# Big Data

Data Warehouse

# Goal of this Module

- Understand how Big Data has been done so far
  - i.e., how to exploit relational database systems
  - which data models to use
  - some interesting algorithms

- Also, understand the limitations and why we need new technology
  - you need to understand the starting point!

# Puzzle of the Day

- There is a jazz festival in Montreux.
- Make sure Migros Montreux has enough beer.

- This is a Big Data problem!
  - how much beer do we need in each store?
- How does Migros solve that problem today?
  - data warehouses (today)
- How could Migros solve that problem in future?
  - data warehouses + event calendar + Facebook + …
  - (coming weeks)

# Selected References on Data Warehouses

- General
  - Chaudhuri, Dayal: An Overview of Data Warehousing and OLAP Technology. SIGMOD Record 1997
  - Lehner: Datenbanktechnologie für Data Warehouse Systeme. Dpunkt Verlag 2003
  - (…)
- New Operators and Algorithms
  - Agrawal, Srikant: Fast Algorithms for Association Rule Mining. VLDB 1994
  - Barateiro, Galhardas: A Survey of Data Quality Tools. Datenbank Spektrum 2005
  - Börszonyi, Kossmann, Stocker: Skyline Operator. ICDE 2001
  - Carey, Kossmann: On Saying Enough Already in SQL. SIGMOD 1997
  - Dalvi, Suciu: Efficient Query Evaluation on Probabilistic Databases. VLDB 2004
  - Gray et al.: Data Cube... ICDE 1996
  - Helmer: Evaluating different approaches for indexing fuzzy sets. Fuzzy Sets and Systems 2003
  - Olken: Database Sampling - A Survey. Technical Report LBL.
  - (…)

# History of Databases

- Age of Transactions (70s - 00s)
  - Goal: reliability - make sure no data is lost
  - 60s: IMS (hierarchical data model)
  - 80s: Oracle (relational data model)
- Age of Business Intelligence (95 -)
  - Goal: analyze the data -> make business decisions
  - Aggregate data for boss.  Tolerate imprecision!
  - SAP BW, Microstrategy, Cognos, … (rel. model)
- Age of „Big Data“ and „Data for the Masses“
  - Goal: everybody has access to everything, M2M
  - Google (text), Cloud (XML, JSON: Services)

# Some Selected Topics

- Motivation and Architecture
- SQL Extensions for Data Warehousing (DSS)
- Algorithms and Query Processing Techniques
- ETL, Virtual Databases (Data Integration)
- Parallel Databases
- Column Stores, Vector Databases
- Data Mining
- Probabilistic Databases
- Temporal Databases

- **This is a whole class for itself (Spring semester)**
  - we will only scratch the surface here

# OLTP vs. OLAP

- OLTP – Online Transaction Processing
  - Many small transactions
    (point queries: UPDATE or INSERT)
  - Avoid redundancy, normalize schemas
  - Access to consistent, up-to-date database
- OLTP Examples:
  - Flight reservation (see IS-G)
  - Order Management, Procurement, ERP
- Goal:  6000 Transactions per second (Oracle 1995)

# OLTP vs. OLAP

- OLAP – Online Analytical Processing
  - Big queries (all the data, joins); no Updates
  - Redundancy a necessity (Materialized Views, special-purpose indexes, de-normalized schemas)
  - Periodic refresh of data (daily or weekly)
- OLAP Examples
  - Management Information (sales per employee)
  - Statistisches Bundesamt (Volkszählung)
  - Scientific databases, Bio-Informatics
- Goal: Response Time of seconds / few minutes

