Database and Information Systems

11. Deductive Databases
12. Data Warehouses and OLAP
13. Index Structures for Similarity Queries
14. Data Mining
15. Semi-Structured Data
16. Document Retrieval
17. Web Mining
18. Content Extraction
19. Multimedia Data

Data Warehouses and OLAP

- Decision support systems
- Data Warehouses
- Dimensional Modelling
- Online Analytical Processing

Data Warehouses and OLAP

Readings:

Heuer, Andreas; Saake, Gunter: Datenbanken - Konzepte und Sprachen, 2nd edition, Thomson Int., 2000, Section 4.6, 10.2.3.
Conolly, Thomas; Begg, Carolyn: Database Systems - A Practical Approach to Design, Implementation, and Management, 3rd edition, Addison Wesley, 2002, Chapter 30-32.
Kifer, Michael; Bernstein, Arthur; Lewis Philip M.: Database Systems - An Application-Oriented Approach. 2nd edition. Pearson Education 2005, Chapter 15.
Dunham, Margaret H.: Data Mining - Introductory and Advanced Topics. Pearson Education, 2003, Chapter 2.

Data Warehouses and OLAP

- Decision support systems
- Data Warehouses
- Dimensional Modelling
- Online Analytical Processing

Decision Support Systems

- also: executive information systems, executive support systems
- purpose:
  assisting managers in making decisions and solving problems
- traditional databases vs. decision support systems?

Decision Support Systems

- traditional databases:
  - task specific collections of operational data
    - billing
    - inventory control
    - payroll
    - procurement
    - manufacturing support
  - typical services
    - online transaction processing
    - batch reporting

Decision Support Systems

- decision support systems:
  - informational data for
    - strategic analysis
    - planning
    - forecasting
  - typical services
    - ad hoc queries
    - customized information
  - data are usually organized along dimensions
  - data warehouse technology is useful but not necessary

Data Warehouses and OLAP

- Decision support systems
- Data Warehouses
- Dimensional Modelling
- Online Analytical Processing

Data Warehouses

- set of data that supports decision support systems and is subject-oriented, integrated, time-variant, and non-volatile
- single repository for corporate-wide data
  - including historical ones
- William Inmom (1995)
  first used: early 1980ies

Data Warehouses

- task specific:
  - traditional databases: operational data for day-to-day needs
    - inventory control, payroll, manufacturing support
    - online transaction processing and batch reporting
  - data warehouse: informational data supporting other functions
    - strategic analysis, planning, forecasting
  - operational data needs to be transformed into informational ones
  - relevant information is precomputed in advance of queries

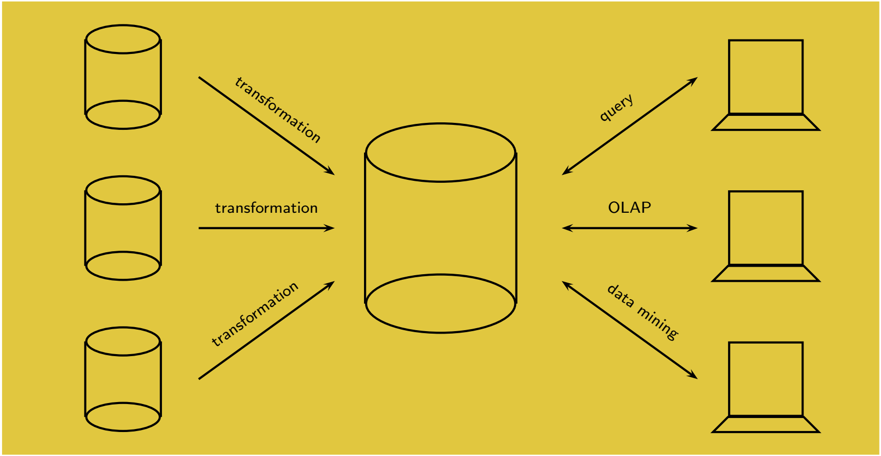
Data Warehouses

- data warehousing is an active approach

active	passive
anticipation of queries	waiting for queries
"eager"	"lazy"
in advance	on demand

