Universität Stuttgart

IPVS Institut für Parallele und Verteilte Systeme

Data-Warehouse-, Data-Mining- und OLAP-Technologien

Extraction, Transformation, Load

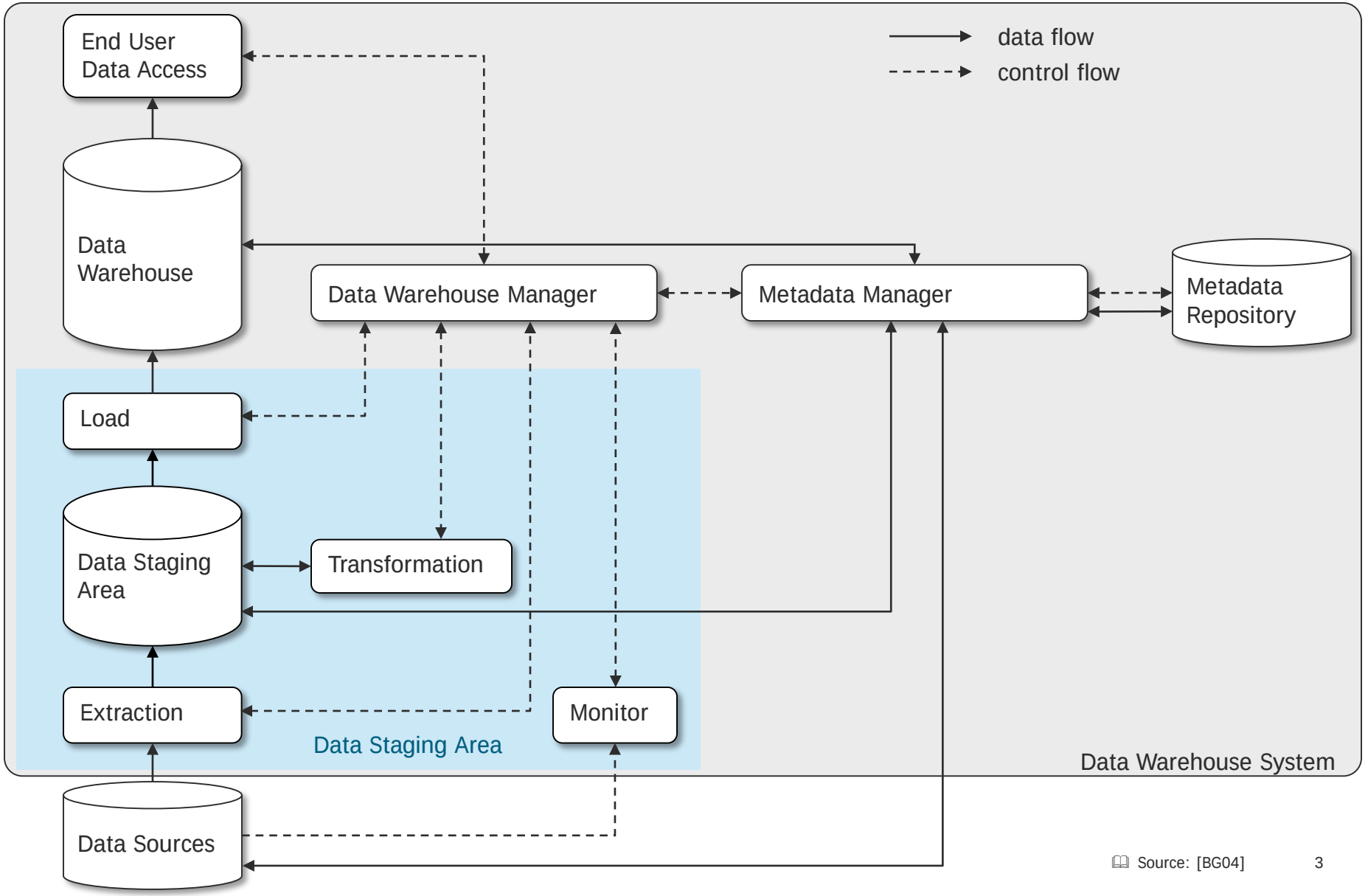
Bernhard Mitschang
Universität Stuttgart

Winter Term 2017/2018

Overview

- Monitoring
- Extraction
  - Export, Import, Filter, Load
  - Direct Integration
- Load
  - Bulk Load
  - Replication
  - Materialized Views
- Transformation
  - Schema Integration
  - Data Integration
  - Data Cleansing
- Tools

Architecture

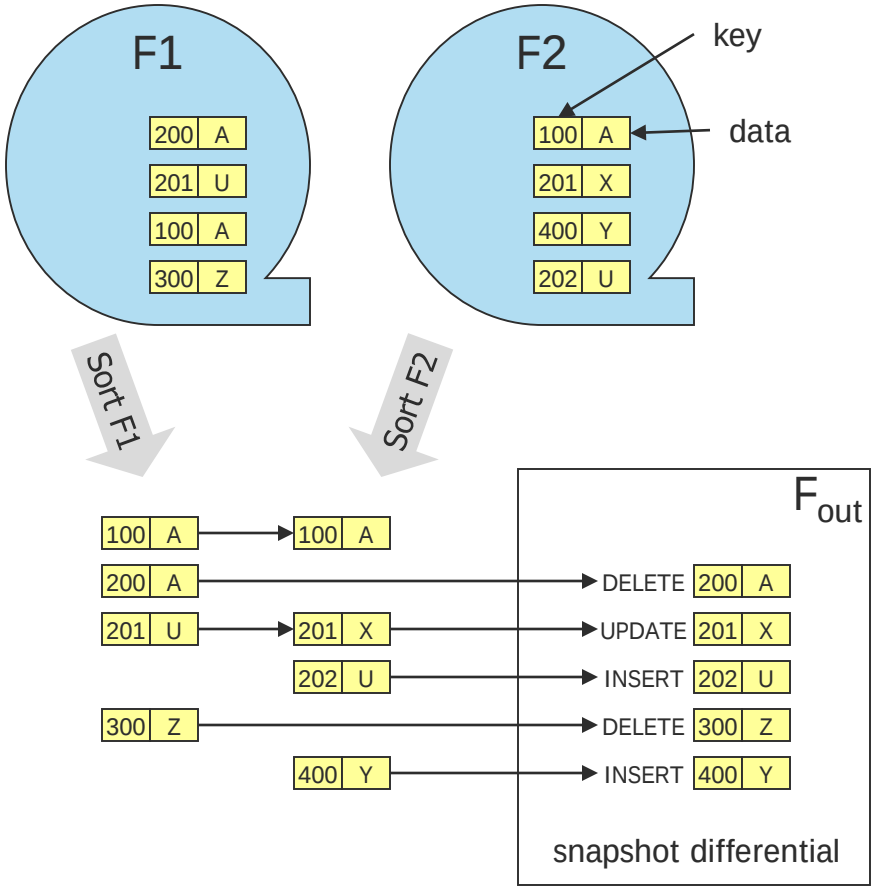


Monitoring

- Goal: Discover changes in data source incrementally
- Approaches

	Based on ...	Changes identified by ...
Trigger	triggers defined in source DBMS	trigger writes a copy of changed data to files
Replica	replication support of source DBMS	replication provides changed rows in a separate table
Timestamp	timestamp assigned to each row	use timestamp to identify changes (supported by temporal DBMS)
Log	log of source DBMS	read log
Snapshot	periodic snapshot of data source	compare snapshots

Snapshot Differentials



- Two snapshot files
  F1 was taken before F2
- Records contain key fields and data fields
- **Goal:** Provide UPDATES/ INSERTS/ DELETES in a shapshot differential file
- Sort Merge Outerjoin
  - sort F1 and F2 on their keys
  - read F1' and F2' and compare records
  - snapshot files may be compresssed
  - snapshots are read multiple times
- Window Algorithm
  - maintain a moving window of records in memory for each snapshot (aging buffer)
  - assumes that matching records are "physically" nearby
  - read snapshots only once