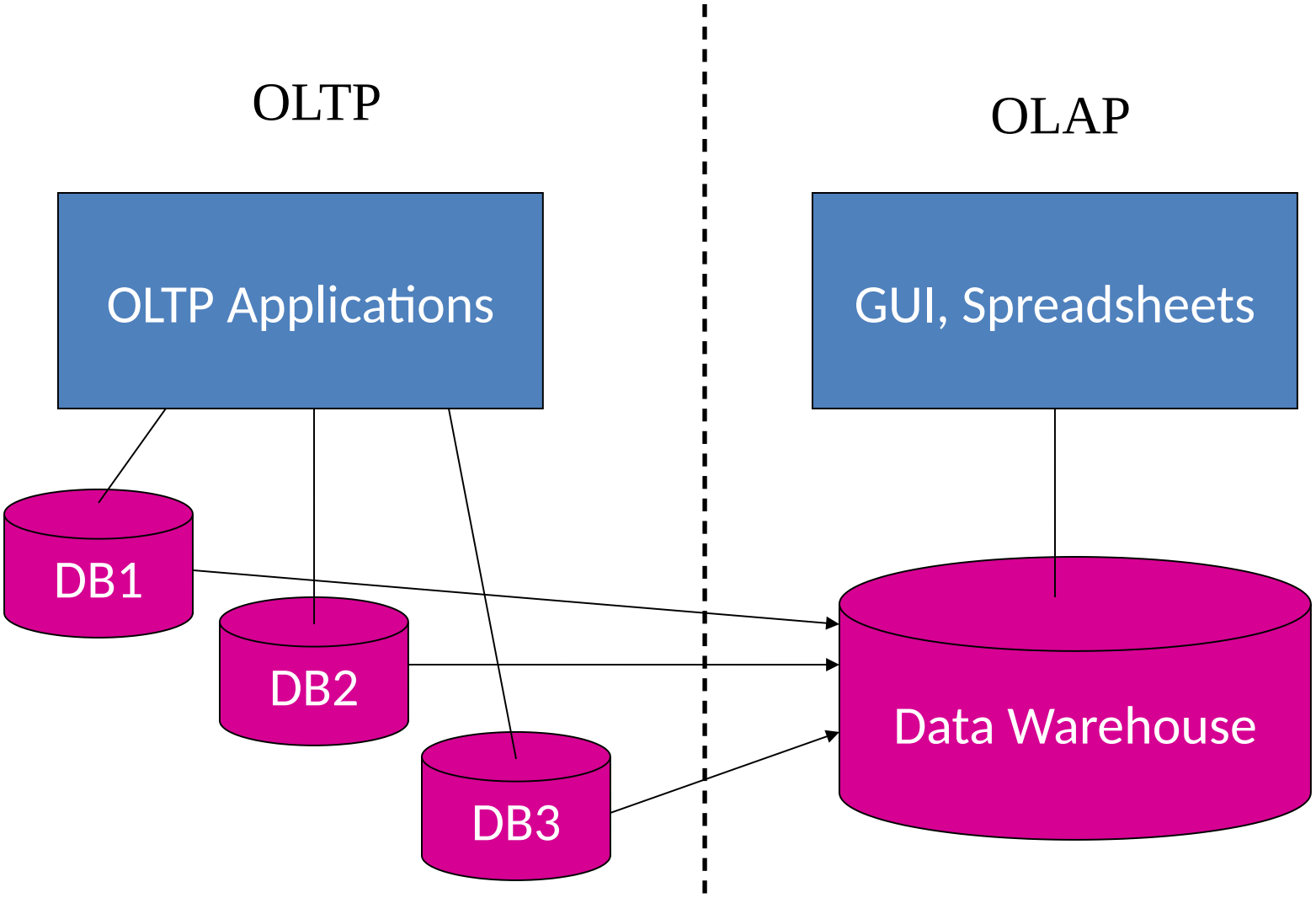
# OLTP vs. OLAP (Water and Oil)

- Lock Conflicts:  OLAP blocks OLTP
- Database design:
  - OLTP normalized, OLAP de-normalized
- Tuning, Optimization
  - OLTP: inter-query parallelism, heuristic optimization
  - OLAP: intra-query parallelism, full-fledged optimization
- Freshness of Data:
  - OLTP:  serializability
  - OLAP:  reproducability
- Precision:
  - OLTP:  ACID
  - OLAP:  Sampling, Confidence Intervals

