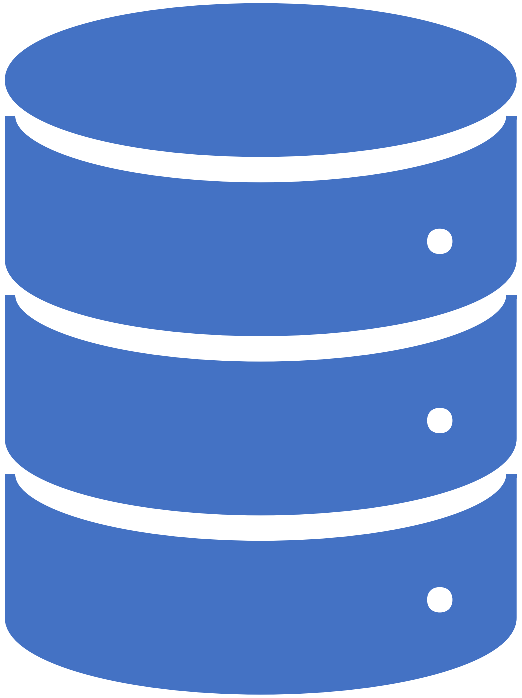


PA220: Database systems for data analytics

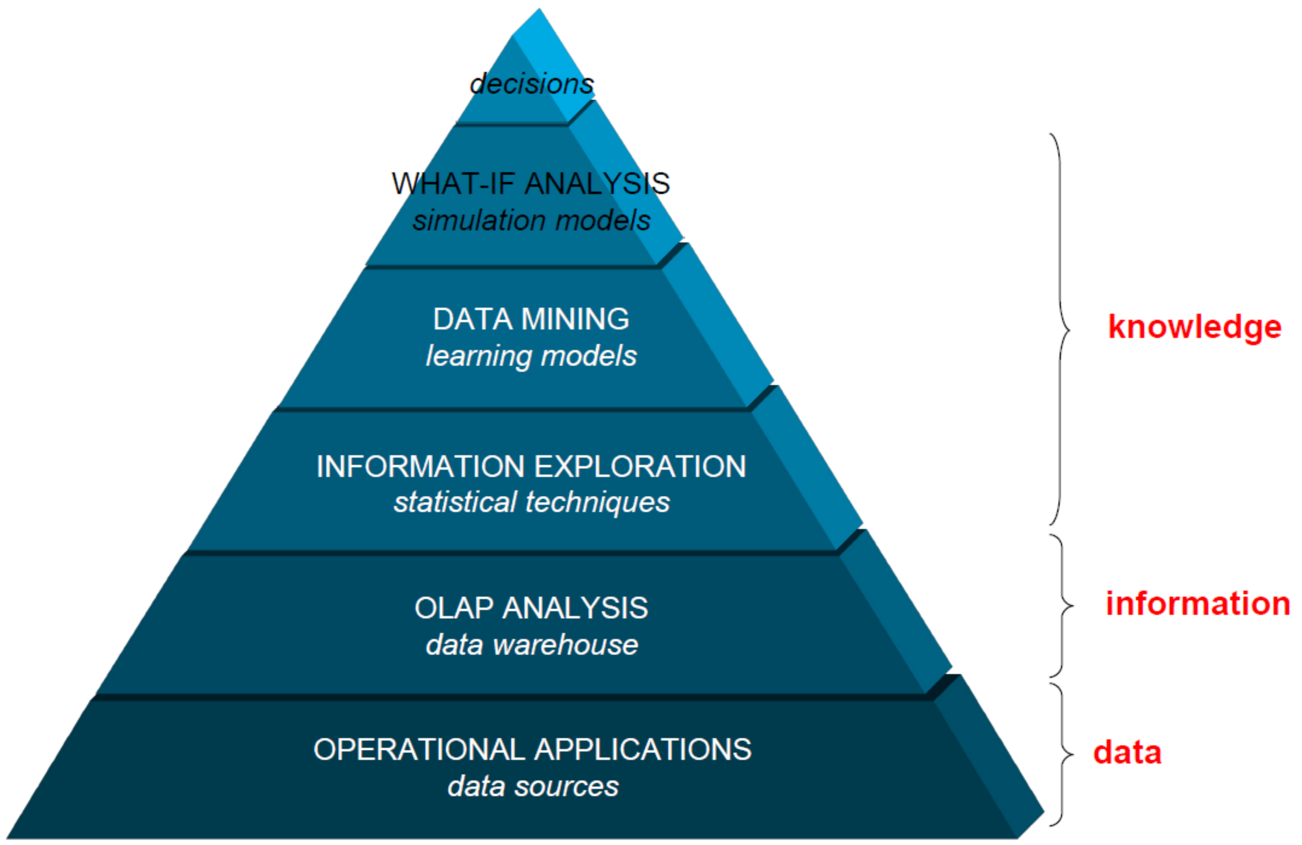
Introduction, Business Intelligence, Data Warehouse

Overview of data warehousing

- motivation
  - business intelligence / data analytics
- data warehouse
  - architecture
  - oltp vs. olap
  - big data

Motivation

- Data production
  - Information systems
  - Monitoring services
  - Sensors, GPS tracking
  - Social networks
- Data processing
  - Storage & archiving
  - Summarization
  - Reporting
  - Visualization
  - Insights
  - Predictions



Business Intelligence pyramid

Business Intelligence

- a process of analyzing data and presenting results to business managers
  - to make informed decisions
- tools and applications
  - to collect data
  - to prepare it for storage and analysis
  - to develop and run queries
  - to create reports and dashboards
  - to visualize data
- evolved from decision support systems
- business analytics / (advanced) data analytics
  - prescriptive analytics

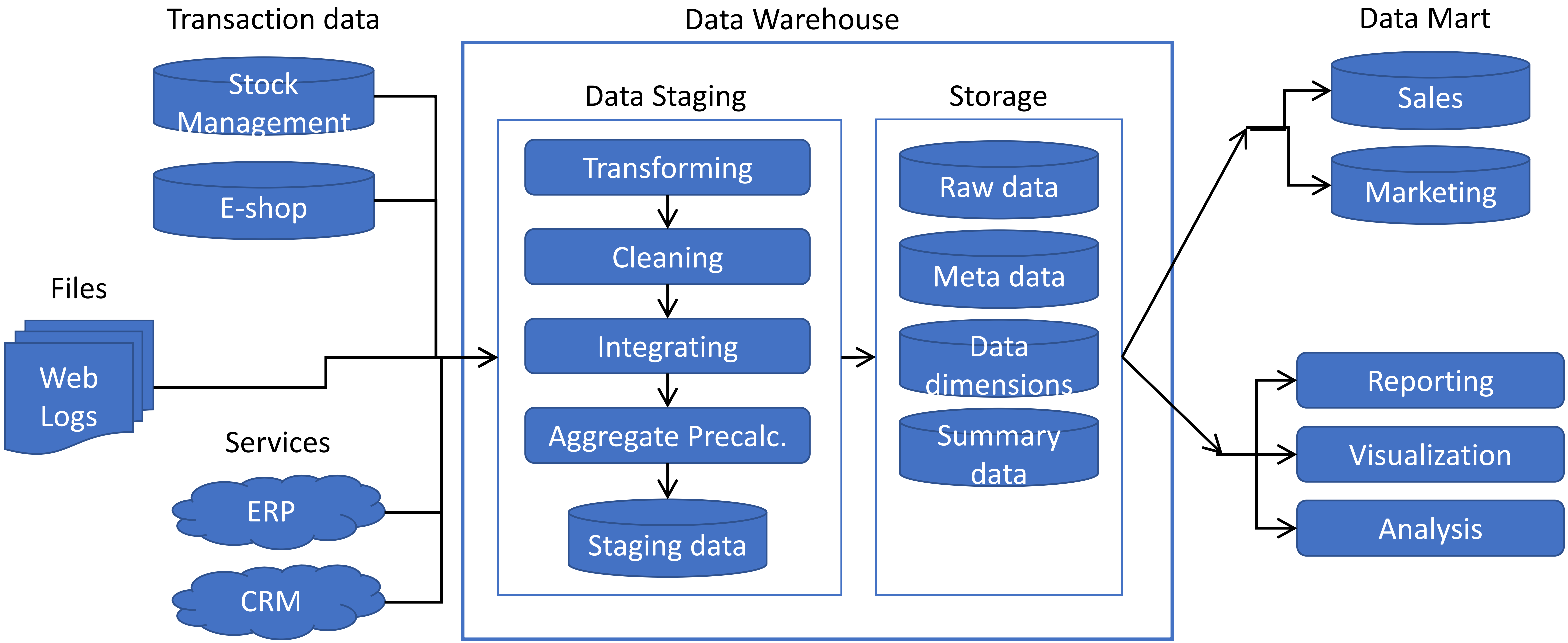
Business Intelligence vs. Artificial Intelligence

- BI is the “opposite” of Artificial Intelligence (AI)
  - AI systems make decisions for the users
  - BI systems help users make the right decisions, based on the available data
  - Many BI techniques have roots in AI, though.

