Data Management for Data Analysis
Data Warehouses

Vera Goebel
Department of Informatics, University of Oslo

Data Analysis (DA) / Machine Learning (ML)
Data Management (DM) for Data Analysis (ML)
OLAP (DA with DBS) versus OLTP (op. DBS)
Data Warehouses (DW)
Data Lakes (DL)

Why Data Analysis?
Can all problems be solved with ML?

Four questions to be answered:

1. Can the problem be clearly defined?
2. Does potentially meaningful data exist?
3. Does the data contain hidden knowledge or useful only for reporting purposes?
4. Will the cost of processing the data be less then the likely increase in profit from the knowledge gained from applying any data analysis?

ML Techniques

- Supervised Learning:
  - Classification
  - Estimation
  - Prediction
- Unsupervised Learning:
  - Clustering
  - Summarization
  - Association

Supervised vs. Unsupervised ML

- Supervised learning (classification, prediction)
  - Supervision: Training data (observations or measurements)
    -> labeled data indicating the class of the data
  - New data is classified based on the training dataset
- Unsupervised learning (summarization, association, clustering)
  - Class labels of training data are unknown
  - Given a set of measurements or observations
    -> establish existence of classes or clusters

ML Types

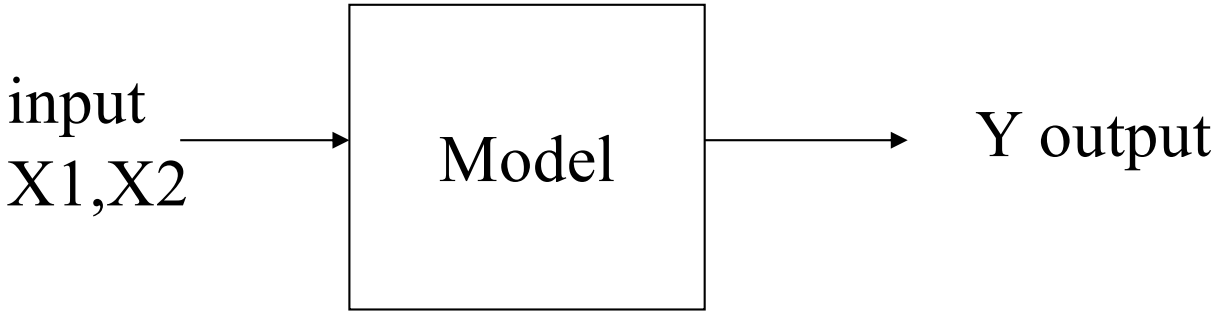
- Descriptive ML
  - characterize the general data properties in DB
  - finds patterns in data
  - user determines which ones are important
- Predictive ML
  - perform inference on the current data to make predictions
  - we know what to predict
- Not mutually exclusive
  - used together
  - descriptive $\rightarrow$ predictive

Descriptive ML

- Discover new patterns in the data
- Used during data exploration
- Typical questions answered:
  - What is in the data?
  - What does it look like?
  - Are there any unusual patterns?
- Patterns at various granularities
- Functionalities of descriptive data mining:
  - Clustering
  - Summarization
  - Visualization
  - Association

Black-box Model

X: vector of independent variables or inputs
Y =f(X) : an unknown function
Y: dependent variables or output
a single variable or a vector



User sees only a black box
Interested in the accuracy of the predictions

Predictive Data Mining

- Using known examples to train model
  - unknown function is *learned from data*
- the more data with known outcomes is available
  - the better the predictive power of the model
- Used to predict outcomes whose inputs are known but the output values are not realized yet
- Never 100% accurate
- Performance of a model on past data is not important
  - to predict the known outcomes
- Performance on unknown data is much more important

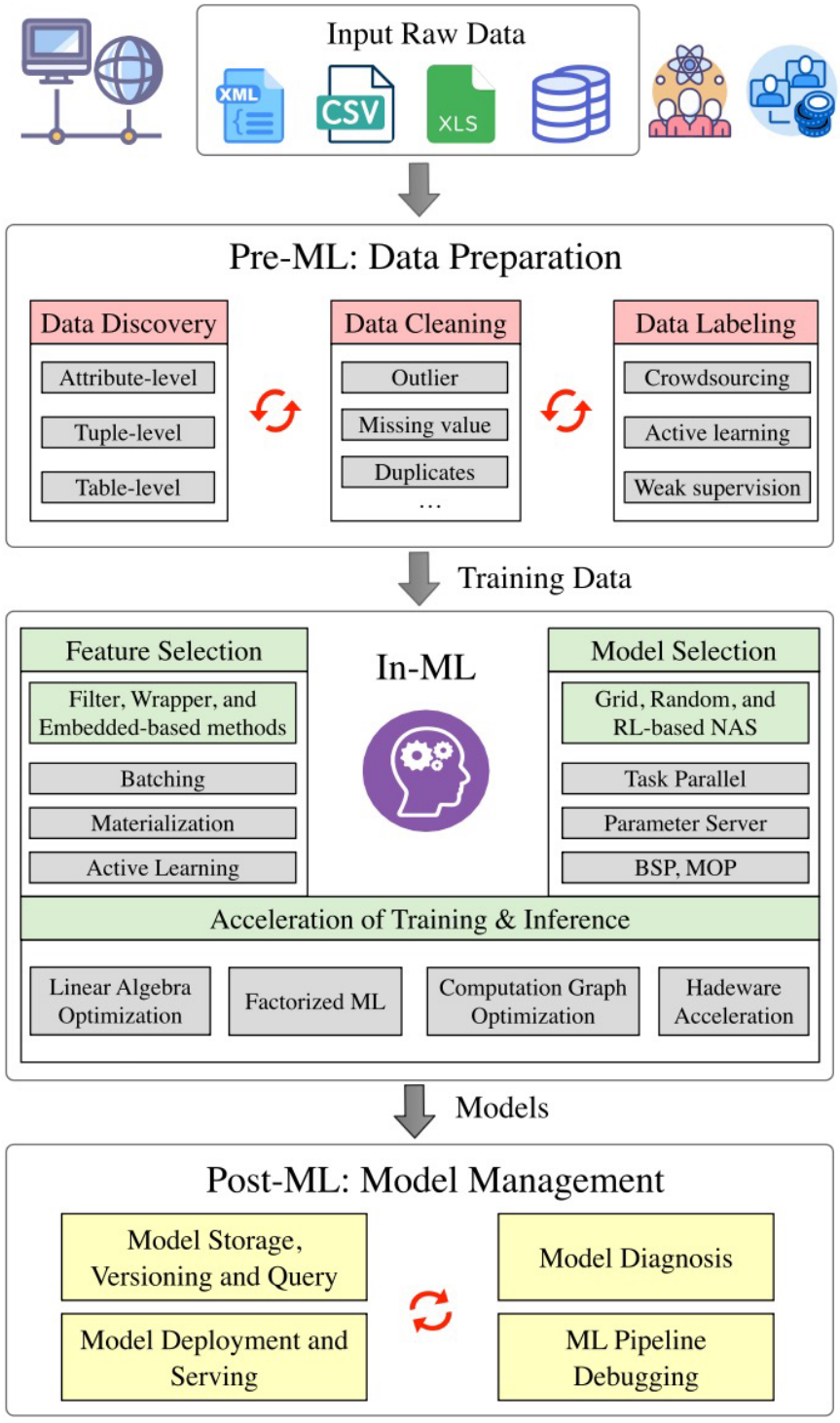
Data Analysis Requirements

- DB schema -> support complex and non-normalized data
  - Summarized and Aggregate data
  - Multiple Relationships
  - Queries must extract multi-dimensional time slices
  - Redundant Data
- Data extraction and filtering -> DB created by extracting data from operational DBS & data imported from external source
  - Need for advanced data extraction & filtering tools
  - Allow batch/scheduled data extraction
  - Support different types of data sources
  - Check for inconsistent data / data validation rules
  - Support advanced data integration / data formatting conflicts
- Analysis tools & user interface
  - advanced data modeling and data presentation tools
- Datasets very large (TB – PB) & advanced hardware

Data Analysis with DW (DB) Why Now?

- Data is being produced -> Big Data!
- Data is being warehoused
- The computing power is available
- The computing power is affordable
- The competitive pressures are strong
- Commercial products are available

Overview - Data Mgnt. for ML



[Chai et al. 2023]

DM Challenges for ML

1. Sufficient high-quality training data is inevitable, data is human expansive to acquire.
2. Large amount of training data and complicated model structures, need end-to-end data analysis pipeline.
3. Given an ML task, need to train many models
   -> hard to manage in real applications

- DB techniques can benefit ML by addressing these challenges.