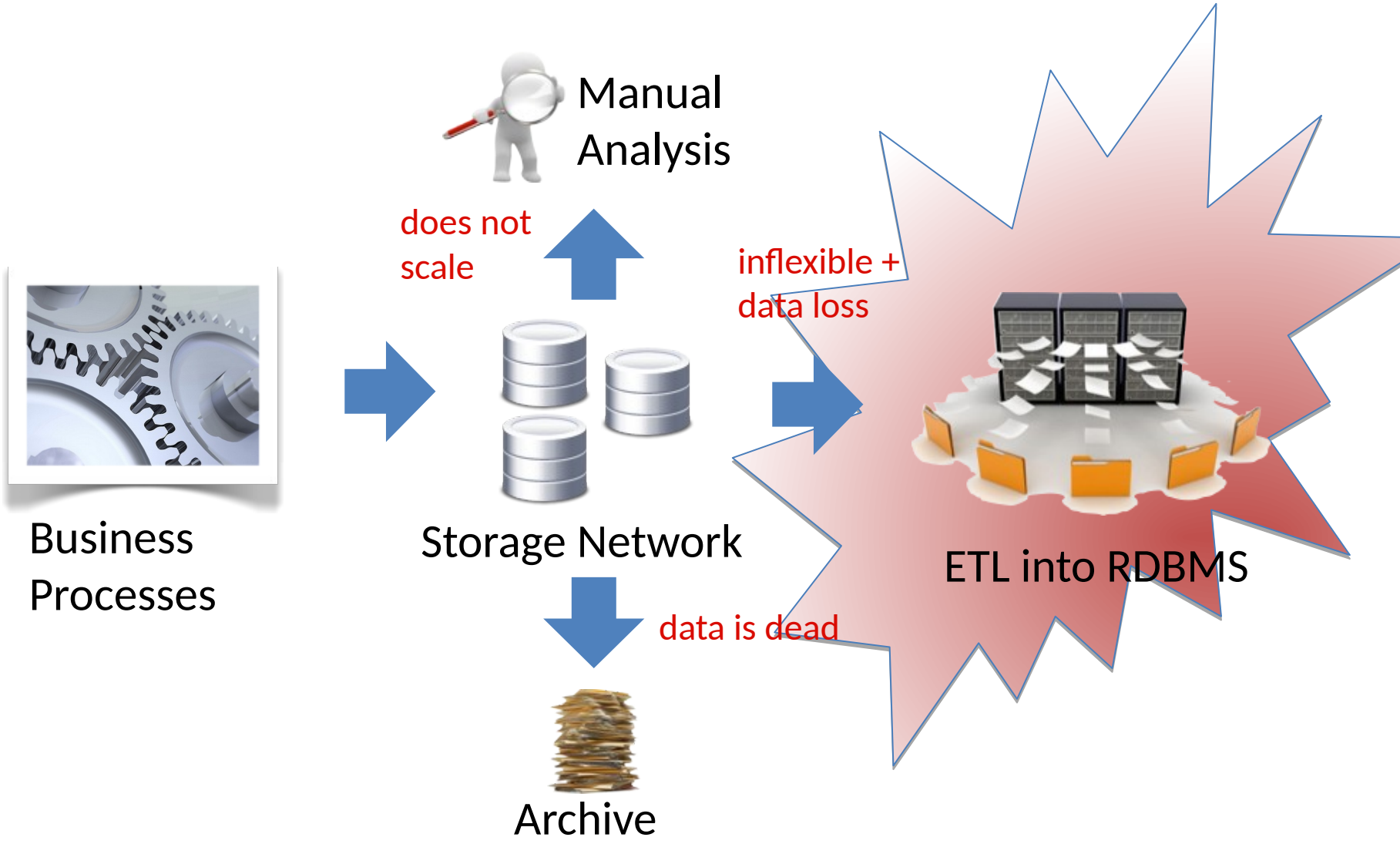
# Solution: Data Warehouse

- Special Sandbox for OLAP
- Data input using OLTP systems
- Data Warehouse aggregates and replicates data (special schema)
- New Data is ***periodically*** uploaded to Warehouse
- Old Data is deleted from Warehouse
  - Archiving done by OLTP system for legal reasons