Business Intelligence: Key Problems

1. Complex and unusable models
   - Many DB models are difficult to understand
   - DB models do not focus on a single clear business purpose
2. Same data found in many different systems
   - Example: customer data in many different systems
   - The same concept is defined differently
3. Data is suited for operational systems
   - Accounting, billing, etc.
   - Do not support analyses across business functions
4. Data quality is bad
   - Missing data, imprecise data, different use of systems
5. Data are ”volatile”
   - Data deleted in operational systems (6 months)
   - Data change over time – no historical information

Business Intelligence Architecture



Data Warehouse

- Typically:
  - One large repository for entire company
  - Dedicated hard- and software
    - Enterprise-grade DBMS
    - Often: database appliances (e.g., Teradata, Oracle Exadata, IBM Netezza, …)
- Goal:
  - Single source of truth for analysis and reporting
  - Requires data cleansing and conflict resolution

Data Warehouse Users

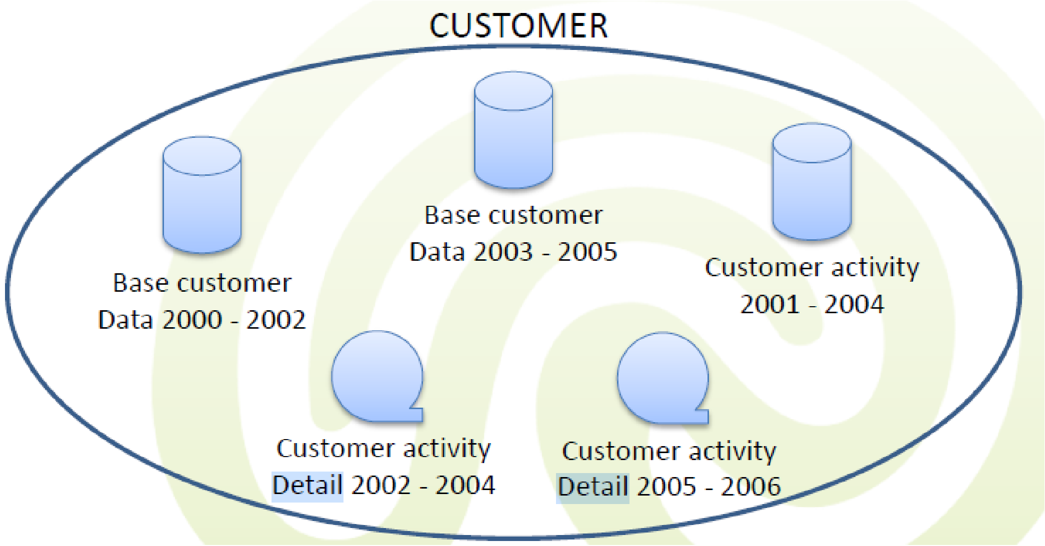
- Business analysts:
  - Explore data to discover information
  - Use for decision making -> “Decision Support System (DSS)”
- Consequences:
  - Workloads and access patterns not known in advance
  - For exploration, data representation must be easy to understand (even by business analysts)
  - Design and usage driven by data, not applications

Definition of Data Warehouse

- Barry Devlin, IBM Consultant
  - A data warehouse is simply a **single**, **complete**, and **consistent** store of data obtained from a **variety** of sources and made available to end users in a way they can **understand** and **use** it in a **business context**.
- Ralph Kimball, The Data Warehouse Toolkit
  - A **copy** of transaction data specifically **structured** for query and **analysis**.
- W. H. Inmon, Building the Data Warehouse
  - A data warehouse is a **subject-oriented**, **integrated**, **time-varying**, **non-volatile** collection of data that is used primarily in organizational **decision making**.

Definition of Data Warehouse (1)

- 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,…

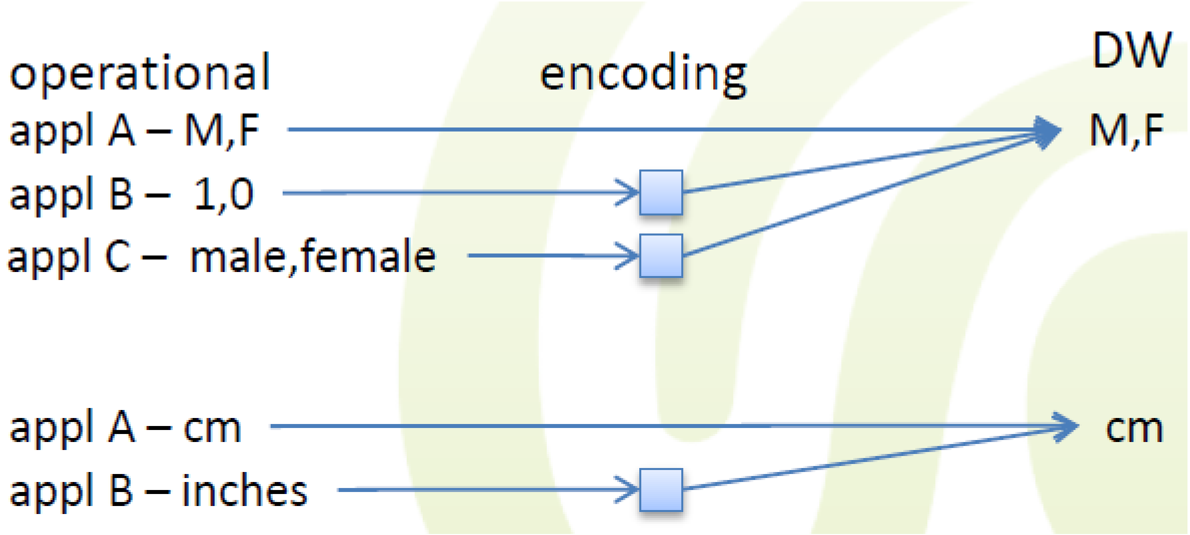


Base tables hold original data and can be different in schema.

Customer activity represents summary (derived) information.

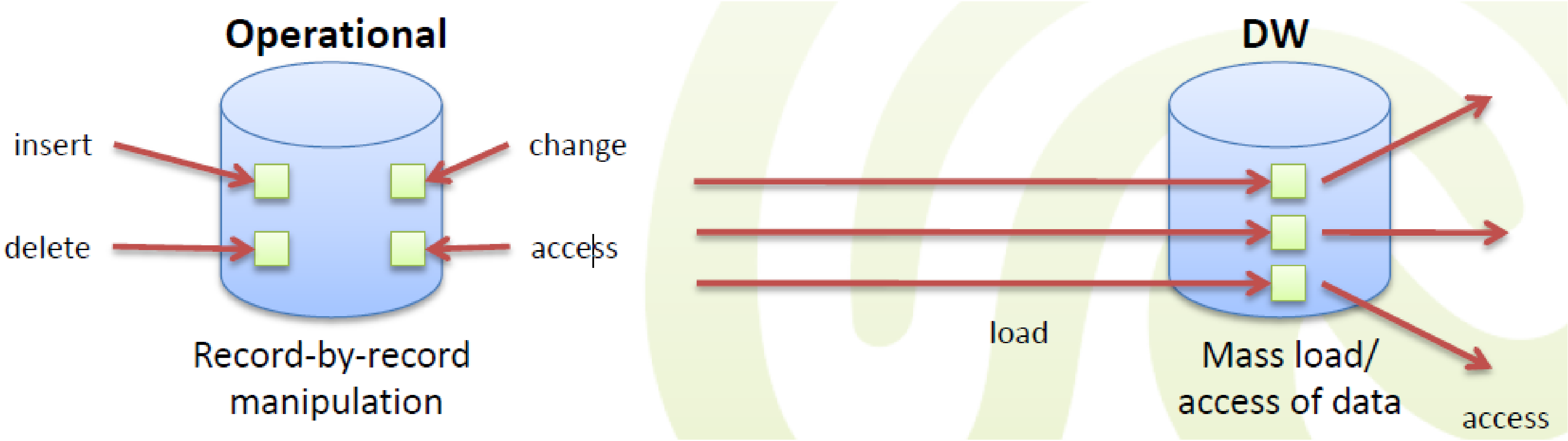
Definition of Data Warehouse (2)