3 Aspects of ML Pipeline

1. ## Data preparation (Pre-ML):

   Data preparation is the act of manipulating raw data from multiple data sources into a form that can be readily analyzed.

   Tasks: data discovery, data cleaning, data labeling

2. ## Model training and inference (In-ML):

   Improve model performance during training (models become larger and deeper), how to accelerate the entire training process, includes feature selection and model selection.

   Tasks: feature selection, model selection, acceleration of model training & inference

3. ## Model management (Post-ML):

   How to store, query, deploy and debug the models after training. Train iteratively, hard because too much data (parameters, model structures, performance). How to manage these models & artifacts?

   Tasks: model storage, versioning and query; model diagnosis; model deployment and serving; ML pipeline debugging

Pre-ML: Data Preparation

1. Data discovery:
   use data warehouse or data lake to build ML model, we need sufficient data for training and testing
   -> problem: lacking attributes, tuples or features categories: attribute/tuple-level and table-level data discovery.
2. Data cleaning:
   cleaning dirty data, always happens in raw data collected from different resources: missing values, outliers, duplicates, inconsistencies, … -> DW!
3. Data labeling:
   label data for training, need high-quality labels to achieve good ML results -> need human experts, expensive!
   Approaches: crowdsourcing active learning, weak supervision build connections between week labels and downstream ML tasks

In-ML: Model Training & Inference

- Feature extraction:
  batching, materialization (DW!), feature pruning, active learning
- Model selection:
  select most appropriate model or parameters during training
- Computing in training/inference:
  accelerate model training and inference -> DB community: parallelism techniques!

Post-ML: Model Management

- Model storage, versioning and querying:
  store, log, search and analyze model variants and metadata efficiently and effectively: need/use column-store, compression and indexing techniques -> DW!
- Model diagnosis:
  touched model parameters and data artifacts, DM challenges: storage and computation, techniques: sampling, materialization, indexing.
- Model deployment and serving:
  how to support low latency and high throughput model prediction, while guaranteeing accuracy of prediction results.
- ML pipeline debugging:
  (1) data debugging -> learn from data, find root cause of unexpected results, (2) model debugging

Data Analysis Process

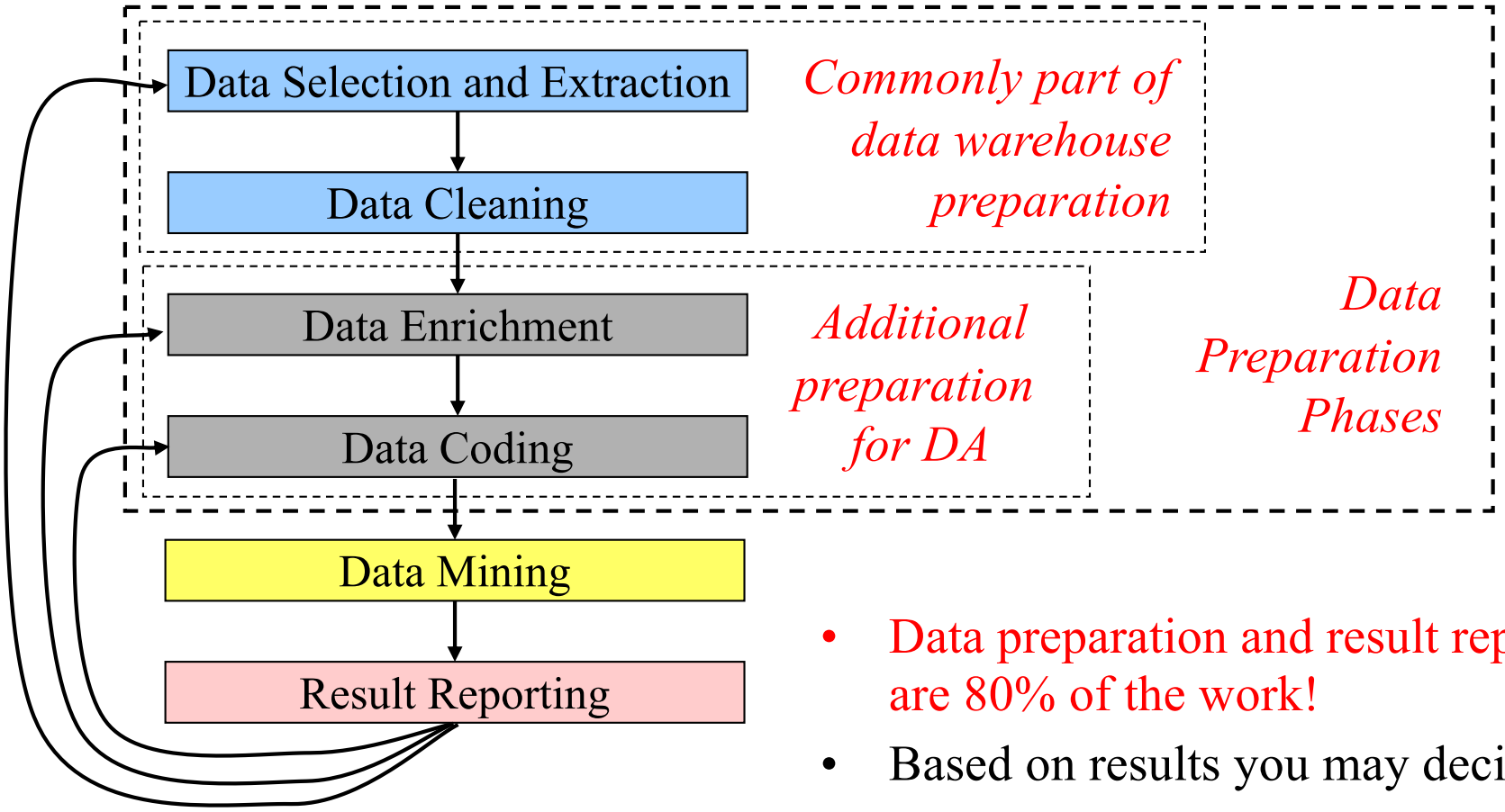
- Problem formulation
- Data collection
  - subset data: sampling might hurt if highly skewed data
  - feature selection: principal component analysis, heuristic search
- Pre-processing: cleaning
  - name/address cleaning, different meanings, duplicate removal, supplying missing values
- Transformation:
  - map complex objects, e.g. time series data, to features, e.g. frequency
- Choosing mining task and mining method
- Result evaluation and visualization

Data Analysis is an iterative process

Data Analysis Process (detailed)

- **1. Goal identification:**
  - Define problem
  - relevant prior knowledge and goals of application
- **2. Creating a target dataset:** data selection
- **3. Data preprocessing:** (80% of effort!) -> good Data Warehouse needed!
  - removal of noise or outliers
  - strategies for handling missing data fields
  - accounting for time sequence information
- **4. Data reduction and transformation:**
  - Find useful features, dimensionality/variable reduction, invariant representation
- **5. Data Mining:**
  - **Choosing functions of data mining:**
    - summarization, classification, regression, association, clustering
  - **Choosing the mining algorithm(s):**
    - which models or parameters
  - **Search for patterns of interest**
- **6. Presentation and evaluation:**
  - visualization, transformation, removing redundant patterns, etc.
- **7. Taking action:**
  - incorporating into the performance system
  - documenting
  - reporting to interested parties

Data Analysis Life Cycle



- Data preparation and result reporting are 80% of the work!
- Based on results you may decide to:
  - Get more data from internal data sources
  - Get additional data from external sources
  - Recode the data

Data Enrichment

- Integrating additional data from external sources
- Sources are public and private agencies
  - Government, Credit bureau, Research lab, …
- Typical data examples:
  - Average income by city, occupation, or age group
  - Percentage of homeowners and car owners by …
  - A person's credit rating, debt level, loan history, …
  - Geographical density maps (for population, occupations, …)
- New data extends each record from internal sources
- Database issues:
  - More heterogenous data formats to be integrated
  - Understand the semantics of the external source data

Data Coding

- Goal: to streamline the data for *effective* and *efficient* processing by the target DA application
- Steps:

  1) Delete records with many missing values
     - But …in fraud detection, missing values are indicators of the behavior you want to discover!

