally) internal
- No need to curate, clean or transform the data
- Analysis queries are interpreted by the middleware
  - No need to create separate data marts
- Not really used in practice
  - Only useful if analysis is extremely restricted



Source: Golfarelli, M., Rizzi, S., Data Warehouse Design, Modern Principles and Methodologies, 978-0-07-161040-7

DATA WAREHOUSE ARCHITECTURE: TWO-LAYER

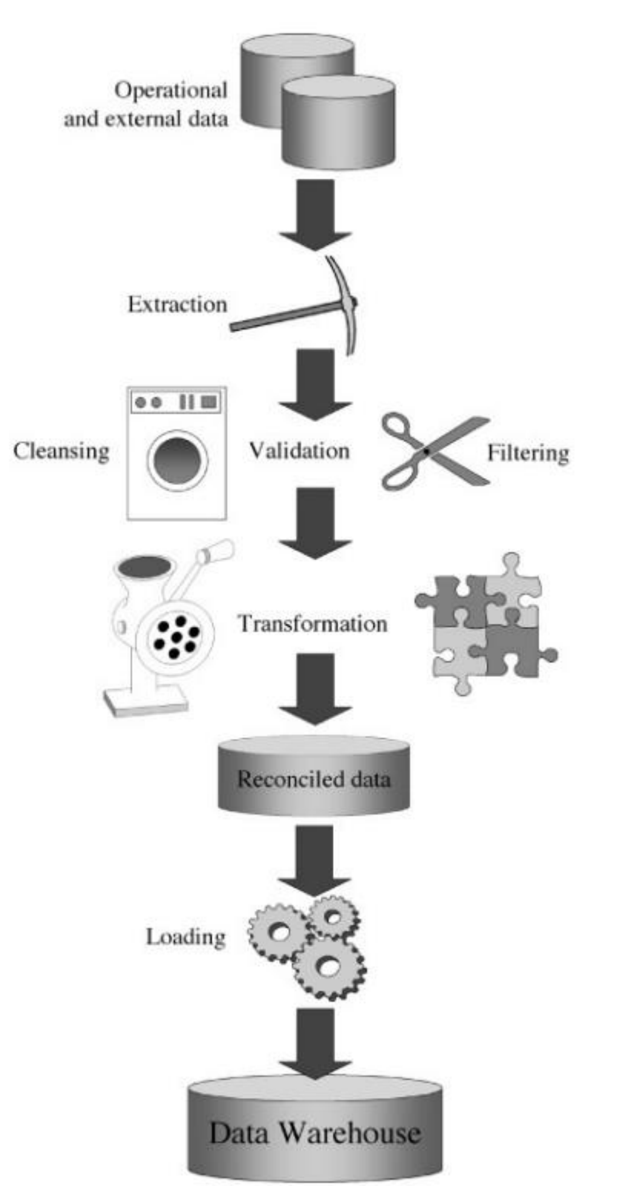
- Data sources are both internal and external
  - Data needs to be curated and transformed with ETL tools
- More common approach for data warehouses
  - Three-layer architecture separates a reference data model after the ETL-processes
- Data warehouse can be directly accessed instead of creating smaller data marts
- There are other possible architectures as well
  - Independent data marts
  - Hub-and-spoke
  - Federated architecture



Source: Golfarelli, M., Rizzi, S., Data Warehouse Design, Modern Principles and Methodologies, 978-0-07-161040-7

EXTRACTION, TRANSFORMATION, LOADING (ETL)