- 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,…



Definition of Data Warehouse (3)

- 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



Definition of Data Warehouse (4)

- 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 daytime horizon is normal, DWs have a 5-to-10-year horizon

General DW Definition

- A large repository of some organization's electronically stored data
- Specifically designed to facilitate reporting and analysis
- Requirements:
  - information easily accessible
  - consistent information
  - present information timely
  - protect the information
  - adapt to change
  - accepted by users

Example: Oracle Exadata X8-2

- Up to 912 CPU cores and 28.5TB memory per rack for database processing
  - Up to 576 CPU cores per rack dedicated to SQL processing in storage
  - From 2 to 19 Database Servers per rack; From 3 to 18 Storage Servers per rack
- Up to 3.0 PB of disk capacity (raw) per rack
  - Up to 920 TB of flash capacity (raw) per rack
  - Ability to perform up to 4.8M 8K database read I/O operations,
  - or 4.3M 8K flash write I/O operations per second per full rack
- Hybrid Columnar Compression (10-15x compression ratio)
- 40 Gb/second (QDR) InfiniBand Network
  - Uncompressed I/O bandwidth of up to 560 GB/second per full rack from SQL
- Complete redundancy for high availability
- Scale by connecting by InfiniBand up to 18 racks

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 milion Sun E10000 multi-CPU server, a $500 000 Teradata license, a book "Database Design for Smarties", and builds a normalized SQL data model

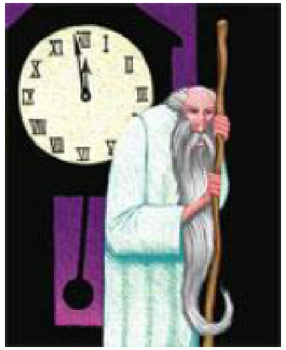
1.1 Use Case

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 computer up for 20 minutes

1.1 Use Case

- 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.*

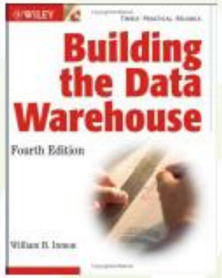
1.1 Use Case

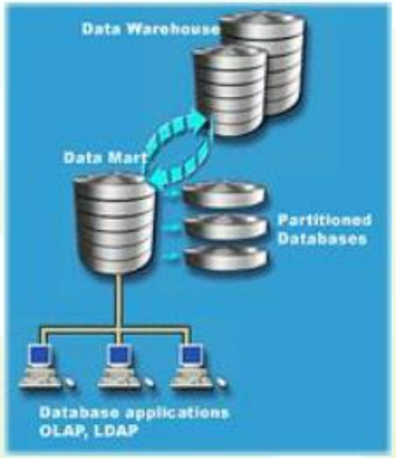
Detour

- 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…





16.09.2021

(Enterprise) Data Warehouse

- integrates data from more sources
- subject-oriented
- stores current and historical data
- answers multidimensional queries
- create aggregated data reports
- is it a relational DBMS?
  - on-line transaction processing (OLTP)
  - on-line analytical processing (OLAP)
    - MOLAP (Multidimensional OLAP)
    - ROLAP (Relational OLAP)

OLTP		DW
few	Indexes	many
many	Joins	some
normalized	Schema	denormalized
rare	Derived data and aggregates	common
update/select	Workload	select
predefined	Typical operations	ad hoc
few days	Historical data	years
indexing / hashing	Data access method	full scans
GB	Data volume	TB
Thousands	Users	Hundreds

OLTP (Online Transaction Processing)

- Day-to-day business operations
  - Mix of insert, update, delete, and read operations
  - e.g., enter orders, maintain customer data, etc.
- System sometimes called operational data store (ODS)
  - Up-to-date state of the data
- From a database perspective:
  - Short-running operations
  - Most queries known in advance
  - Often point access, usually through indexes
  - write access ⇝ ACID principles

OLAP (Online Analytical Processing)

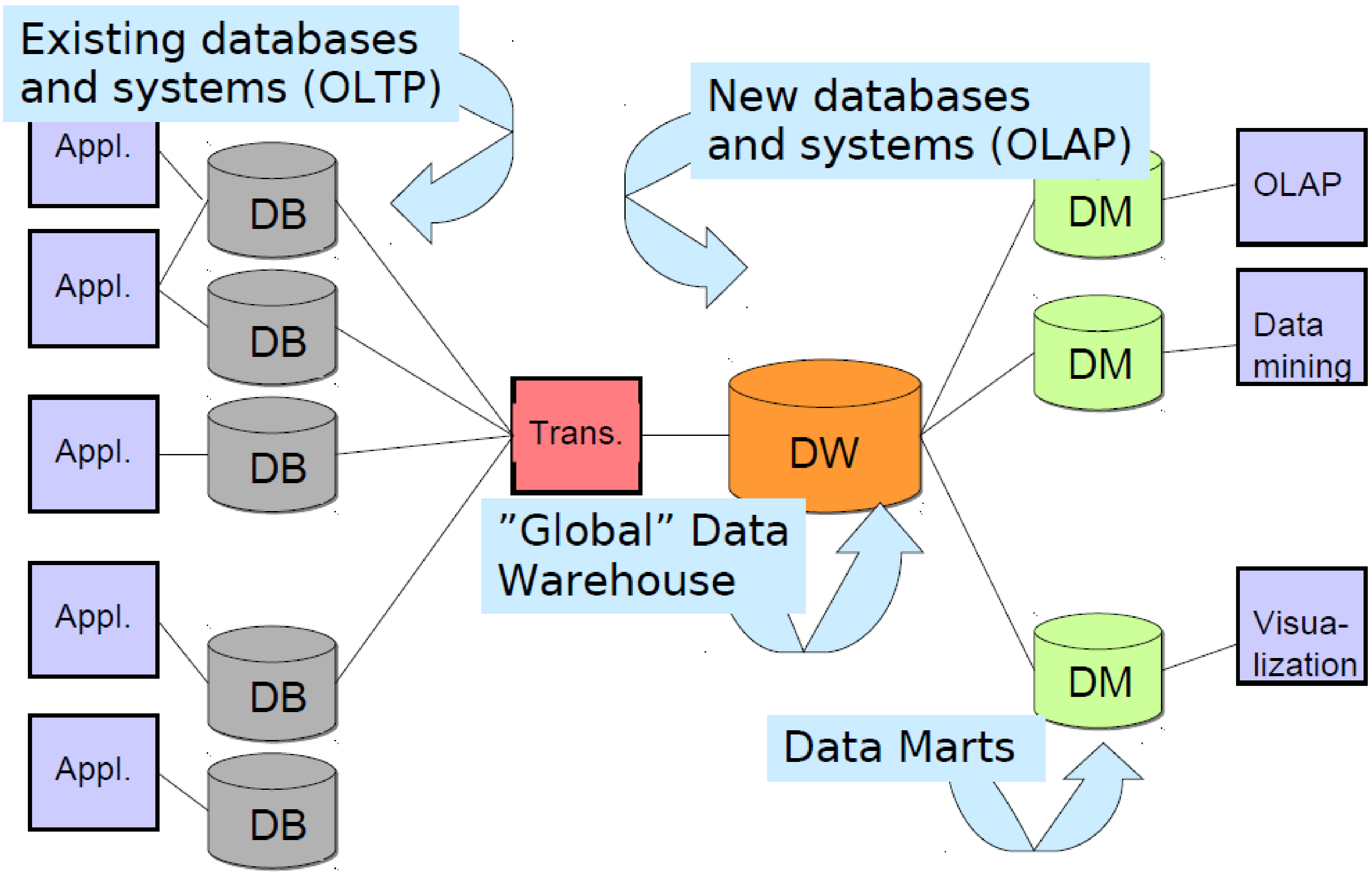
- Provide data for reporting and decision making
  - Mostly read-only access
  - e.g., resource planning, marketing initiatives
- Need archive data; (slightly) outdated information might be okay
  - Report changes over time
  - Can use separate data store (non-ODS)
- From a database perspective:
  - Long-running operations, mostly read-only
  - Queries not known in advance, often complex (⇝indexing?)
  - Might need to scan through large amounts of data
  - Data is (almost) append-only.

