Friedrich-Alexander-Universität
Technische Fakultät

5. Data Warehousing and Online Analytical Processing

Knowledge Discovery in Databases

Dominik Probst, `Dominik.probst@fau.de`
Chair of Computer Science 6 (Data Management), Friedrich-Alexander-University Erlangen-Nürnberg
Summer semester 2024

Outline

1. Data Warehouse: Basic Concepts
2. Data Warehouse Modeling: Data Cube and OLAP
3. Data Warehouse Design and Usage
4. Data Warehouse and Data Mining
5. Summary
6. Appendix

Data Warehouse: Basic Concepts

What is a Data Warehouse?

William “Bill” H. Inmon is commonly referred to as the “father of the data warehouse”.

Data Warehouse

A data warehouse is a **subject-oriented**, **integrated**, **time-variant**, and **nonvolatile** collection of data in support of management’s decision-making process.[1] Common abbreviations: DW or DWH.

Other definitions exist:

- “A data warehouse is a system that extracts, cleans, conforms, and delivers source data into a dimensional data store and then supports and implements querying and analysis for the purpose of decision making.”[2]
- A **decision-support** database that is **maintained separately** from the organization’s operational database.
- Supports information processing by providing a solid platform of **consolidated, historical data** for analysis.

Data warehousing: The process of constructing and using data warehouses.

[1] W. H. Inmon, *Building the Data Warehouse*. Wiley, 2005, 4th edition, ISBN: 978-076459-944-6
[2] R. Kimball and J. Caserta, *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Wiley, 2004, ISBN: 978-0764567575

Data Warehouse – Subject-oriented

- **Organized around major subjects.** Such as customer, product, sales.
- **Focusing on the modeling and analysis of data for decision makers.** Not on daily operations or transaction processing.
- **Provide a simple and concise view around particular subject issues.** By excluding data that are not useful in the decision-support process.

Data Warehouse – Integrated

- **Constructed by integrating multiple heterogeneous data sources.**
  - Relational databases, flat files, online transaction records, . . .
- **Data-cleaning and data-integration techniques are applied.**
  - Ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources.
  - E.g., hotel price: currency, tax, breakfast covered.
  - When data is moved to the data warehouse, it is converted.
  - ETL – Extract, Transform, Load.

Data Warehouse – Time-Variant

- **The time horizon for a data warehouse is significantly longer than that of operational systems.**
  - Operational database: current-value data.
  - Data warehouse: provide information from a historical perspective, e.g. past 5 — 10 years.
- **Every key structure in the data warehouse contains an element of time, explicitly or implicitly.**
- The key of operational data may or may not contain a "time element."

Data Warehouse – Nonvolatile

- **A physically separate store of data.**
  - Transformed from the operational environment.
  - By copying.
- **No operational update of data:**
  - Hence, does not require transaction processing, i.e. no logging, recovery, concurrency control, etc.
  - Requires only three operations:
    1. Initial loading of data.
    2. Refresh (update, often periodically, e.g. over night).
    3. Access of data.

Data Warehouse Usage

Three kinds of data warehouse applications.

1. **Information processing.**
   - Supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs.
2. **Analytical processing.**
   - Multidimensional analysis of data warehouse data.
   - Supports basic OLAP operations such as slicing, dicing, drilling, and pivoting.
3. **Data mining.**
   - Knowledge discovery from hidden patterns.
   - Supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.

Online Transaction Processing vs. Online Analytical Processing

	OLTP	**OLAP**
Users	clerk, IT professional	knowledge worker
Function	day-to-day operations	decision support
DB Design	application-oriented	decision support
Data	current, up-to-date; detailed, flat relational; isolated	historical; summarized, multidimensional, integrated, consolidated
Usage	repetitive	ad-hoc
Access	read/write; index/hash on primary key	lots of scans
Unit of Work	short, simple transaction	complex query
#-Records Accessed	10	10^6
#-Users	1000	100
DB Size	100 MB to GB	100 GB to TB
Quantification	transaction throughput	query throughput, response

OLTP = Online Transaction Processing, OLAP = Online Analytical Processing

Why a Separate Data Warehouse?

High performance for both systems:

- **DBMS**: tuned for OLTP; Access methods, indexing concurrency control, recovery.
- **Data Warehouse**: tuned for OLAP; Complex OLAP queries, multidimensional view, consolidation.

Different functions and different data:

- *Missing data (DBMS):* Decision support (DS) requires **historical data** which operational DBs do not typically maintain.
- *Data consolidation (warehouse):* DS requires **consolidation** (aggregation, summarization) of data from heterogeneous sources.
- *Data quality (warehouse):* Different sources typically use inconsistent data representations, codes and formats which have to be reconciled.

Note

There are more and more systems which perform OLAP analysis directly on relational databases.

Three Data Warehouse Models

1. **Enterprise Warehouse:**
   - Collects all of the information about subjects spanning the entire organization.
2. **Data Mart:**
   - A **subset** of corporate-wide data that is of value to a **specific group of users**.
   - Typically contains (highly) summarized data.
   - Independent vs. dependent (directly from warehouse) data mart.
3. **Virtual Warehouse:**
   - Also known as *data virtualization*.
   - A set of **views** over operational databases.

     As an *operational database* are all data sources considered that summarize, serve, and access up-to-date and real-time data. Generally, these are OLTP systems that provide ACID properties. These systems include, but are not limited to relational databases, NoSQL databases, but also unstructured data.
   - Only some of the possible summary views may be materialized.