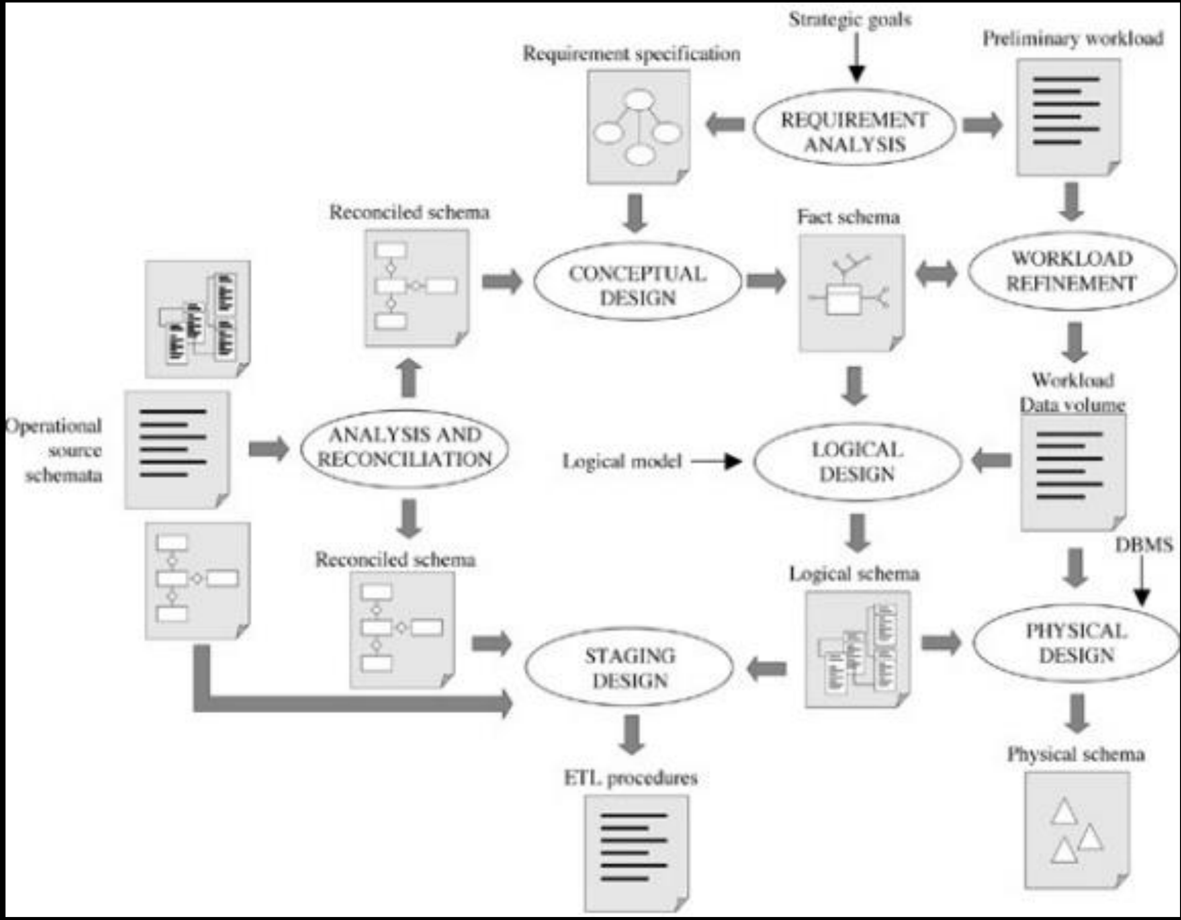
- Extraction
  - Static extraction for initializing, incremental extraction for updates
  - Can be source-driven (when changes happen in the source data)
- Cleansing
  - Duplicates, inconsistencies, missing or unexpected data, etc.
- Transformation (closely related to cleansing)
  - Conversion of data from the source format to the data warehouse format
  - Conversion, matching and selection operations are done
- Loading
  - Refresh: Rewrite the whole data warehouse
  - Update: Modify existing data based on changes in the source data



Source: Golfarelli, M., Rizzi, S., Data Warehouse Design, Modern Principles and Methodologies, 978-0-07-161040-7