# Architecture



# Limitations of State of the Art



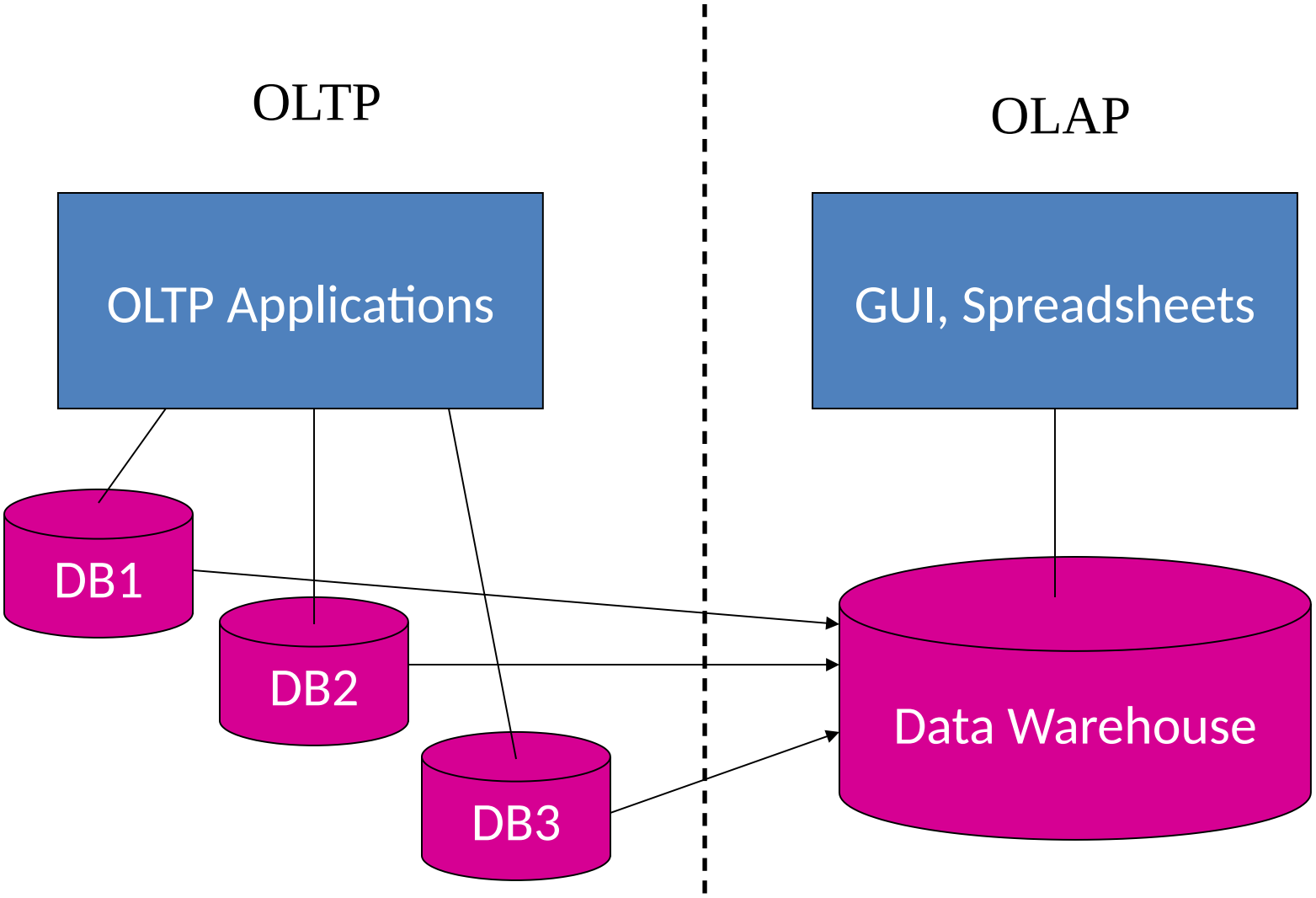
# Data Warehouses in the real World

- First industrial projects in 1995
- At beginning, 80% failure rate of projects
- Consultants like Accenture dominate market
- Why difficult: Data integration + cleaning, poor modeling of business processes in warehous
- Data warehouses are expensive (typically as expensive as OLTP system)
- Success Story: WalMart - 20% cost reduction because of Data Warehouse (just in time...)