Window Algorithm

INPUT: F_1, F_2, n

OUTPUT: F_{out} /* the snapshot differential */

(1) Input $Buffer_1$ ← Read n blocks from F1

(2) Input $Buffer_2$ ← Read n blocks from F2

(3) while ((Input Buffer1 ≠ EMPTY) and
(Input Buffer 2 ≠ EMPTY))

(4) Match Input $Buffer_1$ against Input $Buffer_2$

(5) Match Input $Buffer_1$ against Aging $Buffer_2$

(6) Match Input $Buffer_2$ against Aging $Buffer_1$

(7) Put contents of Input $Buffer_1$
to Aging $Buffer_1$

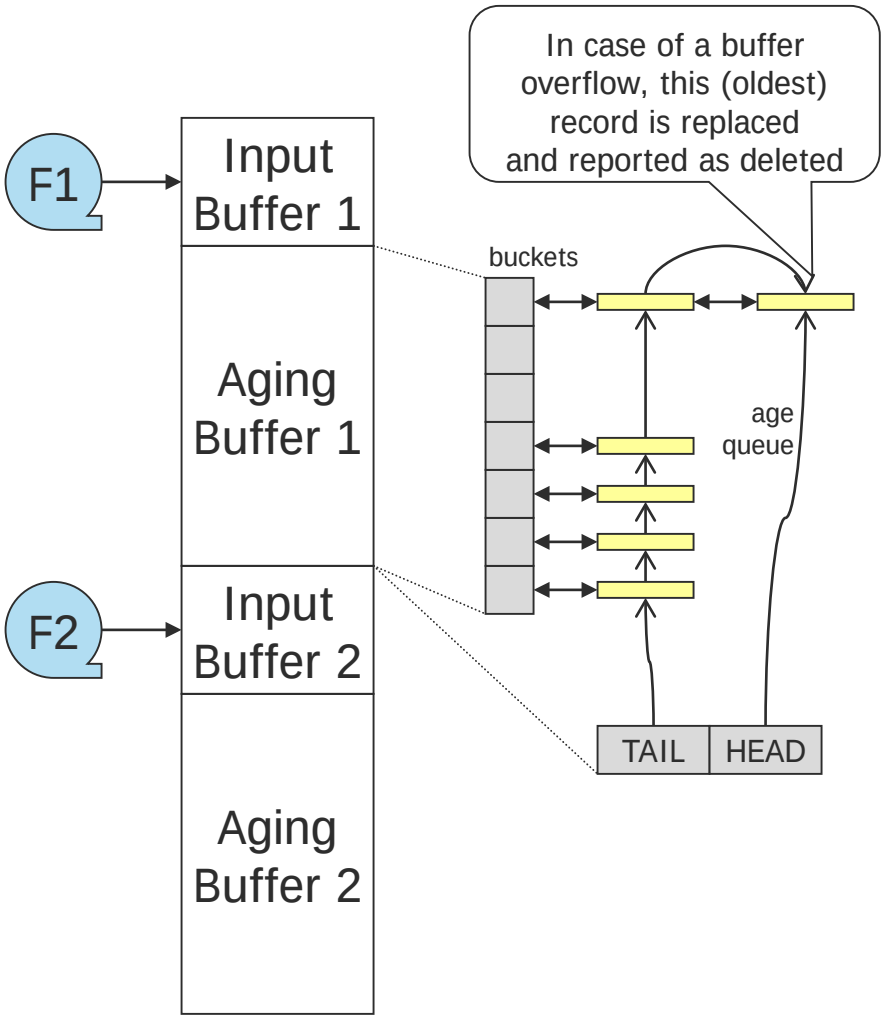
(8) Put contents of Input $Buffer_2$
to Aging $Buffer_2$

(9) Input $Buffer_1$ ← Read n blocks from F_1

(10) Input $Buffer_2$ ← Read n blocks from F_2

(11) Report records in Aging $Buffer_1$ as deletes

(12) Report records in Aging $Buffer_2$ as inserts

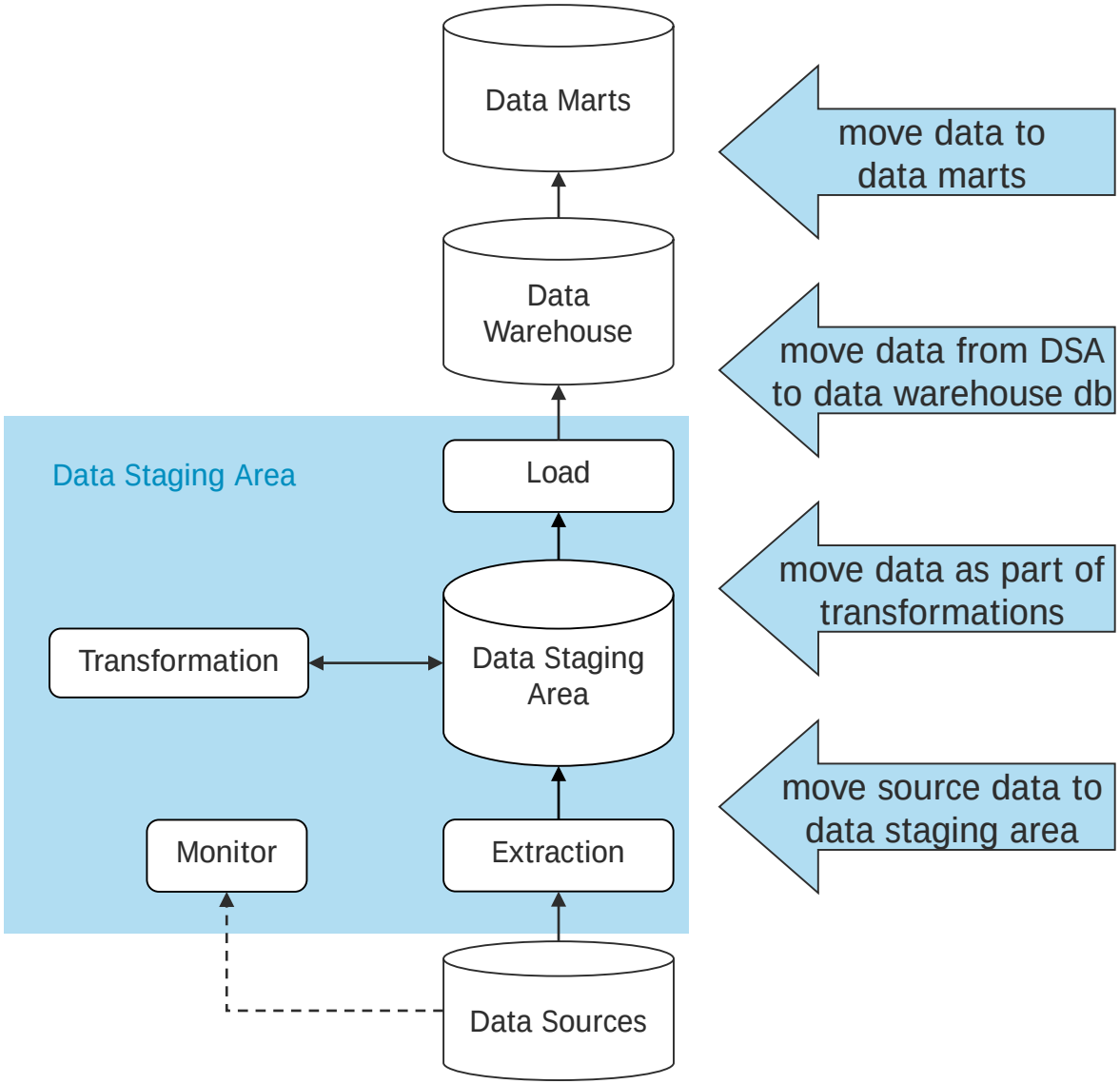


Source: [LG96]