DATA MART DESIGN PHASE

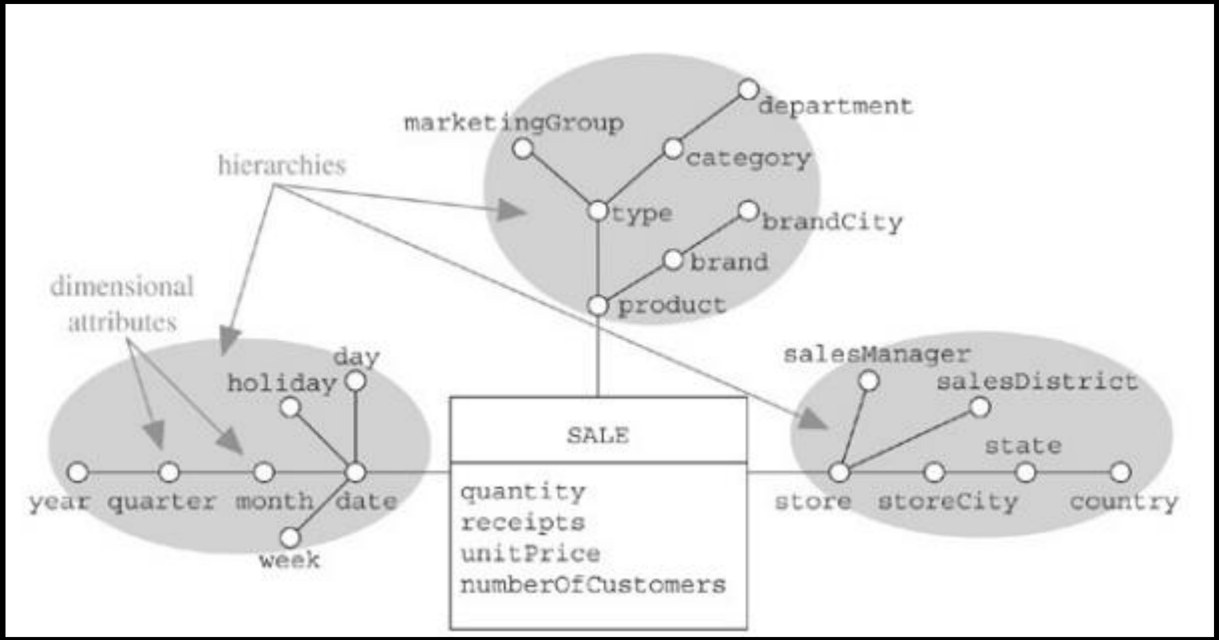
- There are seven phases to designing a data mart
- Each phase involves different people within the company
  - End users, designers, database administrators
- There are different approaches to designing
  - Data-driven
  - Requirement-driven



Source: Golfarelli, M., Rizzi, S., Data Warehouse Design, Modern Principles and Methodologies, 978-0-07-161040-7

DIMENSIONAL FACT MODEL

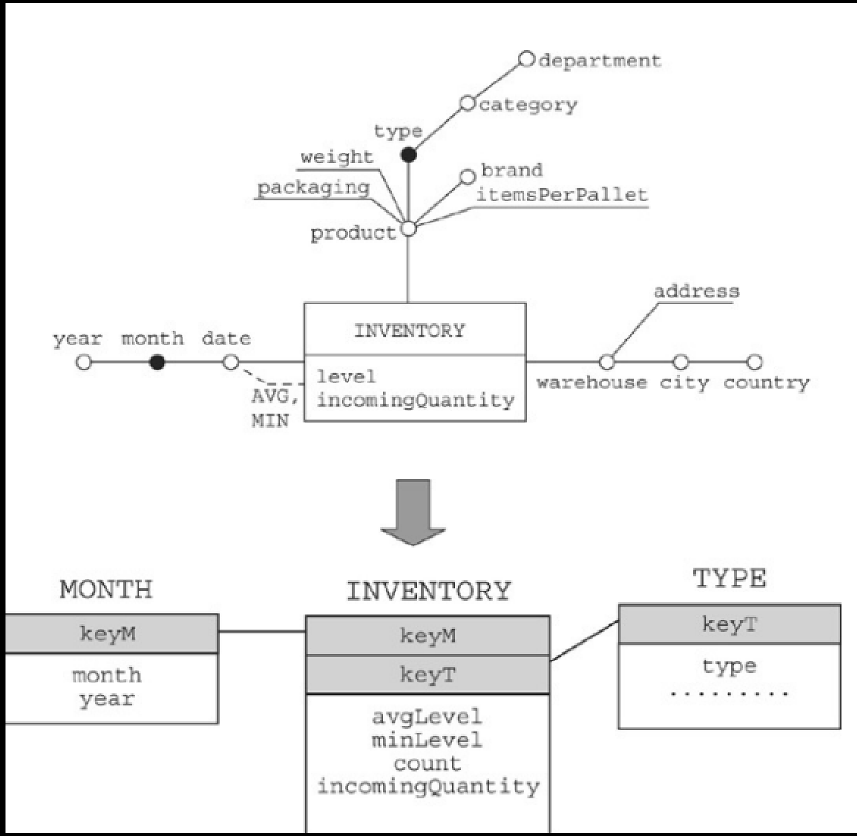
- A conceptual model created for data mart design
- Each DFM has **fact(s)** that the data mart is designed for and the fact has dimensions and attributes
- Fact has measures that are relevant for analysis