Transactional vs. Analytical Workloads

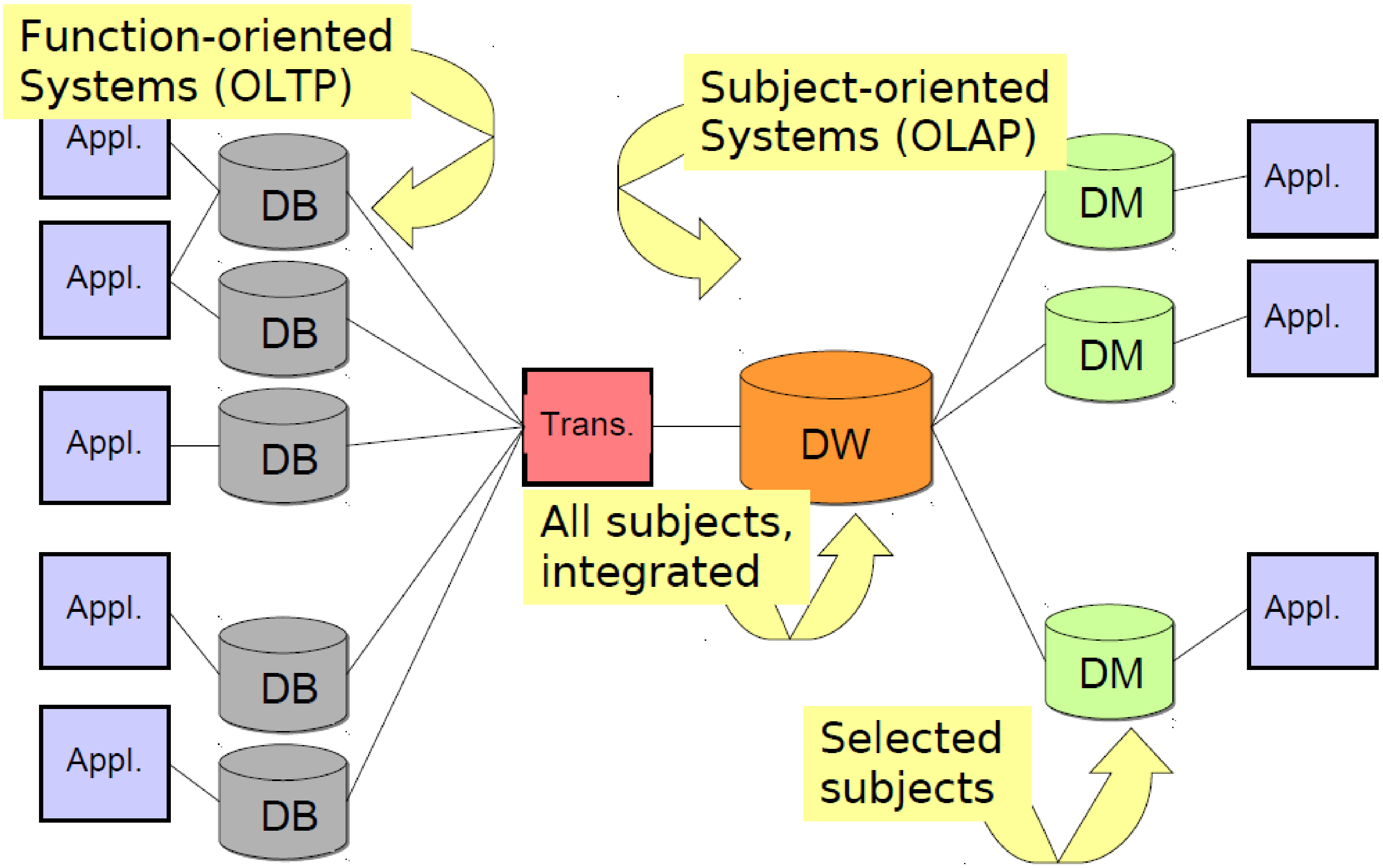
	OLTP	ODS	OLAP	DM / DW
Business Focus	Operational	Oper./Tact.	Tactical	Tact./Strat.
DB Technology	Relational	Relational	Cubic	Relational
Transaction Count	Large	Medium	Small	Small
Transaction Size	Small	Medium	Medium	Large
Transaction Time	Short	Medium	Medium	Long
DB Size in GB	10–400	100–800	100–800	800–80,000
Data Modeling	Trad. ERD	Trad. ERD	N/A	Dimensional
Normalization	3–5 NF	3 NF	N/A	0 NF

source: Bert Scalzo. *Oracle DBA Guide to Data Warehousing and Star Schemas*.

OLTP/OLAP and DW



OLTP/OLAP and DW (2)



Architecture Alternatives

- Variants of the full data warehouse architecture:
  1. Independent data marts (no central warehouse)
     - Populate data marts directly from sources
     - Like several “mini warehouses”
     - Redundancy, no “single source of truth”
  2. Logical data marts (no explicit, physical data marts)
     - Data mart just a logical view on full warehouse
     - Easier to provide integrated, consistent view across the
     - enterprise
- Data marts (and warehouse) might also reside at different geographic locations.

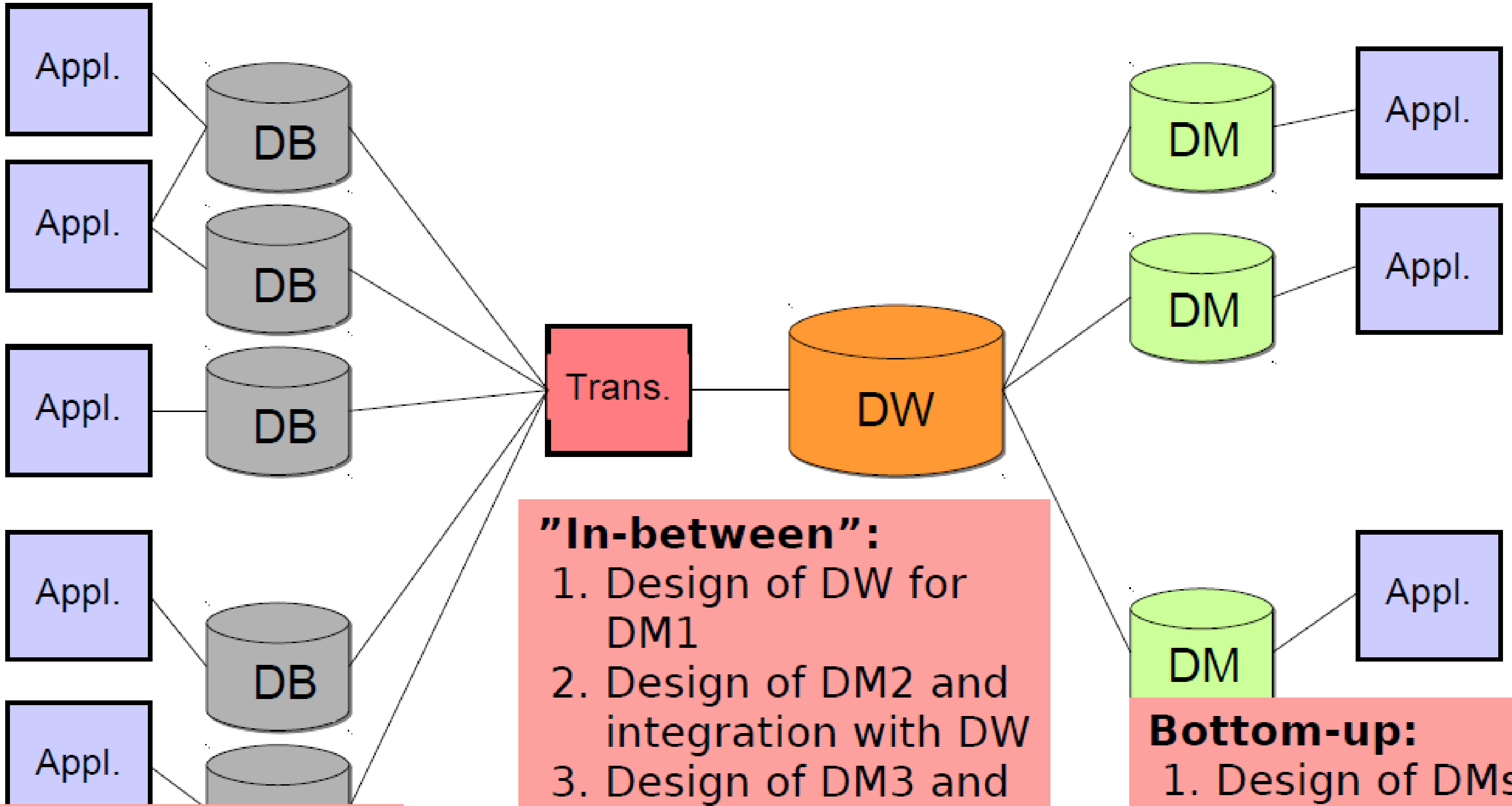
Building DW: Top-down Approach

- Analyze global business needs, plan how to develop a data warehouse, design it, and implement it as a whole
  - This procedure is promising: it is based on a global picture of the goal to achieve, and in principle it ensures consistent, well integrated data warehouses.
  - High-cost estimates with long-term implementations discourage company managers from embarking on these kind of projects.
  - Analyzing and integrating all relevant sources at the same time is a very difficult task, even because it is not very likely that they are all available and stable at the same time.
  - It is extremely difficult to forecast the specific needs of every department involved in a project, which can result in the analysis process coming to a standstill.
  - Since no working system is going to be delivered in the short term, users cannot check for this project to be useful, so they lose trust and interest in it.

