Distributed Data Management

Distributed DBMSs

Thorsten Papenbrock

F-2.04, Campus II

Hasso Plattner Institut

Definition

Distributed Database

- A distributed database is a collection of multiple, logically interrelated databases located at the nodes of a distributed system.

(M. Tamer Özsu, Patrick Valduriez: "Principles of Distributed Database Systems")

Distributed System

- A distributed computing system is a number of autonomous processing elements (not necessarily homogeneous) that are interconnected by a computer network and that cooperate in performing their assigned task.

(M. Tamer Özsu, Patrick Valduriez: "Principles of Distributed Database Systems")

Distributed Database Management System

- A distributed database management system is the software system that permits the management of the distributed database and makes the distribution transparent to the user.

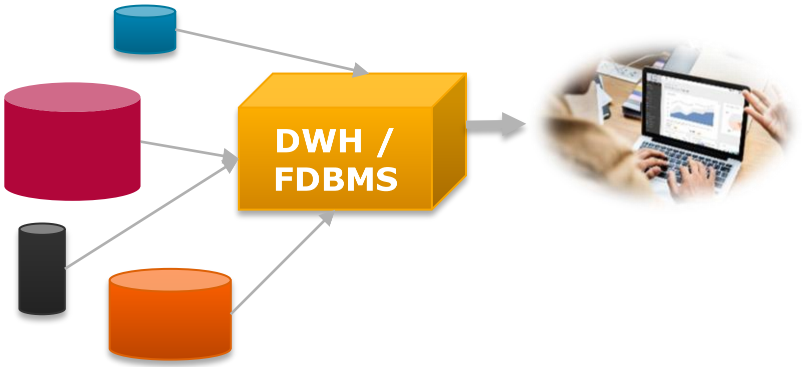
(M. Tamer Özsu, Patrick Valduriez: "Principles of Distributed Database Systems")

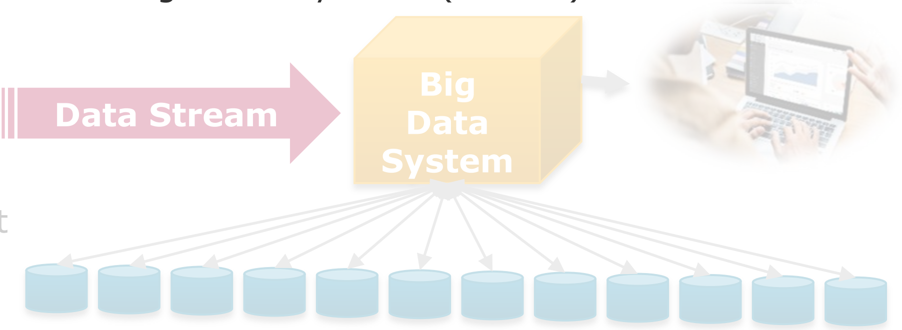
Distributed DBMSs

Drivers of Distribution

Distributed data creation

- Relevant data is created distributedly by independent sources/systems but integrated to enable global analytics.
  - ➢ Traditional cause for data distribution
- Goal: Integrate disconnected data in one central system to satisfy an information need.
- Systems: Data Warehouses (DWH); Federated Database Management Systems (FDBMS)





Distributed data processing

- Relevant data is artificially distributed to independent workers/systems for faster data analytics.
  - ➢ Modern cause for data distribution
- Goal: Partition and distribute large datasets to satisfy storage and analytical needs.
- Systems: Big Data Analytics Systems (sharded DBMSs, batch- and stream systems)

Motivation

Example: Question of a Biologist

Question: “Find all humans ESTs (DNA-sequences), that according to BLAST are at least 60% and across at least 50 amino acids identical with mouse-channel genes in the tissue of the central nervous system.”

- Various sources of information needed:
  - Mouse Genome Database (MGD) @ Jackson Labs
  - SwissProt @ EBI
  - BLAST tool @ NCBI
  - GenBank nucleotide sequence database @ NCBI

Source for example: *A Practitioner’s Guide to Data Management and Data Integration in Bioinformatics,* Barbara A. Eckman in Bioinformatics by Zoe Lacroix and Terence Critchlow, 2003, Morgan Kaufmann.

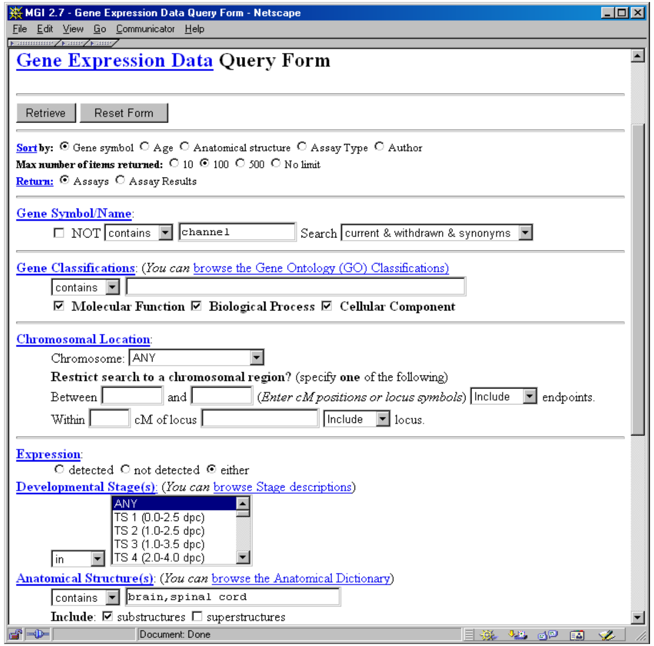
Example: Question of a Biologist

Hasso Plattner Institut

Question: "Find all humans ESTs (DNA-sequences), that according to BLAST are at least 60% and across at least 50 amino acids identical with mouse-channel genes in the tissue of the central nervous system."

Source for example:
A Practitioner's Guide to Data Management and Data Integration in Bioinformatics, Barbara A. Eckman in Bioinformatics by Zoe Lacroix and Terence Critchlow, 2003, Morgan Kaufmann.

1. Find „channel" sequence in tissue of central nervous system in MGD HTML-form


MGI 2.7 - Gene Expression Data Query Form - Netscape

File Edit View Go Communicator Help

Gene Expression Data Query Form

Retrieve | Reset Form

Sort by: ◉ Gene symbol ○ Age ○ Anatomical structure ○ Assay Type ○ Author
Max number of items returned: ○ 10 ◉ 100 ○ 500 ○ No limit
Return: ◉ Assays ○ Assay Results

Gene Symbol/Name:
☐ NOT [contains] [channel] Search [current & withdrawn & synonyms]

Gene Classifications: (You can browse the Gene Ontology (GO) Classifications)
[contains] [ ]
☑ Molecular Function ☑ Biological Process ☑ Cellular Component

Chromosomal Location:
Chromosome: [ANY]
Restrict search to a chromosomal region? (specify one of the following)
Between [ ] and [ ] (Enter cM positions or locus symbols) [Include] endpoints.
Within [ ] cM of locus [ ] [Include] locus.

Expression:
○ detected ○ not detected ◉ either
Developmental Stage(s): (You can browse Stage descriptions)
[in] [ANY / TS 1 (0.0-2.5 dpc) / TS 2 (1.0-2.5 dpc) / TS 3 (1.0-3.5 dpc) / TS 4 (2.0-4.0 dpc)]
Anatomical Structure(s): (You can browse the Anatomical Dictionary)
[contains] [brain, spinal cord]
Include: ☑ substructures ☐ superstructures

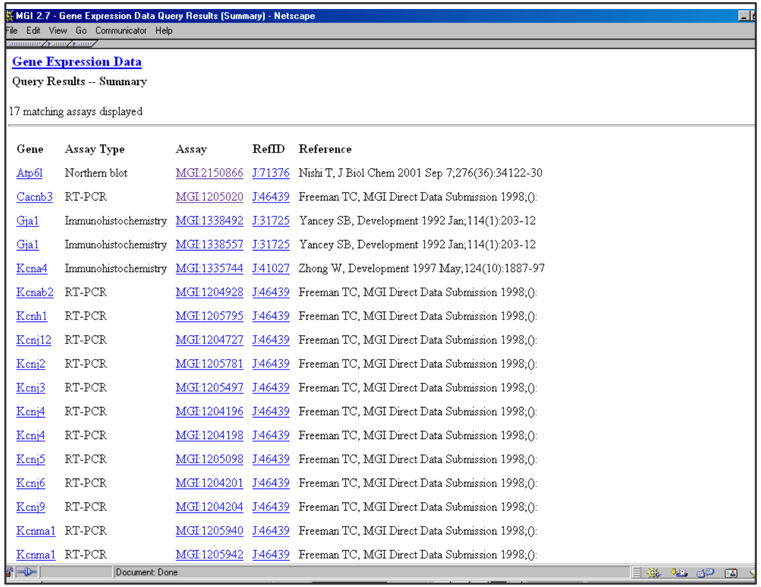
Document: Done

Example: Question of a Biologist

Question: "Find all humans ESTs (DNA-sequences), that according to BLAST are at least 60% and across at least 50 amino acids identical with mouse-channel genes in the tissue of the central nervous system."

1. Find „channel" sequence in tissue of central nervous system in MGD HTML-form

- MGD Result
  - 14 genes from 17 experiments


MGI 2.7 - Gene Expression Data Query Results (Summary) - Netscape

File Edit View Go Communicator Help

Gene Expression Data

Query Results -- Summary

17 matching assays displayed

Gene	Assay Type	Assay	RefID	Reference
Atp6l	Northern blot	MGI:2150866	J:71376	Nishi T, J Biol Chem 2001 Sep 7;276(36):34122-30
Cacnb3	RT-PCR	MGI:1205020	J:46439	Freeman TC, MGI Direct Data Submission 1998;():
Gja1	Immunohistochemistry	MGI:1338492	J:31725	Yancey SB, Development 1992 Jan;114(1):203-12
Gja1	Immunohistochemistry	MGI:1338557	J:31725	Yancey SB, Development 1992 Jan;114(1):203-12
Kcna4	Immunohistochemistry	MGI:1335744	J:41027	Zhong W, Development 1997 May;124(10):1887-97
Kcnab2	RT-PCR	MGI:1204928	J:46439	Freeman TC, MGI Direct Data Submission 1998;():
Kcnh1	RT-PCR	MGI:1205795	J:46439	Freeman TC, MGI Direct Data Submission 1998;():
Kcnj12	RT-PCR	MGI:1204727	J:46439	Freeman TC, MGI Direct Data Submission 1998;():
Kcnj2	RT-PCR	MGI:1205781	J:46439	Freeman TC, MGI Direct Data Submission 1998;():
Kcnj3	RT-PCR	MGI:1205497	J:46439	Freeman TC, MGI Direct Data Submission 1998;():
Kcnj4	RT-PCR	MGI:1204196	J:46439	Freeman TC, MGI Direct Data Submission 1998;():
Kcnj4	RT-PCR	MGI:1204198	J:46439	Freeman TC, MGI Direct Data Submission 1998;():
Kcnj5	RT-PCR	MGI:1205098	J:46439	Freeman TC, MGI Direct Data Submission 1998;():
Kcnj6	RT-PCR	MGI:1204201	J:46439	Freeman TC, MGI Direct Data Submission 1998;():
Kcnj9	RT-PCR	MGI:1204204	J:46439	Freeman TC, MGI Direct Data Submission 1998;():
Kcnma1	RT-PCR	MGI:1205940	J:46439	Freeman TC, MGI Direct Data Submission 1998;():
Kcnma1	RT-PCR	MGI:1205942	J:46439	Freeman TC, MGI Direct Data Submission 1998;():

Document: Done

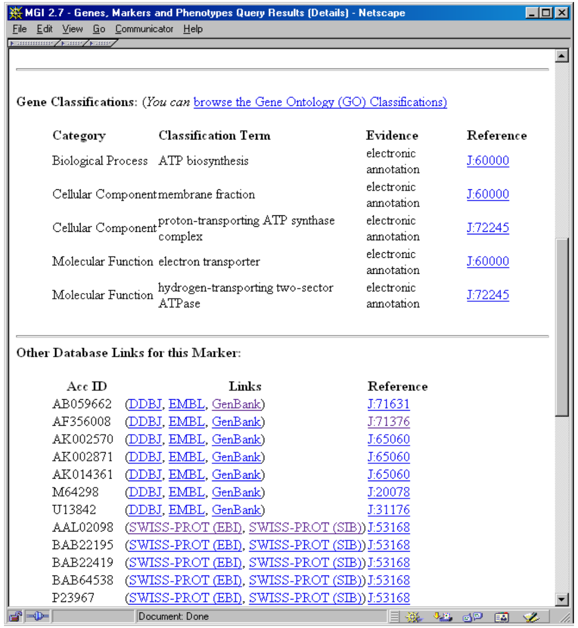
Source for example: *A Practitioner's Guide to Data Management and Data Integration in Bioinformatics,* Barbara A. Eckman in Bioinformatics by Zoe Lacroix and Terence Critchlow, 2003, Morgan Kaufmann.

Motivation

Example: Question of a Biologist

Question: "Find all humans ESTs (DNA-sequences), that according to BLAST are at least 60% and across at least 50 amino acids identical with mouse-channel genes in the tissue of the central nervous system."

1. Find „channel" sequence in tissue of central nervous system in MGD HTML-form
2. Examine the details for each of the 14 genes on SwissProt
   - On average 5 SwissProt links per gene



MGI 2.7 - Genes, Markers and Phenotypes Query Results (Details) - Netscape

File Edit View Go Communicator Help

Gene Classifications: (*You can* browse the Gene Ontology (GO) Classifications)

Category	Classification Term	Evidence	Reference
Biological Process	ATP biosynthesis	electronic annotation	J:60000
Cellular Component	membrane fraction	electronic annotation	J:60000
Cellular Component	proton-transporting ATP synthase complex	electronic annotation	J:72245
Molecular Function	electron transporter	electronic annotation	J:60000
Molecular Function	hydrogen-transporting two-sector ATPase	electronic annotation	J:72245

Other Database Links for this Marker:

Acc ID	Links	Reference
AB059662	(DDBJ, EMBL, GenBank)	J:71631
AF356008	(DDBJ, EMBL, GenBank)	J:71376
AK002570	(DDBJ, EMBL, GenBank)	J:65060
AK002871	(DDBJ, EMBL, GenBank)	J:65060
AK014361	(DDBJ, EMBL, GenBank)	J:65060
M64298	(DDBJ, EMBL, GenBank)	J:20078
U13842	(DDBJ, EMBL, GenBank)	J:31176
AAL02098	(SWISS-PROT (EBI), SWISS-PROT (SIB))	J:53168
BAB22195	(SWISS-PROT (EBI), SWISS-PROT (SIB))	J:53168
BAB22419	(SWISS-PROT (EBI), SWISS-PROT (SIB))	J:53168
BAB64538	(SWISS-PROT (EBI), SWISS-PROT (SIB))	J:53168
P23967	(SWISS-PROT (EBI), SWISS-PROT (SIB))	J:53168

Document: Done

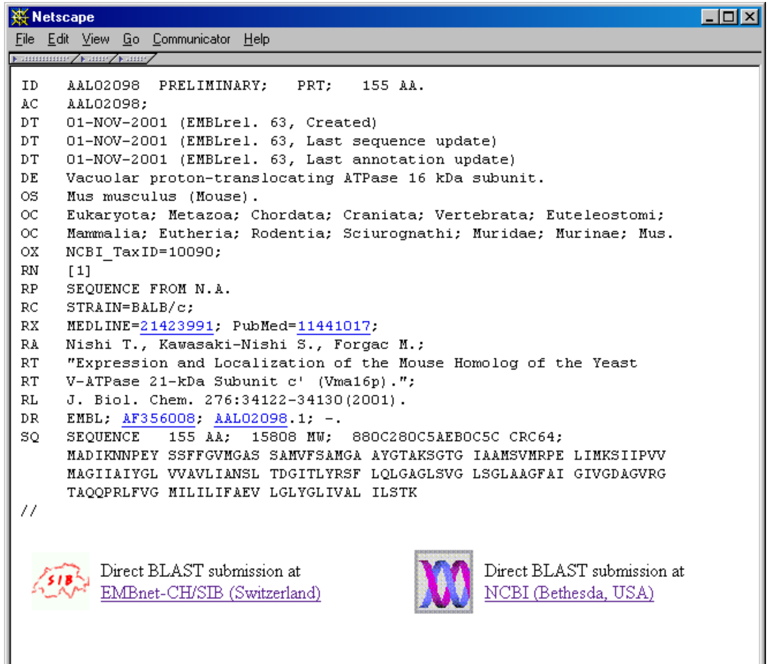
Source for example:
A Practitioner's Guide to Data Management and Data Integration in Bioinformatics, Barbara A. Eckman in Bioinformatics by Zoe Lacroix and Terence Critchlow, 2003, Morgan Kaufmann.

Motivation

Example: Question of a Biologist

Question: “Find all humans ESTs (DNA-sequences), that according to BLAST are at least 60% and across at least 50 amino acids identical with mouse-channel genes in the tissue of the central nervous system.”

1. Find „channel“ sequence in tissue of central nervous system in MGD HTML-form
2. Examine the details for each of the 14 genes on SwissProt
3. Examine each SwissProt entry with the BLAST algorithm


Netscape
File Edit View Go Communicator Help

```
ID   AAL02098  PRELIMINARY;   PRT;   155 AA.
AC   AAL02098;
DT   01-NOV-2001 (EMBLrel. 63, Created)
DT   01-NOV-2001 (EMBLrel. 63, Last sequence update)
DT   01-NOV-2001 (EMBLrel. 63, Last annotation update)
DE   Vacuolar proton-translocating ATPase 16 kDa subunit.
OS   Mus musculus (Mouse).
OC   Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
OC   Mammalia; Eutheria; Rodentia; Sciurognathi; Muridae; Murinae; Mus.
OX   NCBI_TaxID=10090;
RN   [1]
RP   SEQUENCE FROM N.A.
RC   STRAIN=BALB/c;
RX   MEDLINE=21423991; PubMed=11441017;
RA   Nishi T., Kawasaki-Nishi S., Forgac M.;
RT   "Expression and Localization of the Mouse Homolog of the Yeast
RT   V-ATPase 21-kDa Subunit c' (Vma16p).";
RL   J. Biol. Chem. 276:34122-34130(2001).
DR   EMBL; AF356008; AAL02098.1; -.
SQ   SEQUENCE   155 AA;  15808 MW;  880C280C5AEB0C5C CRC64;
     MADIKNNPEY SSFFGVMGAS SAMVFSAMGA AYGTAKSGTG IAAMSVMRPE LIMKSIIPVV
     MAGIIAIYGL VVAVLIANSL TDGITLYRSF LQLGAGLSVG LSGLAAGFAI GIVGDAGVRG
     TAQQPRLFVG MILILIFAEV LGLYGLIVAL ILSTK
//
```

Direct BLAST submission at EMBnet-CH/SIB (Switzerland)

Direct BLAST submission at NCBI (Bethesda, USA)

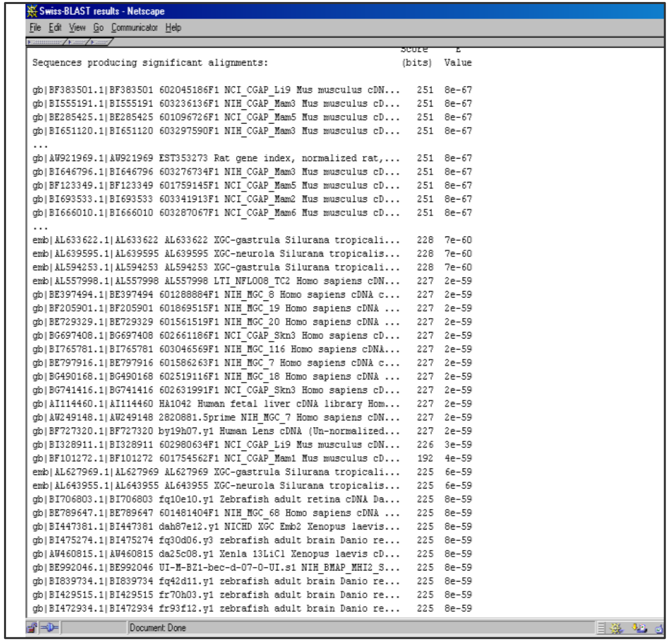
Source for example:
A Practitioner’s Guide to Data Management and Data Integration in Bioinformatics, Barbara A. Eckman in Bioinformatics by Zoe Lacroix and Terence Critchlow, 2003, Morgan Kaufmann.