- components of a data warehouse
  - data migration tools
  - the data warehouse
  - access tools

Data Warehouses



[Berson, Smith 1997]

Data Migration

- ETL: extract, transform, load
  - minimizing latency

- extraction
  - selection of relevant data
  - data profiling, estimation of data quality
  - periodicity: periodic, event-driven, query-driven

- transformation:
  - converting heterogeneous sources into one common schema

- loading
  - combination of snapshots into a historical data base

Data Migration

- syntactic transformations
  - reformatting (date, time, ...)
  - different data types
- semantic transformations
  - encoding conventions
    - code mapping (countries, gender, ...)
    - time zone mapping
    - units of measurement
  - schema mapping
  - harmonization of terminology

Data Migration

- semantic transformations (cont.)
  - inserting derived data
    - relative instead of absolute time information
      e.g. age $\rightarrow$ day of birth
  - cleansing
    - handling of missing and erroneous data
    - elimination of duplicates
  - summarization
    - aggregation of data

Data Warehouses

	Operational Data	Data Warehouse
Application	OLTP	OLAP
Usage	Standard Workflow	ad hoc Queries
Temporal charact.	Snapshot	Historical
Modification	Dynamic	Static
Orientation	Application	Business Enterprise
Data	Operational Values	Integrated
Size	Gigabits	Terabits
Level	Detailed	Summarized
Access	Frequently	Less Frequently
Response	Few Seconds	Minutes
Data schema	Relational	Star / Snowflake

Data Warehouses

- problems in setting up a data warehouse (Greenfield 1996)
  - underestimation of resources for data loading
  - hidden problems with the source systems (e.g. missing data)
  - required data not captured
  - increased end-user demands
  - data homogenization (differences between different source systems are lost)
  - high resource demands
  - conflicts between owners of data
  - high maintenance requirements
  - long-duration project
  - complexity of integration (different requirements, different tools, ...)

Data Warehouses

- increased complexity
- longer lifespan compared to operational data
- data need not be consistent
- derived concepts:
  - data mart: subset of a data warehouse
    - departmental, regional, functional level
  - virtual warehouse: implemented as a view on the operational data

Data Warehouses

- performance improvement
  - summarization:
    - precomputation during data transformation
    - 20 ... 100% increase in storage space $\rightarrow$ 2 .. 10 times speedup [Singh 1998]
- denormalization:
  - reduction of joins
  - update anomalies are not a problem

Data Warehouses

- additional meta data requirements
  - origin of the data
  - changes made to the data during upload
  - aggregation procedures
  - table partitions and partion keys
  - profiling: typical queries for different users and user groups
  - user-group specific meanings of attributes and changes in meaning
  - synchronizing meta data between different systems and tools

Data Warehouses and OLAP

- Decision support systems
- Data Warehouses
- Dimensional Modelling
- Online Analytical Processing

Dimensional Modelling

- analysis-oriented way to represent and query data in a database
  - to be used in decision support systems
- special emphasis: efficient access to dimension-based data
- dimension: collection of logically related attributes
  - regions
  - time intervals
  - product classes
  - organisational hierarchies

  each viewed as an axis for modelling

Example

ProductID	LocationID	Date	Quantity	UnitPrice
176	London	2004-01-05	5	2900
352	Madrid	2004-01-07	9	5400
176	Prague	2004-01-12	3	2500
210	Manchester	2004-01-19	4	1500
176	Munich	2004-01-28	1	2800
176	Munich	2004-01-28	9	2700
317	Dresden	2004-02-04	3	4600
289	Milan	2004-02-06	100	990

Dimensional Modelling

- granularity:
  - unit of measurement, can vary depending on purpose
    - year, quarter, month, decade, week, day, hour, minute, second

  $\rightarrow$ granularity levels of a dimension
- changing the level of granularity:
  - roll up, drill down
- granularity problem:
  - selection of keys depends on the level of granularity
    c.f. `176/Munich/2004-01-28`