# Products and Tools

- Oracle 11g, IBM DB2, Microsoft SQL Server, ...
  - All data base vendors
- SAP Business Information Warehouse (Hana)
  - ERP vendors
- MicroStrategy, Cognos
  - Specialized vendors
  - „Web-based EXCEL“
- Niche Players (e.g., Btell)
  - Vertical application domain

# Architecture



# ETL Process

- Major Cost Factors of Data Warehousing
  - define schema / data model (next)
  - define ETL process
- ETL Process
  - extract: suck out the data from OLTP system
  - transform:  clense it, bring it into right format
  - load: add it to the data warehouse
- Staging areas
  - modern data warehouses keep results at all stages

# Some Details

- Extract
  - easy, if OLTP is a relational database
    - (use triggers, replication facilities, etc.)
  - more difficult, if OLTP data comes from file system
- Transform
  - data clensing: can be arbitrary complicated
    - machine learning, workflow with human input, …
  - structures: many tools that generate code
- Load
  - use bulkloading tools from vendors

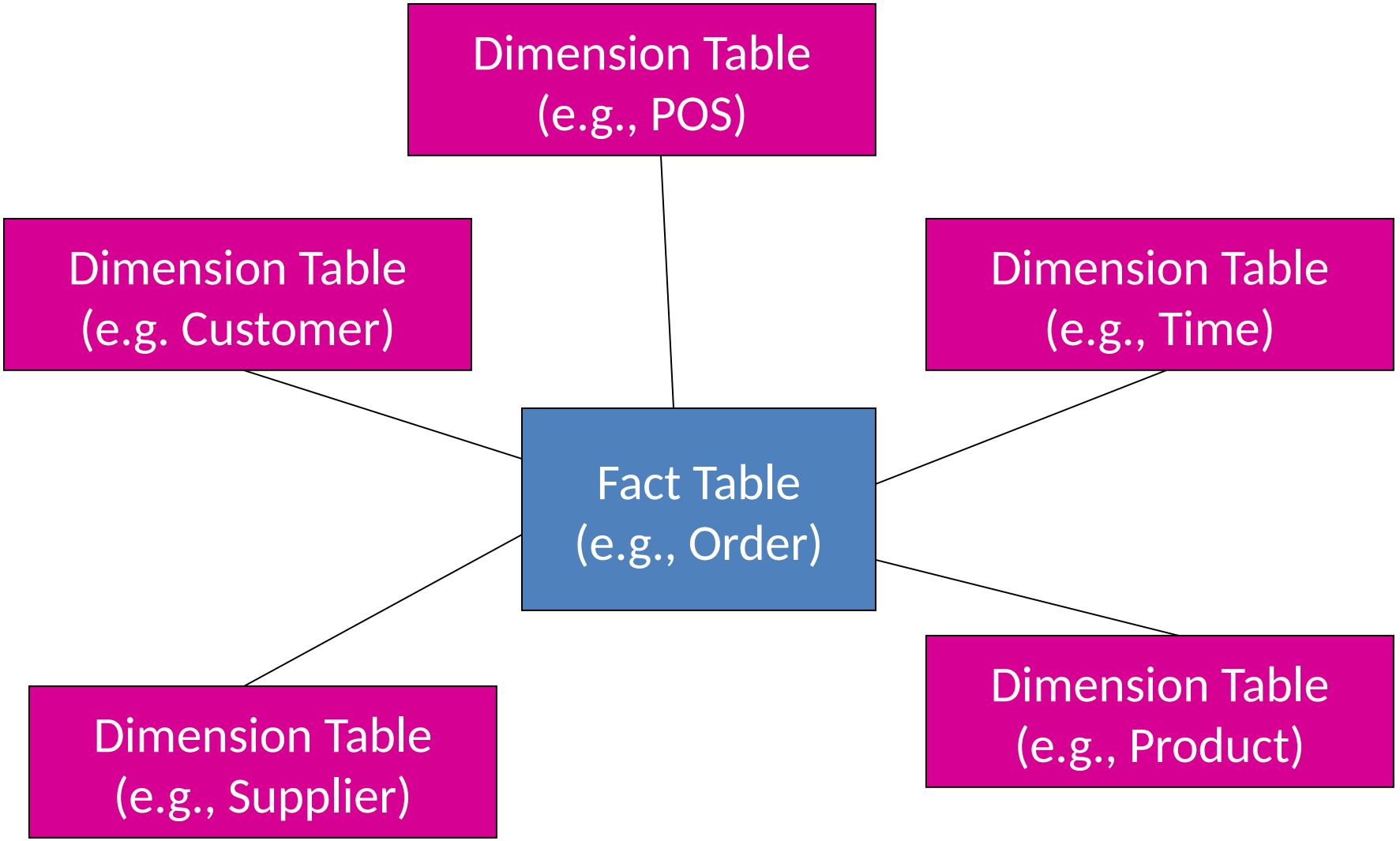
# Some Considerations

- When to ETL data?
  - freshness: periodically vs. continuously
  - consistency: do we need to transact the ETLs
- Granularity of ETL?
  - individual tuples vs. batches
  - cost / freshness / quality tradeoffs
    - often a batch can be better clensed
- Infrastructure?
  - ETL from same machine or even same DB
  - workload / performance separation vs. cost

# ETL vs. Big Data

- ETL is the exact opposite of "modern" Big Data
  - "speed": does not really work for fast data
  - philosophy: change question -> change ETL workflow
- Big Data *prefers* in-situ processing
  - "volume": not all data is worth ETLing