Example: Question of a Biologist

Question: "Find all humans ESTs (DNA-sequences), that according to BLAST are at least 60% and across at least 50 amino acids identical with mouse-channel genes in the tissue of the central nervous system."

Source for example:
A Practitioner's Guide to Data Management and Data Integration in Bioinformatics,
Barbara A. Eckman in Bioinformatics by Zoe Lacroix and Terence Critchlow, 2003, Morgan Kaufmann.

1. Find „channel" sequence in tissue of central nervous system in MGD HTML-form
2. Examine the details for each of the 14 genes on SwissProt
3. Examine each SwissProt entry with the BLAST algorithm
4. Examine each BLAST result to…
    1. eliminate non-human hits
    2. check other predicates (>60% identical, etc.).



```
Swiss-BLAST results - Netscape
File Edit View Go Communicator Help

                                                               Score     E
Sequences producing significant alignments:                    (bits)  Value

gb|BF383501.1|BF383501 602045186F1 NCI_CGAP_Li9 Mus musculus cDN...   251  8e-67
gb|BI555191.1|BI555191 603236136F1 NIH_CGAP_Mam3 Mus musculus cD...   251  8e-67
gb|BE285425.1|BE285425 601096726F1 NCI_CGAP_Mam5 Mus musculus cD...   251  8e-67
gb|BI651120.1|BI651120 603297590F1 NIH_CGAP_Mam3 Mus musculus cD...   251  8e-67
...
gb|AW921969.1|AW921969 EST353273 Rat gene index, normalized rat,...   251  8e-67
gb|BI646796.1|BI646796 603276734F1 NIH_CGAP_Mam3 Mus musculus cD...   251  8e-67
gb|BF123349.1|BF123349 601759145F1 NCI_CGAP_Mam5 Mus musculus cD...   251  8e-67
gb|BI693533.1|BI693533 603341913F1 NCI_CGAP_Mam2 Mus musculus cD...   251  8e-67
gb|BI666010.1|BI666010 603287067F1 NCI_CGAP_Mam6 Mus musculus cD...   251  8e-67
...
emb|AL633622.1|AL633622 AL633622 XGC-gastrula Silurana tropicali...   228  7e-60
emb|AL639595.1|AL639595 AL639595 XGC-neurola Silurana tropicalis...   228  7e-60
emb|AL594253.1|AL594253 AL594253 XGC-gastrula Silurana tropicali...   228  7e-60
emb|AL557998.1|AL557998 AL557998 LTI_NFL008_TC2 Homo sapiens cDN...   227  2e-59
gb|BE397494.1|BE397494 601288884F1 NIH_MGC_8 Homo sapiens cDNA c...   227  2e-59
gb|BF205901.1|BF205901 601869515F1 NIH_MGC_19 Homo sapiens cDNA ...   227  2e-59
gb|BE729329.1|BE729329 601561519F1 NIH_MGC_20 Homo sapiens cDNA ...   227  2e-59
gb|BG697408.1|BG697408 602661186F1 NCI_CGAP_Skn3 Homo sapiens cD...   227  2e-59
gb|BI765781.1|BI765781 603046569F1 NIH_MGC_116 Homo sapiens cDNA...   227  2e-59
gb|BE797916.1|BE797916 601586263F1 NIH_MGC_7 Homo sapiens cDNA c...   227  2e-59
gb|BG490168.1|BG490168 602519116F1 NIH_MGC_18 Homo sapiens cDNA ...   227  2e-59
gb|BG741416.1|BG741416 602631991F1 NCI_CGAP_Skn3 Homo sapiens cD...   227  2e-59
gb|AI114460.1|AI114460 HA1042 Human fetal liver cDNA library Hom...   227  2e-59
gb|AW249148.1|AW249148 2820881.5prime NIH_MGC_7 Homo sapiens cDN...   227  2e-59
gb|BF727320.1|BF727320 by19h07.y1 Human Lens cDNA (Un-normalized...   227  2e-59
gb|BI328911.1|BI328911 602980634F1 NCI_CGAP_Li9 Mus musculus cDN...   226  3e-59
gb|BF101272.1|BF101272 601754562F1 NCI_CGAP_Mam1 Mus musculus cD...   192  4e-59
emb|AL627969.1|AL627969 AL627969 XGC-gastrula Silurana tropicali...   225  6e-59
emb|AL643955.1|AL643955 AL643955 XGC-neurola Silurana tropicalis...   225  6e-59
gb|BI706803.1|BI706803 fq10e10.y1 Zebrafish adult retina cDNA Da...   225  8e-59
gb|BE789647.1|BE789647 601481404F1 NIH_MGC_68 Homo sapiens cDNA ...   225  8e-59
gb|BI447381.1|BI447381 dah87e12.y1 NICHD XGC Emb2 Xenopus laevis...   225  8e-59
gb|BI475274.1|BI475274 fq30d06.y3 zebrafish adult brain Danio re...   225  8e-59
gb|AW460815.1|AW460815 da25c08.y1 Xenla 13LiCl Xenopus laevis cD...   225  8e-59
gb|BE992046.1|BE992046 UI-M-BZ1-bec-d-07-0-UI.s1 NIH_BMAP_MHI2_S...   225  8e-59
gb|BI839734.1|BI839734 fq42d11.y1 zebrafish adult brain Danio re...   225  8e-59
gb|BI429515.1|BI429515 fr70h03.y1 zebrafish adult brain Danio re...   225  8e-59
gb|BI472934.1|BI472934 fr93f12.y1 zebrafish adult brain Danio re...   225  8e-59

Document Done
```

Motivation

Example: Question of a Biologist

Question: "Find all humans ESTs (DNA-sequences), that according to BLAST are at least 60% and across at least 50 amino acids identical with mouse-channel genes in the tissue of the central nervous system."

1. Find „channel" sequence in tissue of central nervous system in MGD HTML-form
2. Examine the details for each of the 14 genes on SwissProt
3. Examine each SwissProt entry with the BLAST algorithm
4. Examine each BLAST result
5. For each remaining result: retrieve EST-sequence from GenBank

Source for example:
A Practitioner's Guide to Data Management and Data Integration in Bioinformatics, Barbara A. Eckman in Bioinformatics by Zoe Lacroix and Terence Critchlow, 2003, Morgan Kaufmann.

Example: Question of a Biologist

Question: "Find all humans ESTs (DNA-sequences), that according to BLAST are at least 60% and across at least 50 amino acids identical with mouse-channel genes in the tissue of the central nervous system."

1. Find „channel" sequence in tissue of central nervous system in MGD HTML-form
2. Examine the details for each of the 4 entries of SwissProt
3. Examine each SwissProt entry with the BLAST algorithm
4. Examine each BLAST result
5. For each remaining result: retrieve EST-sequence from GenBank

Phew!

Source for example: *A Practitioner's Guide to Data Management and Data Integration in Bioinformatics*, Barbara A. Eckman in Bioinformatics by Zoe Lacroix and Terence Critchlow, 2003, Morgan Kaufmann.

Example: Question of a Biologist

Question: "Find all humans ESTs (DNA-sequences), that according to BLAST are at least 60% and across at least 50 amino acids identical with mouse-channel genes in the tissue of the central nervous system."

- If there was an integrated database with all four sources, the following "simple" SQL query does all previous manual steps:

```
SELECT   g.accnum,g.sequence
FROM     genbank g, blast b, swissprot s, mgd m
WHERE    m.exp = "CNS"
AND      m.defn LIKE "%channel%"
AND      m.spid = s.id AND s.seq = b.query
AND      b.hit = g.accnum
AND      b.percentid >= 60 AND b.alignlen >= 50
```

Source for example:
A Practitioner's Guide to Data Management and Data Integration in Bioinformatics, Barbara A. Eckman in Bioinformatics by Zoe Lacroix and Terence Critchlow, 2003, Morgan Kaufmann.

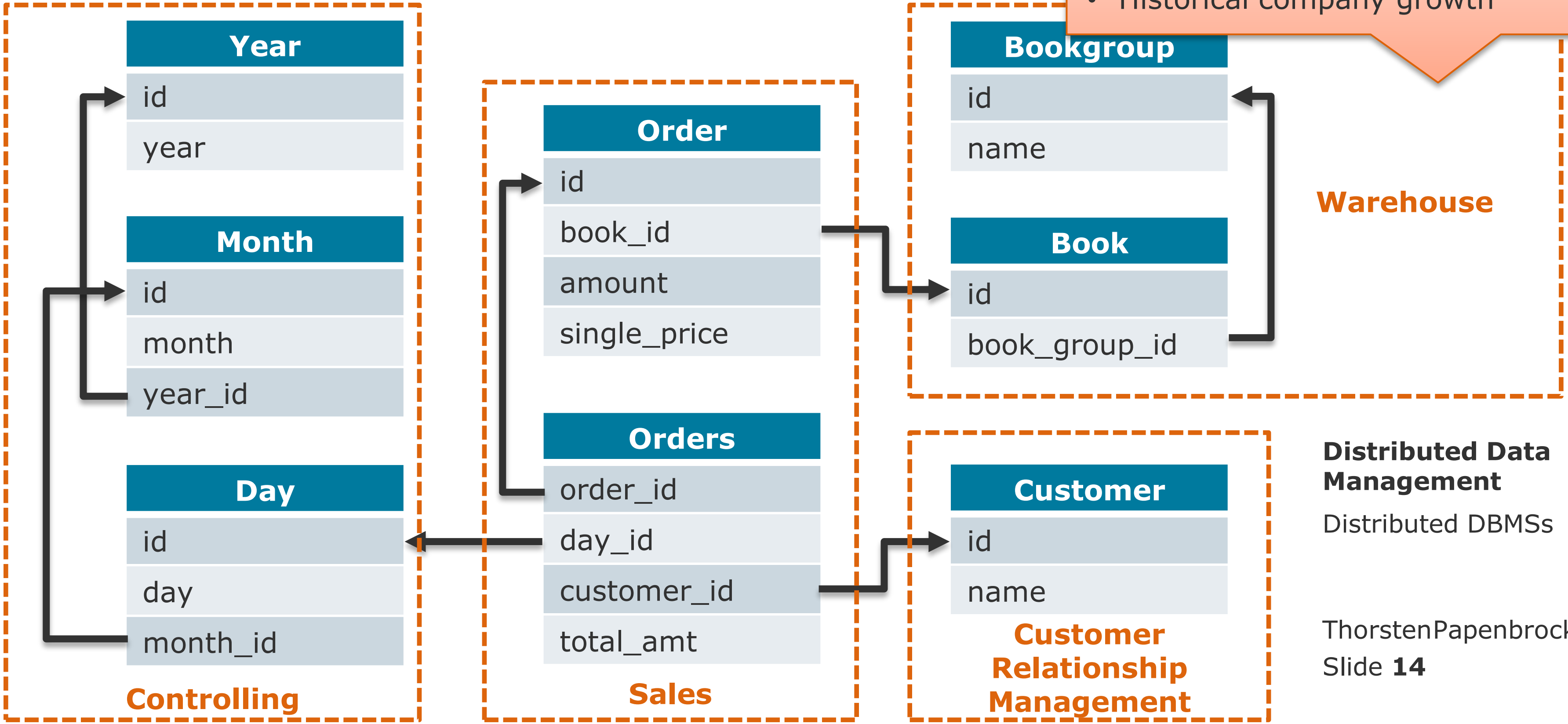
Overview

1. **Distributed DBMSs**
2. Materialized vs. Virtual
3. Data Warehouses
4. Federated Database Management Systems

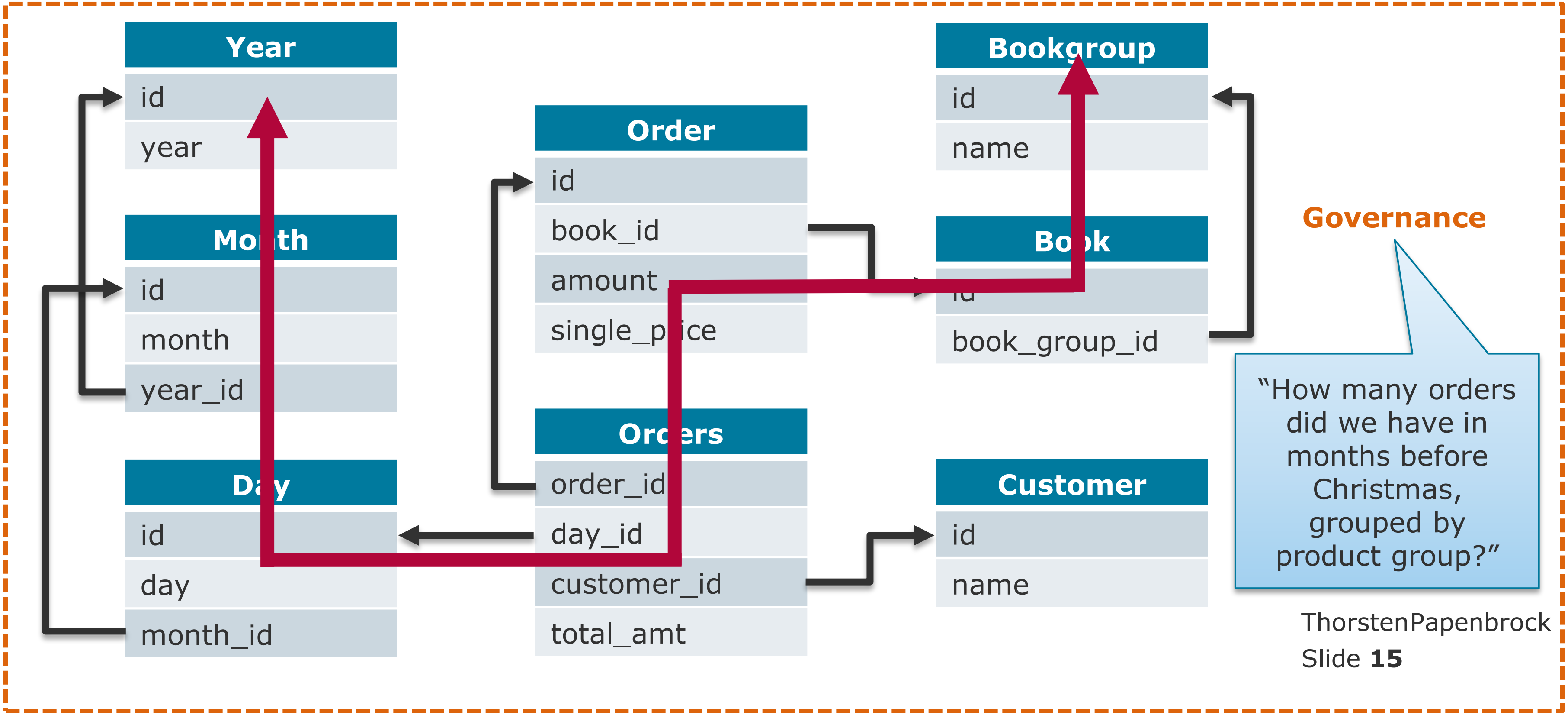
Distributed Data Creation

Why this separation?

- Faster OLTP
- Responsibilities (politics)
- Historical company growth



Distributed Data Creation



Distributed Data Creation

```
SELECT Y.year, BG.name, count(B.id)
FROM   year Y, month M, day D, order O,
       orders OS, book B, bookgroup BG
WHERE  M.year = Y.id
AND    M.id = D.month
AND    O.day_id = D.id
AND    OS.order_id = O.id
AND    B.id = O.book_id
AND    B.book_group_id = BG.id
AND    day < 24 and month = 12
GROUP BY Y.year, BG.name
ORDER BY Y.year
```

Six joins on large tables

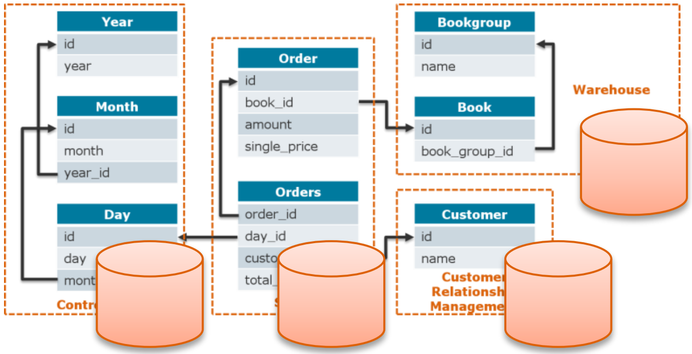
Difficult to optimize; large intermediate tables

Governance

"How many orders did we have in months before Christmas, grouped by product group?"