Dimensional Modelling

- target data:
    - usually numeric values for statistical purposes
        - organized along dimensions
        - need to be stored and queried on all levels
        - can be aggregated

    $\rightarrow$ facts

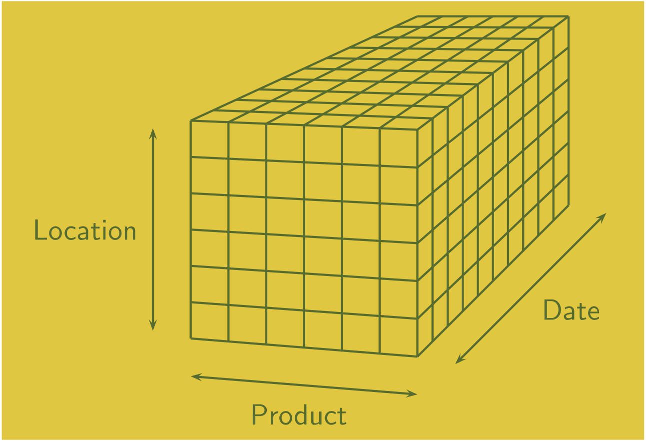
    $\rightarrow$ fact table

Example

ProductID	LocationID	Date	Quantity	UnitPrice
176	London	2004-01-05	5	2900
352	Madrid	2004-01-07	9	5400
176	Prague	2004-01-12	3	2500
210	Manchester	2004-01-19	4	1500
176	Munich	2004-01-28	1	2800
176	Munich	2004-01-28	9	2700
317	Dresden	2004-02-04	3	4600
289	Milan	2004-02-06	100	990

Dimensional Modelling

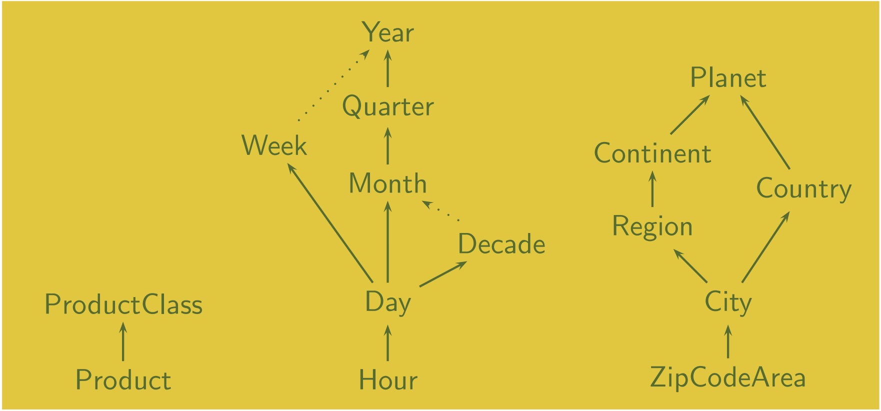
- the data cube



- fast access required
- but possibly extremely sparse

Dimensional Modelling

- dimensional hierarchy:
  - partial ordering of granularity levels according to an inclusion relationship ($<$)



Dimensional Modelling

- aggregation: If $X < Y$ then there is an aggregate type of relationship among the facts, e.g.

$$\text{quantity}(\text{product\_class}) = \sum_{\text{product}_i \in \text{product\_type}} \text{quantity}(\text{product}_i)$$

$$\text{quantity}(\text{month}) = \sum_{\text{day}_i \in \text{month}} \text{quantity}(\text{day}_i)$$