  2) Delete extra attributes
     - Ex: delete customer names if looking at customer classes

  3) Code detailed data values into categories or ranges based on the types of knowledge you want to discover
     - Ex: divide specific ages into age ranges, 0-10, 11-20, …
       map home addresses to regional codes
       convert homeownership to ”yes” or ”no”
       convert purchase date to a month number starting from Jan. 1990

OLTP versus OLAP

- OLTP: On-Line Transaction Processing
  - Processing at operational DBS, short time horizon
  - Optimize throughput (max. # TAs / time unit)
  - Large number of short online TAs
  - ACID transactions
- OLAP: On-Line Analytical Processing
  - Processing at DW, long time horizon
  - Optimize response time (1-n queries / minimal time)
  - Read-only data/queries, periodic refresh/update
  - Complex queries, involve aggregations
  - Store aggregated, historical data in multi-dimensional schemes
  - DW data latency: few hours, Data Marts: 1 day
- 3 aspects: time span, granularity, dimensionality

Aspects

Time span

- Operational DBS / *Data Store:*
  real-time, current transactions, short time frame, specific facts
- Data analysis:
  historic data, long time frame, patterns

Granularity

- Operational DBS / *Data Store:*
  specific Transactions that occur at a given time
- Data analysis:
  shown at different levels of aggregation, different summary levels, decompose (drill down), summarize (roll up)

Dimensionality (Most distinguishing DA characteristic)

- Operational DBS / *Data Store:*
  represents atomic transactions
- Data analysis:
  data is related in many ways, develop the larger picture, multi-dimensional view of data

OLTP versus OLAP

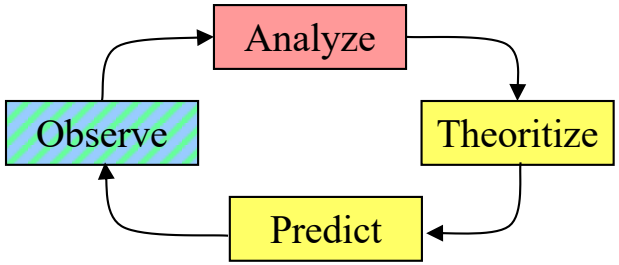
OLTP

- Mostly updates
- Many small transactions
- GB-TB of data
- Raw data
- Clerical users
- Up-to-date data
- Consistency, recoverability critical

OLAP

- Read-only queries
- Periodic updates
- Queries long, complex
- PB-EB+ of data
- Summarized, consolidated data
- ML data scientists

Where SQL Stops Short



ML Tools — *Creates new knowledge in the form of predictions, classifications, associations, time-based patterns*

OLAP Tools — *Analyzes a multi-dimensional view of existing knowledge*

DA-Extended SQL Queries — *Extends the DB schema to a limited, multi-dimensional view*

Complex SQL Queries — *DB schema describes the structure you already know*

Database

- Successful DA uses the entire tool hierarchy

Data Warehouses (DW)

DW is collection of integrated, subject-oriented DBs designed to support data analysis, where each unit of data is non-volatile and relevant to some moment in time [Inmon 1992].

Facts:

- 1990 – IBM, “Business Intelligence”: process of collecting and analyzing
- 1993 – Bill Inmon, “Data Warehouse”
- Growing industry: high RoI (Return of Investment)
- DW solutions offered today by nearly all commercial DBS vendors!
- Expensive to build: 10-1000 million $

What is a Data Warehouse?

- Collection of diverse data
  - subject oriented
  - aimed at executive, decision maker
  - often a copy of operational data
  - with value-added data (e.g., summaries, history)
  - data integrated
  - time variant
  - non-volatile

DW Characteristics

- Subject oriented: data are organized based on how the users refer to them.
- Data integrated: all inconsistencies regarding naming convention and value representations are removed.
- Non-volatile: data are stored in read-only format and do not change over time (except periodic updates/refresh from operational DBS).
- Time variant: data are not current but normally time series.

Data Warehouse (DW)

-> Data Analysis needs multi-dimensional data input

DW: Repository of multiple heterogeneous data sources, organized under a unified multi-dimensional schema at a single site in order to facilitate management decision making.

DW technology includes:

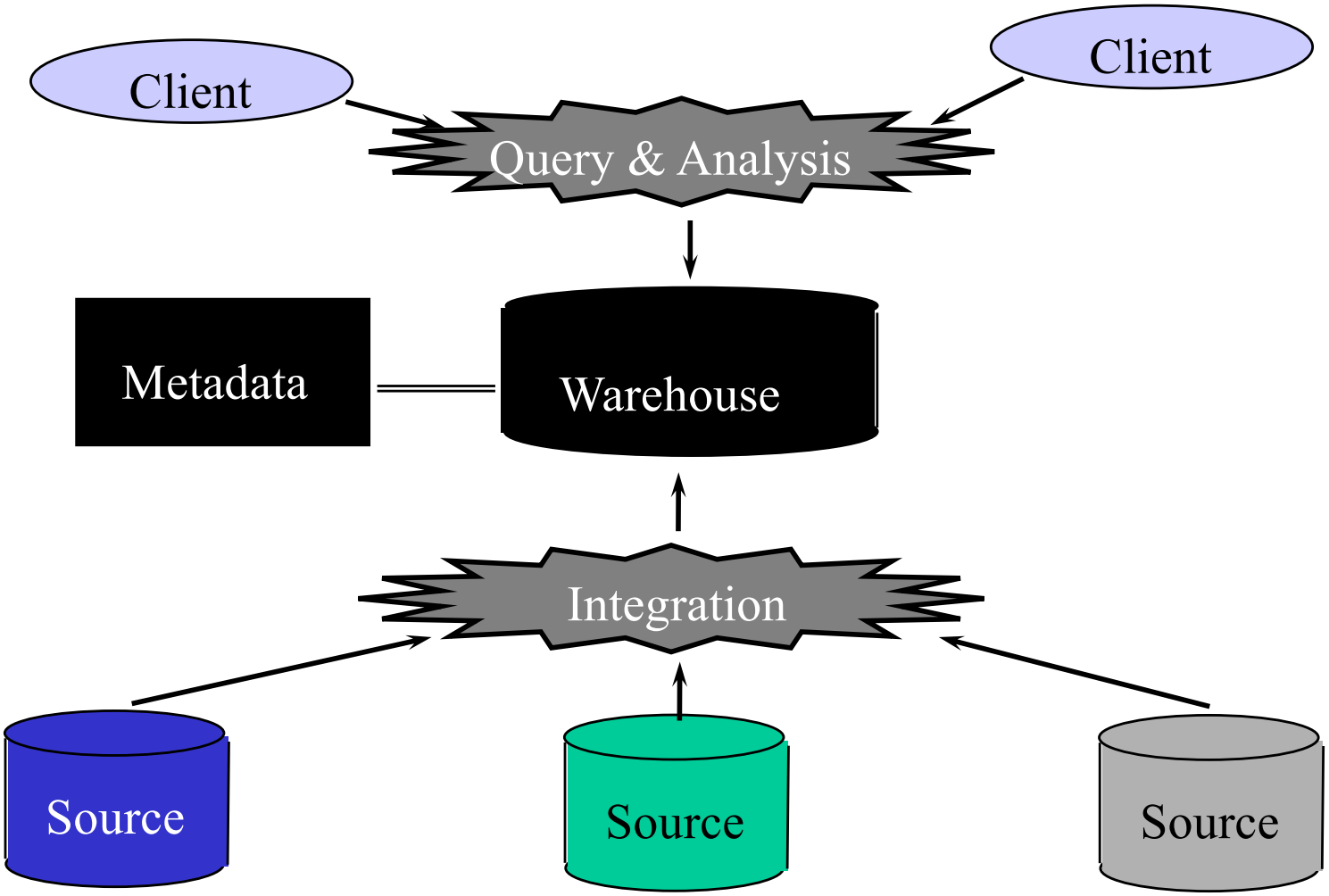
- Data cleaning
- Data integration
- On-Line Analytical Processing (OLAP): Techniques that support multidimensional analysis and decision making with the following functionalities
  - summarization
  - consolidation
  - aggregation
  - view information from different angles
- but additional data analysis tools are needed for
  - classification
  - clustering
  - characterization of data changing over time

DW Characteristics

- Summarized: operational data are mapped into a decision-usable format.
- Large volume: time series data sets are normally quite large.
- Not normalized: DW data can be (often are) redundant.
- Metadata: data about data are stored.
- Data sources: data come from internal and external un-integrated operational systems.