Building DW: Bottom-up Approach

- DWs are incrementally built, and several data marts are iteratively created. Each data mart is based on a set of facts that are linked to a specific department and that can be interesting for a user group.
    - Leads to concrete results in a short time
    - Does not require huge investments
    - Enables designers to investigate one area at a time
    - Gives managers a quick feedback about the actual benefits of the system being built
    - Keeps the interest for the project constantly high
    - May determine a partial vision of the business domain

Building DW: Comparison of Approaches



Top-down:
1. Design of DW
2. Design of DMs

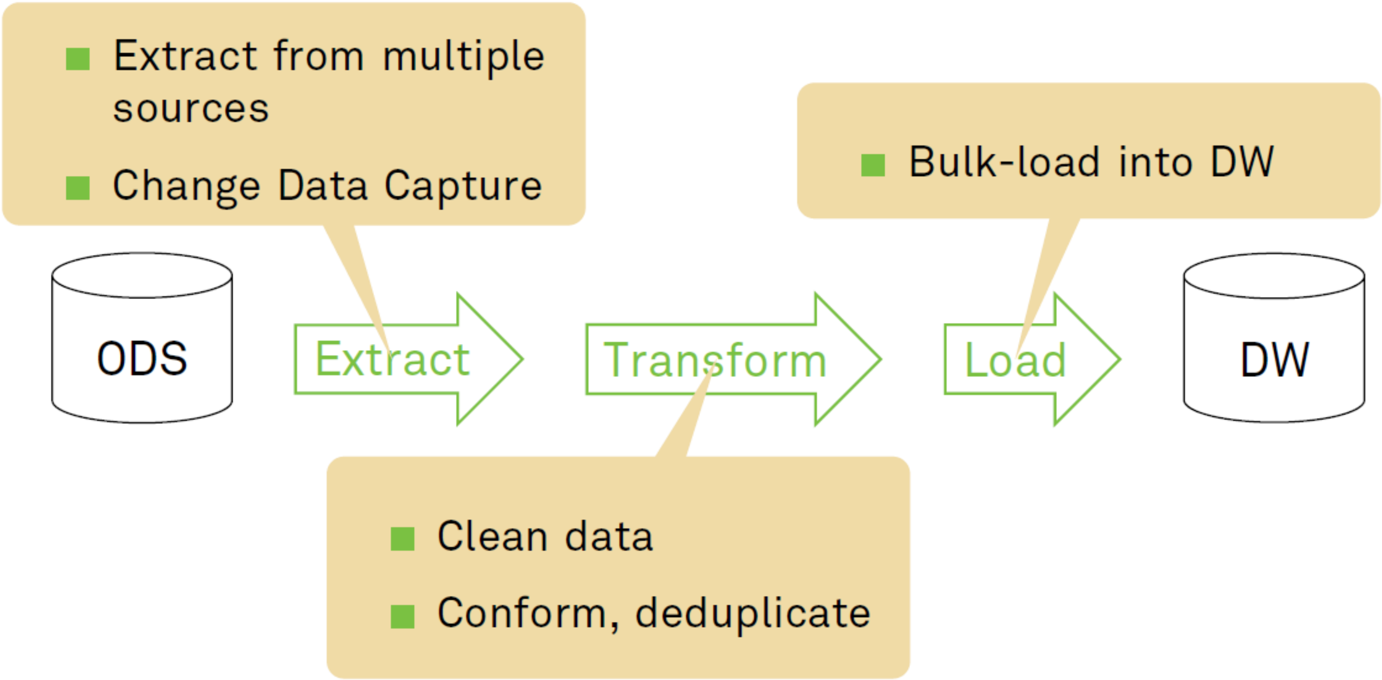
"In-between":
1. Design of DW for DM1
2. Design of DM2 and integration with DW
3. Design of DM3 and integration with DW
4. ...

Bottom-up:
1. Design of DMs
2. Integration of DMs in DW
3. Maybe no physical DW

J. Gamper, Free University of Bolzano, DWDM 2012/13

Data Warehouse Preparation

- Data is periodically brought from the ODS to the data warehouse.



- This is also referred to as ETL Process.

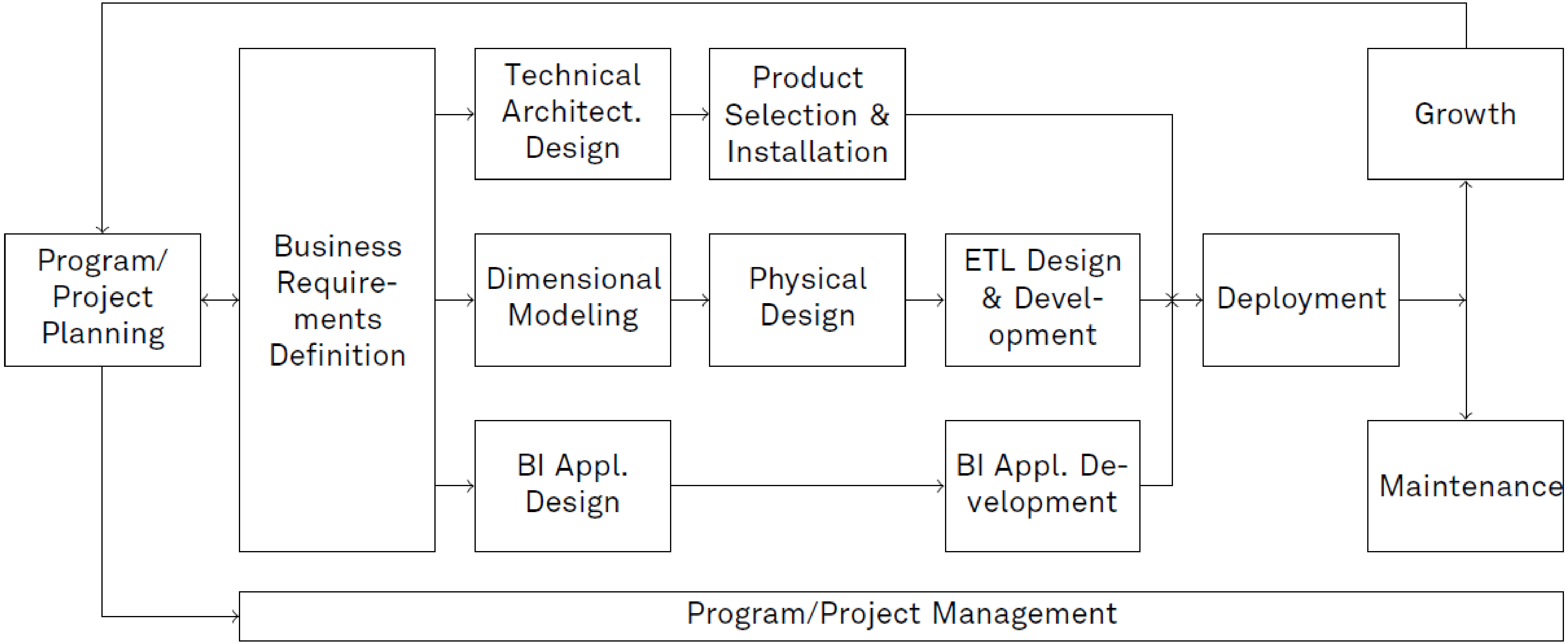
Data Warehouse Preparation - Transformation