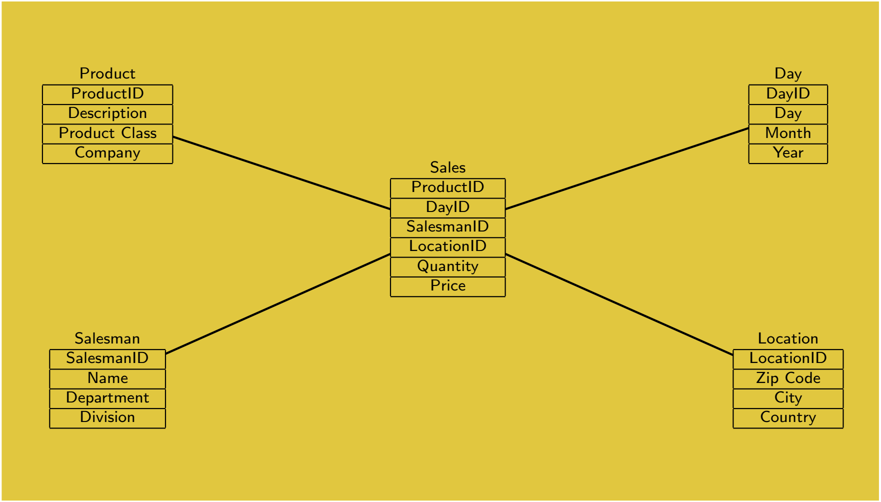
- other aggregate operations: average, maximum, minimum
  - if two levels are on one and the same path, aggregation is additive
  - non-additive dimensions require a more complicated roll up/drill down

Dimensional Modelling

- DB schemas for multidimensional data
  - star schema
  - snowflake schema
  - fact constellation schema

- center: fact tables (major tables)
- periphery: dimension tables (minor tables)

Star Schema


Product
ProductID
Description
Product Class
Company
Day
DayID
Day
Month
Year
Sales
ProductID
DayID
SalesmanID
LocationID
Quantity
Price
Salesman
SalesmanID
Name
Department
Division
Location
LocationID
Zip Code
City
Country

Star Schema

- several fact tables are possible, dimension tables might point to other dimension tables
- fact table can be indexed, but amount of data is usually huge
- aggregation requirements must be supported efficiently

Star schema

- four storage models for dimension tables [Purdy/Brobst 1999]
  - flattened
  - normalized
  - expanded
  - levelized

Flattened Star Schema

- store facts only at the lowest level of granularity
- key: all level attributes for the dimensions

```
sales(ProductID,DayID,SalesmanID,LocationID,
      Quantity,UnitPrice)
product(ProductID,Description,ProductClass,
      Company)
day(DayID,Day,Month,Year)
salesman(SalesmanID,Name,Department,Division)
location(LocationID,ZipCode,City,Country)
```

  - roll up: sum aggregation
  - problems:
    - time requirements
    - redundancies in the dimension tables

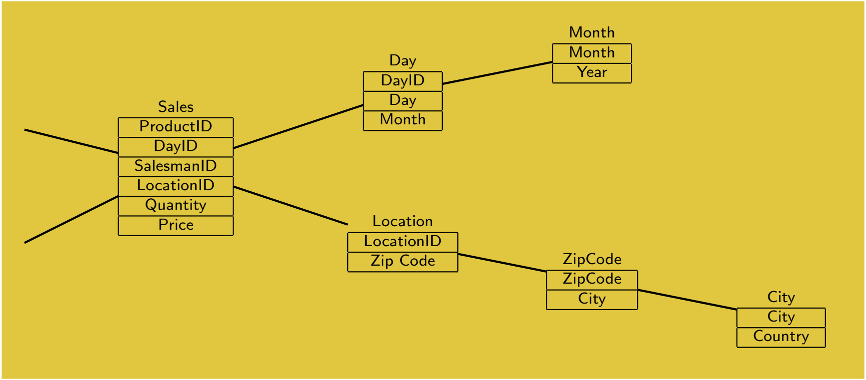
Normalized Star Schema

- dependencies resolved

```
sales(ProductID,DayID,SalesmanID,LocationID,...)
product(ProductID,Description,ProductClass,...)
day(DayID,Day,Month)
month(Month,Year)
salesman(SalesmanID,Name,Department)
department(Department,Division)
location(LocationID,ZipCode)
zipcode(ZipCode,City)
city(City,Country)
```

  - duplication/redundancy is removed

Normalized Star Schema



- expensive access due to joins in the dimension tables
  $\rightarrow$ denormalization