Data Warehouse Architecture

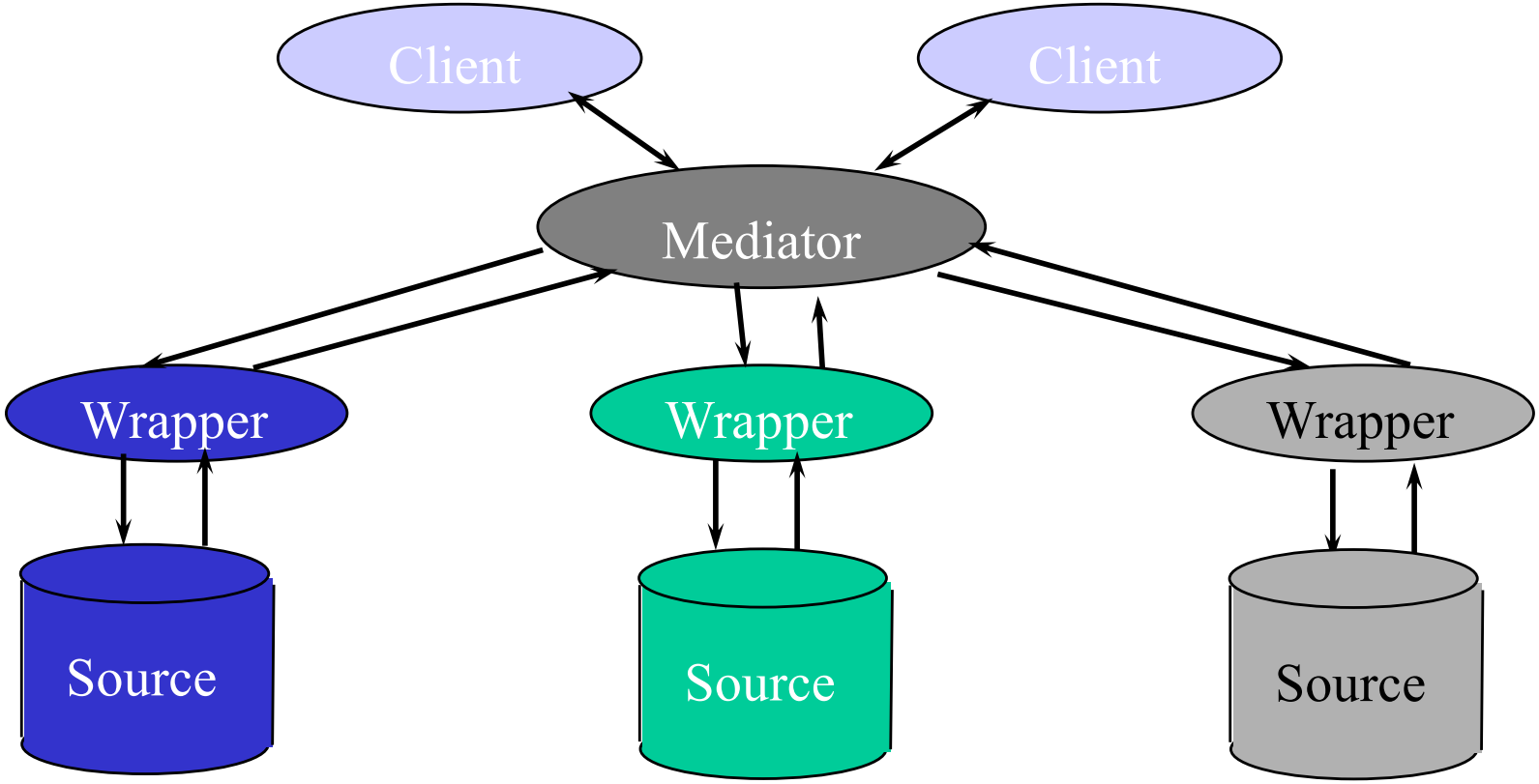
Two Approaches: query-driven (lazy), DW (eager)



Advantages of Data Warehousing

- High query performance
- Queries not visible outside DW
- Local processing at sources unaffected
- Can operate when sources unavailable
- Can query data not stored in a DBMS
- Extra information at warehouse
  - Modify, summarize (store aggregates)
  - Add historical information

Query-Driven Approach



Advantages of Query-Driven Approach

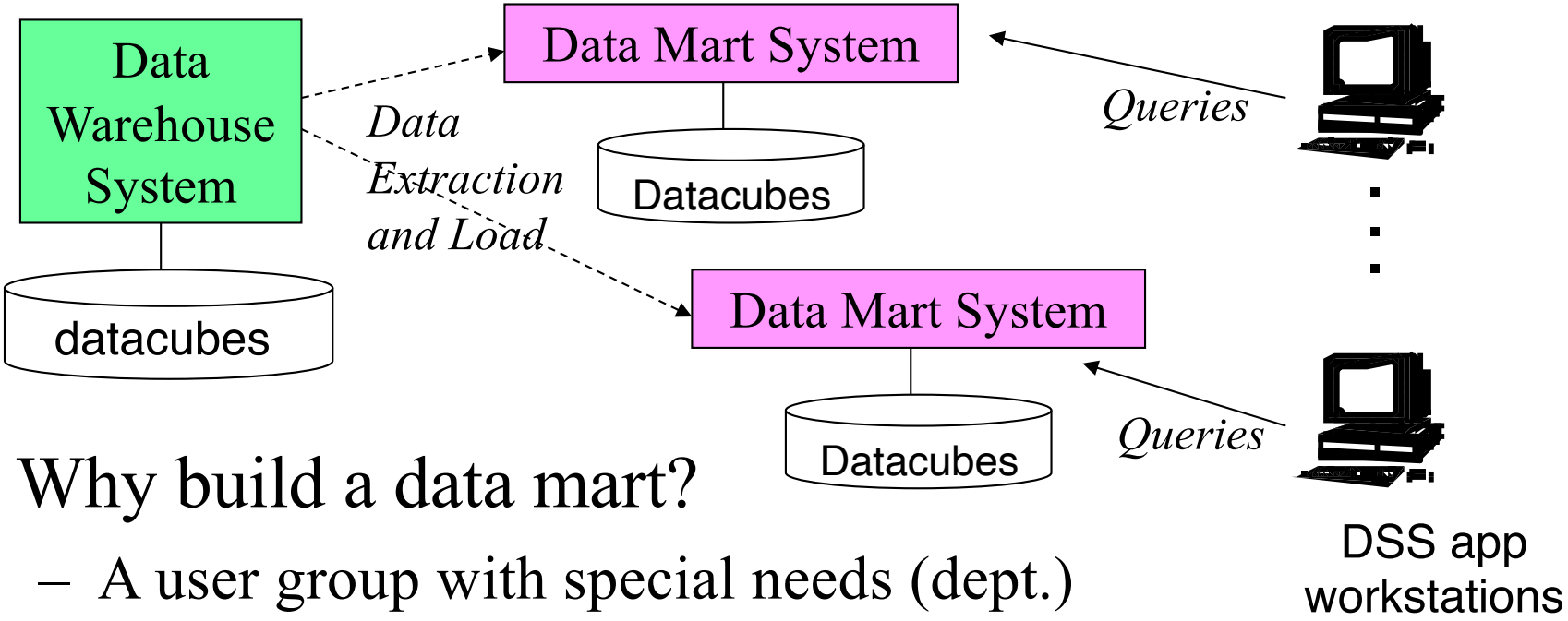
- No need to copy data
  - less storage
  - no need to purchase data
- More up-to-date data
- Query needs can be unknown
- Only query interface needed at sources
- May be less draining on sources

Data Marts

- Smaller data warehouses
- Spans part of organization
  - e.g., marketing (customers, products, sales)
- Do not require enterprise-wide consensus
  - but long term integration problems?