  - "statistics": error may be part of the signal (!)
  - "cost:" why bother if you can have it all in one
    - products like SAP Hana also go into this direction
  - "diversity:" increases complexity of ETL process
- But, Big Data has no magic with regard to quality
  - and ETL great if investment is indeed worth-while
    - valuable data vs. mass data

# Star Schema (relational)



# Fact Table (Order)

| No. | Cust. | Date | ... | POS | Price | Vol. | TAX |
|---|---|---|---|---|---|---|---|
| 001 | Heinz | 13.5. | ... | Mainz | 500 | 5 | 7.0 |
| 002 | Ute | 17.6. | ... | Köln | 500 | 1 | 14.0 |
| 003 | Heinz | 21.6. | ... | Köln | 700 | 1 | 7.0 |
| 004 | Heinz | 4.10. | ... | Mainz | 400 | 7 | 7.0 |
| 005 | Karin | 4.10. | ... | Mainz | 800 | 3 | 0.0 |
| 006 | Thea | 7.10. | ... | Köln | 300 | 2 | 14.0 |
| 007 | Nobbi | 13.11. | ... | Köln | 100 | 5 | 7.0 |
| 008 | Sarah | 20.12 | ... | Köln | 200 | 4 | 7.0 |

# Fact Table

- Structure:
  - key (e.g., Order Number)
  - Foreign key to all dimension tables
  - measures (e.g., Price, Volume, TAX, …)
- Store *moving data (Bewegungsdaten)*
- Very large and normalized

# Dimension Table (PoS)

| Name | Manager | City | Region | Country | Tel. |
|---|---|---|---|---|---|
| Mainz | Helga | Mainz | South | D | 1422 |
| Köln | Vera | Hürth | South | D | 3311 |

- De-normalized: City -> Region -> Country
  - Avoid joins
- fairly small and constant size
- Dimension tables store *master data (Stammdaten)*
- Attributes are called *Merkmale* in German

# Snowflake Schema

- If dimension tables get too large
  - Partition the dimension table
- Trade-Off
  - Less redundancy (smaller tables)
  - Additional joins needed
- Exercise:  Do the math!