Expanded Star Schema

- denormalization of the dimension tables
- store dimensional data for all levels of granularity

```
sales(ProductID,DayID,SalesmanID,LocationID,...)
product(ProductID,Description,ProductClass,...)
day(DayID,Month,Quarter,Year)
month(Month,Quarter,Year)
quarter(Quarter,Year)
salesman(SalesmanID,Department,Division)
department(Department,Division)
location(LocationID,ZipCode,City,Country)
zipcode(ZipCode,City,Country)
city(City,Country)
```

Expanded Star Schema

- even more space expensive than the flattened schema
- substantial amount of redundancy
  - $\rightarrow$ transformation from operational data!
- fast access
  - no join operations for the dimension tables

Levelized Star Schema

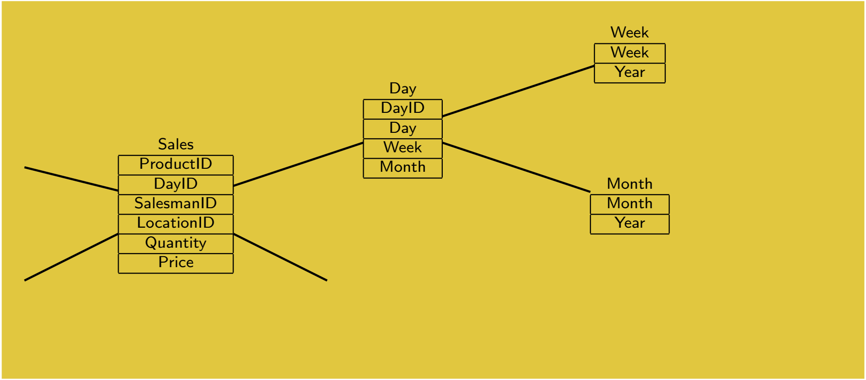
- denormalization of the fact table
- aggregation is precomputed for all granularity levels
- extend dimensional data to also include a level indicator

```
sales(ProductID,TimeID,AgentID,LocationID,...)
product(ProductID,Description,ProductClass,...)
day(TimeID,Day,Month,Quarter,Year,LevelID)
salesman(AgentID,Agent,Dpmt,Division,LevelID)
location(LocationID,ZipCode,City,Country,LevelID)
```

  - one tuple for each instance of each level in the dimension
  - massive redundancy
    - $\rightarrow$ transformation from operational data!
- fast access
  - no join operations for dimension access, no aggregation for roll up

Snowflake Schema

- generalization of the normalized star schema
- aggregation hierarchy is directly represented in the DB schema



normalized star schema is a special case

Dimensional Modelling

- Indexing:
  - bitmap indices: each value in each domain is represented by a bit
    - $\rightarrow$ one bit vector per tuple
    - size of the vector: $\sum_i |Dom(A_i)|$
    - supports efficient join and aggregation through arithmetic operations
    - efficiency gains for attributes with few values
  - join indices: precomputation of tuples that join together
    - e.g. fact and dimension table
  - B-trees
    - more efficient if number of values is high

Data Warehouses and OLAP

- Decision support systems
- Dimensional Modelling
- Data Warehouses
- Online Analytical Processing

OLAP

- OnLine Analytical Processing
- Codd 1993
- no clear definition
- mixture of goals and implementation issues

OLAP

- Codd's rules
  - multi-dimensional conceptual view
  - transparency
  - accessibility
  - consistent reporting performance
  - dynamic sparse matrix handling
  - multi-user support
  - unrestricted cross-dimensional operations
  - intuitive data manipulation
  - flexible reporting
  - unlimited dimensions and aggregation levels

OLAP

- OLAP council white paper
  - multidimensional view of data
  - calculation-intensive capabilities (related to aggregation functions)
  - time intelligence
- FASMI: Fast Analysis of Shared Multidimensional Information
- OLAP is an application view, not a data structure or a schema

OLAP Tools

- MOLAP: multidimensional OLAP
  - modelled, viewed and physically stored in a multidimensional database (MDD)
  - n-dimensional array
  - cube view is stored directly
    + ad-hoc products (no SQL limitations)
    + good mapping with data
    + good performance for small cubes
    - no standard (API may change over time)
    - no common query language
    - storage limitations