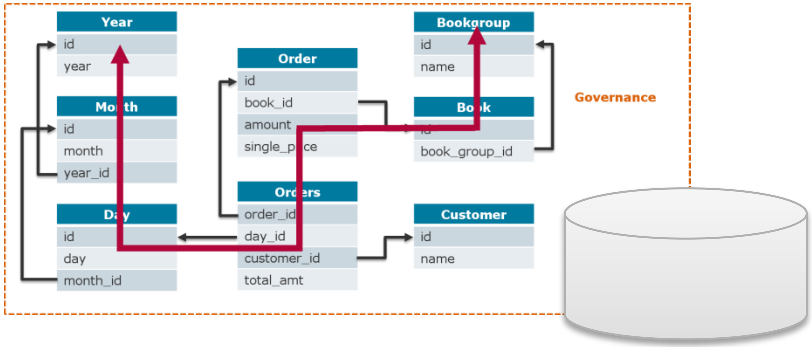
Conflicting Goals

Online Transaction Processing (OLTP)



- Local, isolated databases
- Fast point reads and writes

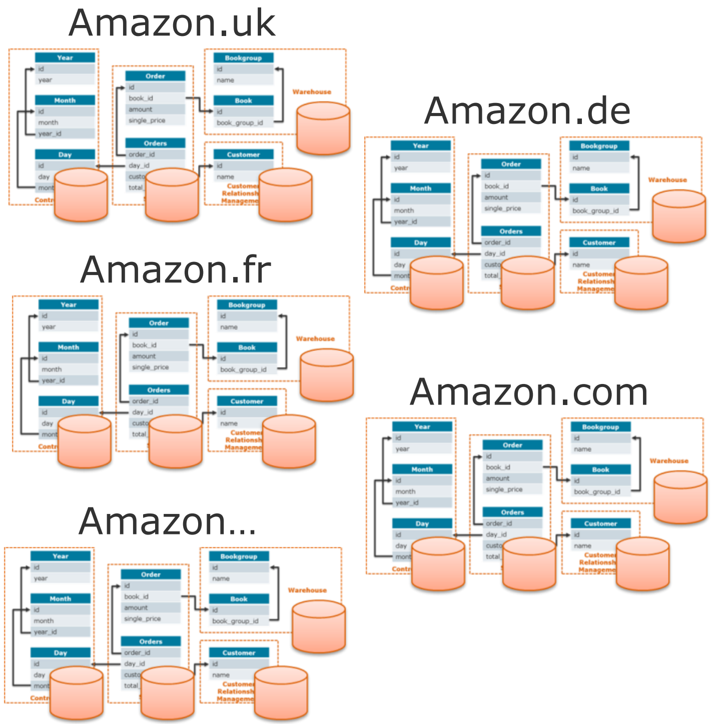
Online Analytics Processing (OLAP)



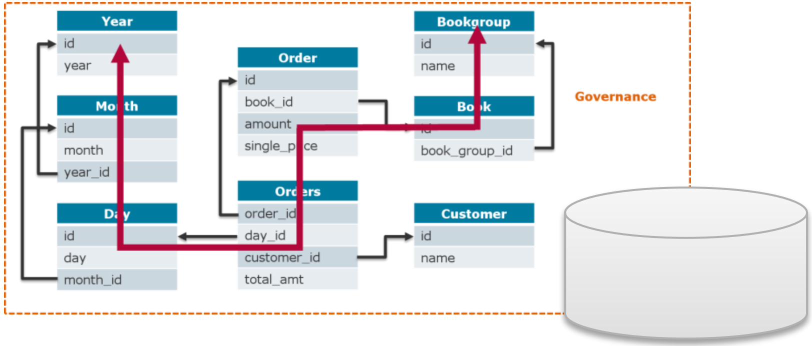
- One integrated database
- Fast aggregations, joins, filters, projections and other complex read operations

Conflicting Goals

Online Transaction Processing (OLTP)



Online Analytics Processing (OLAP)



The reality might be even harder:

Completely independent datasets

OLTP vs. OLAP

Online Transaction Processing (OLTP)

- "Fast processing of operational data, i.e., transactions while maintaining data integrity in multi-access environments"
- Performance characteristic: transactions per second
- Often: time-critical, mixed-workload data with high velocity
- Dominating operations: INSERT, UPDATE, DELETE

Online Analytical Processing (OLAP)

- "Effective answering of analytical queries on already collected data"
  - ➢ Arbitrary, complex, and arbitrarily complex workloads
- Performance characteristic: query response time
- Often: pre-aggregated, multi-dimensional, and historical data
- Dominating operations: SELECT, GROUP, Aggregation

OLTP vs. OLAP

Property	OLTP	OLAP
Main read pattern	Small number of records per query, fetched by key	Aggregate over large number of records
Main write pattern	Random-access, low-latency writes from user input	Bulk import (ETL) or event stream
Primarily used by	End user/customer, via (web) application	Internal analyst, for decision support
What data represents	Latest state of data (current point in time)	History of events that happened over time
Data size	Kilobytes to Gigabytes	Terabytes to petabytes

Distributed DBMSs

OLTP vs. OLAP

	OLTP System Online Transaction Processing (Operational System)	OLAP System Online Analytical Processing (Data Warehouse)
Source of data	Operational data; OLTPs are the original source of the data.	Consolidation data; OLAP data comes from the various OLTP Databases
Purpose of data	To control and run fundamental business tasks	To help with planning, problem solving, and decision support
What the data	Reveals a snapshot of ongoing business processes	Multi-dimensional views of various kinds of business activities
Inserts and Updates	Short and fast inserts and updates initiated by end users	Periodic long-running batch jobs refresh the data
Queries	Relatively standardized and simple queries Returning relatively few records	Often complex queries involving aggregations
Processing Speed	Typically very fast	Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes
Space Requirements	Can be relatively small if historical data is archived	Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP
Database Design	Highly normalized with many tables	Typically de-normalized with fewer tables; use of star and/or snowflake schemas
Backup and Recovery	Backup religiously; operational data is critical to run the business, data loss is likely to entail significant monetary loss and legal liability	Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method

source: www.rainmakerworks.com

What is "Analytics"?

"Calculate the number of sales per year and state."

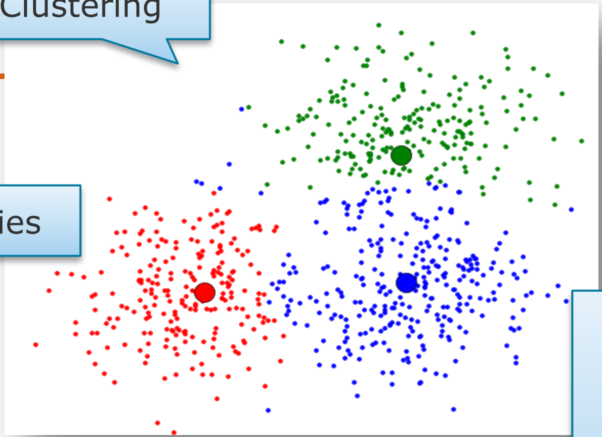
SELECT SUM(S.sales)

FROM Sales S, Times T, Locations L

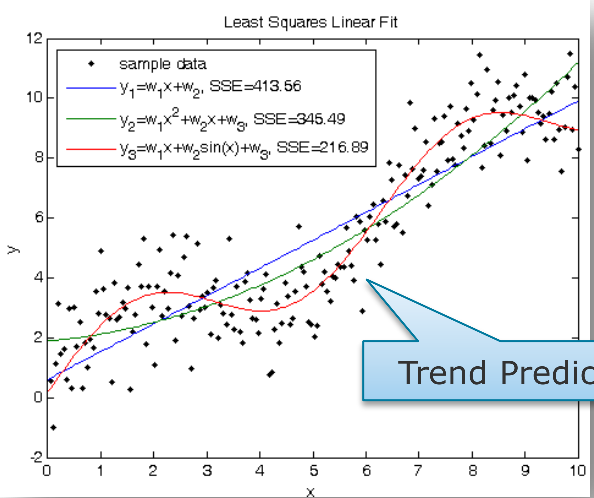
WHERE S.timeid = T.timeid AND S.timeid = L.timeid

Aggregate Queries

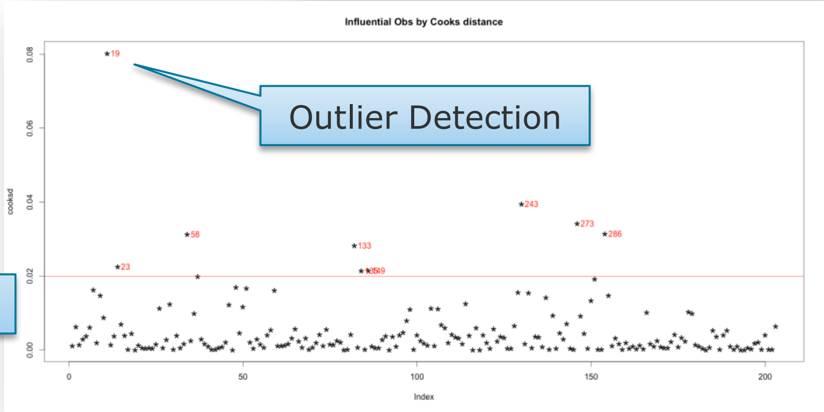
Clustering



And many further questions to datasets!



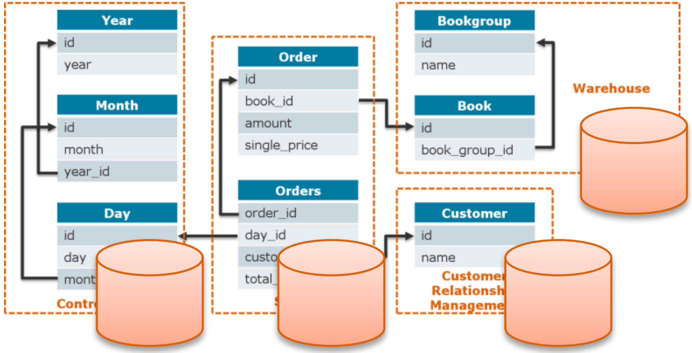
Trend Prediction



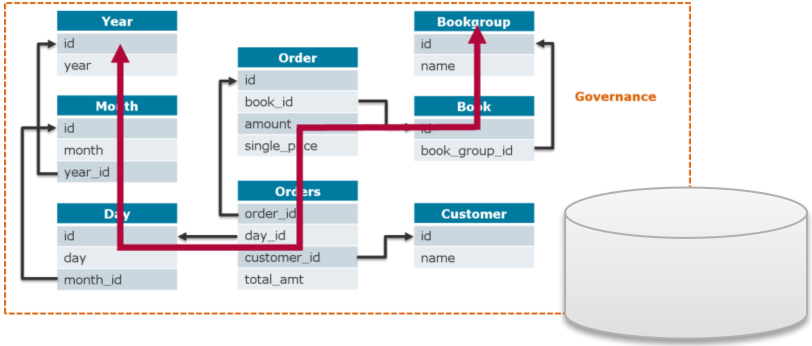
Outlier Detection

Conflicting Goals

Online Transaction Processing (OLTP)

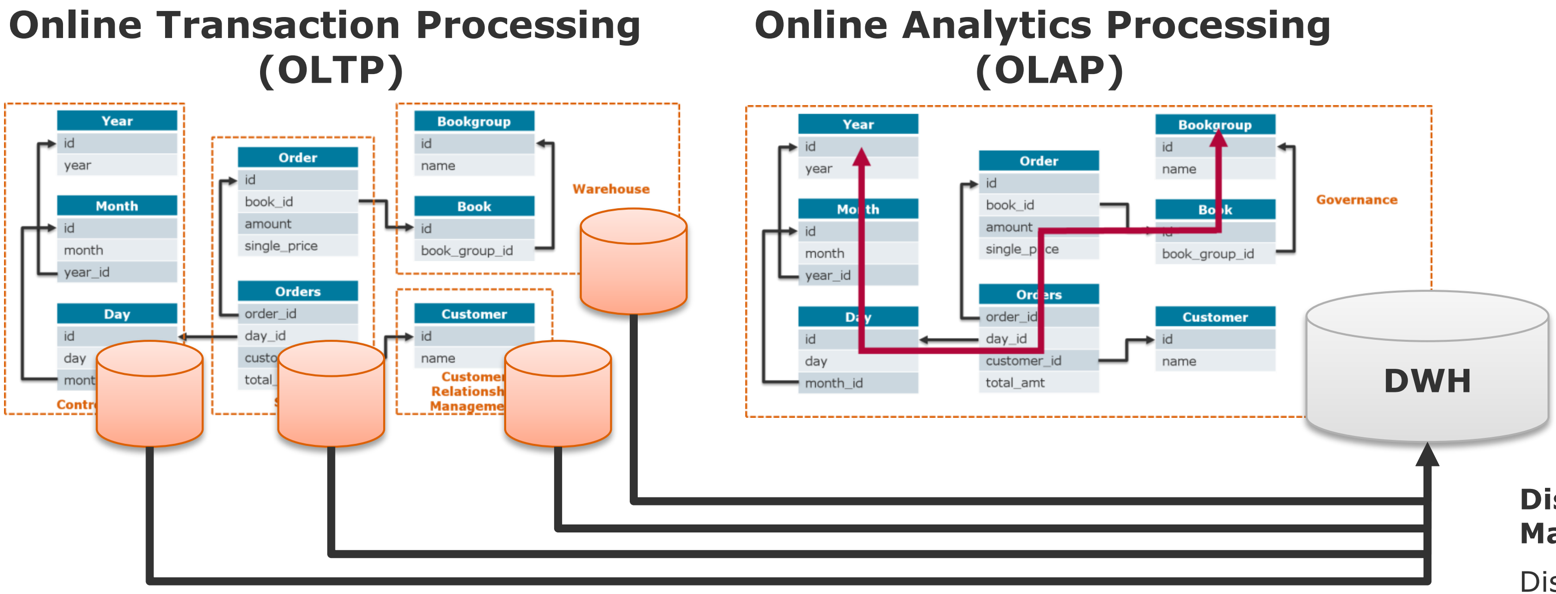


Online Analytics Processing (OLAP)



So since OLTP and OLAP workloads do exist, how do we solve this conflict?

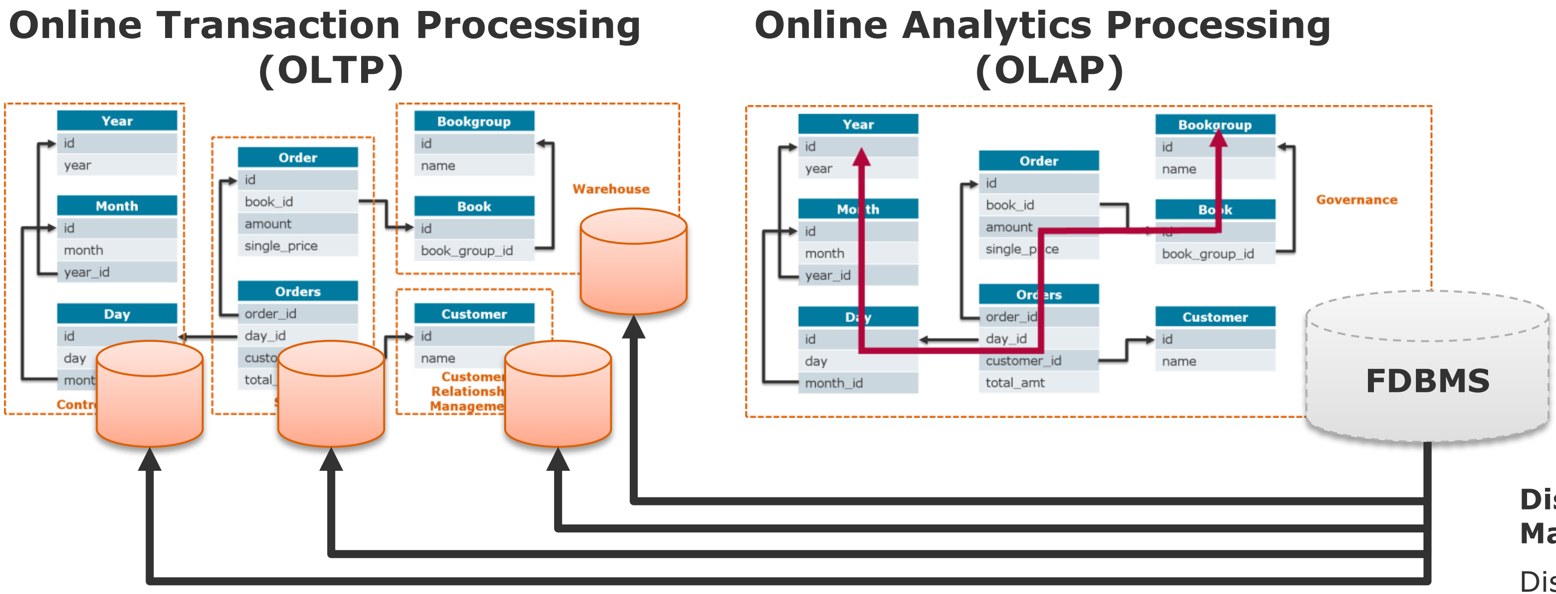
Solution A: Materialized Integration



Frequently copy data from OLTP systems over to OLAP systems.

➢ Data Warehouse (DWH)

Solution B: Virtual Integration



The OLAP systems defines analytical views that fetch the data on demand.

➢ Federated Database Management Systems (FDBMS)

Distributed DBMSs

Overview

1. Distributed DBMSs
2. **Materialized vs. Virtual**
3. Data Warehouses
4. Federated Database Management Systems

Materialized vs. Virtual

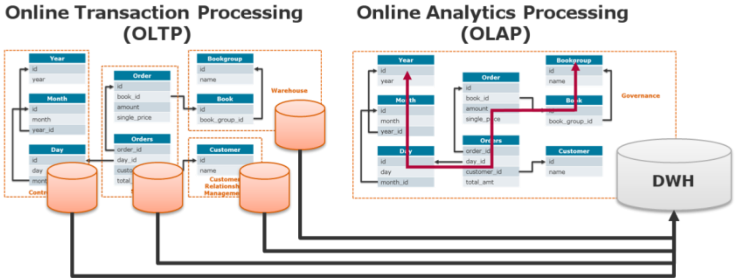
Two Flavors of Integration

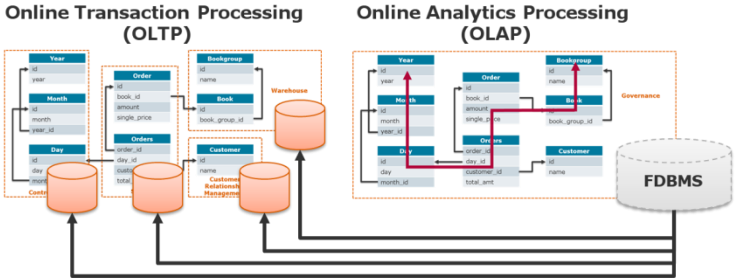
Materialized

- A-priori integration
- Centralized data store
- Centralized query processing
- Typical example:
  data warehouse

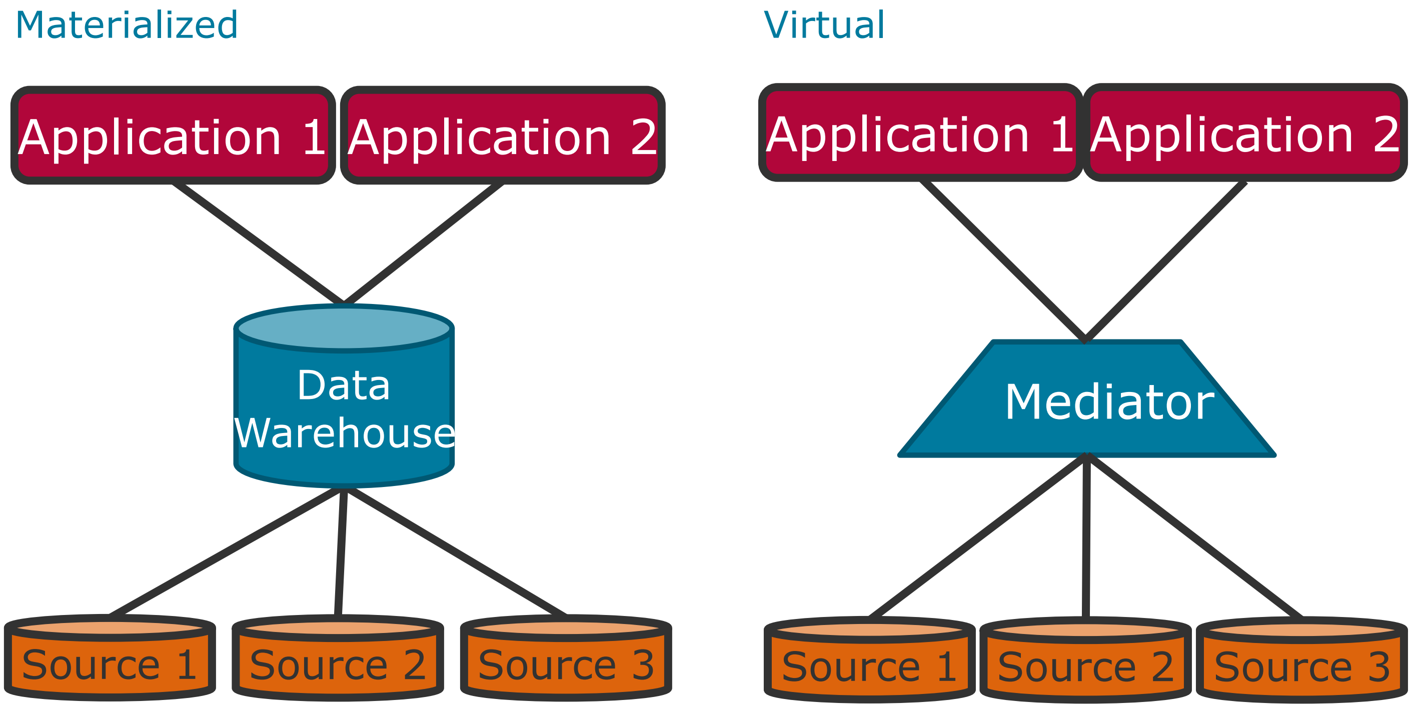
Virtual

- On-demand integration
- Decentralized data
- Decentralized query processing
- Typical example:
  mediator-based information system