- Query-Driven Data Integration
  - Data is integrated on demand (lazy)
  - PROS
    - Access to most up-to-date data (all source data directly available)
    - No duplication of data
  - CONS
    - Delay in query processing
      - Slow (or currently unavailable) information sources
      - Complex filtering and integration
    - Inefficient and expensive for frequent queries
    - Competes with local processing at sources
    - Data loss at the sources (e.g., historical data) cannot be recovered
  - Has not caught on in industry

Data Warehouse Preparation - Transformation

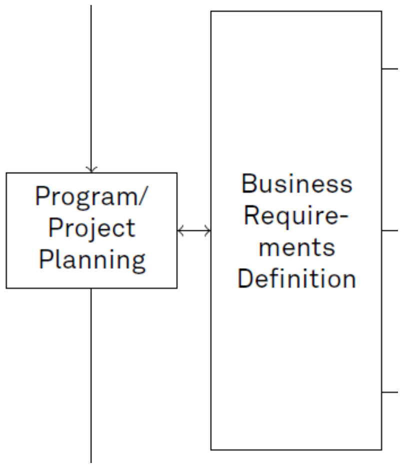
- Warehouse-Driven Data Integration
  - Data is integrated in advance (eager)
  - Data is stored in DW for querying and analysis
  - PROS
    - High query performance
    - Does not interfere with local processing at sources
      - Assumes that data warehouse update is possible during downtime of local processing
      - Complex queries are run at the data warehouse
      - OLTP queries are run at the source systems
  - CONS
    - Duplication of data
    - The most current source data is not available
  - Has caught on in industry

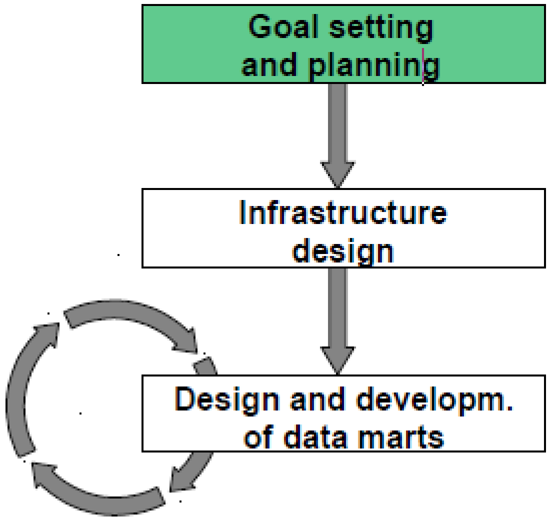
Data Warehouse Lifecycle



↗ Kimball *et al.* The Data Warehouse Lifecycle Toolkit.

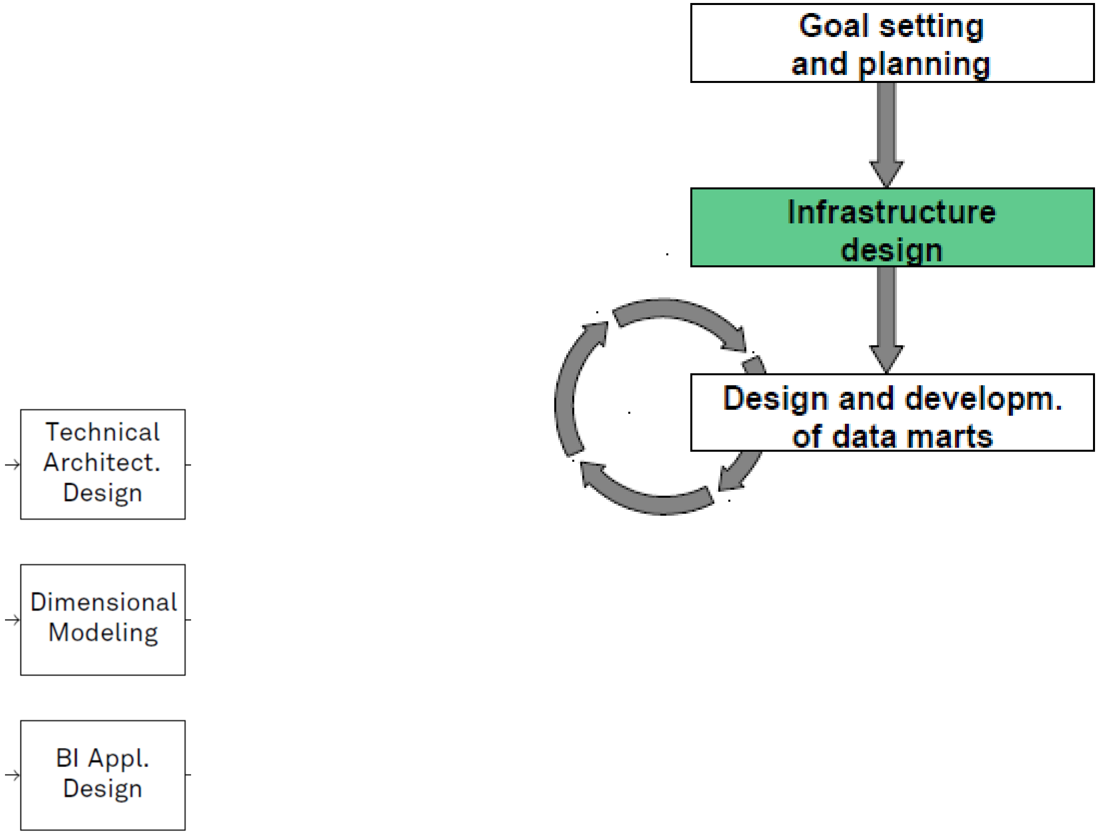
Data Warehouse – Lifecycle (1)





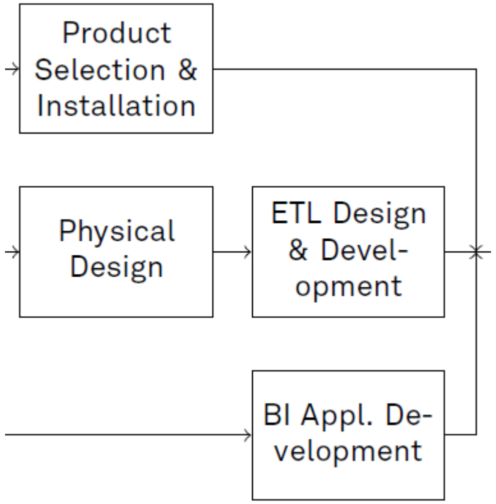
- gather system goals, borders, and size & prioritize them
- estimate costs and benefits
- analyze risks and expectations
- examine the skills of the working team

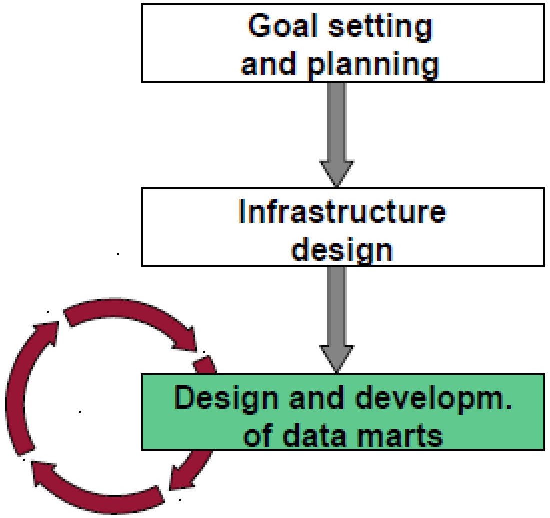
Data Warehouse – Lifecycle (2)



- analyze and compare the possible architectural solutions
- assess the available technologies and tools
- select an approach for design and implementation
- create a preliminary plan of the whole system

Data Warehouse – Lifecycle (3)

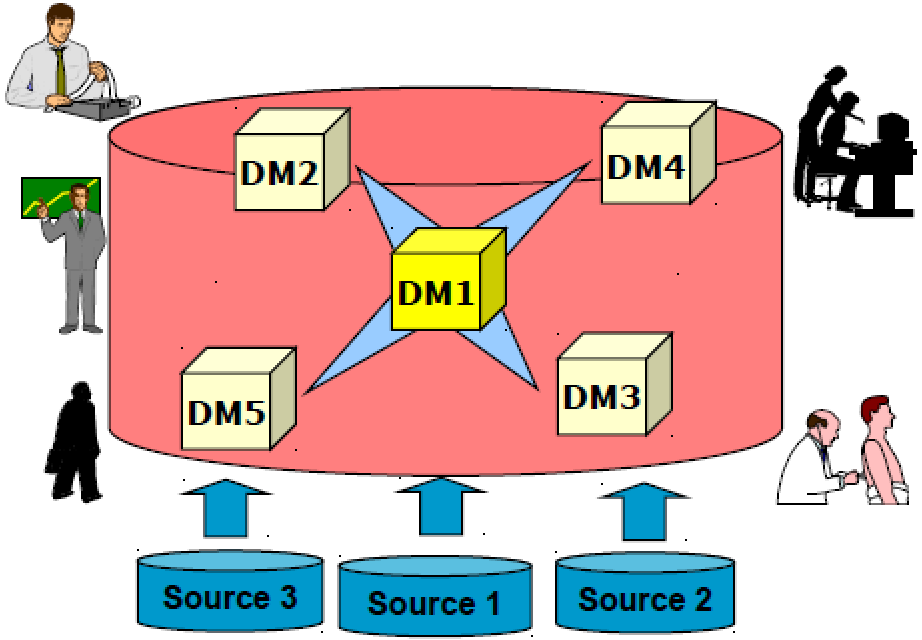




- Every iteration causes a new data mart and new applications to be created and progressively added to the DW system.