Source: Golfarelli, M., Rizzi, S., Data Warehouse Design, Modern Principles and Methodologies, 978-0-07-161040-7

DIMENSIONAL FACT MODEL TO DATABASE SCHEMA

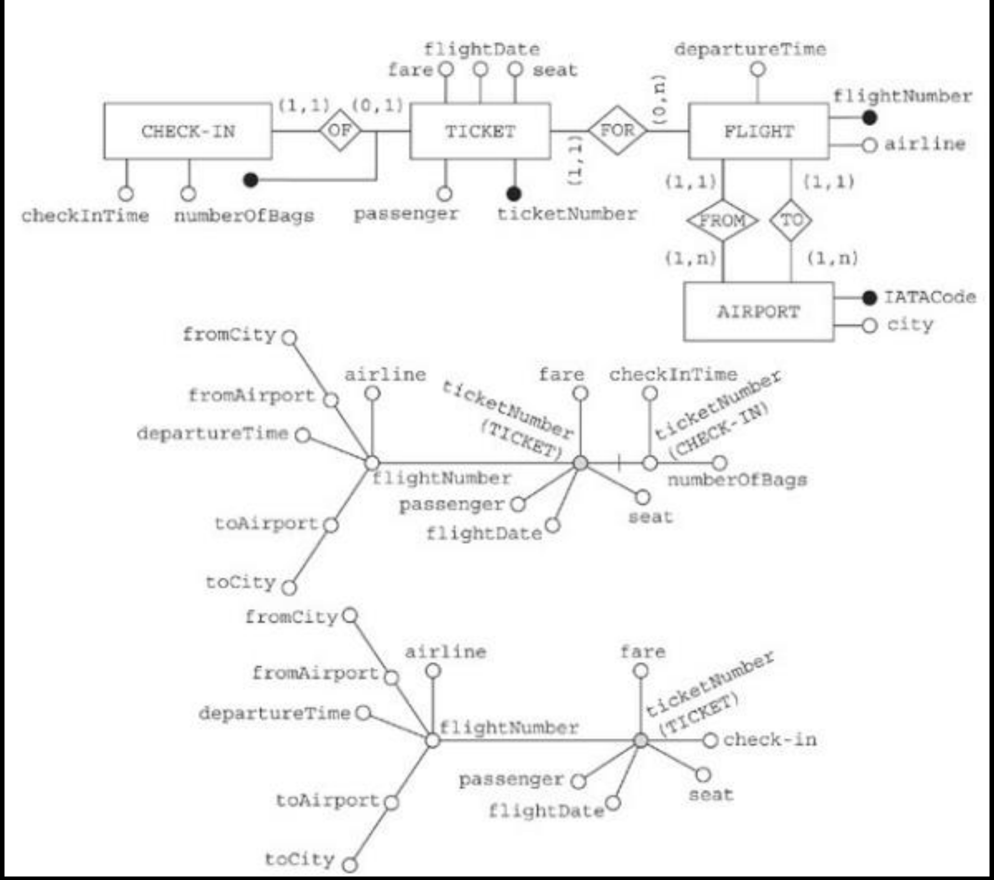
- Fact model can be transformed to an ER-model or a database schema
  - In ER-model, fact is a relationship with attributes
- Fact(s) and dimensions are transformed into their own tables
- Connecting attributes are transformed to attributes of the tables



Source: Golfarelli, M., Rizzi, S., Data Warehouse Design, Modern Principles and Methodologies, 978-0-07-161040-7

ATTRIBUTE TREE

- An attribute tree is basically a DFM in a different format
- Attribute tree is used to modify the design by adding or removing attributes
  - If a detailed attribute is needed
  - If a more general attribute is needed
  - If an attribute is repetitive
- An attribute tree (like DFM) can be transformed into an ER-model