Data Marting

- What: Stores a second copy of a subset of a DW



- Why build a data mart?
  - A user group with special needs (dept.)
  - Better performance accessing fewer records
  - To support a "different" user access tool
  - To enforce access control over different subsets
  - To segment data over different hardware platforms

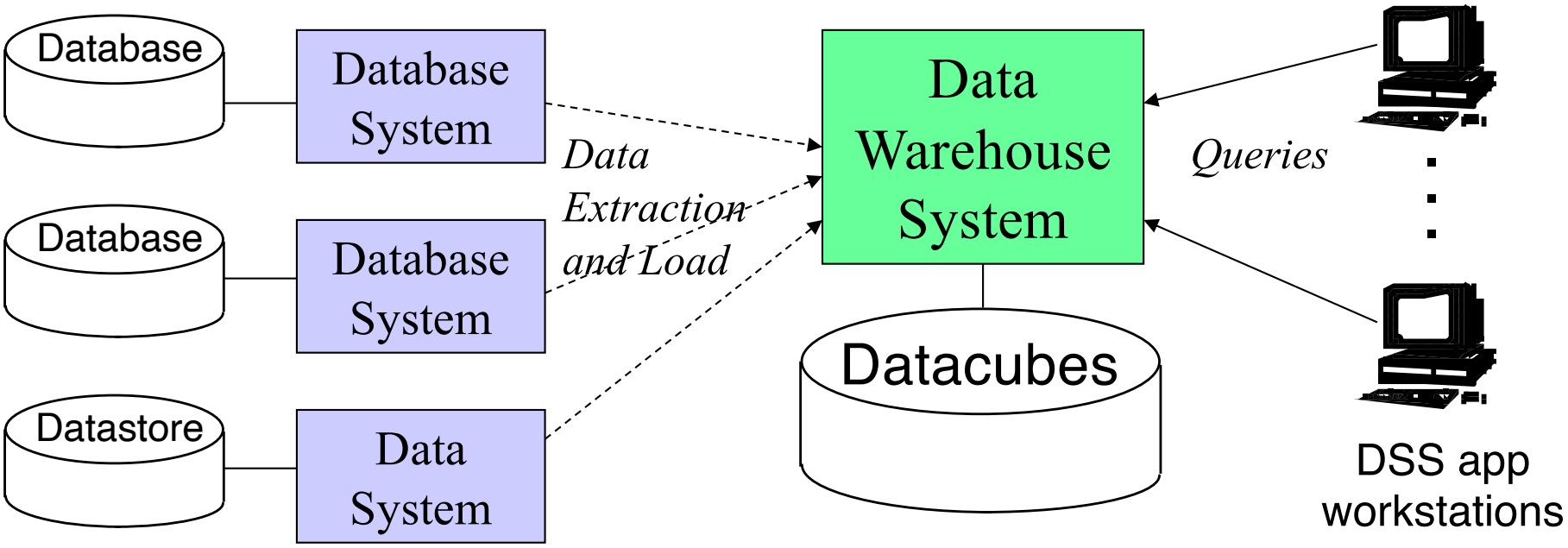
Costs and Benefits of Data Marting

- System costs:
  - More hardware (servers and networks)
  - Define a subset of the global data model
  - More software to:
    - Extract data from the warehouse
    - Load data into the mart
    - Update the mart (after the warehouse is updated)
- User benefits:
  - Define new *measures* not stored in the DW
  - Better performance (mart users and DW users)
  - Support a more appropriate user interface
    - Ex: a browser with forms versus SQL queries
  - Company achieves more reliable access control

DW Models & Operators

- Data Models
  - relations
  - Star schema & snowflake schema
  - Data Cubes
- Operators
  - slice & dice
  - roll-up, drill down
  - pivoting
  - other

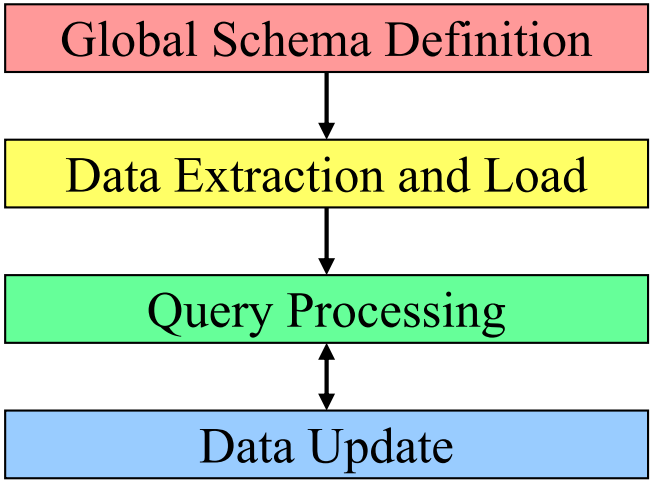
What and Why of Data Warehousing



- What: A very large database containing materialized views of multiple, independent source databases.
  The views generally contain aggregation data (aka datacubes).
- Why: The data warehouse (DW) supports read-only queries for new applications, e.g., ML, OLAP.

DW Life Cycle

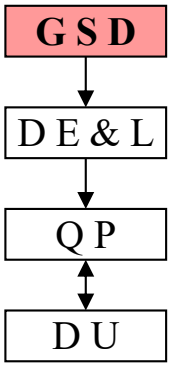
- The Life Cycle:



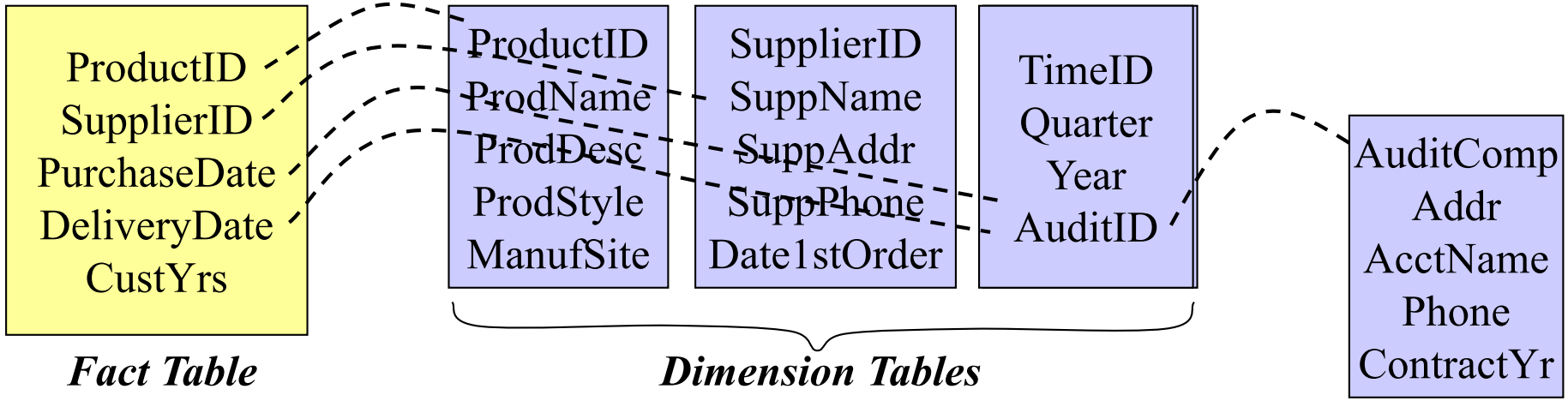
- General Problems:
  - Heavy user demand
  - Problems with source data
    - ownership, format, heterogeneity
  - Underestimating complexity & resources for all phases

- Boeing Computing Services – DW for DSS in airplane repair
  - DW size: 2-3 terabytes
  - Online query services: 24×7 service
  - Data life cycle: retain data for 70+ years (until the airplane is retired)
  - Data update: No "nighttime"; concurrent refresh is required
  - Access paths: Support new and old methods for 70+ years

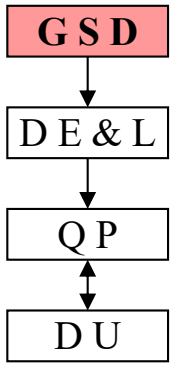
Global Schema Design – Base Tables



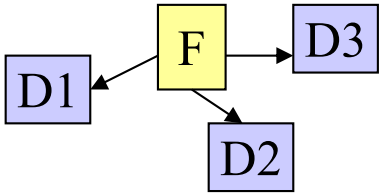
- Fact Table
  - Stores basic facts from the source databases (often denormalized)
  - Data about past events (e.g., sales, deliverings, factory outputs, ...)
  - Have a time (or time period) associated with them
  - Data is very unlikely to change at a data source; no updates
  - Very large tables (up to 1 TB)
- Dimension Table
  - Attributes of one dimension of a fact table (typically denormalized)
  - A chain of dimension tables to describe attributes on other dimension tables, (normalized or denormalized)
  - Data can change at a data source; updates executed occasionally