OLAP Tools

- ROLAP: relational OLAP
  - data stored in a relational database
  - ROLAP server creates the multidimensional view
    - \+ support of RDBMS
    - – relation has no inherent order, array has
    - – virtual cube + meta data
    - – time requirements (joins)
    - – higher storage requirements (for fact table)

$$
\begin{aligned}
|\text{fact(MOLAP)}| &= |d_1| \cdot ... \cdot |d_n| \cdot |value| \\
|\text{fact(ROLAP)}| &= |d_1| \cdot ... \cdot |d_n| \cdot |[k_1, ..., k_n, value]| \\
&= (n+1) \cdot |\text{fact(MOLAP)}|
\end{aligned}
$$

OLAP

- HOLAP: hybrid OLAP
  - combination of MOLAP and ROLAP
  - full data repository as a ROLAP database
  - partitioning: data subsets are downloaded to a MOLAP workplace
    - data cube tailored to specific analysis needs
    - easier access to less complex data
    - efficiency advantages of MOLAP are optimally used

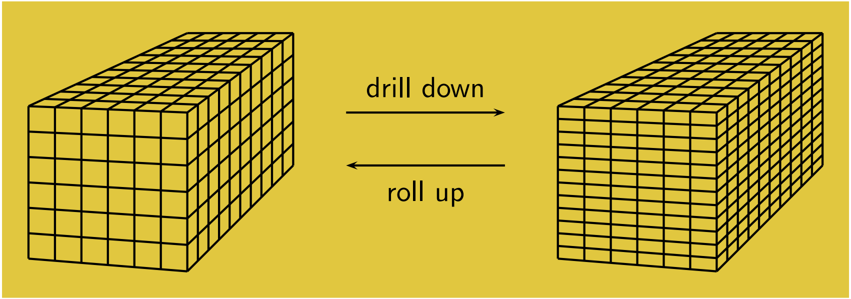
OLAP

- case study from Colliat (Sahuguet 1997)

	ROLAP	MOLAP
disk space (Go)	17	10
fast query	240s	1s
complex query	237h	2h

OLAP Operations

- drill down: zooming into a finer granularity level
- roll up: zooming out to a more coarse granularity level (aggregation)



OLAP Operations

- cube: precomputation of a full data cube
  - generalized roll up
  - n attributes
    $\rightarrow$ aggregated values for 2^n attribute combinations
    `group by` –;
    `group by` a_1;
    `group by` a_2;
    `group by` a_3;
    `group by` a_1, a_2;
    `group by` a_2, a_3;
    `group by` a_1, a_3;
    `group by` a_1, a_2, a_3;

OLAP Operations

- cube corresponds to a (n-dimensional) cross tabulation

	small	medium	large	total
budget	24	31	12	67
premium	11	15	17	43
total	35	46	29	100

OLAP Operations

- relational representation of the cube

quality	size	amount
budget	small	24
budget	medium	31
budget	large	12
budget	all	67
premium	small	11
premium	medium	15
premium	large	17
premium	all	43
all	small	35
all	medium	46
all	large	29
all	all	100

OLAP Operations

- slice: dimension reduction by value selection
- rotate: rotation of a cube
  - only for navigation purposes
- window: range query
- ranking: sorting fact values along a dimension
- visualisation (playing around with data)

OLAP Extentions to SQL (RISQL)

- Decode: replace internal codes by readable versions
- Cume: computes a running (or cumulative) total of an attribute
- MovingAvg(n): computes the moving average of an attribute with a window size of n
- MovingSum(n): computes the moving sum of an attribute with a window size of n
- Rank ... When: compute the ranking of the top n or bottom n tuples according to the values of an attribute
- RatioToReport: percentage of an attribute value with respect to the total for that attribute
- Tertile: three valued binning (high, medium, low) with respect to the values of an attribute
- CreateMacro: define a parameterized macro for repeated use