Overview

- Monitoring
- Extraction
  - Export, Import, Filter, Load
  - Direct Integration
- Load
  - Bulk Load
  - Replication
  - Materialized Views
- Transformation
  - Schema Integration
  - Data Integration
  - Data Cleansing
- Tools

ETL Processing



Requirements:

- files vs. tables
- copy vs. transformation
- same system vs. heterogeneous systems
- automatic vs. user-driven
- availability of source and target system

Extraction

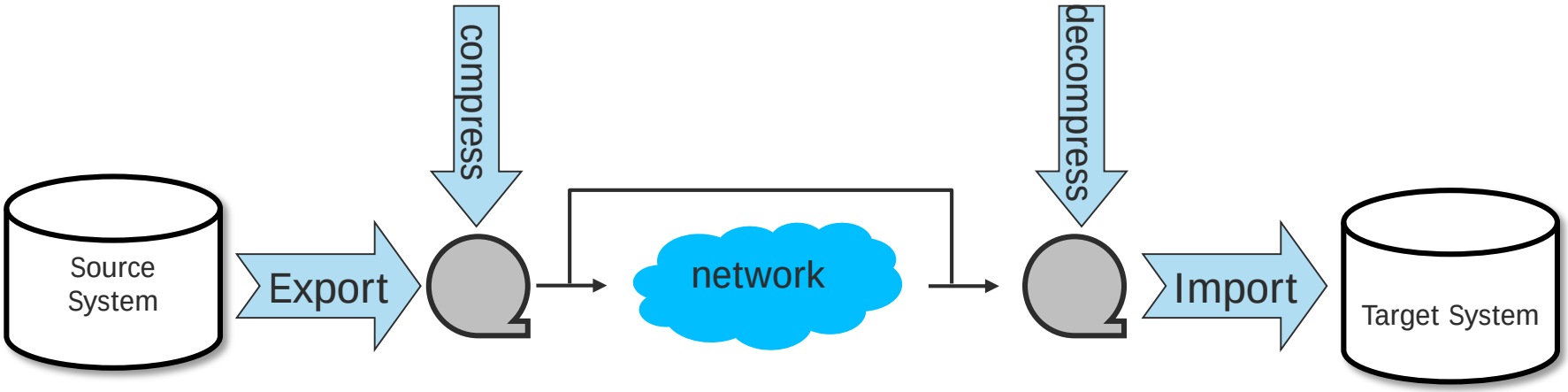
heterogeneous source systems

performance

- support of monitoring and extraction
  - replica
  - active db / trigger
  - snapshot
  - export / db dump
  - logging
  - no support
- accessing data sources
  - application / application API
  - database API
  - log files

- extract current data
- limited time frame
- service of source system should not be restricted

Export and Import




```
EXPORT TO       c:\cust_berlin_I\cust.data
OF              DEL
MODIFIED BY     COLDEL |
MESSAGES        c:\cust_berlin_I\msg1.txt
  SELECT        *
  FROM          customer_data
  WHERE         new_customer = true
```

(DB2)

- Export
  - ASCII files
  - proprietary format
- Import
  - import command
  - bulk load

Import vs. Load

```
IMPORT FROM      c:\cust_berlin_I\cust.data
OF               DEL
MODIFIED BY      COLDEL |
COMMITCOUNT      1000
MESSAGES         c:\cust_berlin_I\msg2.txt
INSERT INTO      cust_berlin_1
```

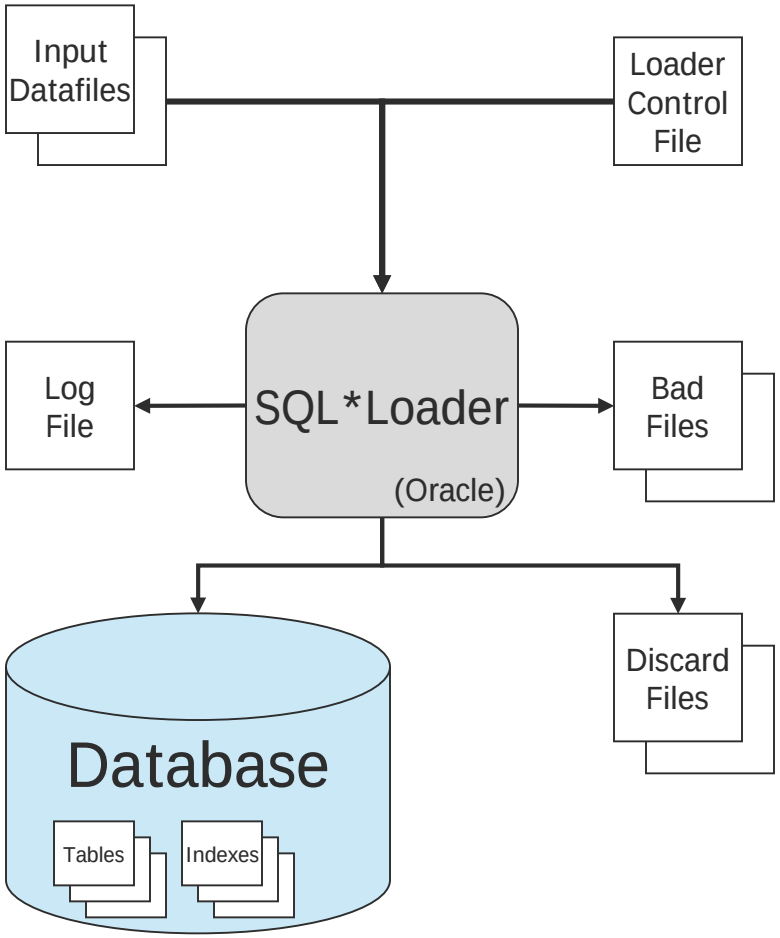
(DB2)

```
LOAD FROM        c:\cust_berlin_I\cust.data
OF               DEL
MODIFIED BY      COLDEL |
SAVECOUNT        1000
MESSAGES         c:\cust_berlin_I\msg3.txt
REPLACE INTO     cust_berlin_1
STATISTICS       YES
```

(DB2)

	IMPORT	LOAD
COMMIT	explicit	automatic
Logging	complete, mandatory	optional
Integrity	check all constraints	check local constraints only
Trigger	all	none
Locking	read access possible	table lock

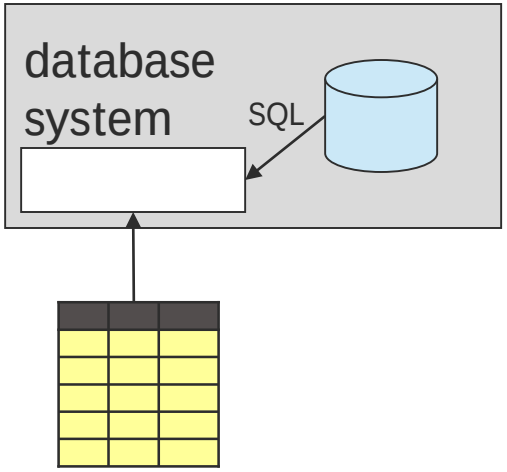
Filter and Load



- Bulk load tools provide some filter and transformation functionality
  - load data from multiple datafiles during the same load session.
  - load data into multiple tables during the same load session
  - specify the character set of the data
  - selectively load data, i.e. load records based on the records' values
  - manipulate the data before loading it, using SQL functions
  - generate unique sequential key values in specified columns