Schema Design Patterns

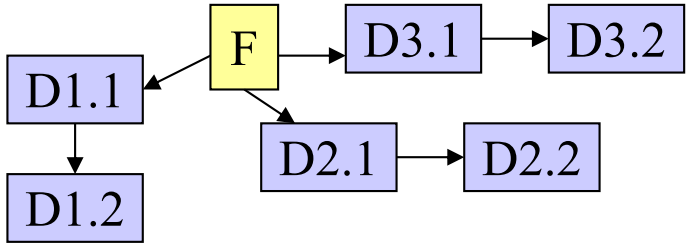


- Star Schema



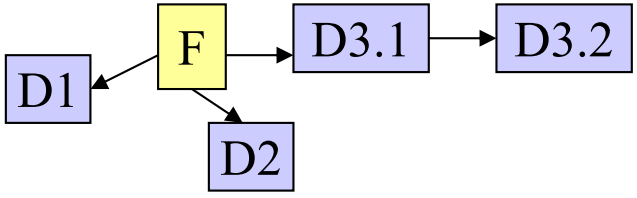
D1, D2, D3 are denormalized

- Snowflake Schema



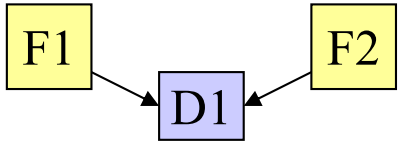
D1, D2, D3 are normalized

- Starflake Schema



D3 may be normalized or denormalized

- Constellation Schema

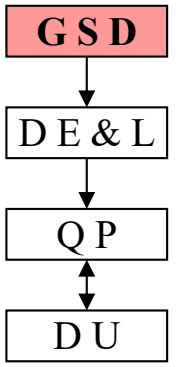


D1 stores attributes about a relationship between F1 and F2

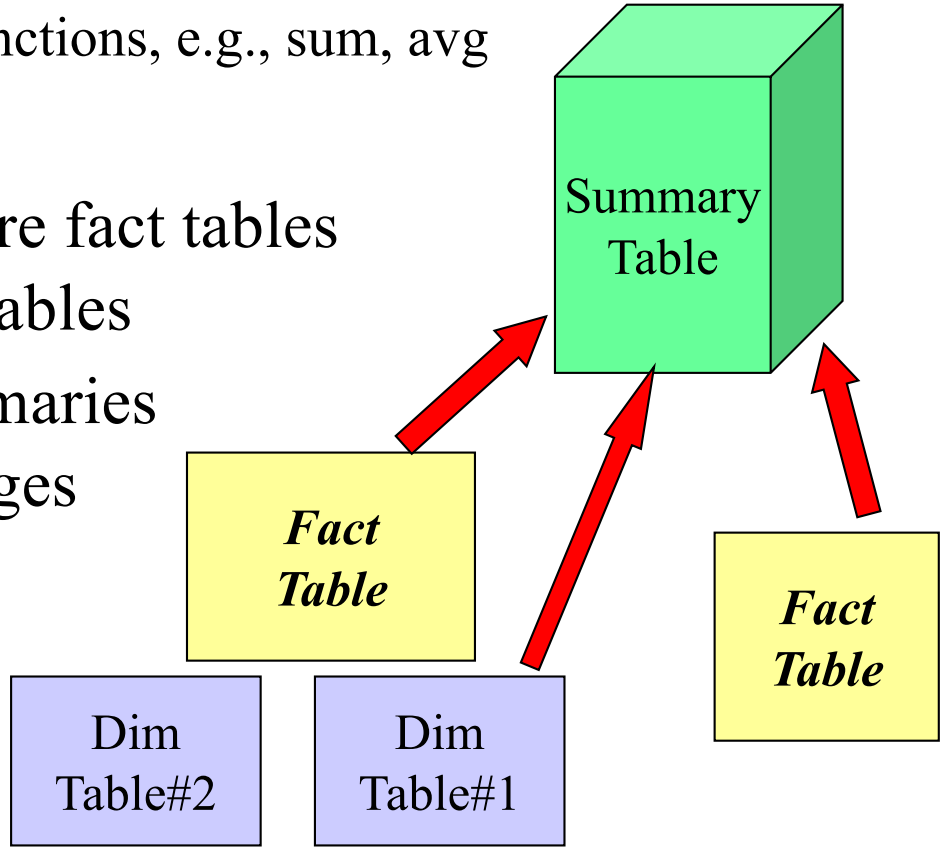
Design Methodologies

- Bottom-up:
  - Data Marts created first
  - Integrate data marts to create DW
  - Bus architecture: collection of conformed dimensions & facts
- Top-down:
  - Use normalized enterprise data model
  - Atomic data, (data at lowest level of detail) stored in DW
  - Dimensional data marts containing data needed for spec. business process, created from DW

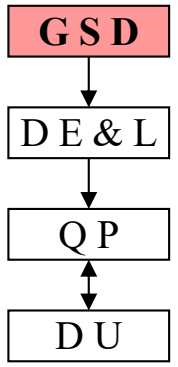
Summary Tables



- aka ***datacubes*** or multidimensional tables
- Store precomputed query results for *likely queries*
  - Reduce on-the-fly join operations
  - Reduce on-the-fly aggregation functions, e.g., sum, avg
- Stores denormalized data
- Aggregate data from one or more fact tables and/or one or more dimension tables
- Discard and compute new summaries as the set of *likely queries* changes



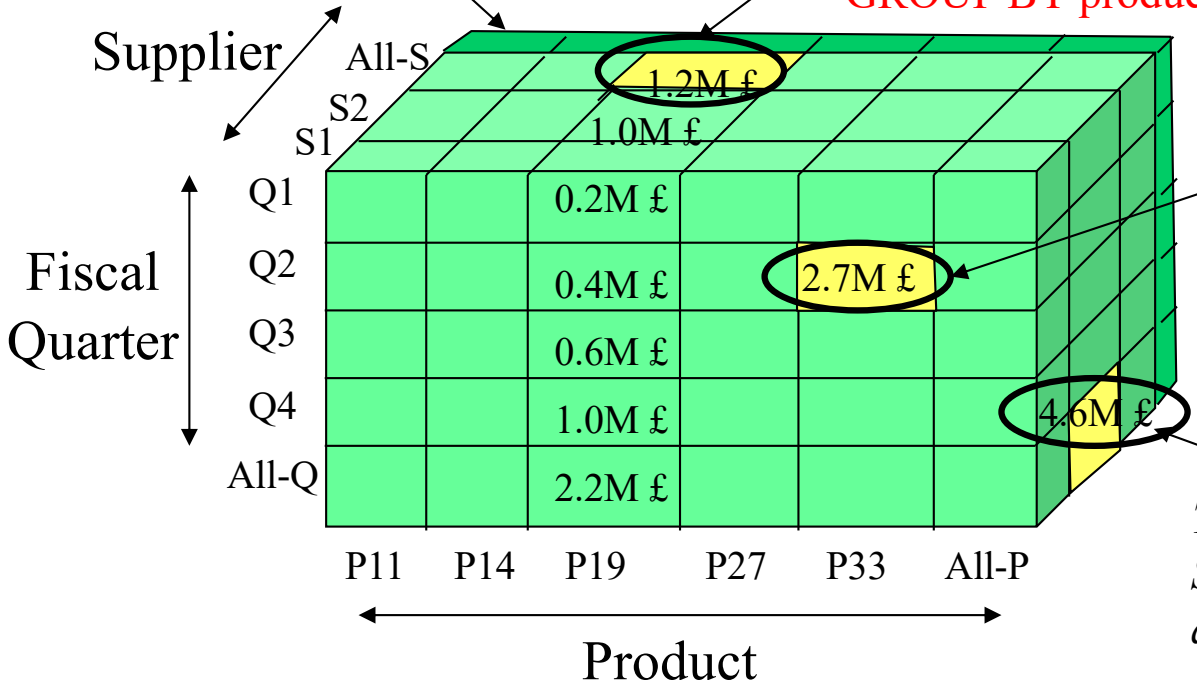
Summary Tables = Datacubes



Total Expenses for Parts by Product, Supplier and Quarter

Average Price paid to all suppliers of parts for Product P11 in the 1st quarter
GROUP BY product, quarter

Total Expenses paid to all suppliers of parts for Product P19 in the 1st quarter
GROUP BY product, quarter