# Typical Queries

```
SELECT  d1.x, d2.y, d3.z, sum(f.z1), avg(f.z2)
FROM    Fact f, Dim1 d1, Dim2 d2, Dim3 d3
WHERE  a < d1.feld < b  AND  d2.feld = c AND
        Join predicates
GROUP BY d1.x, d2.y, d3.z;
```

- Select by Attributes of Dimensions
  - E.g.,  region = „south“
- Group by Attributes of Dimensions
  - E.g., region, month, quarter
- Aggregate on measures
  - E.g., sum(price * volumen)

# Example

```
SELECT  f.region, z.month, sum(a.price * a.volume)
FROM    Order a, Time z, PoS f
WHERE   a.pos = f.name AND a.date = z.date
GROUP BY f.region, z.month
```

| | | |
|---|---|---|
| South | May | 2500 |
| North | June | 1200 |
| South | October | 5200 |
| North | October | 600 |

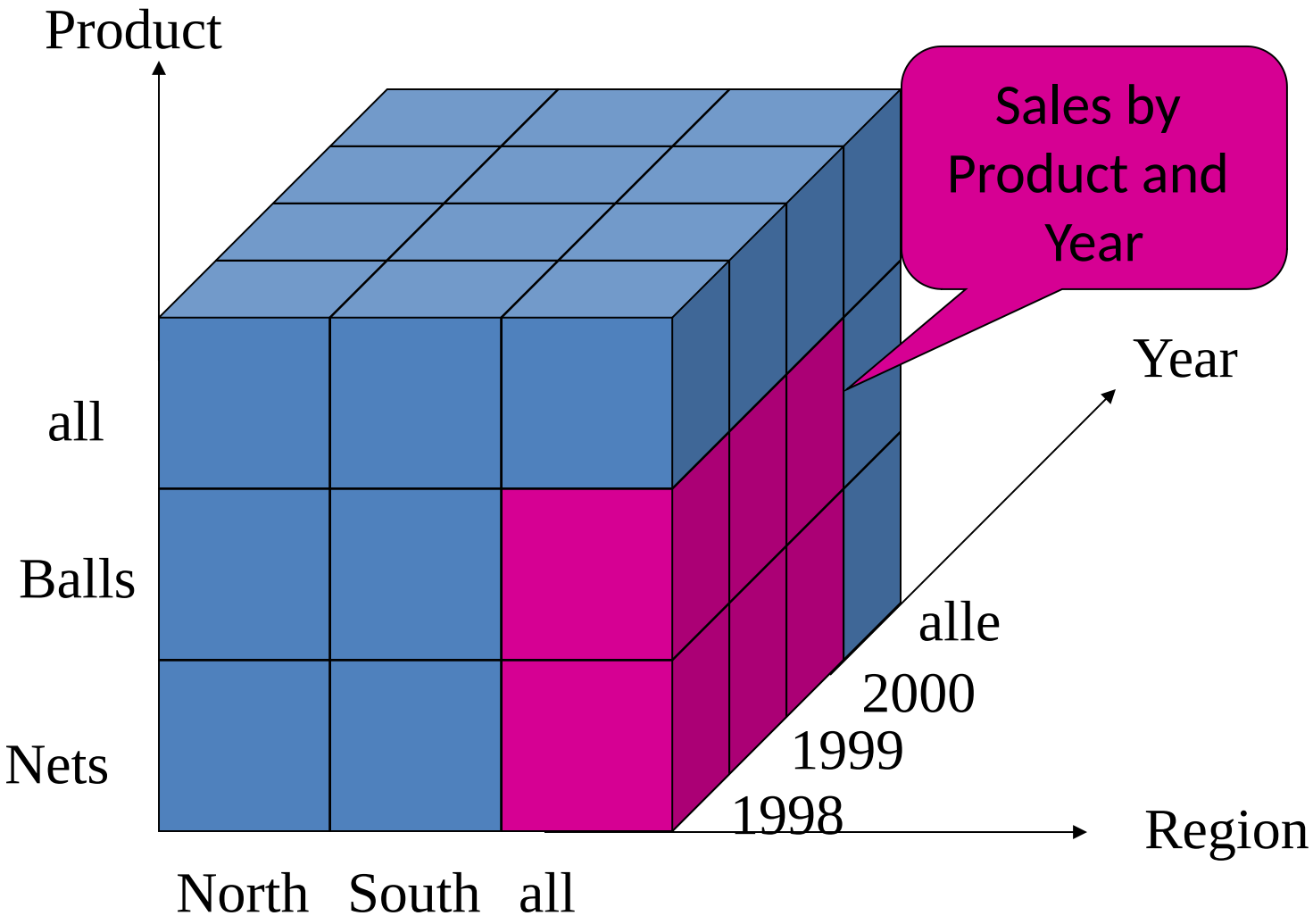
# Star Schema vs. Big Data

- Star Schema designed for specific questions
  - define “metrics” and “dimensions” upfront
  - thus, define questions you can ask upfront
  - great for operational BI
  - bad  for ad-hoc questions (e.g., disasters)
  - breaks philosophy of Big Data (collect, then think)
    - e.g., health record:  is “disease” metric or dimension?
- Poor on diversity
  - even if you know all the questions upfront, you may end up with multiple Star schemas

# Drill-Down und Roll-Up

- Add attribute to GROUP BY clause
  - More detailed results (e.g., more fine-grained results)
- Remove attribute from GROUP BY clause
  - More coarse-grained results (e.g., big picture)
- GUIs allow „Navigation“ through Results
  - Drill-Down: more detailed results
  - Roll-Up: less detailed results
- Typical operation, drill-down along hierarchy:
  - E.g., use „city“ instead of „region“

# Data Cube


Product
Sales by Product and Year
Year
all
Balls
Nets
alle
2000
1999
1998
Region
North
South
all

# Moving Sums, ROLLUP

- Example:
  GROUP BY ROLLUP(country, region, city)
  Give totals for all countries and regions
- This can be done by using the ROLLUP Operator