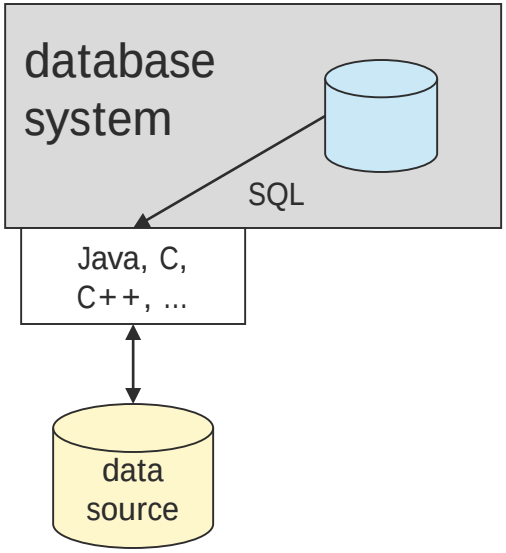
Direct Integration

external tables



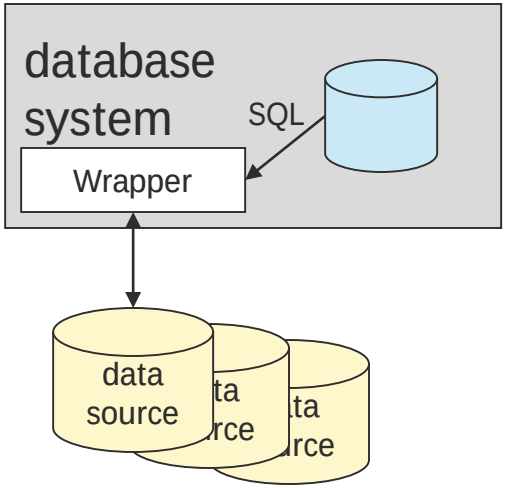
- register external data as tables
- allows to read external data
- no query optimization

table functions



- user-defined functions provides a table as result
- function reads data from external sources

federated database



- register external data as tables
- define wrapper for access to external data
- exploit capabilities of external source for query optimization

Overview

- Monitoring
- Extraction
  - Export, Import, Filter, Load
  - Direct Integration
- Load
  - Bulk Load
  - Replication
  - Materialized Views
- Transformation
  - Schema Integration
  - Data Integration
  - Data Cleansing
- Tools

Load

Transfer data from the data staging area into the data warehouse and data marts

- Bulk load is used to move huge amounts of data
- Data has to be added to existing tables
  - add rows
  - replace rows or values
- Flexible insert mechanism is needed to
  - add and update rows based on a single data source
  - add rows for a single data source to multiple tables in the data warehouse
- Consider complex criteria in load processing
  - write application program
  - use procedural extensions of SQL

Update and Insert

```
IMPORT FROM      c:\cust_berlin_I\cust.data
OF               DEL
MODIFIED BY      COLDEL |
COMMITCOUNT      1000
MESSAGES         c:\cust_berlin_I\msg2.txt
INSERT INTO      cust_berlin_1
                                    (DB2)
```

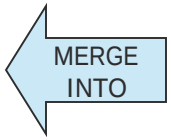
Insert Semantics?

- INSERT
  - adds the imported data to the table without changing the existing table data
- INSERT_UPDATE
  - adds rows of imported data to the target table, or updates existing rows (of the target table) with matching primary keys
- REPLACE
  - deletes all existing data from the table by truncating the data object, and inserts the imported data. The table definition and the index definitions are not changed.
- REPLACE_CREATE
  - if the table exists, deletes all existing data from the table by truncating the data object, and inserts the imported data without changing the table definition or the index definitions
  - if the table does not exist, creates the table and index definitions, as well as the row contents

MERGE INTO

customer

custkey	Name	Address
100	Ortmann	Rauchstr.
101	Martin	Pariser Platz
105	Fagiolo	Hiroshimastr.
106	Byrt	Lassenstr.



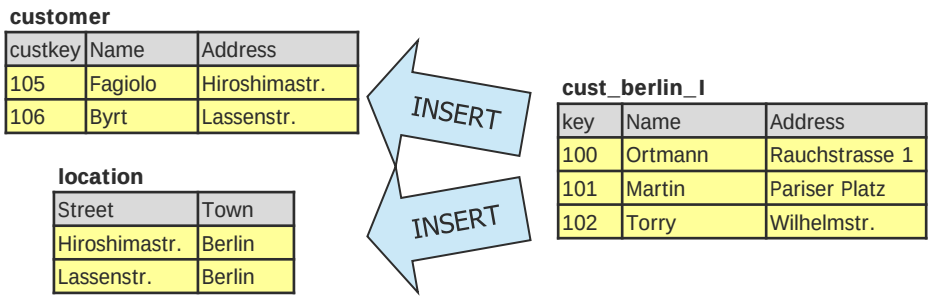
cust_berlin_I

key	Name	Address
100	Ortmann	Rauchstrasse 1
101	Martin	Pariser Platz
102	Torry	Wilhelmstr.

```
MERGE INTO      customer AS c1
USING (  SELECT  key, name, address, ...
         FROM    cust_berlin_I
         WHERE   ...) AS c2
ON ( c1.custkey = c2.key )
WHEN MATCHED THEN
         UPDATE SET c1.address = c2.address
WHEN NOT MATCHED THEN
         INSERT (custkey, name, address, ...
         VALUES (key, name, address ,...)
```

- 'transaction table' (cust_berlin I) contains updates to existing rows in the data warehouse and/or new rows that should be inserted
- MERGE Statement of SQL:2003 allows to
  - update rows that have a matching counterpart in the master table
  - insert rows that do not have a matching counterpart in the master table

Multiple Inserts

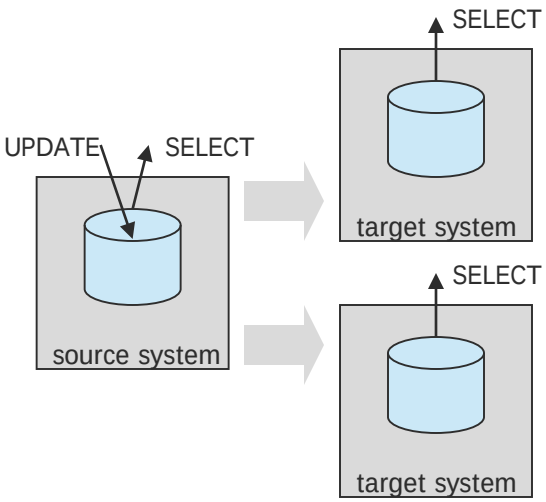


- Insert rows (partly) into several target tables
- Allows to insert the same row several times
- Allows to define conditions to select the target table
- INSERT FIRST defines that the row is inserted only once

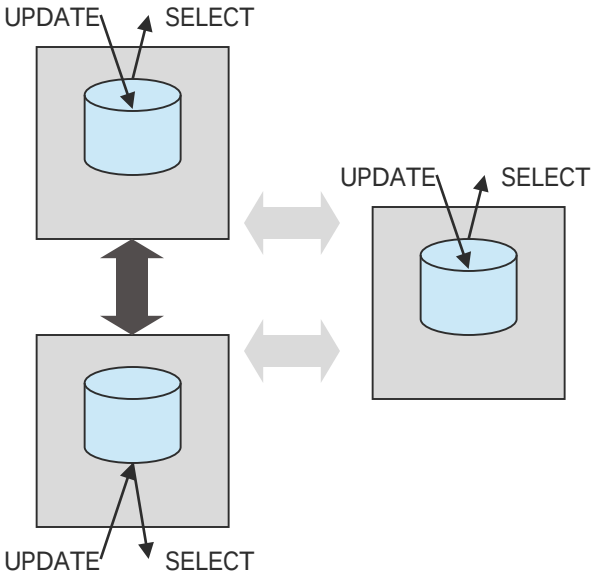
```
INSERT ALL
        INTO customer
        VALUES (key, name, address)
        INTO location
        VALUES (address, 'Berlin')
SELECT * FROM   cust_berlin_I WHERE...
                                        (Oracle)

INSERT ALL        /* INSERT FIRST */
WHEN key < 100
        INTO customer
        VALUES (key, name, address)
WHEN key < 1000
        INTO location
        VALUES (address, 'Berlin')
SELECT * FROM   cust_berlin_I WHERE...
                                        (Oracle)
```