Source: Golfarelli, M., Rizzi, S., Data Warehouse Design, Modern Principles and Methodologies, 978-0-07-161040-7

CT60A4304 - BASICS OF DATABASE SYSTEMS

DATA QUALITY

Lecture

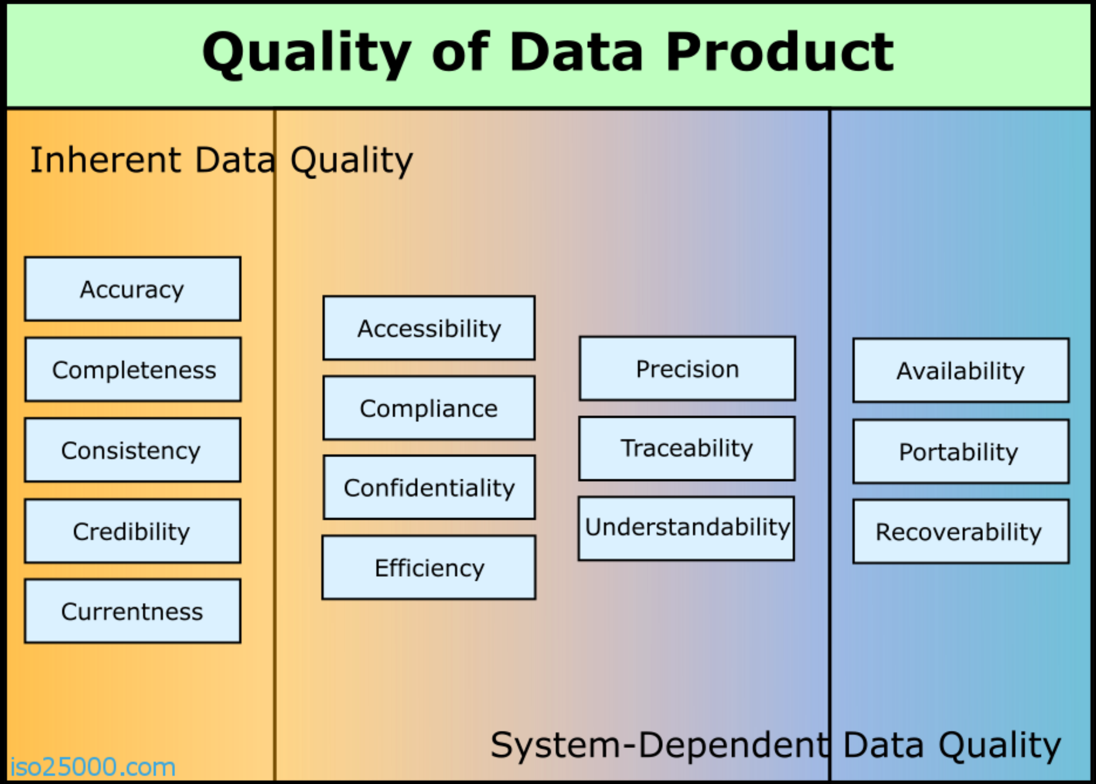
Jiri Musto, D.Sc.