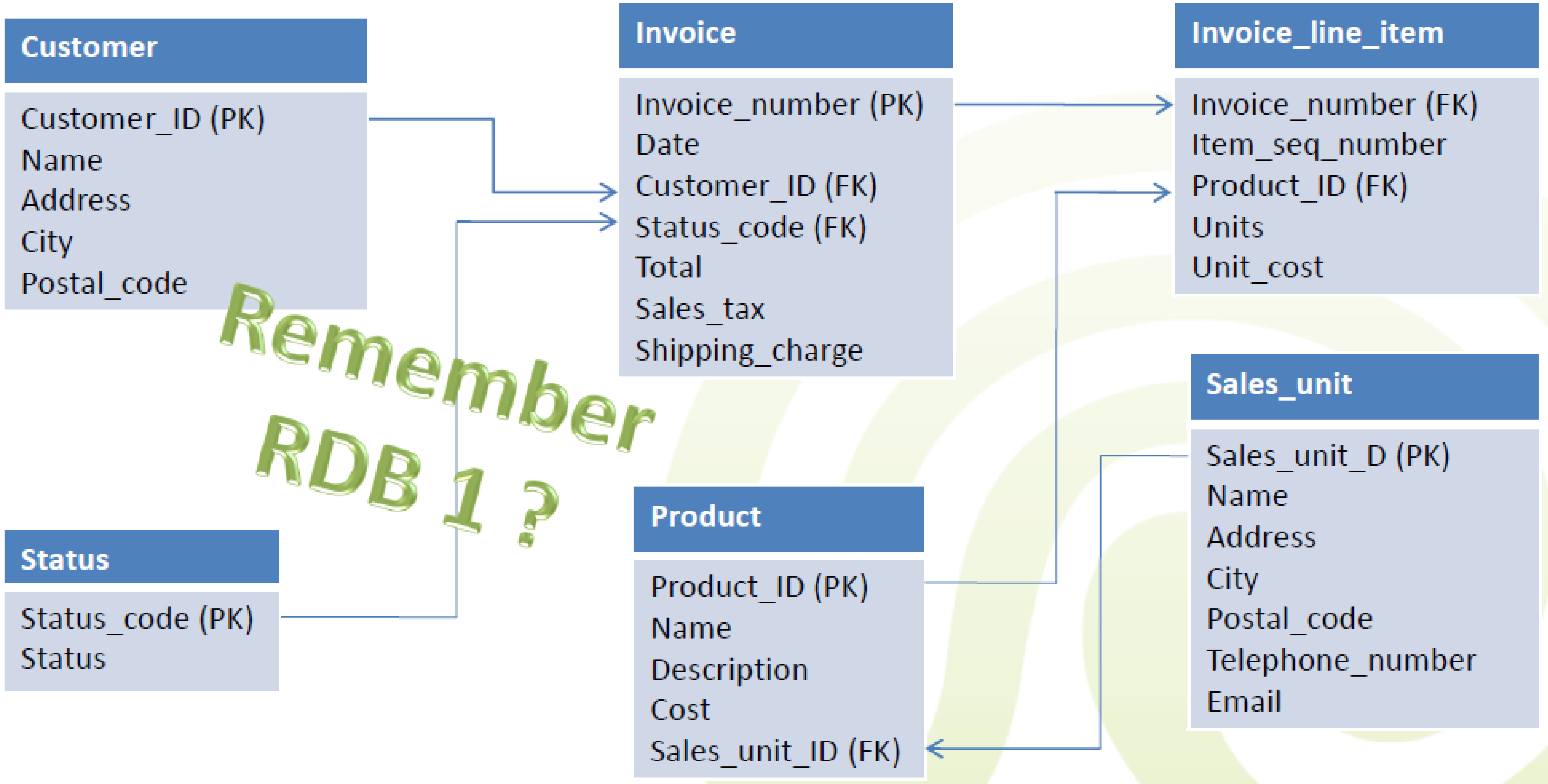
First Data Mart

- is the one playing the most strategic role for the enterprise
- should be a backbone for the whole DW
- should lean on available and consistent data sources



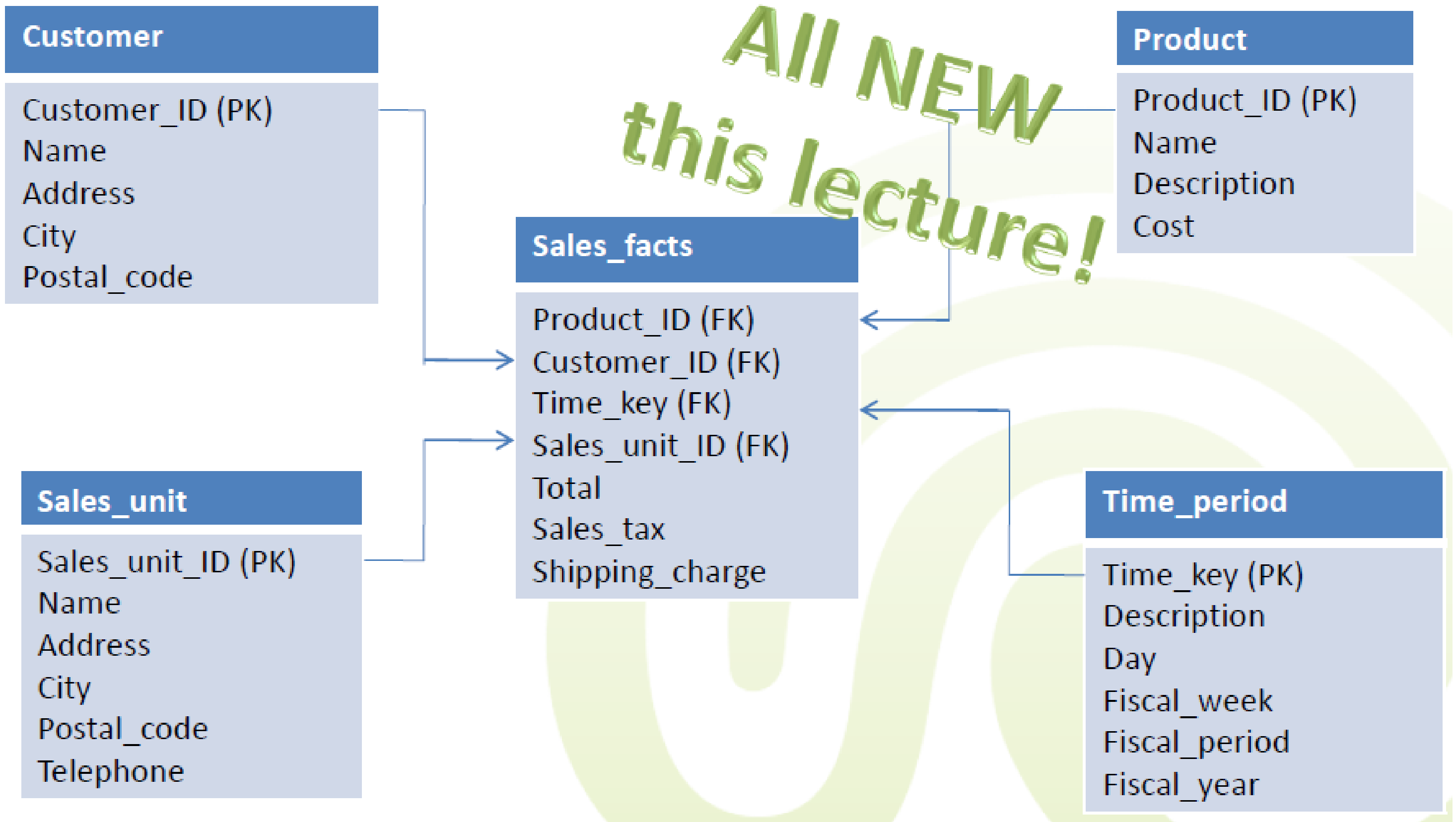
Modelling and Data Warehouse

- Operational stores – normalized database
- E.g., e-shop



Modelling and Data Warehouse

- Data Warehouse is subject-oriented -> sales are important



Data Warehouse – Data Model

- star schema – fact table and dimension tables
  - one purpose-focus analytics
  - not very good for complex analytics
- fact table
  - data measurements/metrics
  - numeric values and foreign keys to dimensions
  - atomic level of detail
- dimension table
  - description of fact
  - many attributes