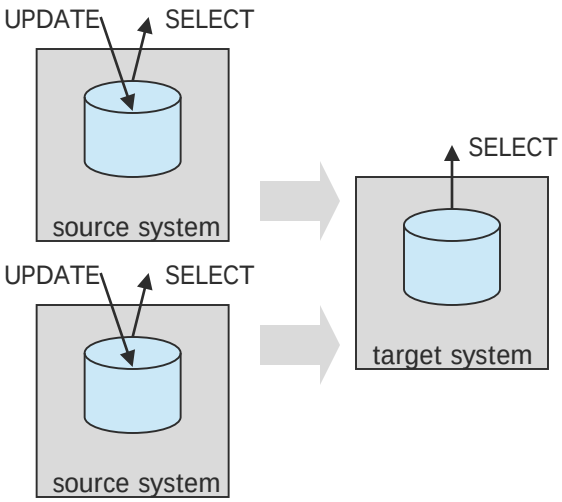
Replication



data distribution

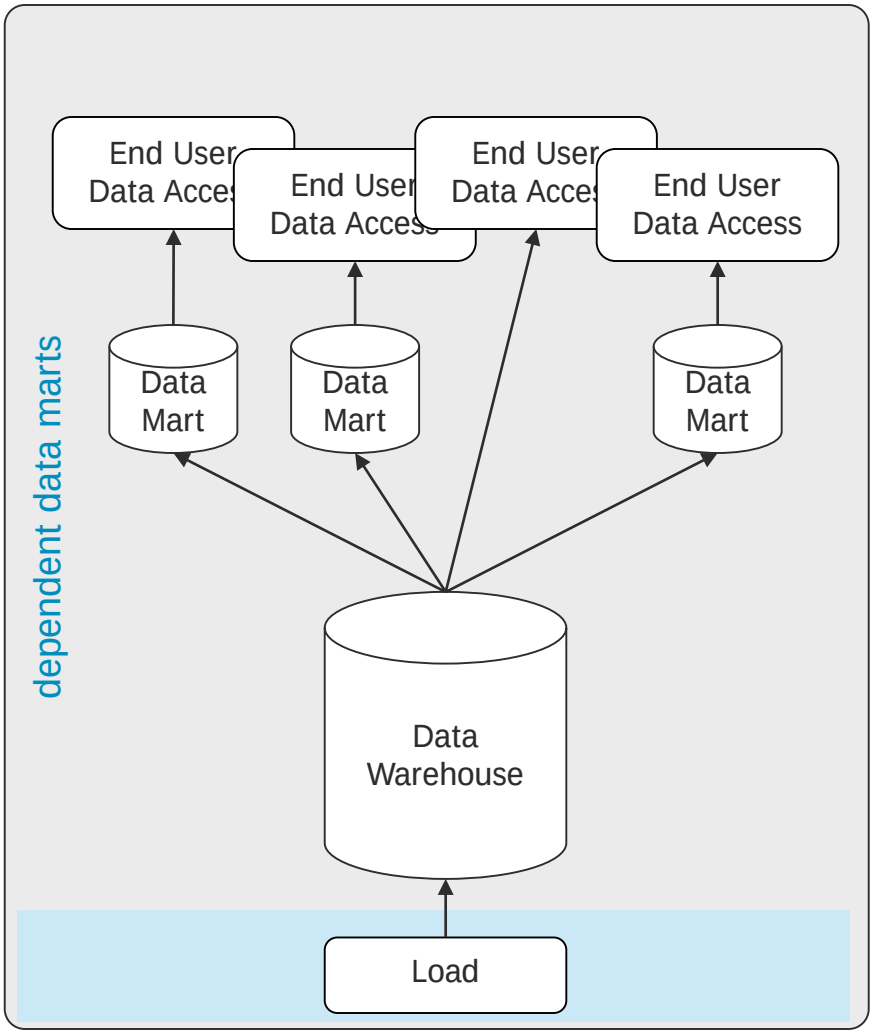


update anywhere



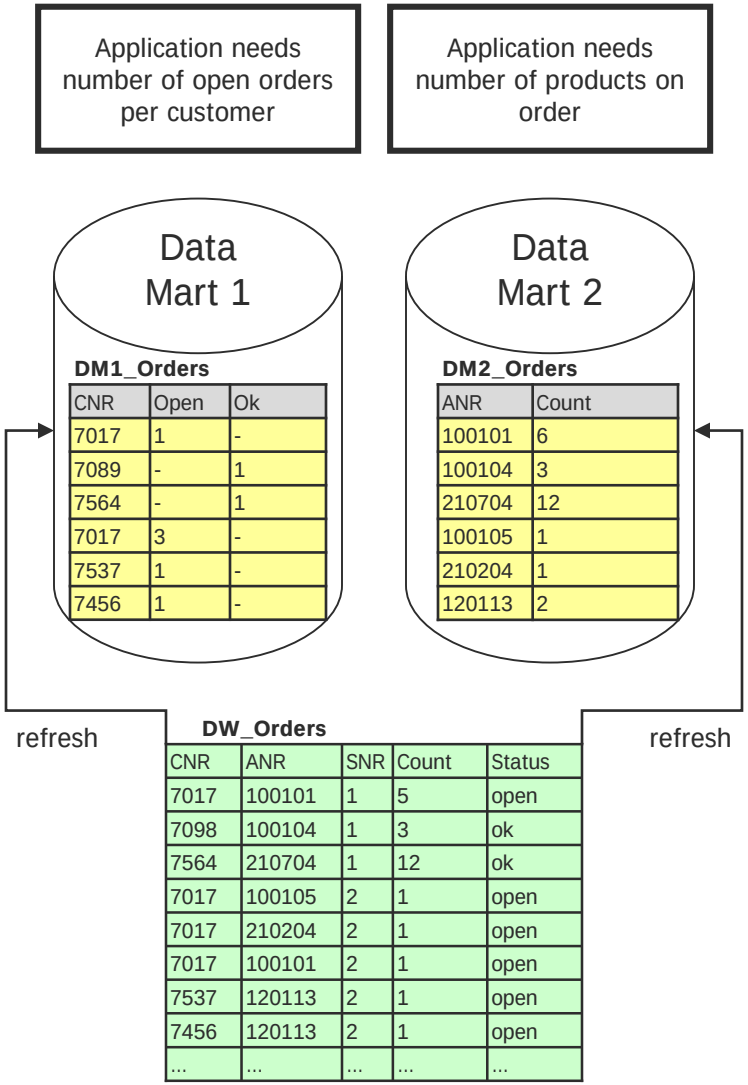
data consolidation

Materialized Views



- Data marts provide extracts of the data warehouse for a specific application
- Applications often need aggregated data
- Materialized Views (MV) allow to
  - define the content of each data mart as views on data warehouse tables
  - automatically update the content of a data mart
- Important Issues
  - MV selection
  - MV refresh
  - MV usage

Materialized Views




```
CREATE  TABLE DM2_Orders AS (
        SELECT  ANR, SUM(Count)
        FROM    DW_Orders
        GROUP BY ANR )
DATA INITIALLY DEFERRED
REFRESH DEFERRED;

REFRESH TABLE DM2_Orders;
```

(DB2)

- Materialized views are created like views
- A strategy for refreshing has to be specified
  - DEFERRED: Use REFRESH TABLE statement
  - IMMEDIATE: As part of the update in the source table

Overview

- Monitoring
- Extraction
  - Export, Import, Filter, Load
  - Direct Integration
- Load
  - Bulk Load
  - Replication
  - Materialized Views
- Transformation
  - Schema Integration
  - Data Integration
  - Data Cleansing
- Tools

Transformation

Convert the data into something representable to the users and valuable to the business

Transformation of structure **and** content

- Semantics → identify proper semantics
- Structure → schema integration
- Data → data integration and data cleansing