DATA QUALITY IN GENERAL

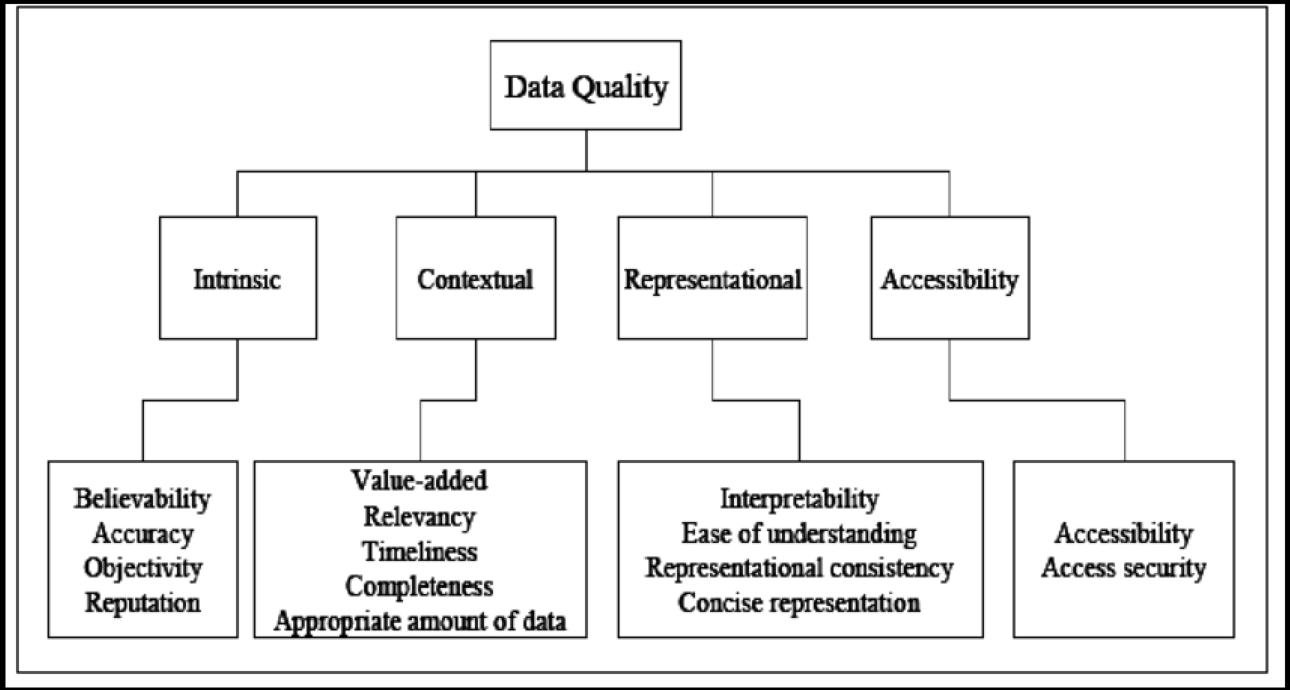
- Data quality is an abstract term
  - Quality can mean different things for each person
- Needs to be defined by dividing quality into separate characteristics / dimensions
  - Accuracy, completeness, precision, credibility, etc.
- Data quality is defined for each case specifically
  - Each case specifies what characteristics/dimensions to use and what to emphasize
- There are multiple definitions and standards for data quality
  - ISO has at least three different data quality standards for different domains