- Attention: The order of dimensions in the GROUP BY clause matters!!!
- Again: Spreadsheets (EXCEL) are good at this
- The result is a table! (Completeness of rel. model!)

# ROLLUP alla IBM UDB

```
SELECT Country, Region, City, sum(price*vol)
FROM    Orders a, PoS f
WHERE  a.pos = f.name
GROUP BY  ROLLUP(Country, Region, City)
ORDER BY Country, Region, City;
```

Also works for other aggregate functions; e.g., avg().

# Result of ROLLUP Operator

| | | | |
|---|---|---|---|
| D | North | Köln | 1000 |
| D | North | (null) | 1000 |
| D | South | Mainz | 3000 |
| D | South | München | 200 |
| D | South | (null) | 3200 |
| D | (null) | (null) | 4200 |

# Summarizability (Unit)

- Legal Query

```
SELECT product, customer, unit, sum(volume)
FROM Order
GROUP BY product, customer, unit;
```

- Legal Query (product -> unit)

```
SELECT product, customer, sum(volume)
FROM Order
GROUP BY product, customer;
```

- Illegal Query (add „kg“ to „m“)!!!

```
SELECT customer, sum(volume)
FROM Order
GROUP BY customer;
```

# Summarizability (de-normalized data)

| *Region* | *Customer* | *Product* | *Volume* | *Populat.* |
|---|---|---|---|---|
| South | Heinz | Balls | 1000 | 3 Mio. |
| South | Heinz | Nets | 500 | 3 Mio. |
| South | Mary | Balls | 800 | 3 Mio. |
| South | Mary | Nets | 700 | 3 Mio. |
| North | Heinz | Balls | 1000 | 20 Mio. |
| North | Heinz | Nets | 500 | 20 Mio. |
| North | Mary | Balls | 800 | 20 Mio. |
| North | Mary | Nets | 700 | 20 Mio. |

Customer, Product -> Revenue

Region -> Population

# Summarizability (de-normalized data)

- What is the result of the following query?

```
SELECT region, customer, product, sum(volume)
FROM Order
GROUP BY ROLLUP(region, customer, product);
```

- All off-the-shelf databases get this wrong!
- Problem:  Total Revenue is 3000 (not 6000!)
- BI Tools get it right: keep track of functional dependencies
- Problem arises if reports involve several unrelated measures.