Transformation

Semantics

- Information on the same object is covered by several data sources
- E.g., customer information is provided by several source systems
- Identify synonyms

car	automobile
student	pupil
baby	infant

Indentify homonyms

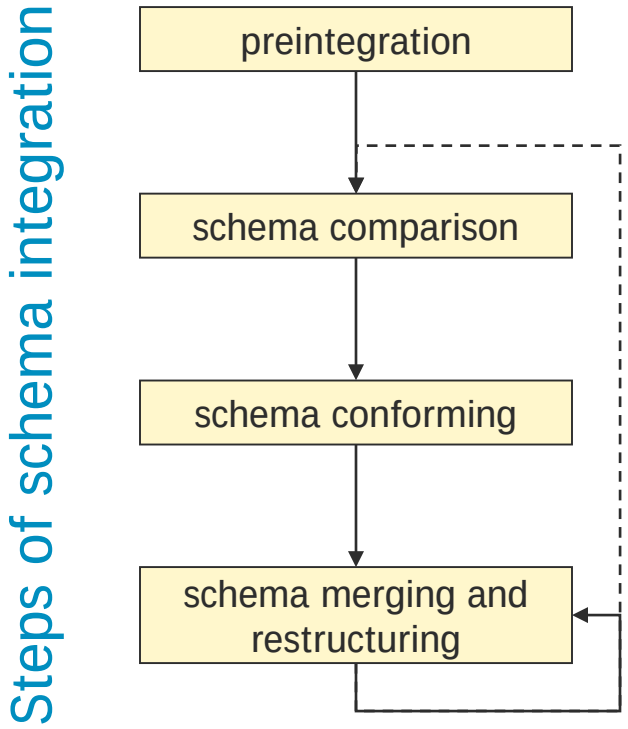
cash	cache
bare	bear
sight	site

- Identifying the proper semantics depends on the context
- Users have to define the proper semantics for the data warehouse
- Describe semantics in the metadata repository

Schema Integration

Activity of integrating the schemata of various sources to produce a homogeneous description of the data of interest

- Properties of the integrated schema
  - completeness
  - correctness
  - minimality
  - understandability

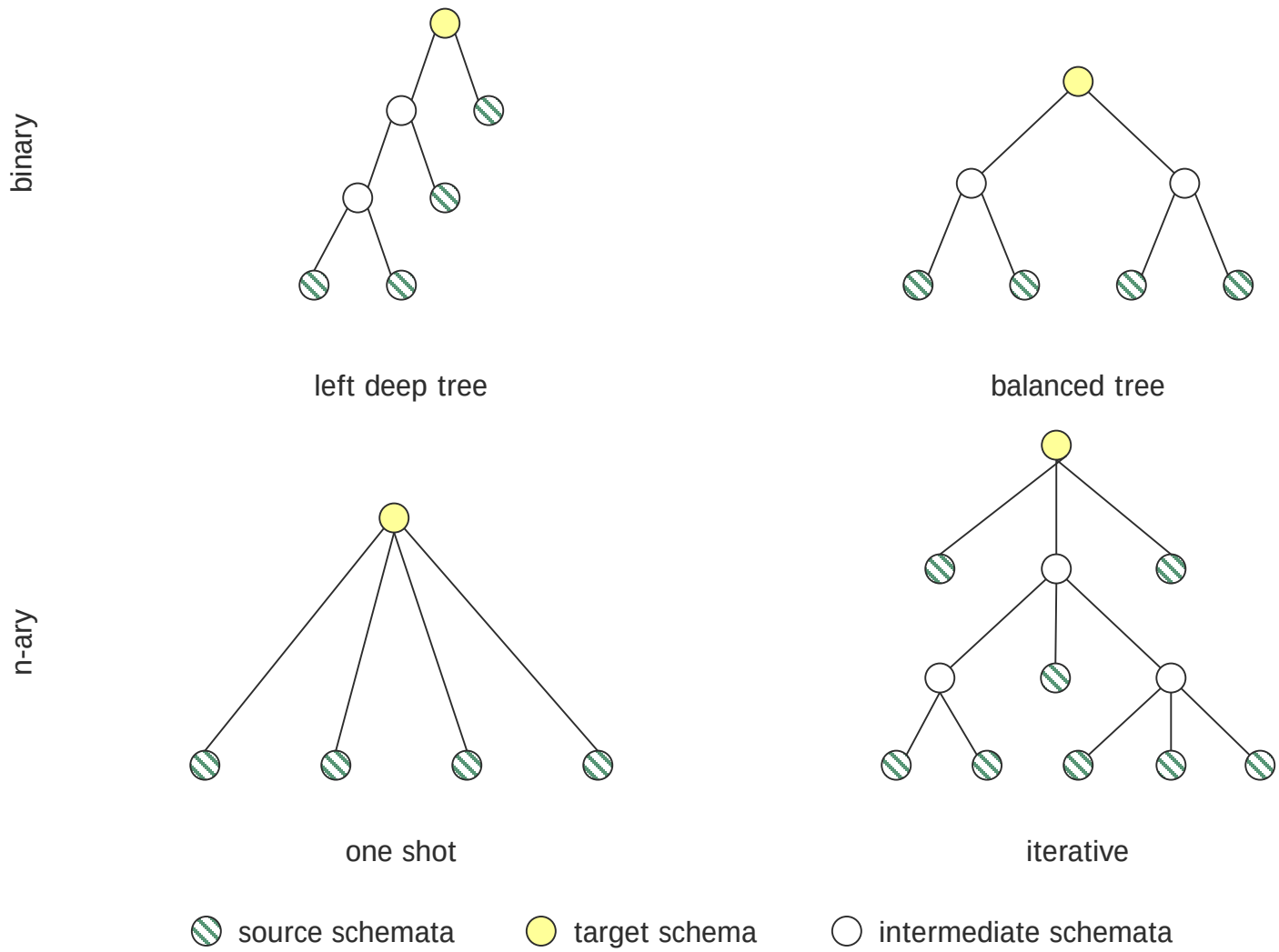


Pre-Integration

Analysis of the schemata to decide on the general integration policy

- Decide on
  - schemata to be integrated
  - order of integration / integration process
  - preferences

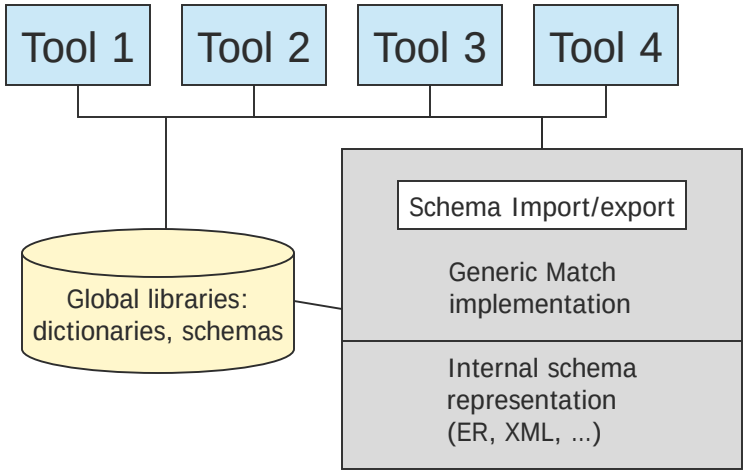
Schema Integration Process



Schema Matching

Take two schemas as input and produce a mapping between elements of the two schemas that correspond semantically to each other