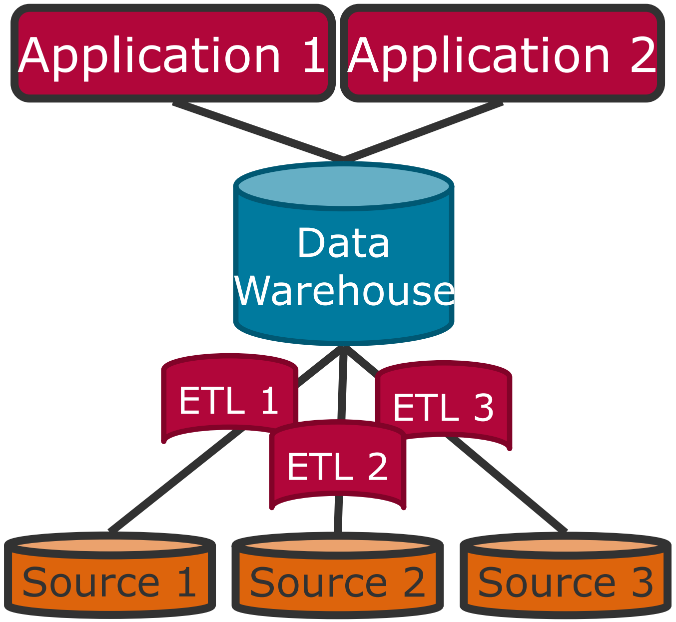
Two Flavors of Integration



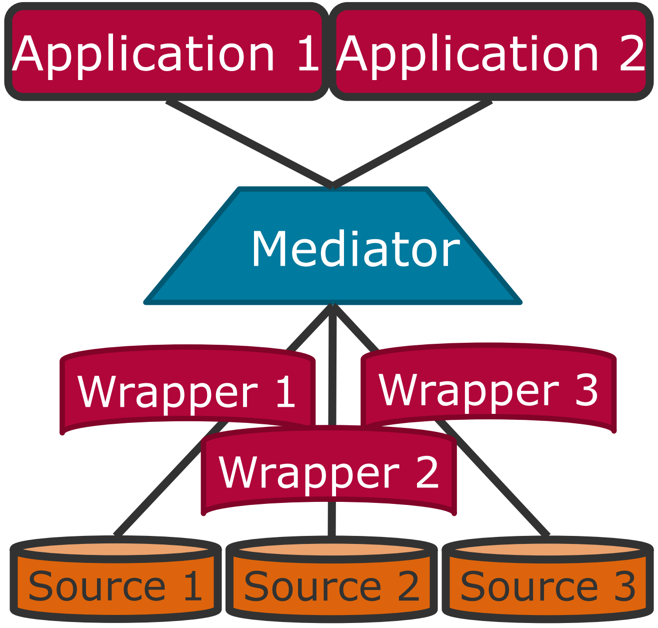
Materialized vs. Virtual

Two Flavors of Integration

Materialized

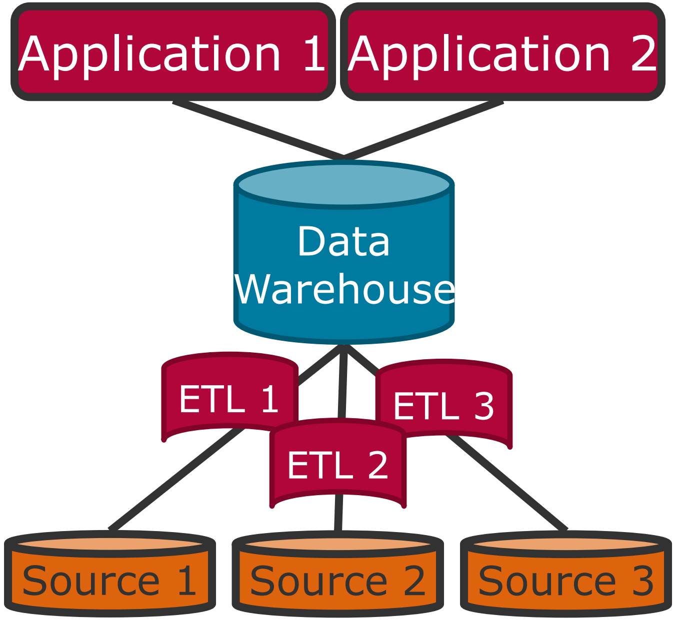


Virtual



Two Flavors of Integration

Materialized

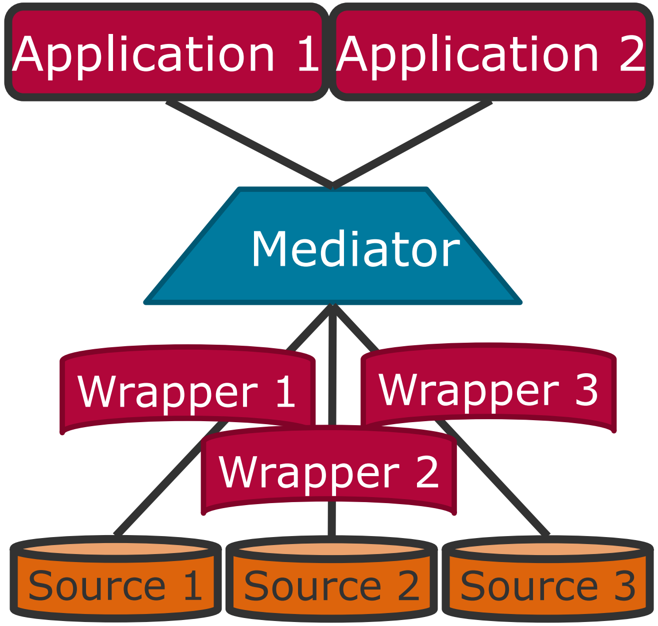


- Push-model for Data
  - Periodically import data
  - Redundant storage
  - Materialized (aggregate) views
- Schema design
  - Bottom up
  - Schema integration
- Query processing as usual
  - OLAP queries
  - Star schema

Two Flavors of Integration

Virtual

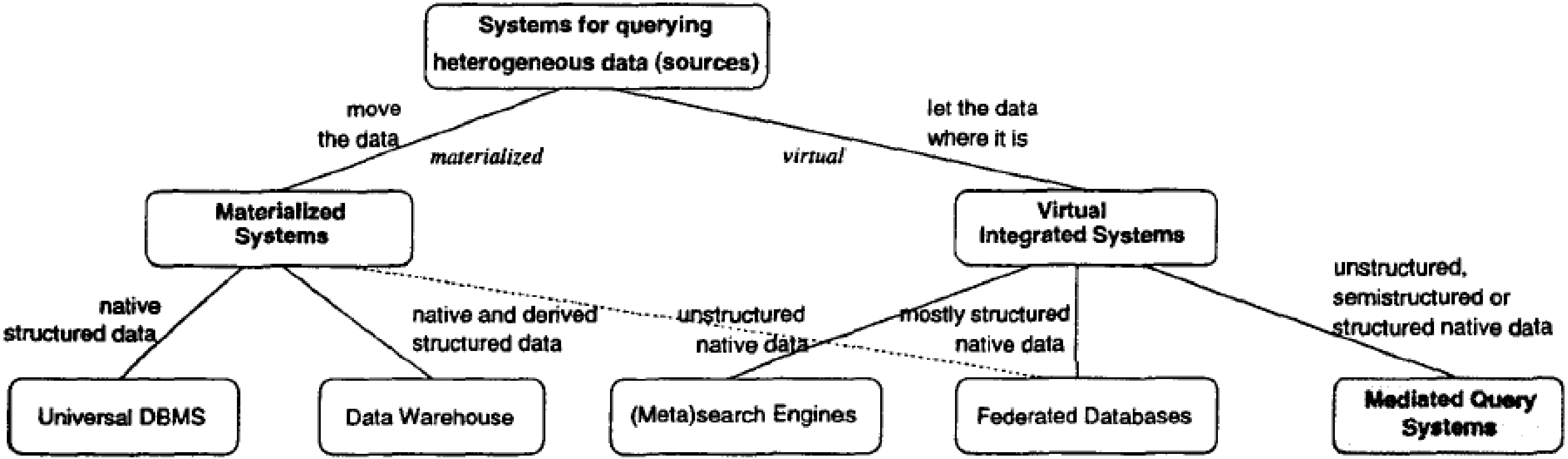
- Pull-model for data
  - Only transfer data for query at hand
- Schema design
  - Top down
  - Schema mapping
- Query processing
  - Difficult optimization
  - Heterogeneous costs and abilities of sources



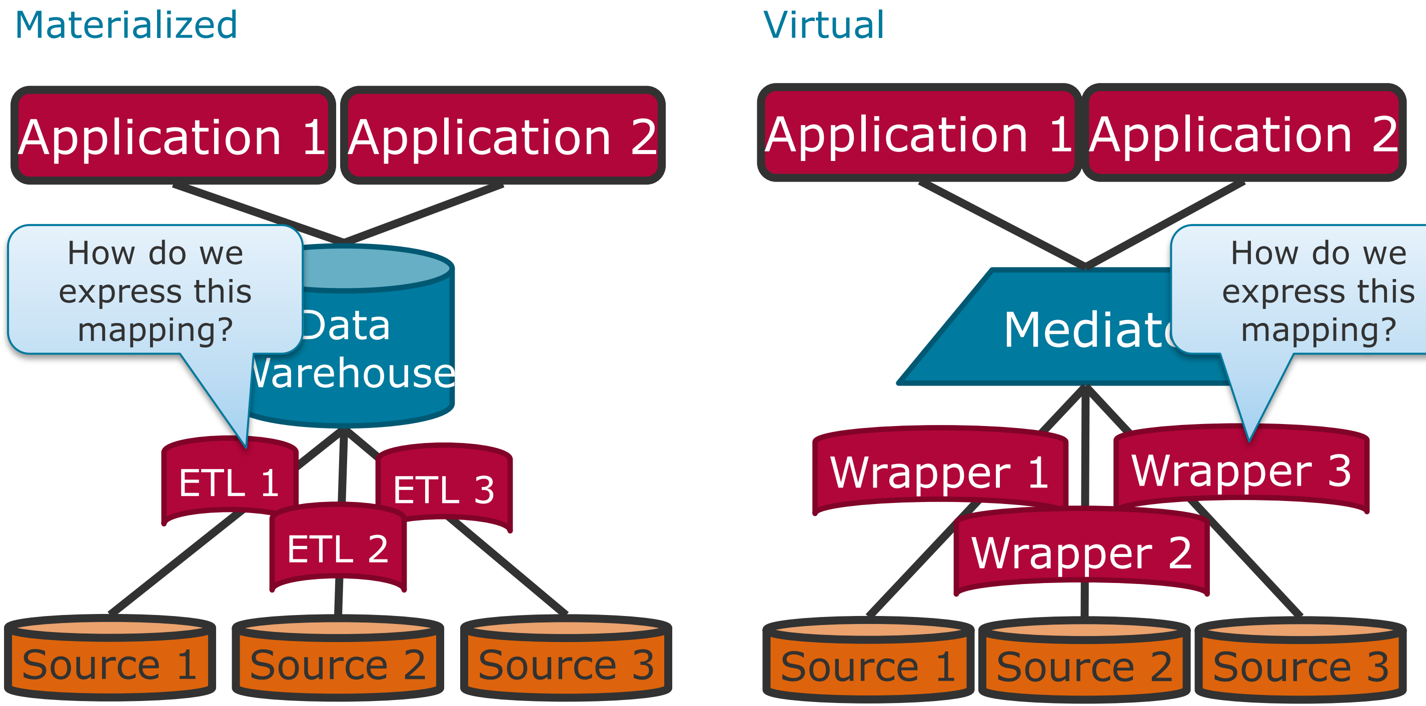
Materialized vs. Virtual
Taxonomy

An Overview and Classification of Mediated Query Systems

Ruxandra Domenig*, Klaus R. Dittrich
Department of Information Technology, University of Zurich
{domenig|dittrich}@ifi.unizh.ch



Two Flavors of Integration

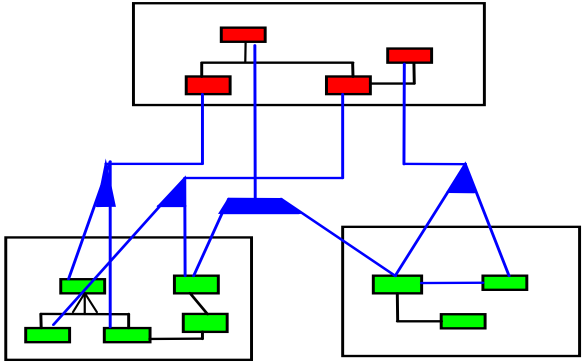


Materialized vs. Virtual

Global as View / Local as View

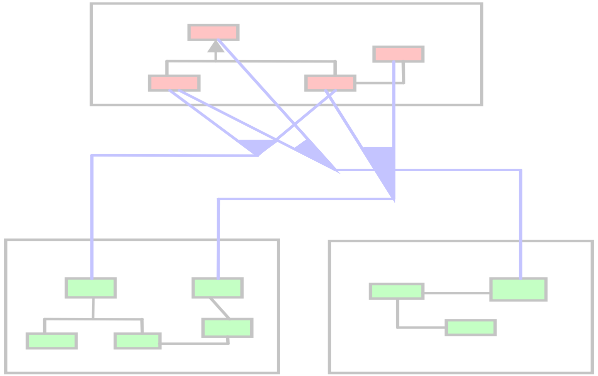
Global as View (GaV)

- Relations of the global integrated schema are expressed as views on the local source schemata.
- Idea: Views tell where the global relations get their data from.

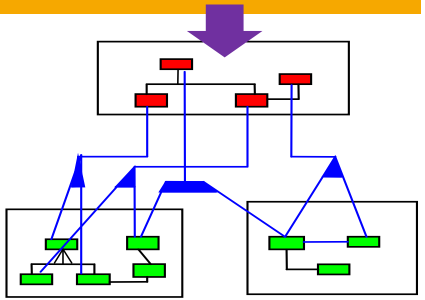


Local as View (LaV)

- Relations of the local source schemata are expressed as views on the global integrated schema.
- Idea: Views tell what parts of the global relations can be found in each local relation.



GaV Query Processing



Given:

- A query against the global integrated schema (in particular: against relations of the global schema)
- For each global relation, a view on local relation(s)

Find:

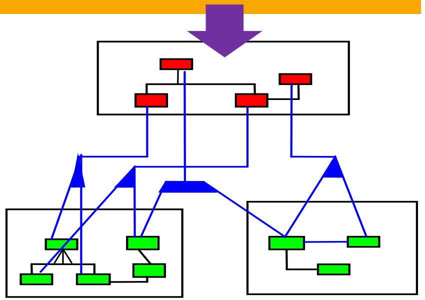
- All valid tuples for the query
- But: Data is stored in local sources.

Idea:

- Replace each relation in the query by the corresponding view definition: „view expansion" or „query unfolding"
- Result: A nested query

Materialized vs. Virtual

GaV Query Processing Example



```
SELECT F.Title, P.Cinema
FROM   Film F, Program P
WHERE  F.Title = P.Title
AND    P.Time > 20:00
```

```
S1: IMDB(Title, Director, Year, Genre)
S2: RegieDB(Title, Director)
S3: GenreDB(Title, Year, Genre)
S7: KinoDB(Title, Cinema, Genre, Time)
```

```
SELECT F.Title, P.Cinema
FROM (     SELECT * FROM IMDB
                      UNION
           SELECT R.Title, R.Director, G.Year, G.Genre
           FROM RegieDB R, GenreDB G
           WHERE R.Title = G.Title
       ) AS F,
       (   SELECT *
           FROM KinoDB
       ) AS P
WHERE F.Title = P.Title
AND      P.Time > 20:00
```

= view for Film

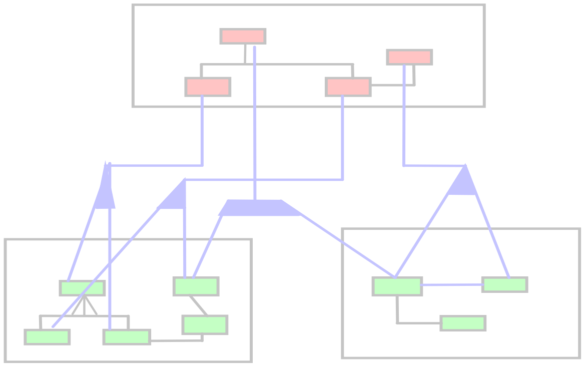
= view for Program

See „Information Integration“ lecture by Prof. Naumann on how to come up with such views.

Global as View / Local as View

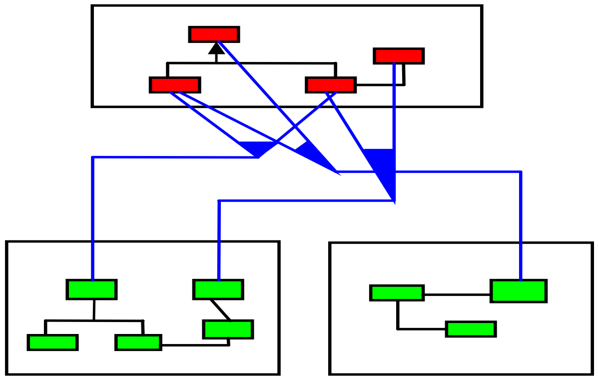
Global as View (GaV)

- Relations of the global integrated schema are expressed as views on the local source schemata.
- Idea: Views tell where the global relations get their data from.



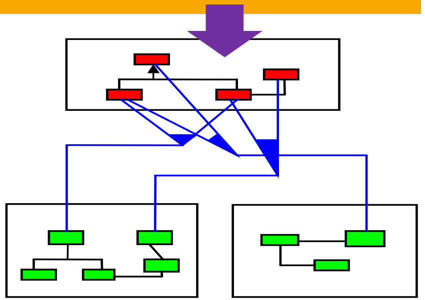
Local as View (LaV)

- Relations of the local source schemata are expressed as views on the global integrated schema.
- Idea: Views tell what parts of the global relations can be found in each local relation.



Materialized vs. Virtual

LaV Query Processing



Given:

- A query against the global integrated schema (in particular: against relations of the global schema)
- For each local relation, a view on global relation(s)

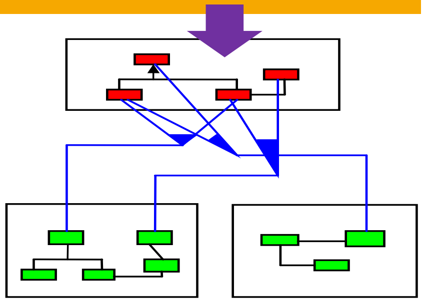
Find:

- All valid tuples for the query
- But: Data is stored in local sources.

Idea:

- Run the query against all local relations whose views can contribute to the result of the query; join/union all local results.