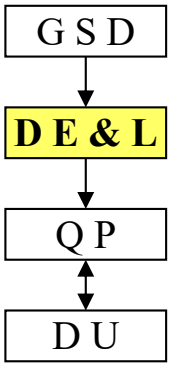


Total Expenses paid to Supplier S1 for parts for Product P33 in 2nd quarter
GROUP BY supplier, product, quarter

Total Expenses paid to Supplier S2 for parts for all products in all quarters
GROUP BY supplier

- Typical, pre-computed *Measures* are:
  - Sum, percentage, average, std deviation, count, min-value, max-value, percentile

Data Extraction and Load



Step1: Extract and clean data from all sources

- Select source, remove data inconsistencies, add default values

Step2: Materialize the views and measures

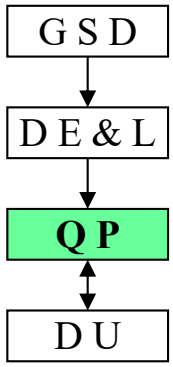
- Reformat data, recalculate data, merge data from multiple sources, add time elements to the data, compute measures

Step3: Store data in the DW

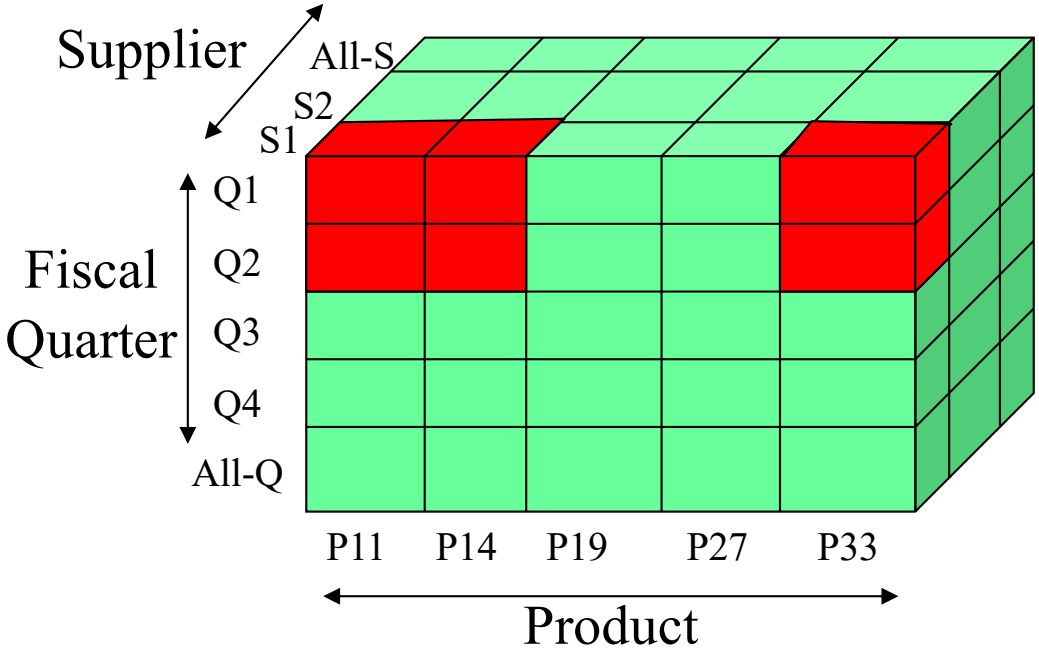
- Create metadata and access path data, such as indexes

- Major Issue: ***Failure during extraction and load***
- Approaches:
  - UNDO/REDO logging
    - Too expensive in time and space
  - Incremental Checkpointing
    - When to checkpoint? Modularize and divide the long-running tasks
    - Must use UNDO/REDO logs also; Need high/performance logging

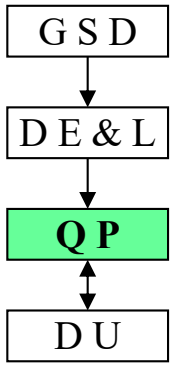
User Queries



- Retrieve pre-computed data or formulate new measures not materialized in the DW.
- New user operations on logical datacubes:
  - Roll-up, Drill-down, Pivot/Rotate
  - Slicing and Dicing with a "data blade"
  - Sorting
  - Selection
  - Derived Attributes

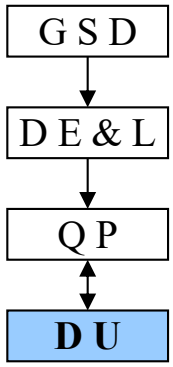


Query Processing



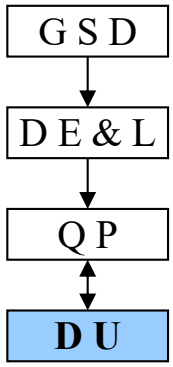
- Traditional query transformations
- Index intersection and union
- Advanced join algorithms
- Piggy-backed scans
  - Multiple queries with different selection criteria
- SQL extensions => new operators
  - Red Brick Systems has proposed 8 extensions, including:
    - MovingSum and MovingAvg
    - Rank … When
    - RatioToReport
    - Tertiles
    - Create Macro

Data Update



- Data sources change over time
- Must "refresh" the DW
  - Adds new historical data to the fact tables
  - Updates descriptive attributes in the dimension tables
  - Forces recalculation of measures in summary tables
- Issues:
  1. Monitoring/tracking changes at the data sources
  2. Recalculation of aggregated measures
  3. Refresh typically forces a shutdown for DW query processing

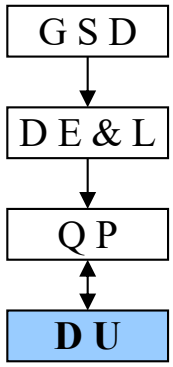
Monitoring Data Sources



Approaches:

1. Value-deltas - Capture before and after values of all tuples changed by normal DB operations and store them in differential relations.
   - Issues: must take the DW offline to install the modified values
2. Operation-deltas – Capture SQL updates from the transaction log of each data source and build a new log of all transactions that effect data in the DW.
   - Advantages: DW can remain online for query processing while executing data updates (using traditional concurrency control)
3. Hybrid – use value-deltas and operation-deltas for different data sources or a subset of the relations from a data source.

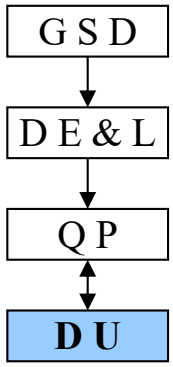
Creating a Differential Relation



Approaches at the Data Source:

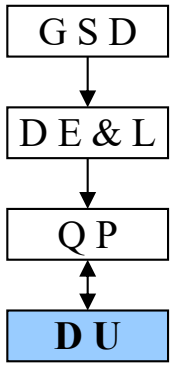
1. Execute the update query 3 times
   - (1) Select and record the before values;
     (2) Execute the update;
     (3) Select and record the after values
   - Issues: High cost in time & space;
     reduces autonomy of the data sources
2. Define and insert DB triggers
   - Triggers fire on "insert", "delete", and "update" operations; Log the before and after values
   - Issues: Not all data sources support triggers;
     reduces autonomy of the data sources

Creating Operation-Deltas



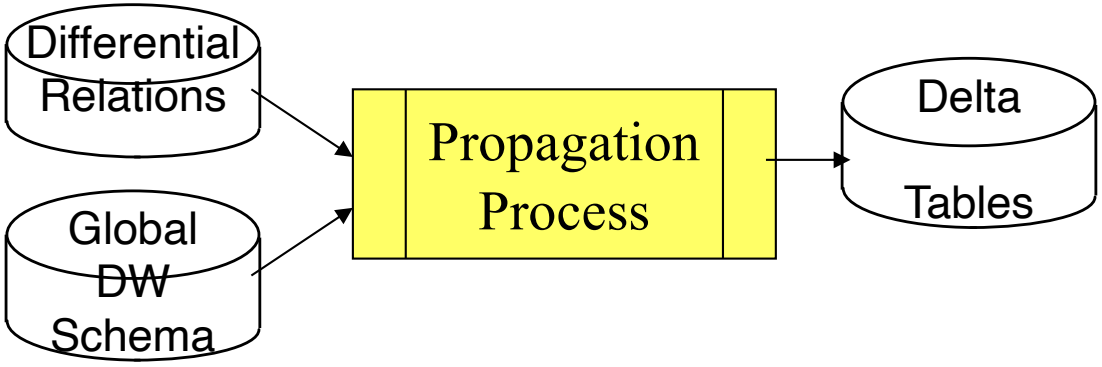
- The process:
  - Scan the transaction log at each data source
  - Select pertinent transactions and delta-log them
- Advantage:
  - Op-delta is much smaller than the value-delta
- Issues:
  - Must transform the update operation on the data source schema into an update operation on the DW schema – not always possible.
    Hence can not be used in all cases.