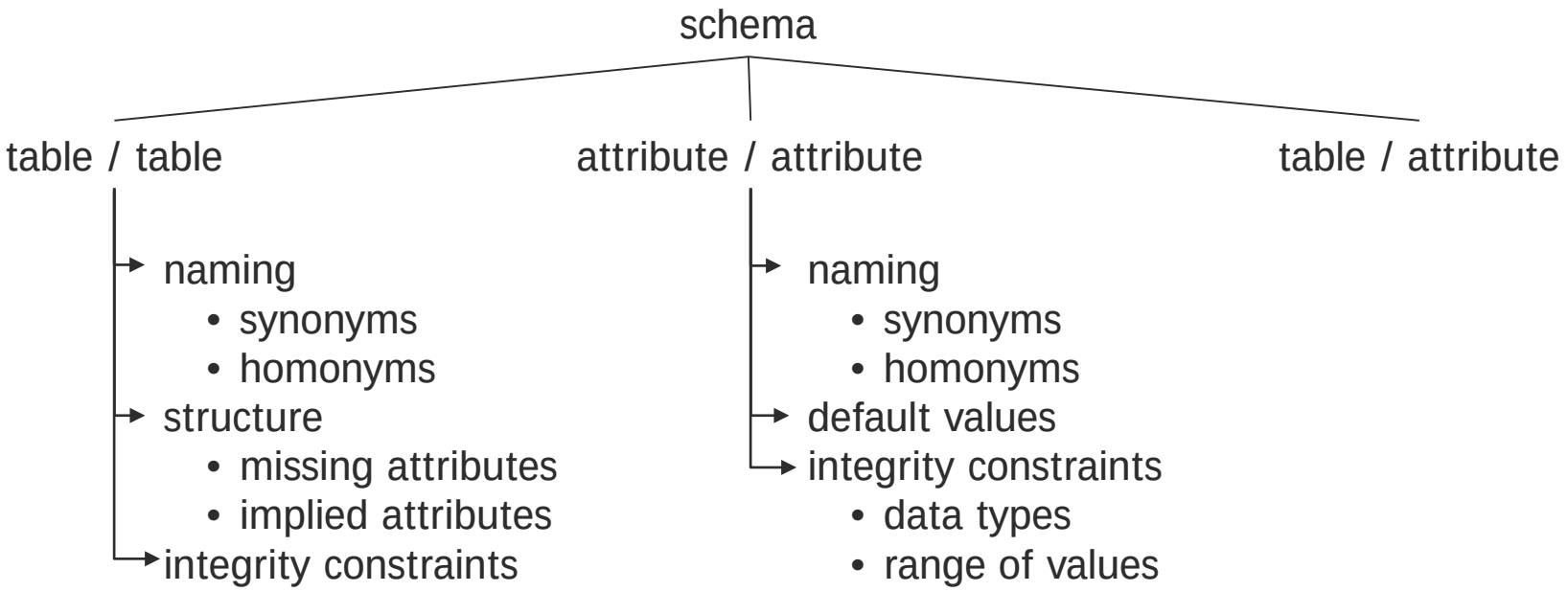
- Typically performed manually, supported by a graphical user interface.
  - tedious, time-consuming, error-prone, expensive
- General architecture of generic match
  - tools = schema-related apps.
  - internal schema representation + import and export needed
  - use libraries to help find matches
  - only determine match candidates
  - user may accept or reject



Schema Comparison

Analysis to determine the correlations among concepts of different schemata and to detect possible conflicts

- Schema Matching is part of this step
- Types of conflicts in relational systems

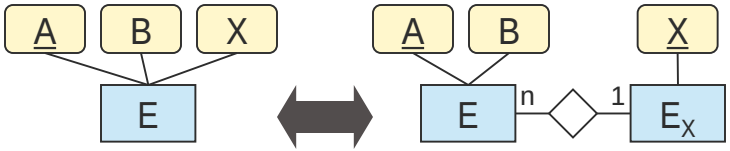


Schema Conforming

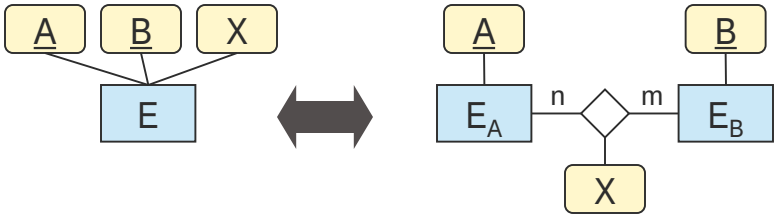
Conform and align schemata to make them compatible for integration

- Conflict resolution
  - based on the application context
  - cannot be fully automated
  - human intervention supported by graphical interfaces
- Sample steps
  - Attributes vs. Entity Sets
  - Composite Primary Keys
  - Redundancies
  - Simplification

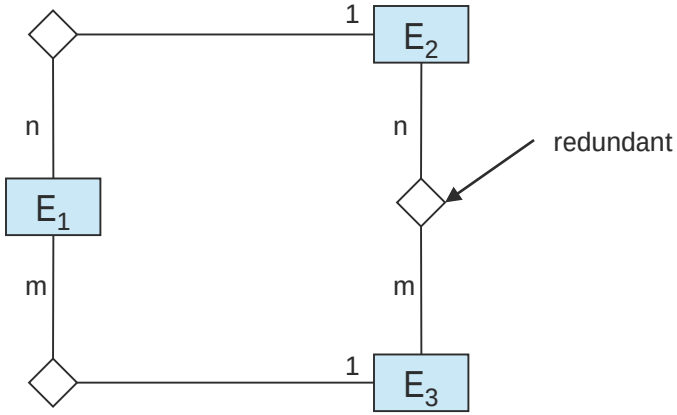
Schema Conforming



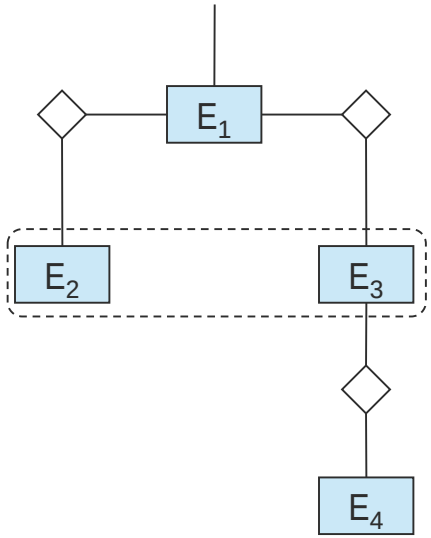
Attribute vs. Entity Set

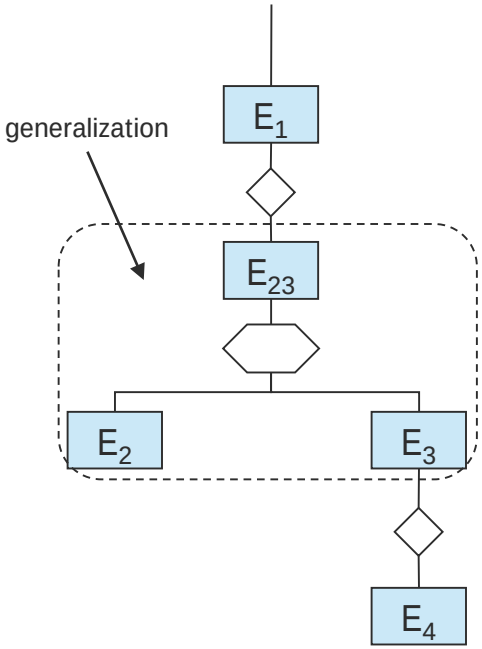


Composite Primary Keys



Redundancies



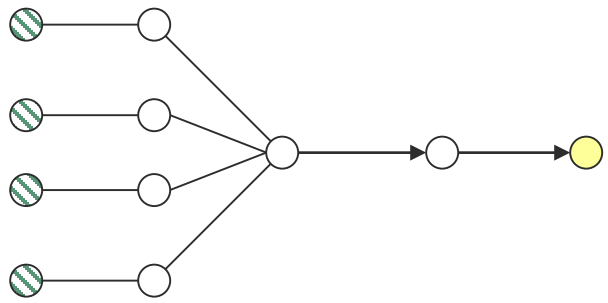


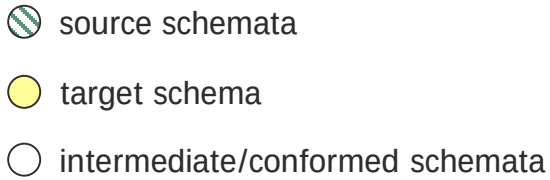
Simplification

Schema Merging and Restructuring

Conformed schemata are superimposed, thus obtaining a global schema

- Main steps
  - superimpose conformed schemata
  - quality tests against quality dimensions (completeness, correctness, minimality, understandability, …)
  - further transformation of the obtained schema