LaV Query Processing (formal)



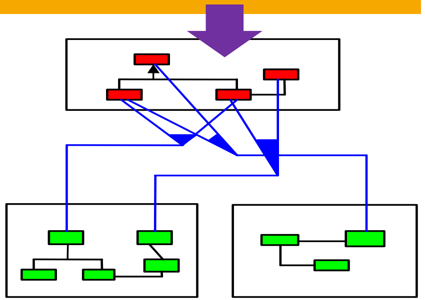
Given:

- A query q against the global integrated schema (in particular: against relations of the global schema)
- For each local relation, a view on global relation(s)

Find:

- Sequence of queries $q_1 \diamond \ldots \diamond q_n$
- Each q_i can be executed by a single view
- Suitable combination of queries $q_1, \ldots, q_n$ answers q
- Within a plan, use joins: $\diamond \rightarrow \bowtie$
- Multiple plans are combined by UNION : $\diamond \rightarrow \cup$
- Tuples created by $q_1 \bowtie \ldots \bowtie q_k$ are valid result tuples for q.

LaV Query Processing Example



Global schema

Lehrt(prof,kurs_id, sem, eval, univ)
Kurs(kurs_id, titel, univ)

Source 1: All database courses

```
CREATE VIEW DB-kurs AS
SELECT  K.titel, L.prof, K.kurs_id, K.univ
FROM    Lehrt L, Kurs K
WHERE   L.kurs_id = K.kurs_id
AND     L.univ = K.univ
AND     K.titel LIKE „%_Datenbanken“
```

Source 2: All HPI lectures

```
CREATE VIEW HPI-VL AS
SELECT K.titel, L.prof, K.kurs_id, K.univ
FROM    Lehrt L, Kurs K
WHERE  L.kurs_id = K.kurs_id
AND     K.univ = „HPI“
AND     L.univ =  „HPI“
AND     K.titel LIKE „%VL_%“
```

Express the content of each local relation as a view on the global schema.

Global query

```
SELECT titel, kurs_id
FROM Kurs K
WHERE K.univ = „HPI“
```

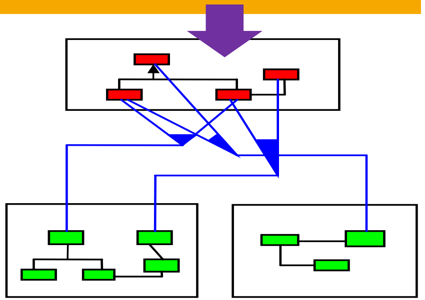
Rewrite query using all views on Kurs

Rewritten query

```
(SELECT titel, kurs_id
 FROM DB-kurs D
 WHERE D.univ = „HPI“)
            UNION
(SELECT titel, kurs_id
 FROM HPI-VL)
```

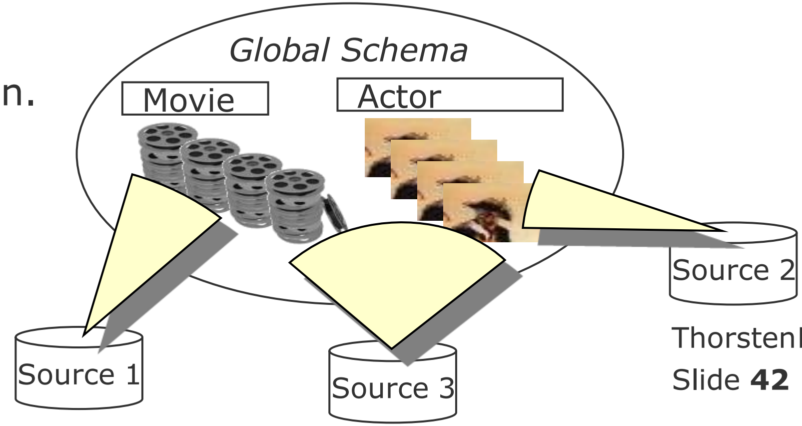
See „Information Integration“ lecture by Prof. Naumann for more details on the rewriting algorithm.

Why LaV 1: Data Integration



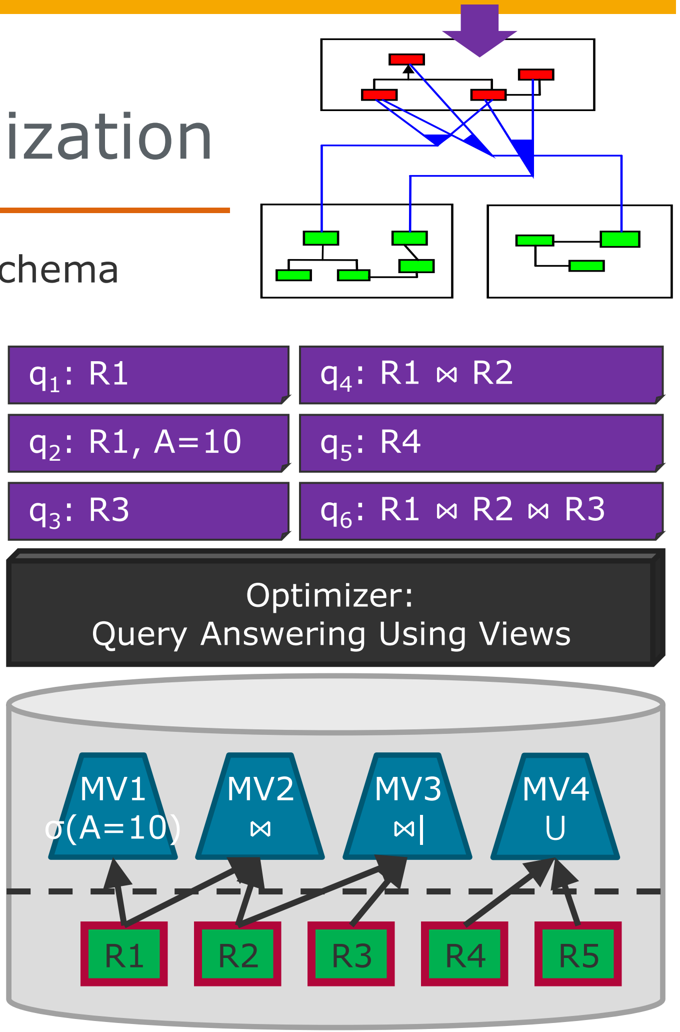
- The global schema models the world (e.g. the entire domain of movies).
- In theory, this establishes the extension, i.e., the content of the database.
- In practice, there exist many databases on movies, actors etc. But nobody knows (or has) this extension.
  - ➢ Information integration tries to collect whatever is available.

- Every source stores a part of the extension.
- Every source describes its content as views on the global schema.
  - ➢ Because the source provides the schema, it is easy to add more sources over time.

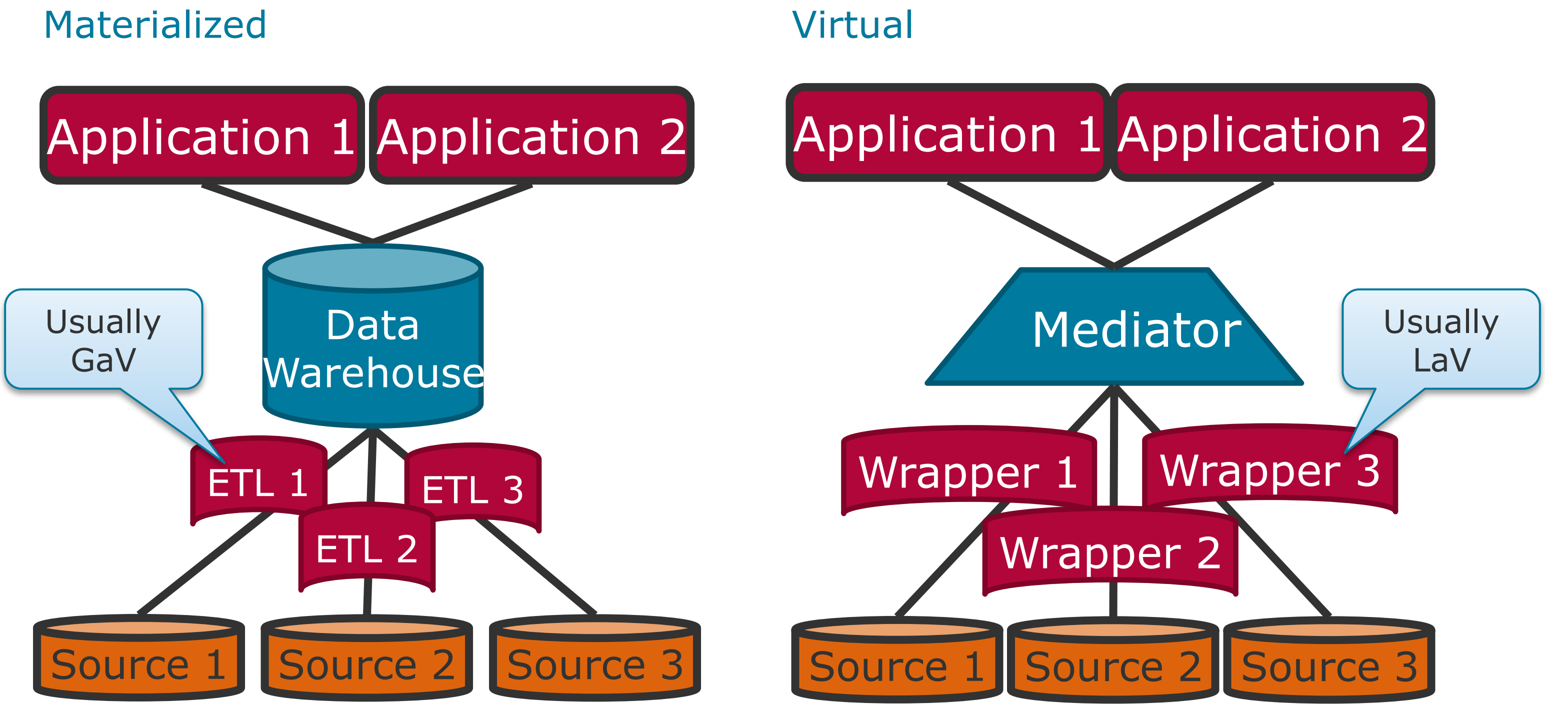


Why LaV 2: Query Optimization

- Materialized views (MVs) on database schema
  - Aka. Materialized Query Table
  - Aka. Advanced Summary Table
- Which MVs (and their pre-calculated intermediate results) can help in determining a query result?
- Challenges:
  - It is not always better to use an MV over e.g. indixes.
  - MVs need to be kept up-to-date.



Back to Materialized vs. Virtual Integration



Back to Materialized vs. Virtual Integration

	Materialized	Virtual
Up-to-date	-	+
Response time	+	-
Flexibility	- (usually GaV)	+ (usually LaV)
Query processing complexity	-	--
Source-autonomy	-	+
Query capabilities	+	-
Read/Write	+/+	+/-
Storage requirement	-	+
Completeness	+	? (OWA, CWA)
Data cleansing	+	-
Information quality	+	-

Distributed DBMSs

Overview

1. Distributed DBMSs

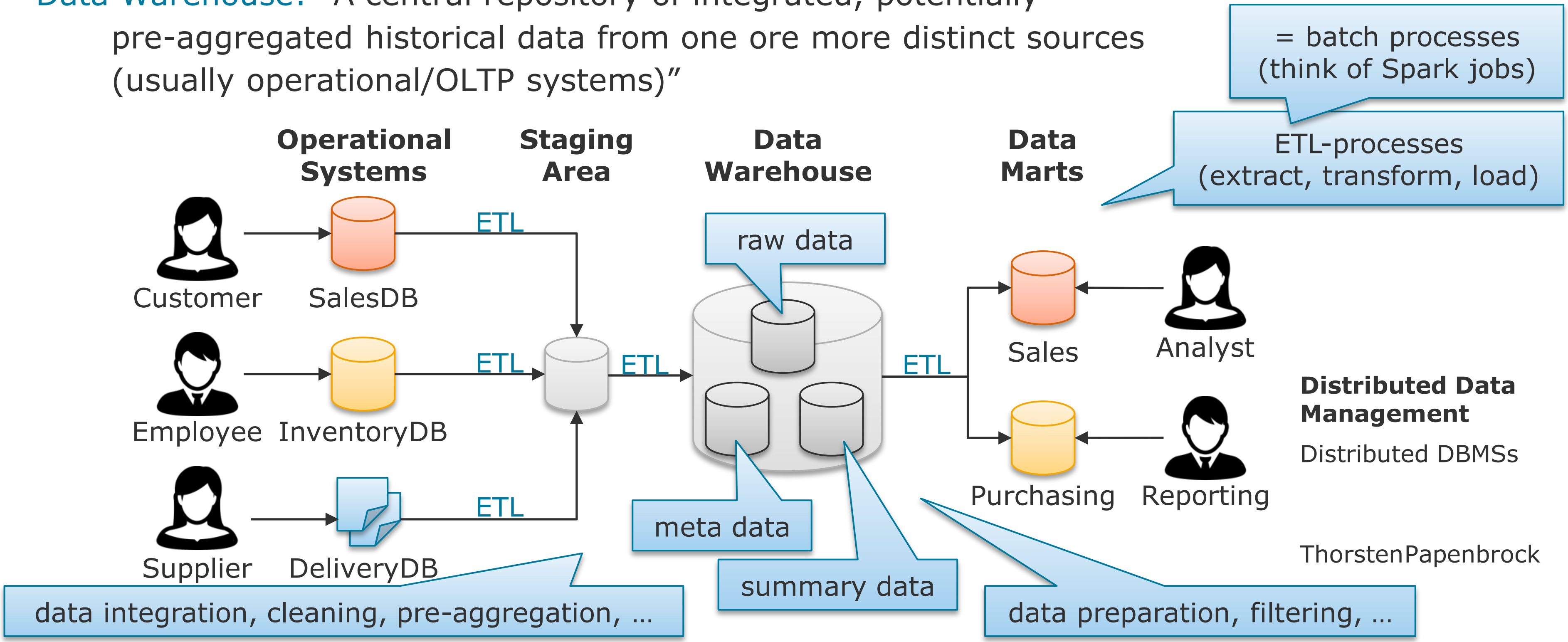
2. Materialized vs. Virtual

3. **Data Warehouses**

4. Federated Database Management Systems

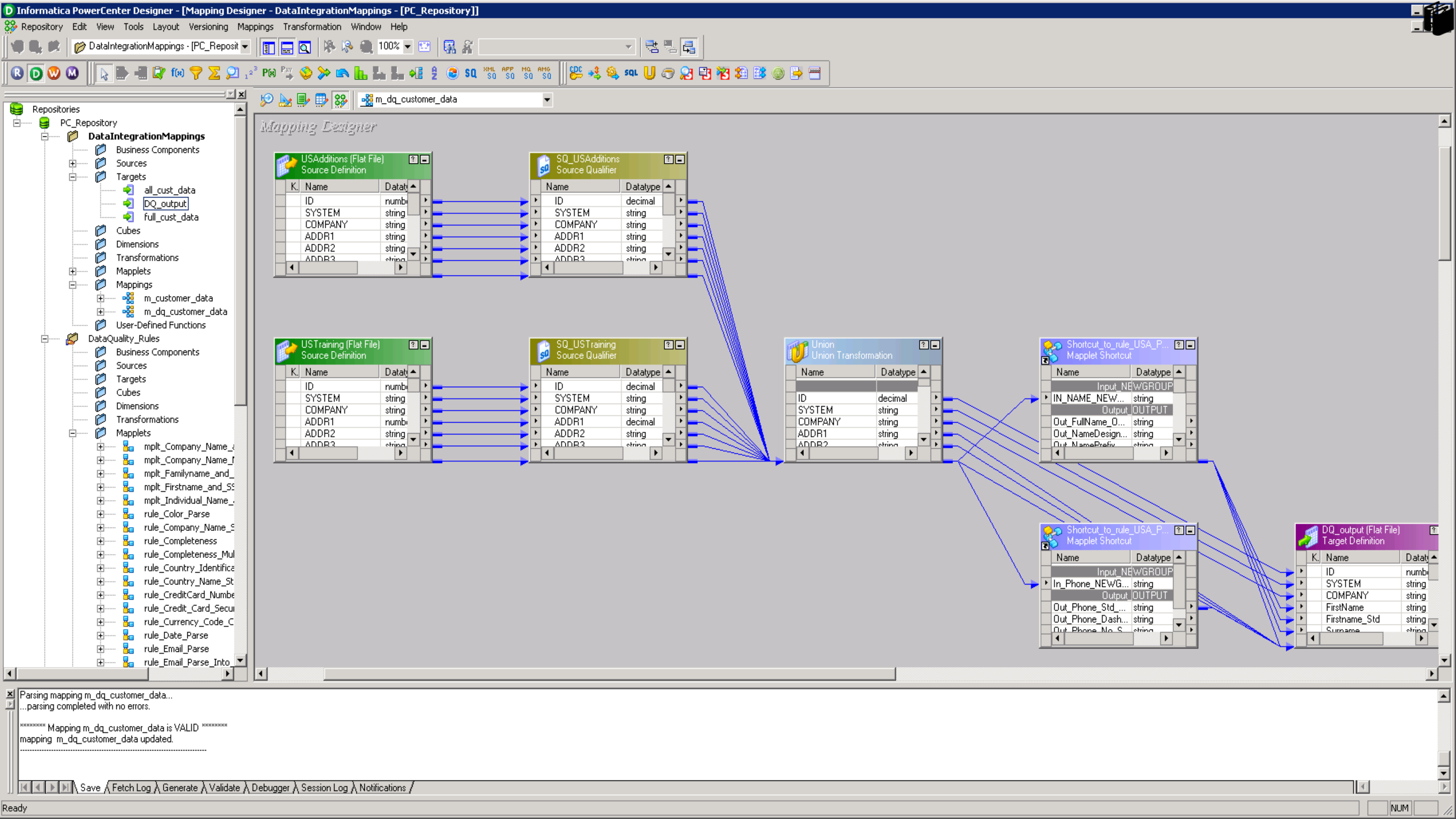
Architecture

Data Warehouse: "A central repository of integrated, potentially pre-aggregated historical data from one ore more distinct sources (usually operational/OLTP systems)"



Extract-Transform-Load (ETL)

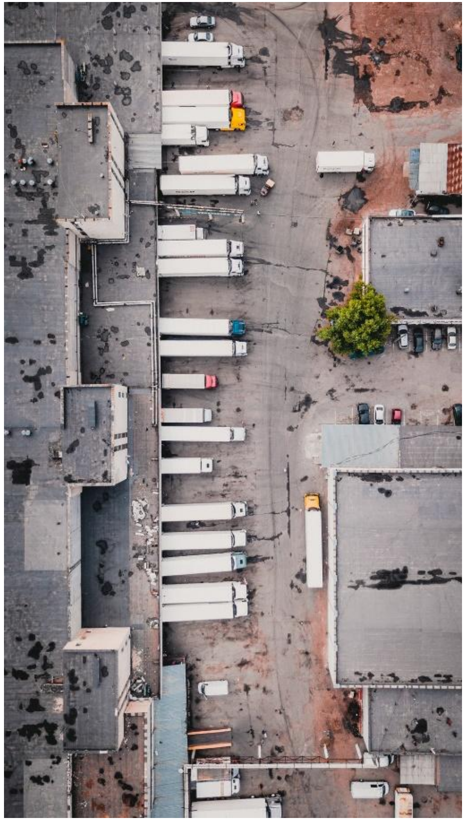
- ETL processes take data from a source, convert it into data analytics-friendly representations and sink the result into a data warehouse.
- ETL processes are …
  - batch processes comparable to modern Spark-jobs.
  - the equivalent to schema mappings in virtual integration.
  - functional/procedural implementations of the views in the GaV model.
  - more powerful than simple views, i.e., can express more complex logic (data cleaning, data encoding, side effects, machine learning etc.).
- ETL processes offer …
  - import filters (read and convert data from sources)
  - standard transformations (join, aggregate, filter, convert, …)
  - de-duplication (find and merge multiple records referring to the same entity)
  - aggregations (simple aggregates, sketches, histograms, …)
  - quality management (test against master data, business rules, constraints, …)