ISO/IEC 25012

ISO/IEC 25012 DATA QUALITY MODEL

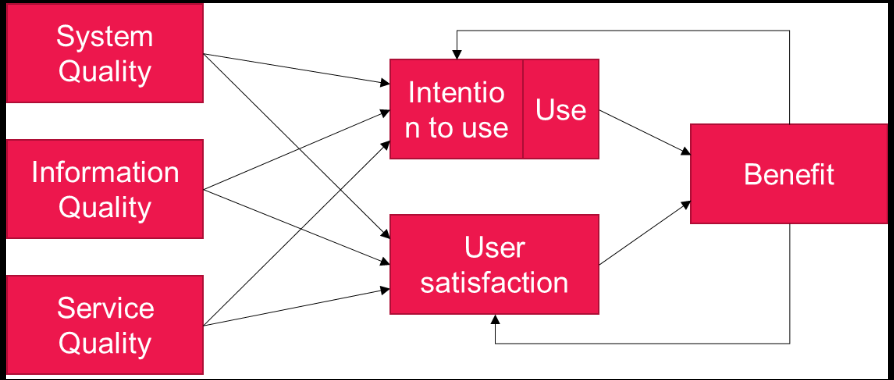


WANG AND STRONG, 1996, BEYOND ACCURACY: WHAT DATA QUALITY MEANS TO DATA CONSUMER



WHY QUALITY IS IMPORTANT

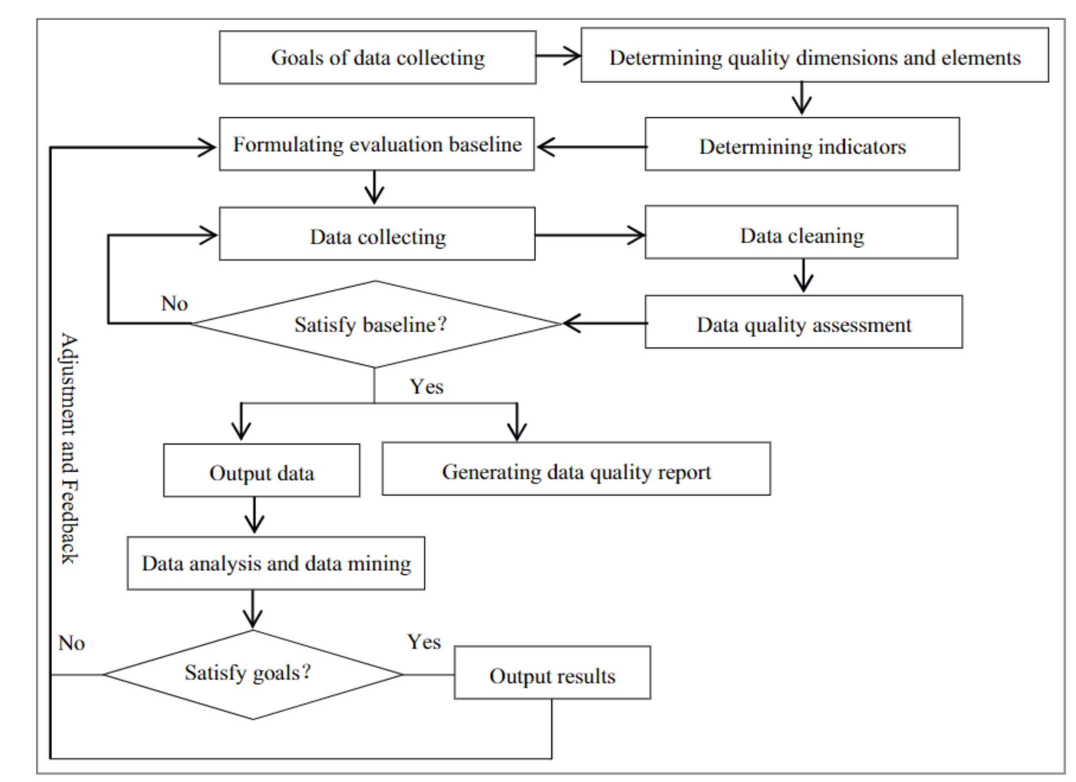
- Many decision are based on data
  - If data quality is poor, decision is poorly justified
  - Low data quality leads to low information quality
- If data comes from unknown sources, the quality of data is unknown
  - How credible the source is?
- Low quality data can cost trillions of USD each year



ERRORS IN QUALITY

- Systemic errors
  - Based on data exploration, data re-usage, data merging, falsifications, biases, wrong models
- Systematic errors
  - Due to abstractions, restrictions in accessible data,dirty data, approximations, computations, wild integration, missing provenance
- Stochastic errors
  - Based on assumptions for occurrence of errors, their distribution functions and their contribution within the model and to the variables
- Modelling errors
  - Quantitative instead qualitative, empiricism, muddling, data first and thought later

ASSESSING QUALITY: PROCESS



Cai and Zhu: The Challenges of Data Quality and Data Quality Assessment in the Big Data Era

ASSESSING QUALITY: MEASURES

- Quality of data can be measured numerically
- Each characteristic/dimension needs to be individually evaluated
- Some are more simple to evaluate, others are more complex
- For example:
- $syntactic\ accuracy = \frac{\sum_i^K closeness\ (w_i, V)}{K}$
- Semantic accuracy using object identification: $\langle \alpha, \beta \rangle \in \begin{cases} M\ if\ p(M|\underline{x}) \geq p(U|\underline{x}) \\ U\ otherwise \end{cases}$
- $currentness = Age + (DeliveryTime - InputTime)$

POSSIBLE SOLUTIONS FOR QUALITY

- Cleaning, filtering, repairing of data
  - Ensures that data fulfills data quality requirements
  - Makes modifications to data → may lead to wrong results when using during analysis
- Discarding low-quality data
  - Ensures that used data fulfills data quality requirements
  - Reduces the amount of data available and may lead to skewed results
- Evaluating data quality and storing the information
  - Data quality information can be stored in metadata (data over data)
  - Can do analysis with varying levels of data quality (can choose the required quality)
  - Expensive resource-wise, may be difficult to fully implement


LUT
University