Dim_store
id (PK)
id_store
name
street
city
state
country
region
manager

Fact_sales
id_time
id_store
id_product
quantity
price

Dim_product
id (PK)
id_product
name
brand
category

Dim_time
id
date
day
month
year
day_of_week

Nature of Current Data and Processing

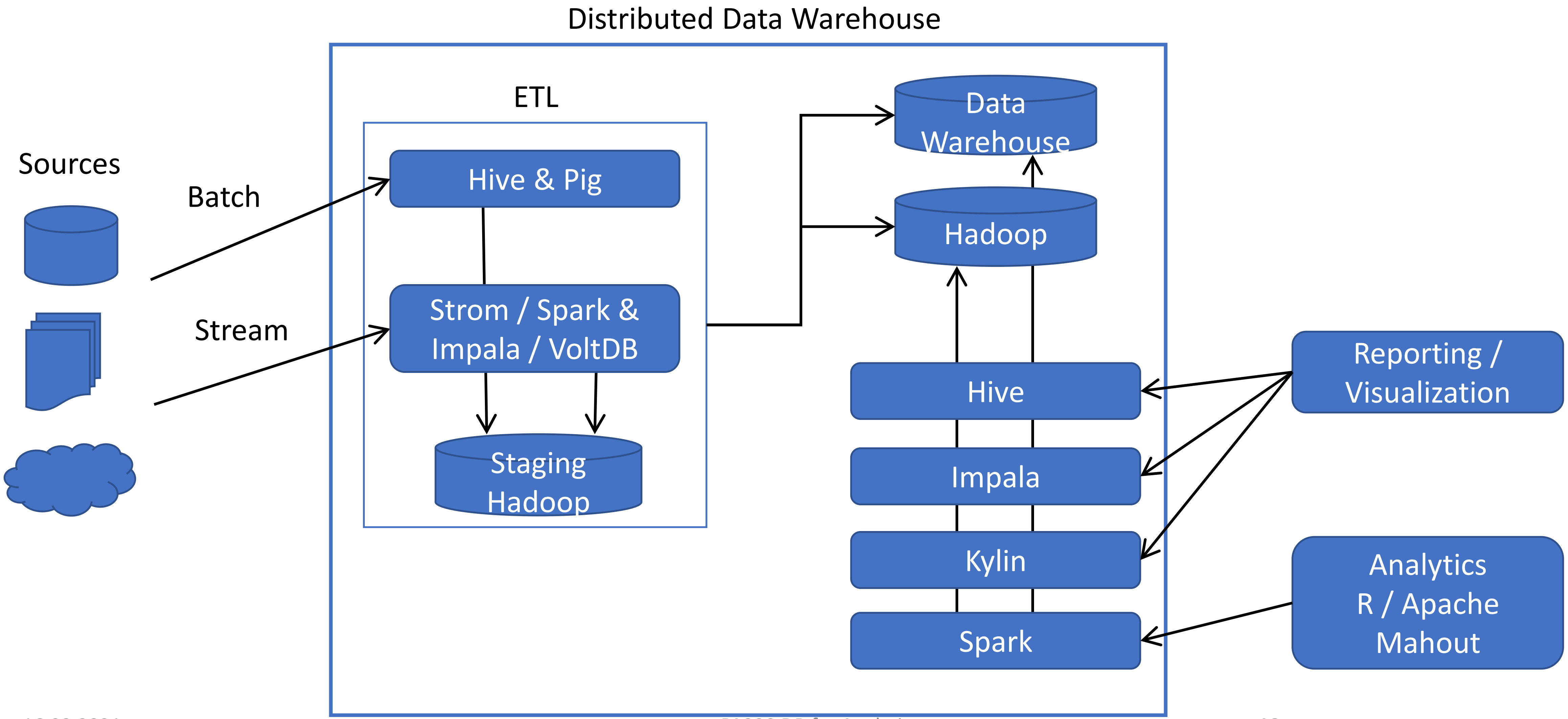
- Volume
  - the amount of data increases tenfold every five years
- Velocity
  - continuous data flow from sensors, social networks, …
- Variety
  - data structure can change, text, multimedia, …
- Veracity
  - with different data sources, it is getting more difficult to maintain data certainty

- Real-time processing

Distributed Data Warehouse

- new platforms
  - columnar DBMSs (e.g., C-Store)
  - HDFS & MapReduce (e.g., Hadoop)
  - NoSQL platforms (e.g., HBase)
  - in-memory DBMSs (e.g., VoltDB)
- horizontal scaling instead of vertical scaling
  - ETL in Hadoop
  - Storing raw data in HDFS
  - SQL-based analytics in columnar DBMSs

Distributed Data Warehouse



Hadoop Platform

- SW library for distributed processing of large data sets
  - across clusters of computers
- High-availability achieved on application layer by replication
  - tasks run / data stored on unreliable HW
- MapReduce – programming model for large scale data processing
  - Map() – filtering and sorting, outputs "key,value" pairs
  - Reduce() – summarizing Map() results by their keys

Map(k1,v1) → list(k2,v2)

Reduce(k2, list (v2)) → list(v3)

- HDFS – distributed high-throughput file system
  - designed for mostly immutable files
  - concurrent write not supported
  - cooperation with MapReduce – data & computation locality

Distributed Data Warehouse

- Apache Hive – data warehouse for large datasets
  - projects structure onto the data in HDFS
  - manages and queries data using HiveQL
    - converts them to Map-Reduce jobs
  - supports indexing
  - DML operations
    - UPDATE & DELETE at row level (new since end of 2014)

Apache Hive

Dim_product
id (PK)
id_product
name
brand
category

Fact_sales
id_time
id_store
id_product
quantity
price

```
CREATE TABLE dim_product ( id INT,
                           id_product INT,
                           name STRING,
                           brand STRING,
                           category STRING );

LOAD DATA LOCAL INPATH '…/file.txt' OVERWRITE INTO TABLE dim_product PARTITION (ds='2014-12-15');

INSERT OVERWRITE TABLE dim_product
  SELECT TRANSFORM (id, id_product, name, brand, category) USING 'python cleaning_mapper.py'
         AS (id, id_product, name, brand, category)
  FROM tmp_product;

SELECT p.name, SUM(s.quantity * price) AS income
  FROM dim_product p LEFT JOIN fact_sales s ON (p.id = s.id_product)
  GROUP BY p.name;
```

Apache Impala

- similar to Apache Hive
- native analytical database for Hadoop
- low-latency queries
  - no translation to Map-Reduce jobs
  - in-memory data in Parquet format
- preference to numeric values than strings
- joins done in memory

Apache Kylin

- framework for OLAP in Hadoop
  - uses Hive
  - precalculates aggregations
  - query engine translation
    - exploit prepared aggregations
- integrate with your favorite BI tools like Tableau and Power BI

Summary

- BI is well-recognized and is a combination of a number of techniques to support decision making.
- DW is at the core of BI that
  - provides a complete, consistent, subject-oriented and time-varying collection of the data;
  - allows to separate OLTP from OLAP.
- Applications that use the DW include OLAP, data mining, visualization
- BI can provide many advantages to an organization
  - Creates added value by transforming data into information
  - Provides comprehensive knowledge about your business
  - A good DW is a prerequisite for BI
  - But a DW is a means rather than a goal … it is only a success if it is heavily used
- Following a clear design methodology is important.

What next?

- Modeling for Data Warehouse