Informatica PowerCenter Designer - [Mapping Designer - DataIntegrationMappings - [PC_Repository]]
Repository Edit View Tools Layout Versioning Mappings Transformation Window Help
DataIntegrationMappings - [PC_Reposit
100%
m_dq_customer_data
Repositories
PC_Repository
DataIntegrationMappings
Business Components
Sources
Targets
all_cust_data
DQ_output
full_cust_data
Cubes
Dimensions
Transformations
Mapplets
Mappings
m_customer_data
m_dq_customer_data
User-Defined Functions
DataQuality_Rules
Business Components
Sources
Targets
Cubes
Dimensions
Transformations
Mapplets
mplt_Company_Name_
mplt_Company_Name_
mplt_Familyname_and_
mplt_Firstname_and_S
mplt_Individual_Name_
rule_Color_Parse
rule_Company_Name_S
rule_Completeness
rule_Completeness_Mul
rule_Country_Identific
rule_Country_Name_St
rule_CreditCard_Numbe
rule_Credit_Card_Secu
rule_Currency_Code_C
rule_Date_Parse
rule_Email_Parse
rule_Email_Parse_Into
Mapping Designer
USAdditions (Flat File)
Source Definition
SQ_USAdditions
Source Qualifier
USTraining (Flat File)
Source Definition
SQ_USTraining
Source Qualifier
Union
Union Transformation
Shortcut_to_rule_USA_P...
Mapplet Shortcut
DQ_output (Flat File)
Target Definition
Name
Datatype
ID
SYSTEM
COMPANY
ADDR1
ADDR2
ADDR3
decimal
string
Input_NEWGROUP
IN_NAME_NEW...
Output OUTPUT
Out_FullName_O...
Out_NameDesign...
Out_NamePrefix
In_Phone_NEWG...
Out_Phone_Std_...
Out_Phone_Dash...
Out_Phone_No_S
FirstName
Firstname_Std
Surname
Parsing mapping m_dq_customer_data...
...parsing completed with no errors.
******** Mapping m_dq_customer_data is VALID ********
mapping m_dq_customer_data updated.
Save
Fetch Log
Generate
Validate
Debugger
Session Log
Notifications
Ready
NUM

Application Areas

- Customer Relationship Management (CRM)
  - Premium customer identification
  - Personalization
  - Mass-marketing
- Controlling / Accounting
  - Cost center discovery
  - Organizational units analysis
  - Human resources management
- Logistics
  - Fleetmanagement and -tracking
- Digital health
  - Experimental studies
  - Patient monitoring



Popular DWH DBMSs

Commercial

- Microsoft SQL Server
- SAP HANA
- Teradata
- Vertica
- ParAccel
- …

Open-Source

- Apache Hive
- Spark SQL
- Cloudera Impala
- Facebook Presto
- Apache Tajo
- Apache Drill
- …

Most of these are **Hadoop**-based and use some kind of **MapReduce** paradigm.

The Pros and Cons

Advantages

- Compute-intense analytical queries do not interfere with operational business.
- Data is static and does not change while queries are run.
- Data can be sorted by certain keys that are often queried as range or group (e.g. timestamps, version-IDs, tags, or country codes), whereas operational data is usually not sorted for better insert performance.
- Data can be compressed more aggressively to improve read performance.
- Analytics-friendly data layouts, e.g., star-schemata, data cubes, or materialized views as well as indexes for analytical query patters are possible.

Disadvantages

- Data is not up-to-date (one ETL-cycle ≈ one day old)

Stars and Snowflakes

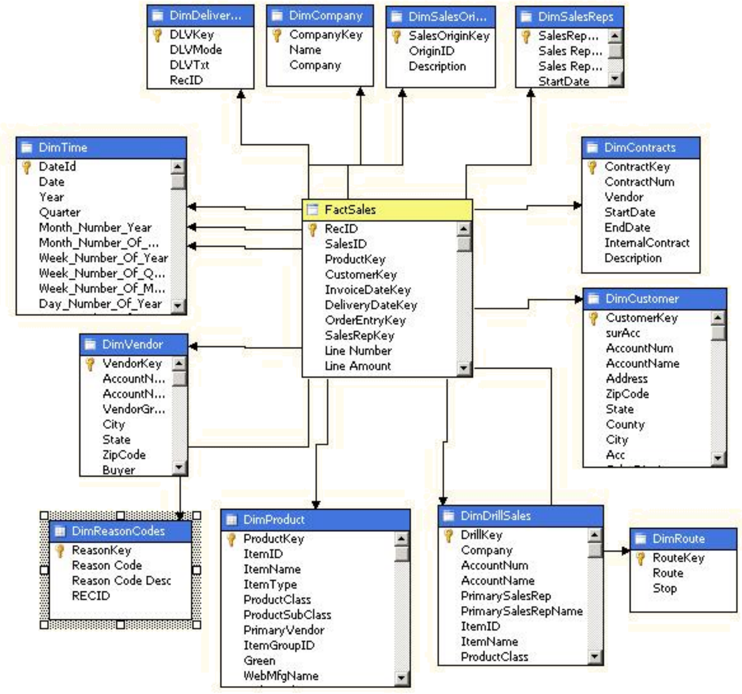
Star Schema

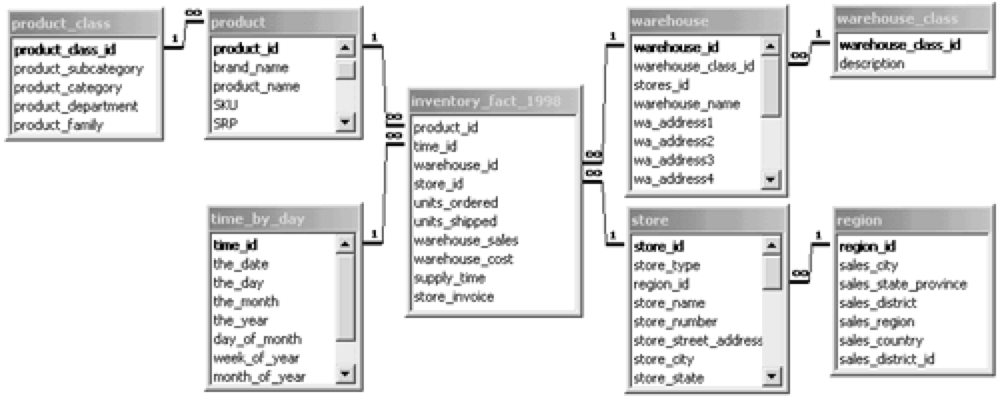
- An acyclic graph of relational tables with depth 1
  - Root = "fact table"
  - Leaves = "dimension tables"
- Fact tables: contain events (transactions, measurements, snapshots,…)
  - Usually very large tables (long and wide)
  - Examples: sales, page views, clicks, shippings, sensor readings, …
- Dimension tables: contain entity data and descriptive information
  - Usually small tables with fixed domain
  - Examples: products, employees, customers, dates, locations, …
- Answer for each event: who, what, where, when, how, or why

Snowflake Schema

- Same as star schema, but with arbitrary depth, i.e., dimension tables might have further dimension tables

Stars and Snowflakes – Examples





Stars and Snowflakes – Benefits

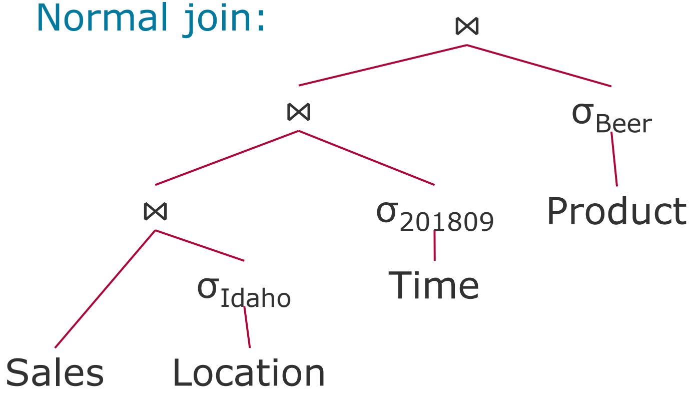
- Improved query performance for (most) aggregation queries:
  - Better join-performance than normalized data
  - Better scan performance than one big table
  - May contain pre-aggregated data
- Simpler queries:
  - Clear join logic and manageable number of joins
- Redundancy reduction via data integration:
  - Redundant information in different sources is consolidated into same tables
  - Redundant information in one source table might get normalized into one dimension table

Stars and Snowflakes – Star Join

```
SELECT *
FROM   Sales S, Location L, Time T, Product P
WHERE  S.L_ID = L.ID AND S.T_ID = T.ID AND S.P_ID = P.ID
AND    L.state = ‚Idaho‘
AND    Year_Month(T.Date) = 201809
AND    P.Category = ‚Beer‘
```

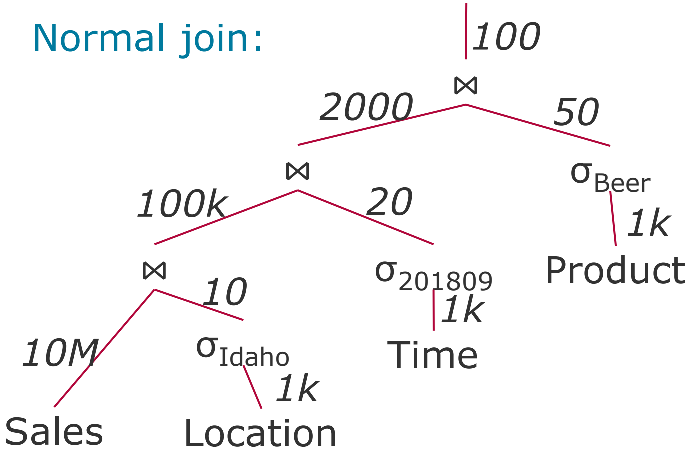
- Some sizes
  - Sales: 10,000,000
  - Locations: 1,000
    - 10 in Idaho
  - Times: 1,000 days
    - 20 in September 2018
  - Products: 1,000
    - 50 Beers

Normal join:

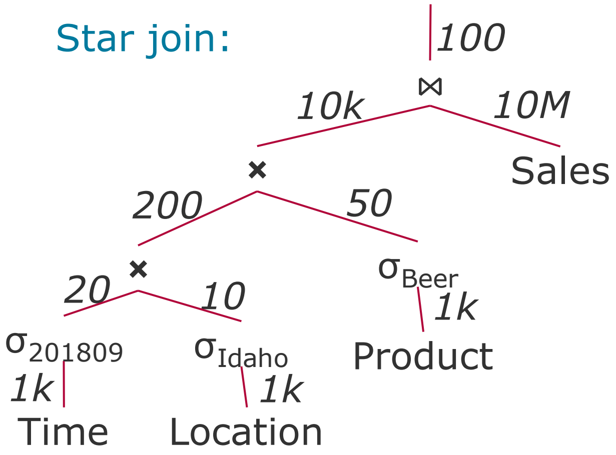


Stars and Snowflakes – Star Join

Normal join:



Star join:



- Intermediate result sizes
  - Normal joins: 100k + 2000
  - Star join: 200 + 10k

Pre-Aggregation

Observation

- Most analytical queries in data warehouses are aggregate queries.
- Aggregation patterns repeat frequently.
  - ➢ Pre-calculate common aggregates!

Materialized Views

- Are query results that are written back to disk.
- DBMS query optimizers can automatically use these views to answer queries or parts of queries (see LaV).
- Strategies to improve data analytics:
  - Pre-calculation: estimate common aggregates and pre-calculate them
  - Lazy pre-calculation: store each aggregate once it was queried

Pre-Aggregation – Data Cubes

Data Cube

- A set of materialized views for multi-dimensional aggregates (i.e., a grid of aggregates grouped by different dimensions)

```
SELECT product_sk, date_sk,
       sum(net_price)
FROM fact_sales
GROUP BY product_sk, date_sk
```

Example

product_sk

date_key	32	33	34	35	…	total
140101	149.60	31.01	84.58	28.18	…	40710.53
140102	132.18	19.78	82.91	10.96	…	73091.28
140103	196.75	0.00	12.52	64.67	…	54688.10
140104	178.36	9.98	88.75	56.16	…	95121.09
…	…	…	…	…	…	…
total	14967.09	5910.43	7328.85	6885.39	…	5365M

In general more than two dimensions

Pre-Aggregation – Data Cubes

```
SELECT SUM(net_price)
FROM fact_sales
WHERE product_sk = 32
AND net_price > 100.00
```

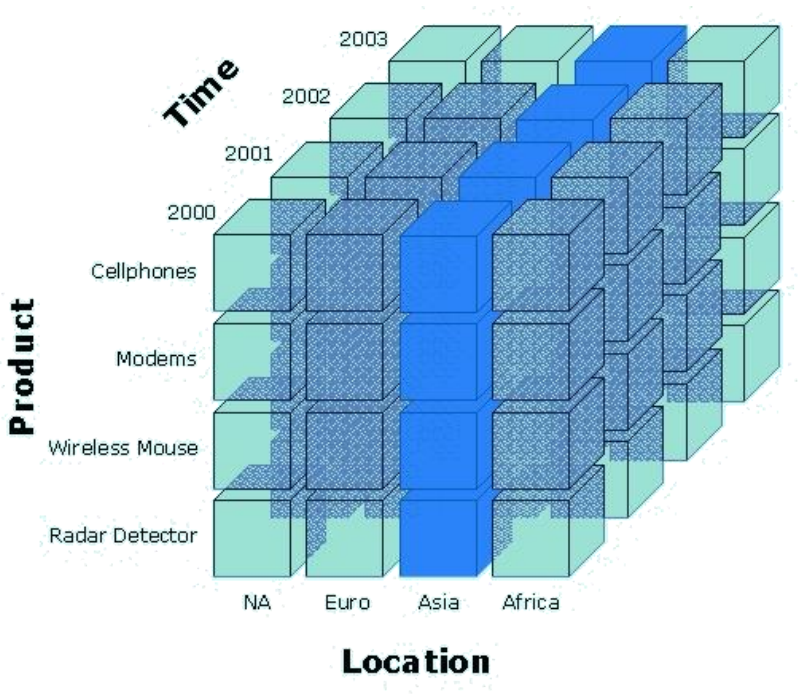
```
SELECT SUM(net_price)
FROM fact_sales
WHERE product_sk = 33
AND date_sk = 140101
```

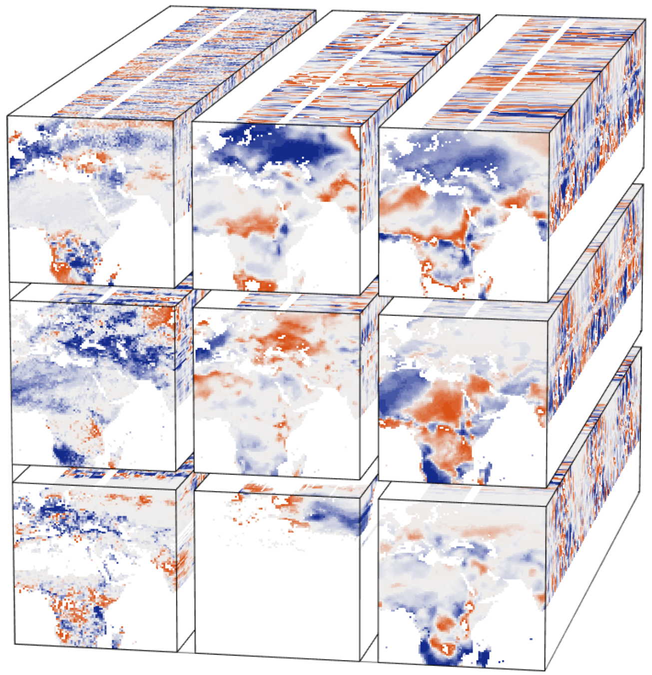
```
SELECT SUM(net_price)
FROM fact_sales
WHERE date_sk = 140101
```

Example

date_key \ product_sk	32	33	34	35	…	total
140101	149.60	31.01	84.58	28.18	…	40710.53
140102	132.18	19.78	82.91	10.96	…	73091.28
140103	196.75	0.00	12.52	64.67	…	54688.10
140104	178.36	9.98	88.75	56.16	…	95121.09
…	…	…	…	…	…	…
total	14967.09	5910.43	7328.85	6885.39	…	5365M

Pre-Aggregation – Data Cubes: Dimension

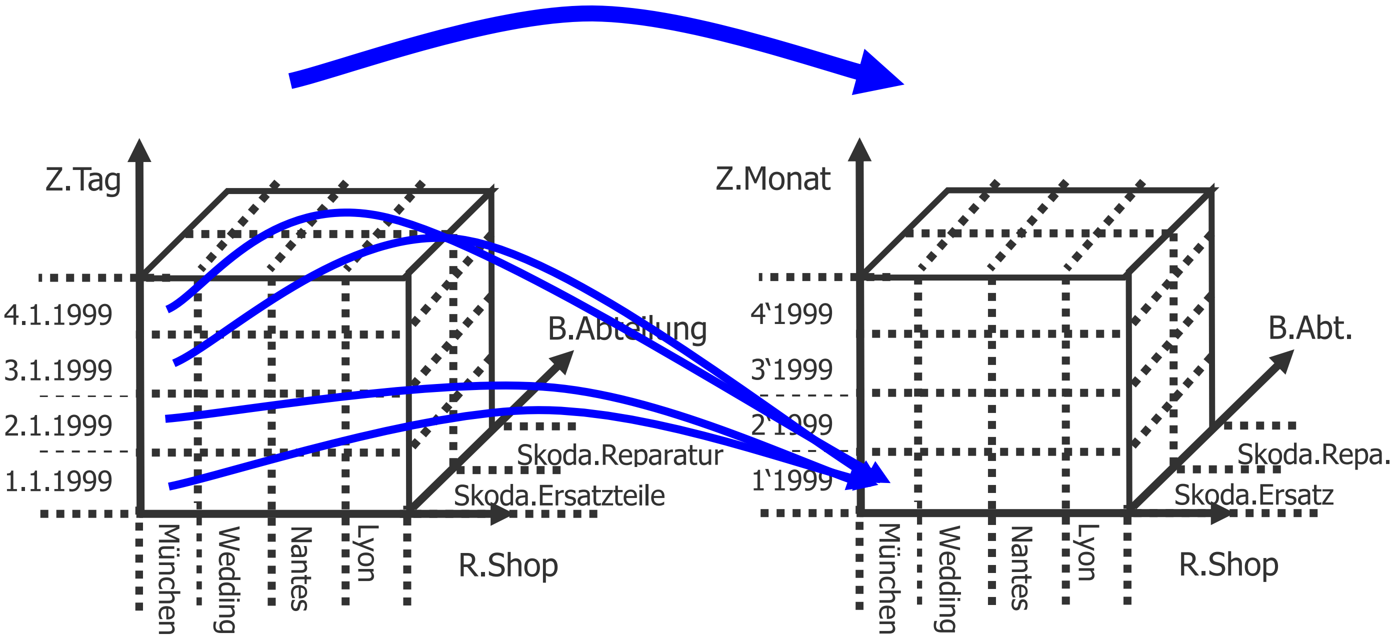




Pre-Aggregation – Data Cubes: Operations

Roll-Up

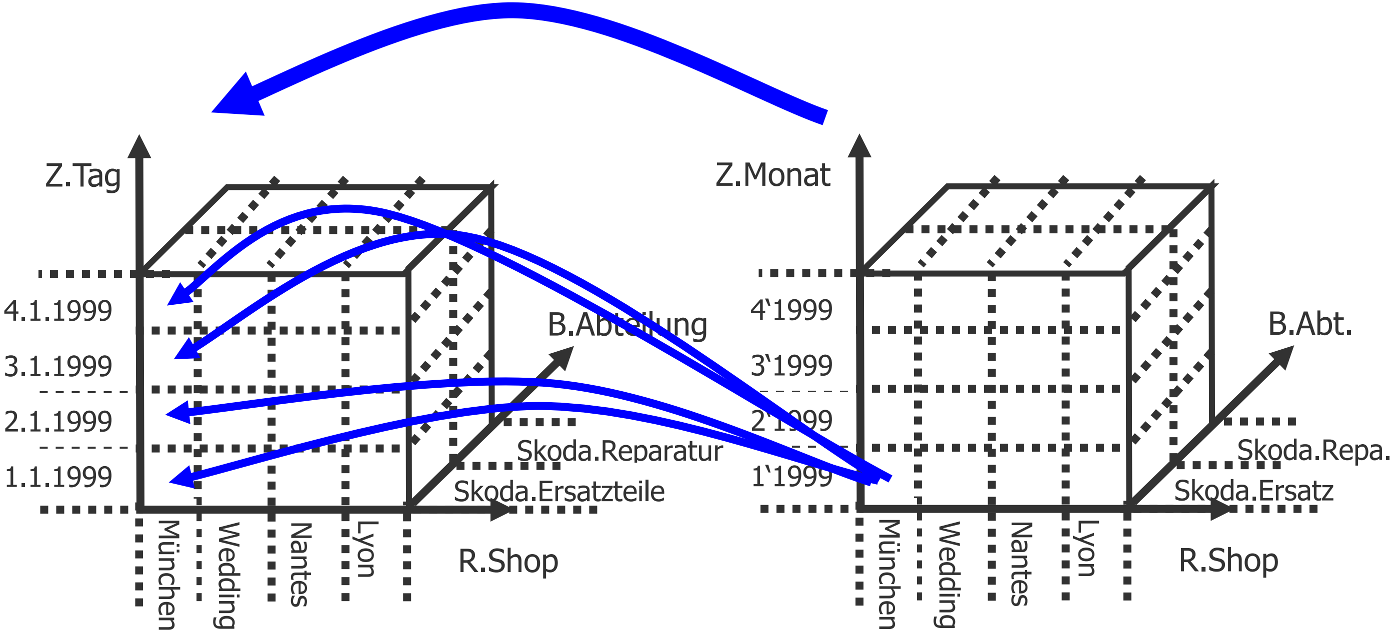
- Aggregate one dimension of a data cube.