Recalculating Aggregated Measures



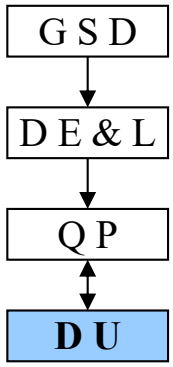
- Delta Tables
  - Assume we have differential relations for the base facts in the data sources (i.e., value deltas)
  - Two processing phases (Propagation & Refresh):

    1) Propagation – pre-compute all new tuples and all replacement tuples and store them in a delta table

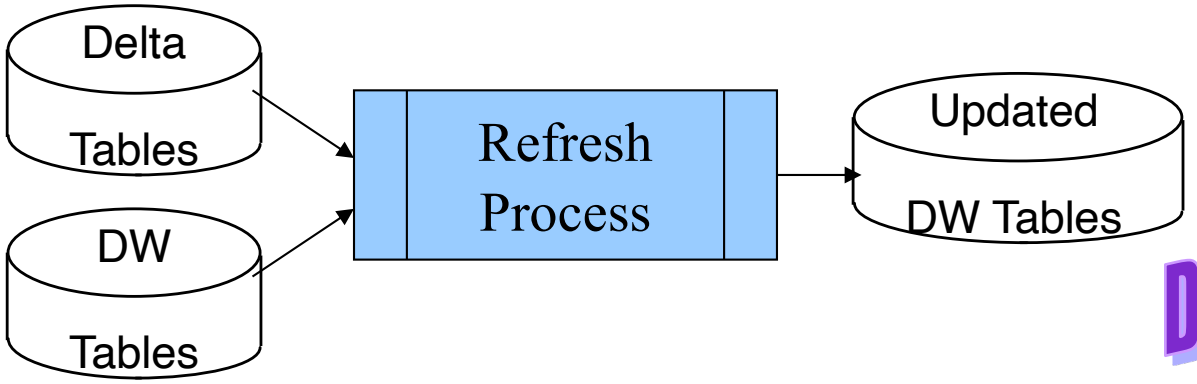


DW is online in phase #1

Recalculate Aggregated Measures



2) Refresh – Scan the DW tuples, replace existing tuples with the pre-computed tuple values, insert new tuples from the delta tables



DW is offline
ONLY in phase #2

Issue:
Can not pre-compute Delta Table for non-commutative measures
Ex: average (without #records), percentiles
Must compute these during the refresh phase.

Data Lake (DL)

- Collection of multi-modal data stored in their raw formats.
  - Each element has a unique identifier and metadata
  - For each business question, you can find the relevant data set to analyze it
- Often based on Hadoop software
  - Enterprise Hadoop

DL Advantages

- Schema on read
  - Write the data as they are, read them according to a diagram (e.g. code of the Map function)
  - More flexibility, multiple views of the same data
- Multi-workload data processing
  - Different types of processing on the same data
  - Interactive, batch, real time
- Cost-effective data architecture
  - Excellent cost/performance and RoI ratio with SN cluster and open source technologies

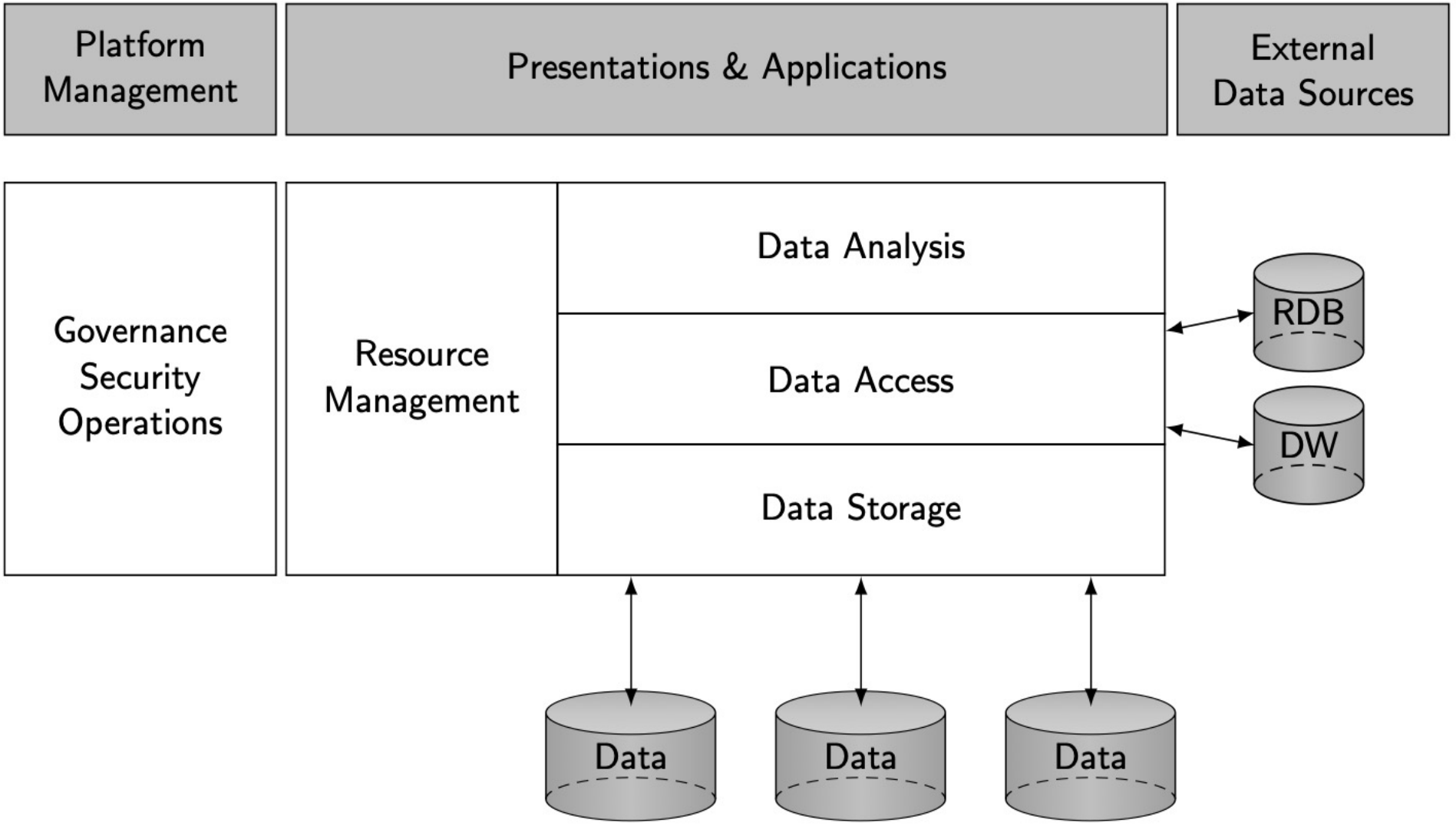
DL Principles

- Collect all useful data
  - Raw data, transformed data
- Dive from anywhere
  - Users from different business units can explore and enrich the data
- Flexible access
  - Different access paths to shared infrastructure
    - Batch, interactive (OLAP and BI), real-time, search,.....

Main Functions

- Data management, to store and process large amounts of data
- Data access: interactive, batch, real time, streaming
- Governance: load data easily, and manage it according to a policy implemented by the *data steward*
- Security: authentication, access control, data protection
- Platform management: provision, monitoring and scheduling of tasks (in a cluster)

DL Architecture



DL versus DW

Data Lake

- Data from het. various data sources
- Multi-workload processing
- Shorter development process
- Storing data objects as: id, metadata, original data
- No cleaning or transformation
- Schema-on-read
- Cost-effective architecture
- Easy to adapt & extend to changes
- Problems: acquire metadata, no data cleaning, consistency & integration, QP&opt., data quality, governance, security, …

Data Warehouse

- Physical DB integration
- ML (OLAP) workloads
- Long development process
- Upfront precise global schema and data modeling
- DB consistency
- Schema-on-write
- Query processing (QP) efficient
- Complex ETL process: extract, transform (clean, …), load
- Difficult & costly to adapt to changes
- Difficult schema and data updates