Schema Integration in Data Warehousing

Integration in data warehousing

- schema integration and data integration
- schema integration is a prerequisite for data integration
- schema integration is mainly used for the data staging area
- final data warehouse schema is defined from a global point of view, i.e., it is more than only integrating all source schemata
- schema matching between source schema and data warehouse schema provides the basis for defining transformations

Integration in federated systems

- focus on schema integration
- integrated schema is used

Data Integration

- Normalization / denormalization
  - depending on the source schema and the data warehouse schema
- Surrogate keys
  - keys should not depend on the source system
- Data type conversion
  - if data type of source attribute and target attribute differ
- Coding
  - text → coding; coding → text; coding A → coding B

examples

customer

system	local key	global key
1	107	5400345
1	109	5401340
2	107	4900342
2	214	5401340

character→ date
character → integer
'MM-DD-YYYY' → 'DD.MM.YYYY'

gross sales → 1
net sales → 2
3 → price
2 → GS

Data Integration

- Convert strings
  - standardization
- Convert date to date format of the target system
- Convert measures
- Combine / separate attributes
- Derived attributes
- Aggregation

examples

'Video' → ' video'
'VIDEO' → 'video'
'Miller, Max' → 'Max Miller'

2004, 05, 31 → 31.05.2004
04, 05, 31 → 31.05.2004
'05/31/2004' → 31.05.2004

inch→ cm
km → m
mph → km/h

2004, 05, 31 → 31.05.2004
'video', 1598 → 'video 1598'

net sales + tax → gross sales
on_stock - on_order → remaining

sales_per_day → sales_per_month

Data Cleansing

- Elementizing
  - identify fields

David and Clara Miller
Ste. 116
13150 Hiway 9
Box 1234 Boulder Crk
Colo 95006

first name 1: David
last name 1: Miller
first name 2: Clara
last name 2: Miller
suite: 116
number: 13150
street: Hiway 9
post box: 1234
city: Boulder Crk
state: Colo
zip: 95006

- Standardizing
  - format, coding

first name: David
last name: Miller
first name 2: Clara
last name 2: Miller
suite: 116
number: 13150
street: Highway 9
post box: 1234
city: Boulder Creek
state: Colorado
zip: 95006

- Verification
  - contradictions?
  - should lead to corrections in source system(s)

first name: David
last name: Miller
first name 2: Clara
last name 2: Miller
suite: 116
number: 13150
street: Highway 9
post box: 1234
city: Boulder Creek
state: California
zip: 95006

- Matching
  - is 'David Miller' and/or 'Clara Miller' already present in data warehouse?
  - if so, are there changed fields?
- Householding
  - 'David Miller' and 'Clara Miller' constitute a household
- Documenting

Dimensions of Data Cleansing

	single source	multiple sources
single record	• attribute dependencies (contradictions) • spelling mistakes • missing values • illegal values	• duplicates / matching • householding • contradictions • standardization, coding
multiple records	• primary key foreign key • duplicates / matching • householding	

Data Quality

consistency	• Are there contradictions in data and/or metadata?
correctness	• Do data and metadata provide an exact picture of the reality?
completeness	• Are there missing attributes or values?
exactness	• Are exact numeric values available? Are different objects identifiable? Homonyms?
reliability	• Is there a Standard Operating Procedure (SOP) that describes the provision of source data?
understandability	• Does a description for the data and coded values exist?
relevance	• Does the data contribute to the purpose of the data warehouse?

Improving Data Quality

- Assumption
  - Various projects can be undertaken to improve the quality of warehouse data
  - Goal: Identify the data quality enhancement projects that maximize value to the users of data
- Tasks for the data warehouse manager
  - Determine the organizational activities the data warehouse will support
  - Identify all sets of data needed to support the organizational activities
  - Estimate the quality of each data set on each relevant data quality dimension
- Identify a set of potential projects (and their cost) that could be undertaken for enhancing or affecting data quality
- Estimate for each project the likely effect of that project on the quality of the various data sets, by data quality dimension
- Determine for each project, data set, and relevant data quality dimension the change in utility should a particular project be undertaken

Improving Data Quality

I: Index of organizational activities supported by a data warehouse

J: Index for data sets

K: Index for data quality attributes or dimensions

L: Index for possible data quality projects P(1) … P(S)

Current quality: CQ(J, K)

Required quality: RQ(I, J, K)

Anticipated quality: AQ(J, K, L)

Priority of organizational activities: Weight(I)

Cost of data quality enhancement: Cost(L)

Value added: Utility(I, J, K, L)

Value of Project L = $\sum_{All\ I} Weight(I) \cdot \sum_{All\ J} \sum_{All\ K} Utility(I, J, K, L)$

Maximize: total value from all projects $\sum_{All\ L} X(L) * Value(L)$

Resource Constraint: $\sum_{All\ L} X(L) * Cost(L) \leq Budget$

Exclusiveness Constraint: $X(P(1)) + X(P(2)) + \ldots + X(P(S)) \leq 1$

Interaction Constraint: $X(P(1)) + X(P(2)) + X(P(3)) \leq 1$

$$X(L) = \begin{cases} 1, \text{ if Project L is selected} \\ 0, \text{ otherwise} \end{cases}$$

Overview

- Monitoring
- Extraction
  - Export, Import, Filter, Load
  - Direct Integration
- Load
  - Bulk Load
  - Replication
  - Materialized Views
- Transformation
  - Schema Integration
  - Data Integration
  - Data Cleansing
- ➡ Tools

ETL Market

- Vendors coming from several different backgrounds and perspectives

"Pure Play" Vendors

- ETL represents a core competency
- ETL accounts for most of the license revenue
- This class of vendors is driving the bulk of innovation and "mind share" in the ETL market

Database Management System (DBMS) Vendors

- database vendors have an increasing impact on this market as they continue to bundle ETL functionality closer to the relational DBMS

Business Intelligence Vendors

- Business intelligence tools and platforms are their core competency
- For most of these vendors, ETL technology plays a supporting role to their flagship business intelligence offerings, or is one component of a broad offering including business intelligence and ETL

Other Infrastructure Providers

- They provide various types of technical infrastructure components beyond the DBMS
- ETL is typically positioned as yet another technical toolset in their portfolios

ETL Tools

- Informatica – PowerCenter
- Talend – Talend Open Studio for Data Integration
- Oracle – Data Integrator
- Microsoft – SQL Server Integrated Services
- IBM – Infosphere Information Server
- SAS – Data Integration studio
- SAP – BusinessObjects Data Integrator
- Clover ETL – CloverETL
- Pentaho – Pentaho Data Integration
- AB – Initio

Gartner Magic Quadrant for Data Integration Tools

Summary

- Moving data is part of most steps of the ETL process
  - extraction
  - transformation
  - loading data warehouse and data marts
- Several approaches available
  - export, import, load
  - direct integration
  - replication
  - materialized views
- Transformation steps include
  - semi-automatic schema matching and integration
  - data integration steps
  - data cleansing

Papers

[BT99] D. Ballou, G. K. Tayi: Enhancing Data Quality in Data Warehouse Environments. Communications of the ACM, Vol. 42, No. 1, 1999.

[LG96] W. Labio, H. Garcia-Molina: Efficient Snapshot Differential Algorithms for Data Warehousing. Proc. of 22th International Conference on Very Large Data Bases, Mumbai (Bombay), India, 1996.

[RB01] E. Rahm, P. Bernstein: A survey of approaches to automatic schema matching. VLDB Journal 10:334-350 (2001)