Pre-Aggregation – Data Cubes: Operations

Drill-Down

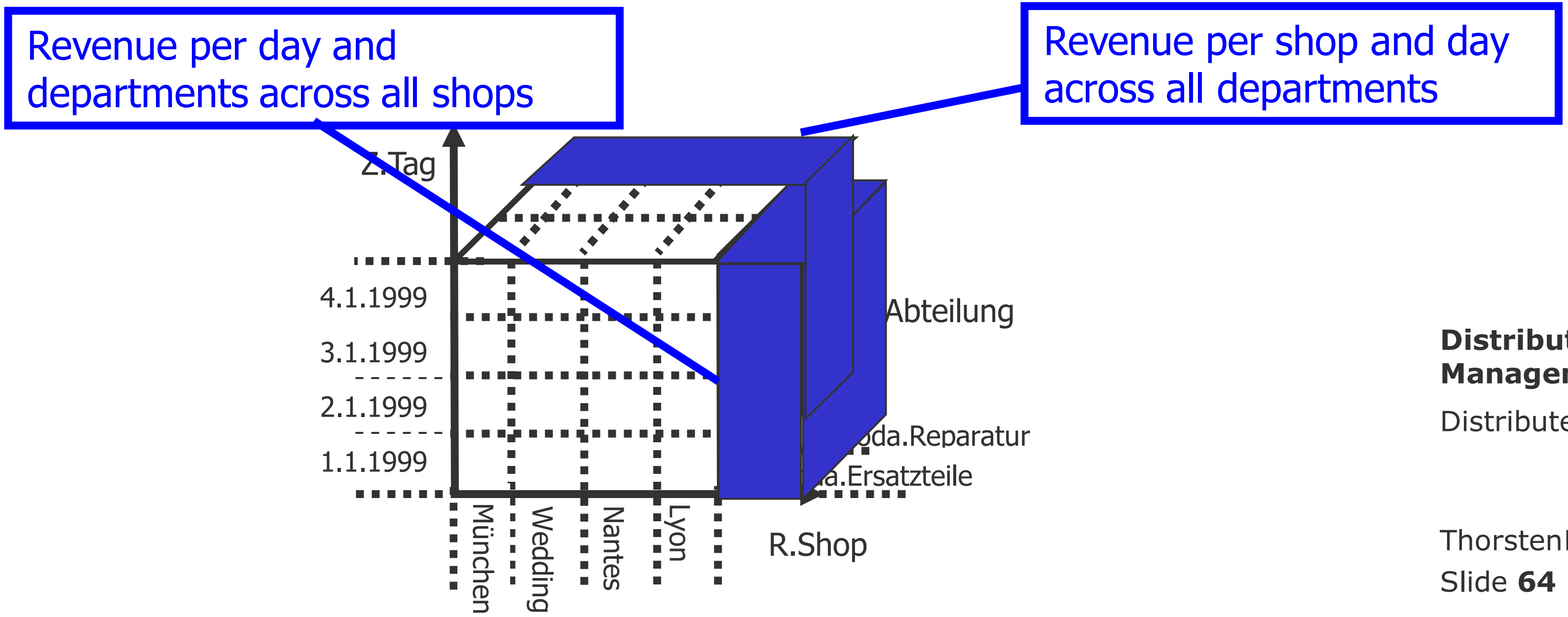
- Unfold one dimension of a data cube.



Pre-Aggregation – Data Cubes: Operations

Aggregate-to-TOP

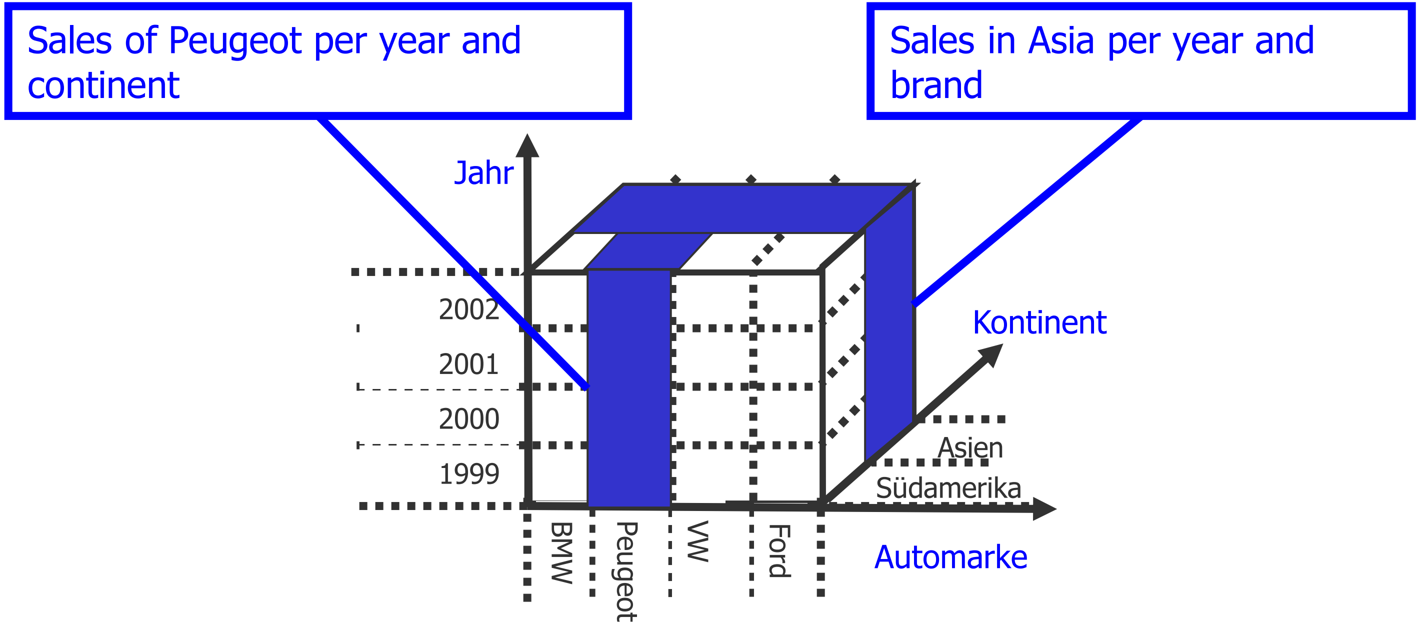
- Aggregate all values in one dimension; reduce cube dimensionality by 1.



Pre-Aggregation – Data Cubes: Operations

Slicing

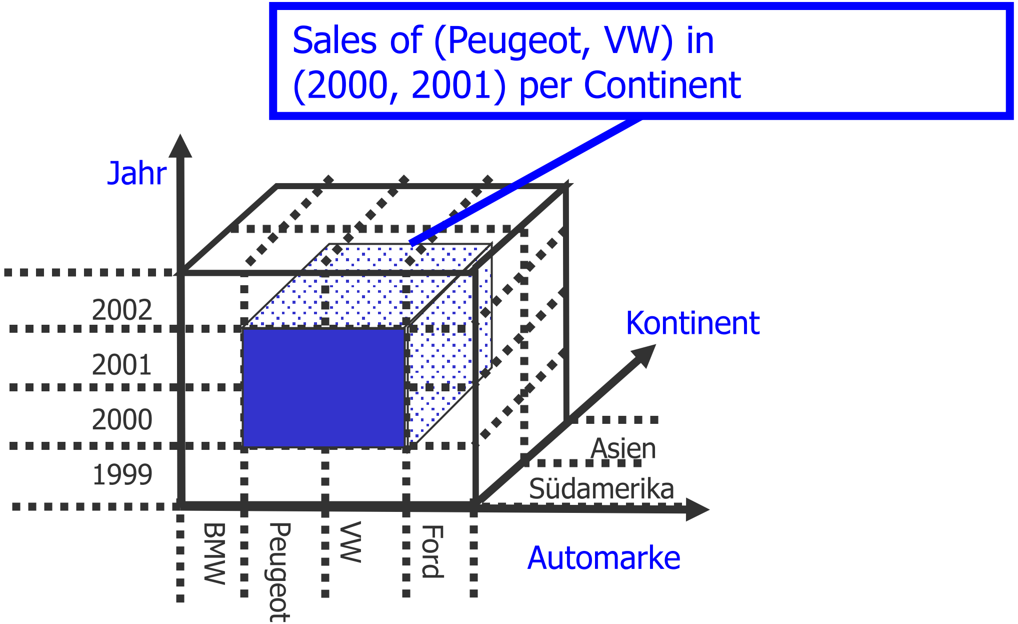
- Select/filter a value for one dimension; reduce cube dimensionality by 1.



Pre-Aggregation – Data Cubes: Operations

Dicing

- Select/filter some values for multiple dimension; make cube smaller.



Column-oriented Storage

Observation

- Data warehouse tables are often very wide (>100 columns), but analytical queries access only very few columns.
  - Usually 4 to 5 and rarely "SELECT *"
- Most data models (relational, key-value, column family, document) store data record-wise and, hence, must read, parse and filter all data for analytical queries.

Column-oriented Storage

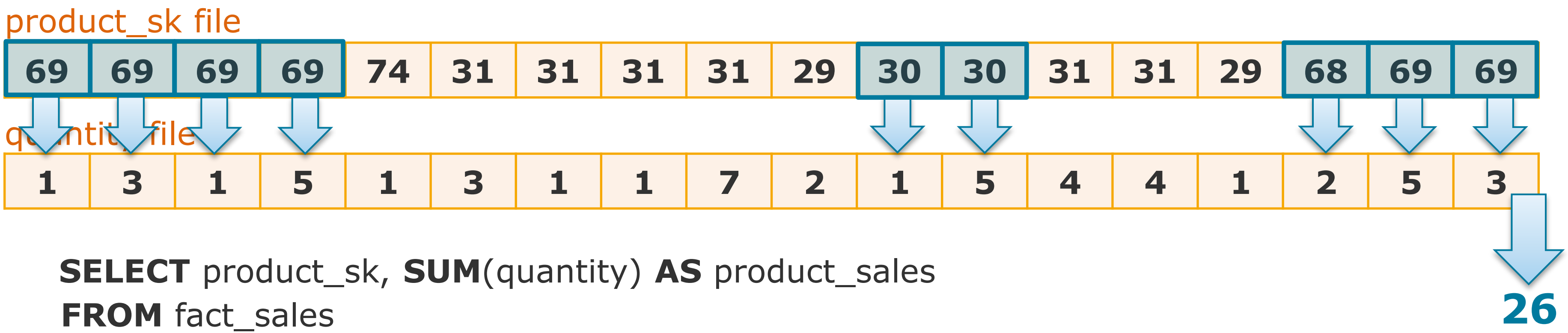
- Store data attribute-wise instead of record-wise:
  - One file per attribute
  - Values ordered by record index in each file
  - For each query: scan only required attribute files

Column-oriented Storage – Example

fact_sales table

date_key	product_sk	store_sk	promotion_sk	customer_sk	quantity	net_price	discount_price
140102	69	4	NULL	NULL	1	13.99	13.99
140102	69	5	19	NULL	3	14.99	9.99
140102	69	5	NULL	191	1	14.99	14.99
140102	74	3	23	202	5	0.99	0.89
140103	31	2	NULL	NULL	1	2.49	2.49
140103	31	3	NULL	NULL	3	14.99	9.99
140103	31	3	21	123	1	49.99	39.99
140103	31	8	NULL	233	1	0.99	0.99
file 1	file 2	file 3	file 4	file 5	file 6	file 7	file 8

Column-oriented Storage – Example



SELECT product_sk, **SUM**(quantity) **AS** product_sales
FROM fact_sales
WHERE product_sk **IN** (30, 68, 69)
GROUP BY product_sk;

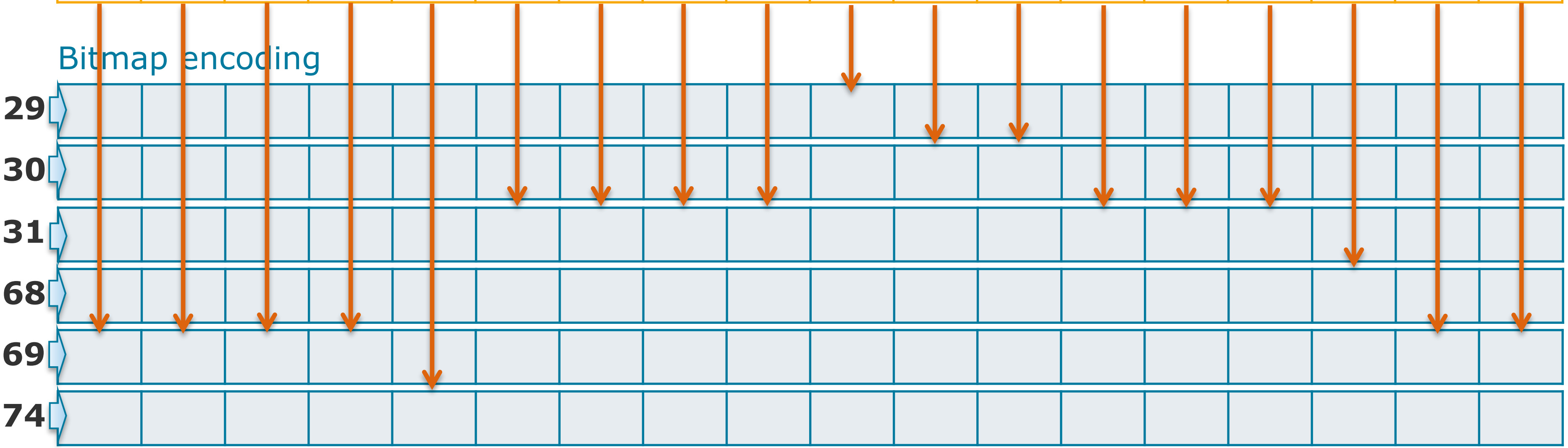
1. Scan the product_sk file for values 30, 68, and 69; remember the position (=row) of each occurrence.
2. Read the quantities at the retrieved positions in the quantity file and sum them up.

We can do even better!
➢ **Compression**

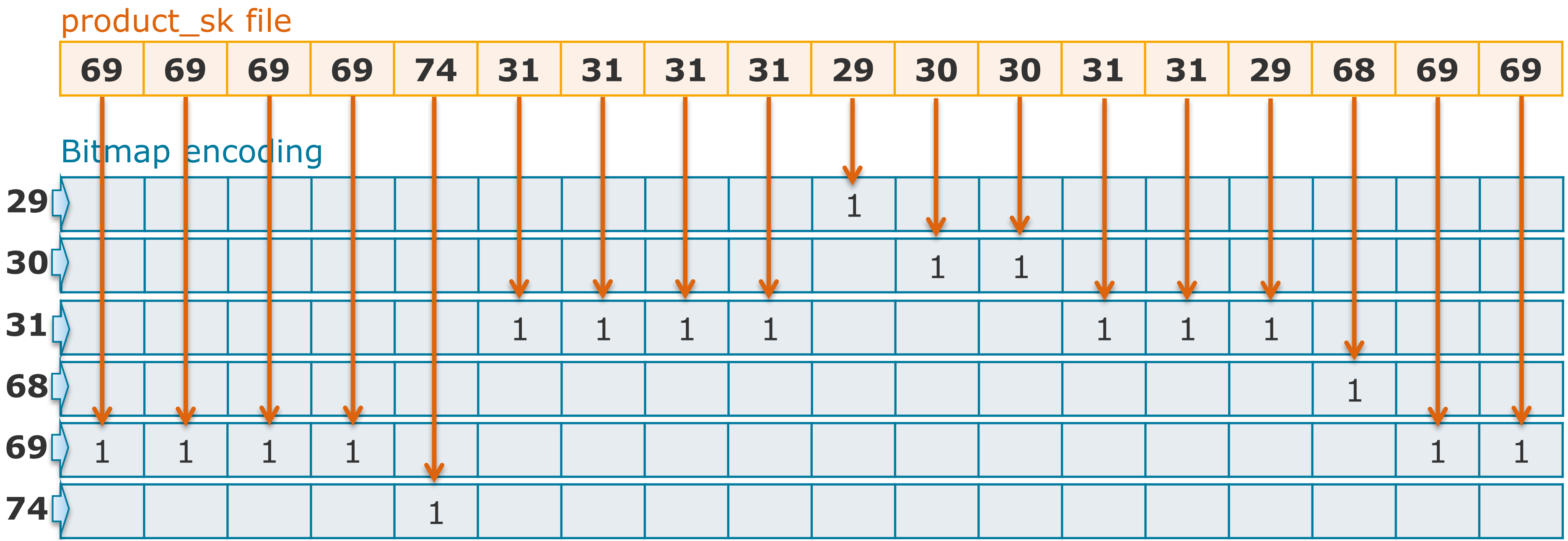
Column-oriented Storage – Example

product_sk file

69	69	69	69	74	31	31	31	31	29	30	30	31	31	29	68	69	69



Column-oriented Storage – Example



Column-oriented Storage – Example

product_sk file

69	**69**	**69**	**69**	**74**	**31**	**31**	**31**	**31**	**29**	**30**	**30**	**31**	**31**	**29**	**68**	**69**	**69**

Bitmap encoding

29	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
31	0	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0	0	0
68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
69	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
74	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Smaller? … Here, yes:

18 · 32 Bit = **576 Bit** vs. 6 · (32 Bit + 32 Bit) = **384 Bit**

Column-oriented Storage – Example

```
SELECT product_sk, SUM(quantity) AS product_sales
FROM fact_sales
WHERE product_sk IN (30, 68, 69)
GROUP BY product_sk;
```

product_sk file

Bitmap encoding

29	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
31	0	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0	0	0
68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
69	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
74	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit-wise OR

Column-oriented Storage – Example

For more compression techniques see:

Daniel Abadi et. al. “The Design and Implementation of Modern Column-Oriented Database Systems”, 2013.

product_sk file

Bitmap encoding

29	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
31	0	0	0	0	0	1	1	1	1	0	0	0	1	1	1	0	0	0
68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
69	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
74	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Problem:

Bitmaps are very long and sparse in practice!

Run-length encoding

29	9	1			9 zeros, 1 ones, rest zeros
30	10	2			10 zeros, 2 ones, rest zeros
31	5	4	3	3	5 zeros, 4 ones, 3 zeros, 3 ones, rest zeros
68	15	1			15 zeros, 1 ones, rest zeros
69	0	4	12	2	0 zeros, 4 ones, 12 zeros, 2 ones, rest zeros
74	4	1			4 zeros, 1 ones, rest zeros

See also: Roaring Bitmaps http://roaringbitmap.org/

Distributed DBMSs
Overview

1. Distributed DBMSs
2. Materialized vs. Virtual
3. Data Warehouses
4. **Federated Database Management Systems**

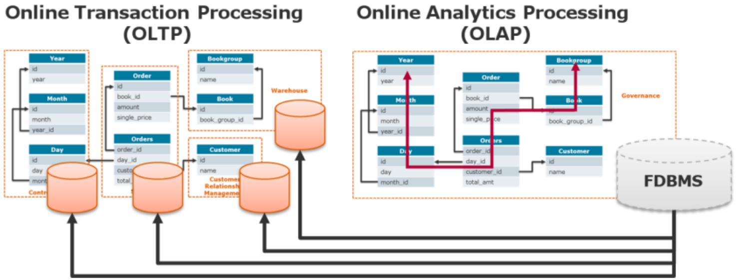
Recap

Materialized

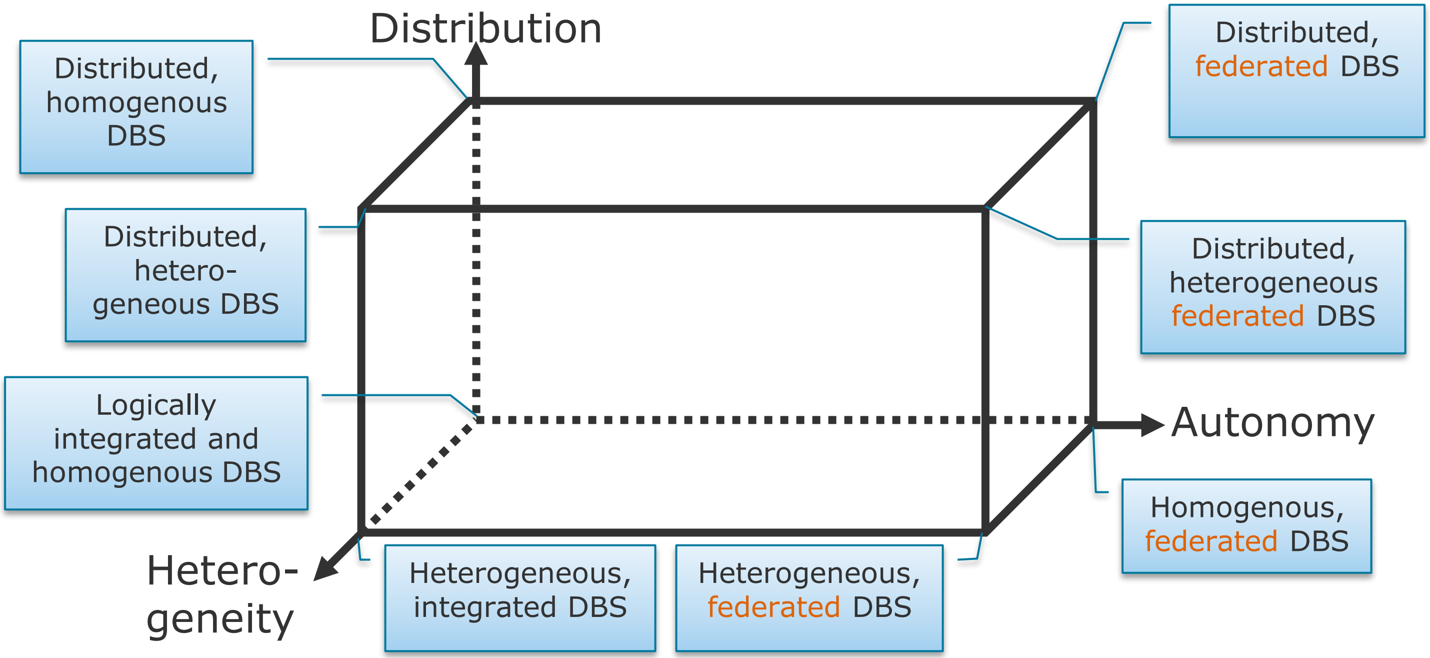
- A-priori integration
- Centralized data store
- Centralized query processing
- Typical example:
  data warehouse

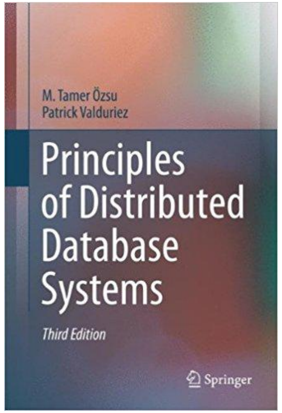
Virtual

- On-demand integration
- Decentralized data
- Decentralized query processing
- Typical example:
  mediator-based information system

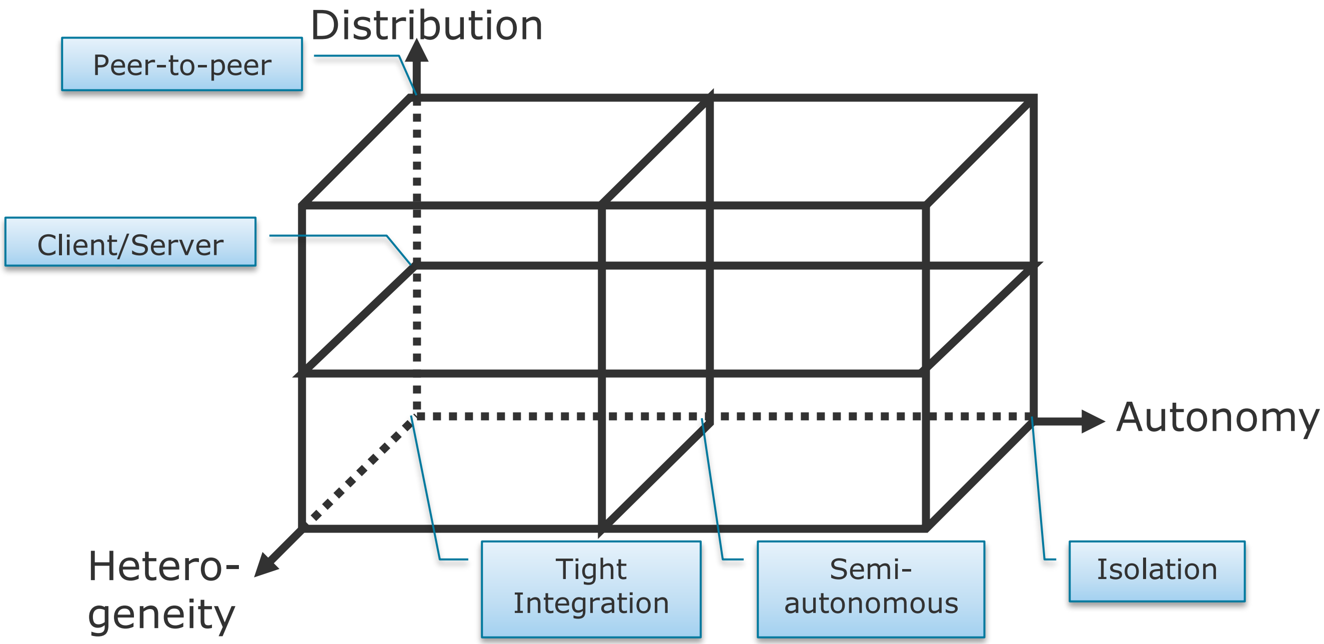


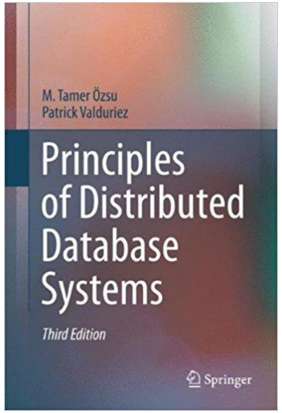
Classification of Distributed DB Systems



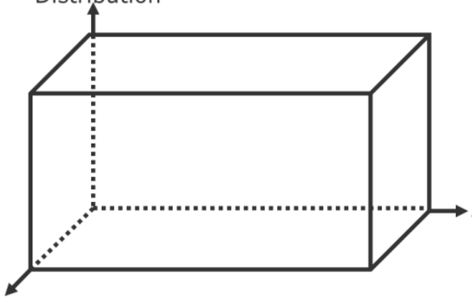


Classification of Distributed DB Systems



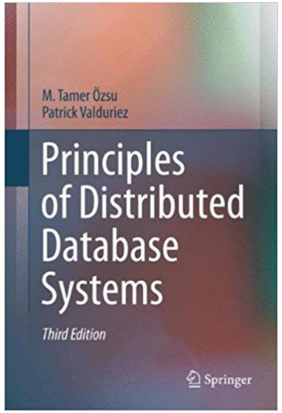


Connecting the three Dimensions



- Distribution leads to Autonomy:
  - Intra-organisation: Historically
  - Inter-organisation: Internet & WWW

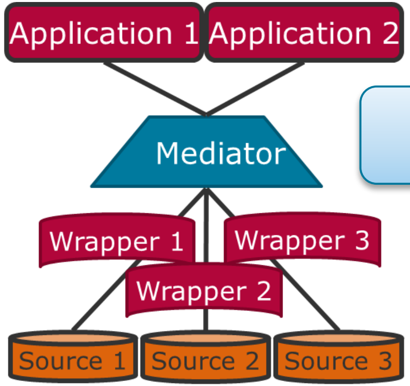
- Autonomy leads to Heterogeneity:
  - Responsibility is with local admins.



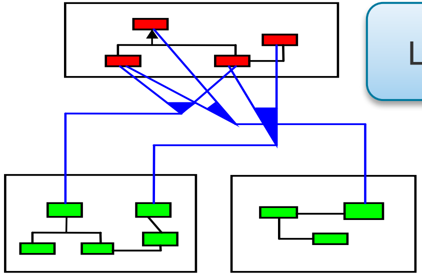
The Federated Approach

Hasso Plattner Institut

- Create the global schema (schema integration).
  - Store it as DBMS schema.
- Create wrappers for each data source that …
  - map from local schemata to the global schema.
  - model the query capabilities of each source.
- Data remains at the sources.
- Data sources remain autonomous.
  - Are not even aware of participation
- Global schema takes declarative queries that are transparently mapped to wrappers.
- Query execution is as distributed as possible.
  - Send sub-queries to sources; wait for results.
  - Federated system replaces missing capabilities.



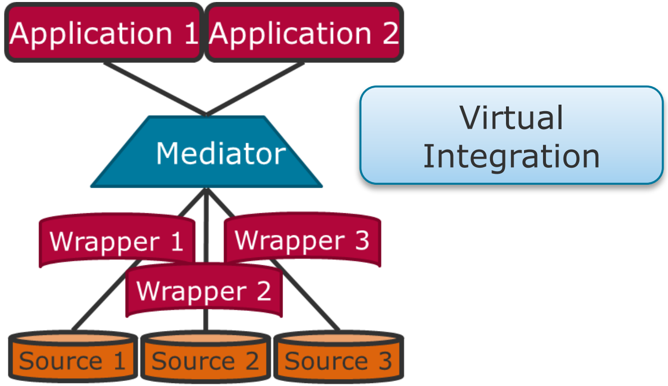
Virtual Integration

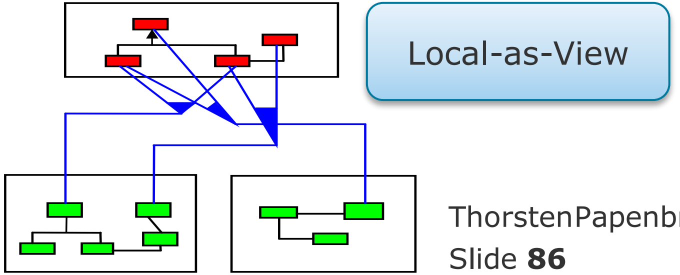


Local-as-View

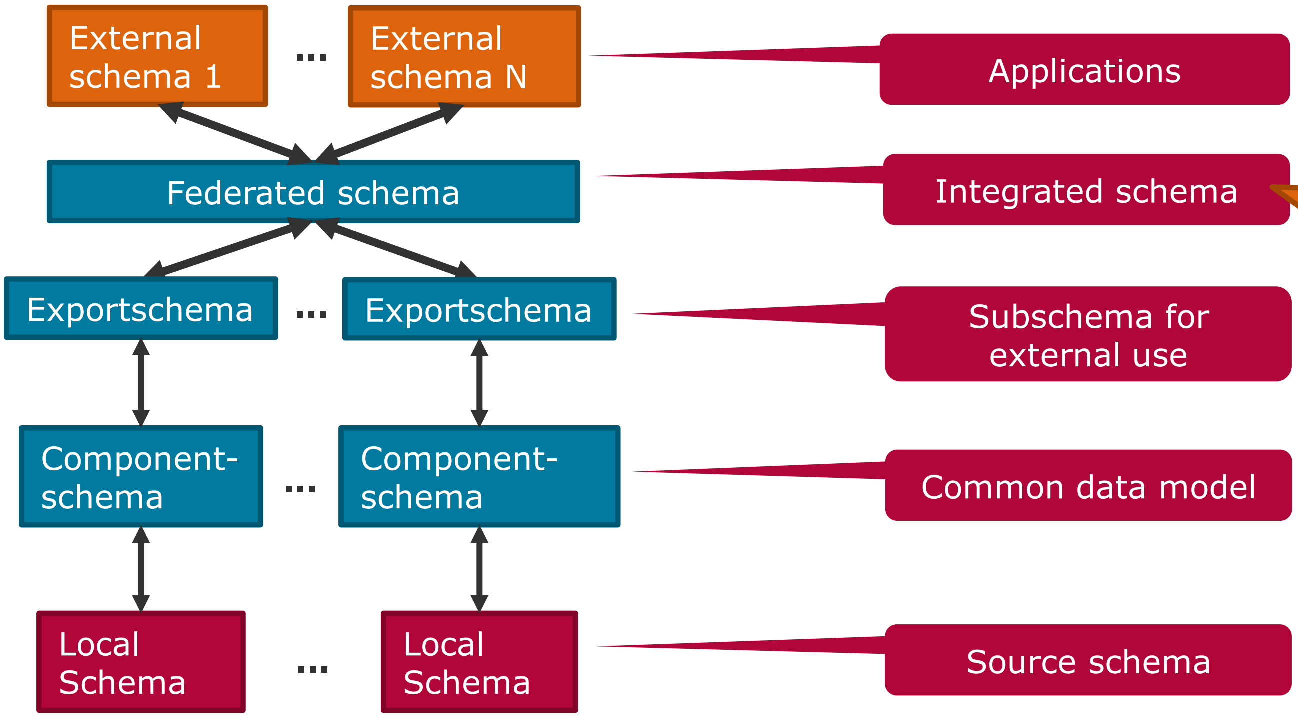
The Federated Approach – Applications

- Meta-search engines
- Company mergers
  - Customer data
  - HR data
- Clinical information systems
  - X-ray/CRT images
  - Medial charts
  - Administrative information
  - Insurance information
- …



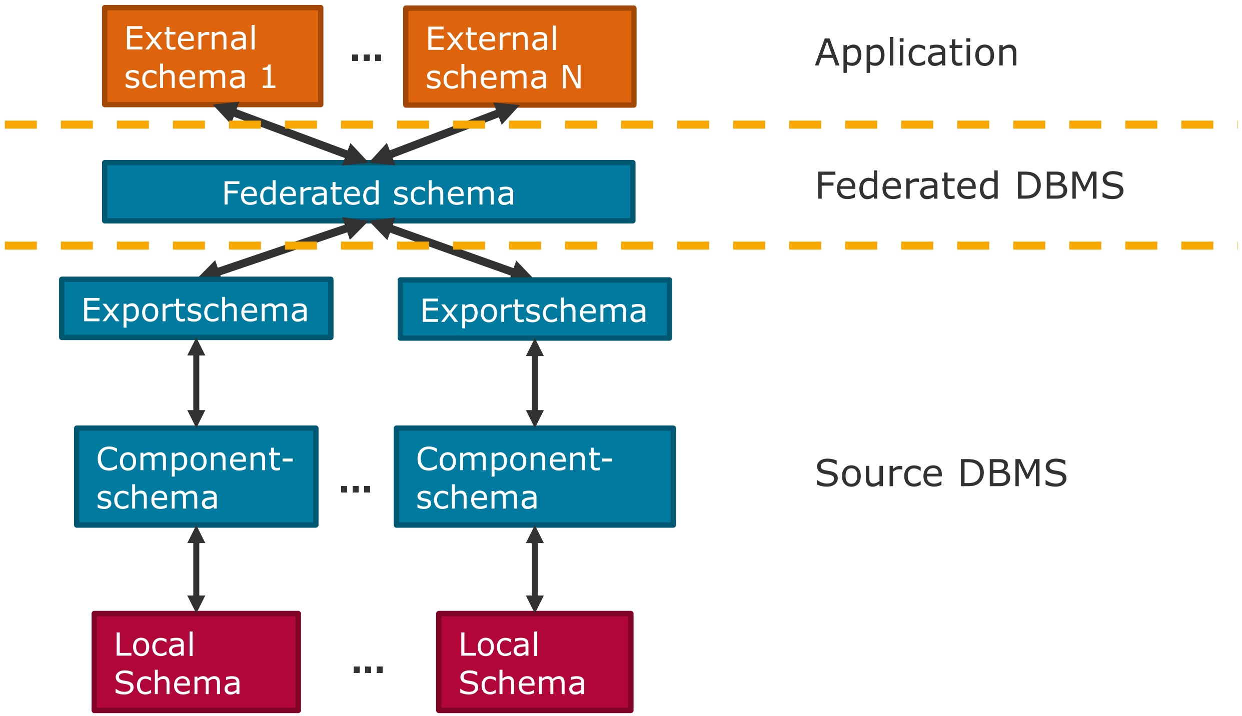


5-Layer Architecture



aka.
- Import Schema
- Globales Schema
- Enterprise Schema
- Unified Schema
- Mediator Schema

5-Layer Architecture



Exercise

1. Why does query optimization using materialized views resemble Local as View and not Global as View?

2. Provide a brief explanation as to why star schemes are typically not suitable for OLTP.

3. When is bitmap compression most effective?

4. Apply bitmap compression to the string "CABBBBCCBCDBDAA" and give the result.