Extract, Transform, and Load (ETL)

- **Extract** Data:
  - Get data from multiple, heterogeneous, and external sources.
- **Clean** Data:
  - Detect errors in the data and rectify them if possible.
- **Transform** Data:
  - Convert data from legacy or host format to warehouse format.
- **Load** Data:
  - Sort, summarize, consolidate, compute views, check integrity, and build indexes and partitions.
- **Refresh** Data:
  - Propagate only the updates from the data sources to the warehouse.

Metadata[3] Repository

Generally speaking:

Metadata

Data about data.

Three types: *business*, *process execution*, and *technical* metadata.

Business Metadata

- Business terms and definitions.
- Logical data mapping.
- Data ownership.
- Charging policies.

Process Execution Metadata

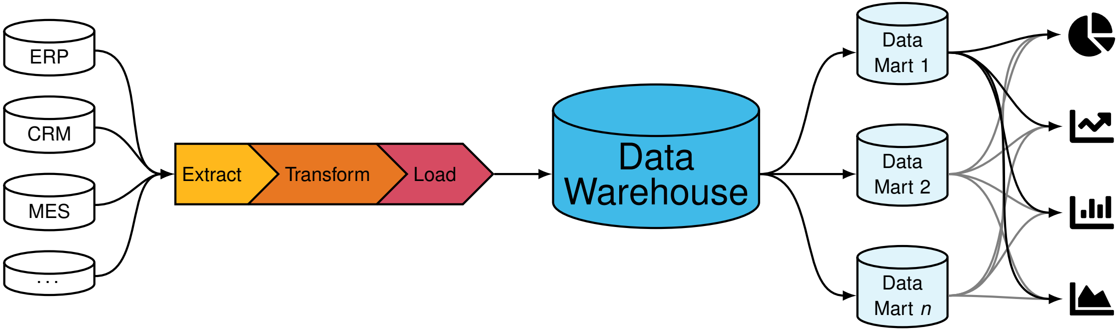
- Data acquisition schedule.
- Data-cleaning specifications.
- Aggregate specifications.
- Slowly changing dimensions policies.
- Duration of ETL, rows rejected and successful.

Technical Metadata

- Table structures and table attributes.
- Derived data definitions.
- Results from data profiling.
- Data lineage.

[3] C.f. chapter 9 of R. Kimball and J. Caserta, *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Wiley, 2004, ISBN: 978-0764567575

Data Warehouse Reference Overview

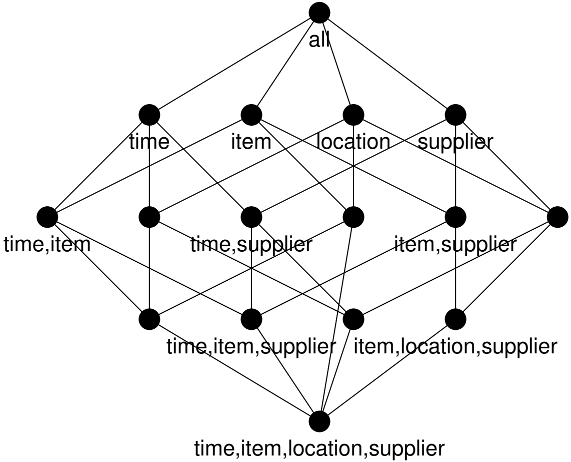


Data Warehouse Modeling: Data Cube and OLAP

From Tables and Spreadsheets to Data Cubes

- Data warehouse is based on a **multidimensional data model** which views data in the form of a **data cube**.
- A **data cube** allows data (example: sales) to be modeled and viewed in multiple dimensions.
  - Defined by dimension and facts.
  - **Dimension tables:** such as: `item (item_name, brand, type)`, or: `time (day, week, month, quarter, year)`.
  - **Fact table:** Contains **measures** (such as `dollars_sold`) and references (`foreign keys`) to each of the related dimension tables.
- ***n*-dimensional base cube.**
  - Called a base cuboid in data warehousing literature.
- **Top most** 0**-dimensional cuboid.**
  - Holds the highest-level of summarization.
  - Called the apex cuboid.
- **Lattice of cuboids.** (Forms a data cube)

Cube: A Lattice of Cuboids



0-dimensional (apex) cuboid

1-dimensional cuboid

2-dimensional cuboid

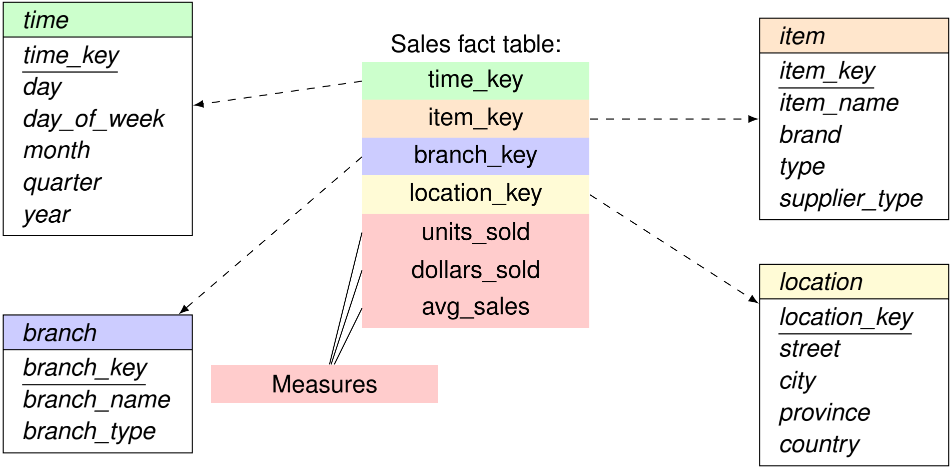
3-dimensional cuboid

4-dimensional (base) cuboid

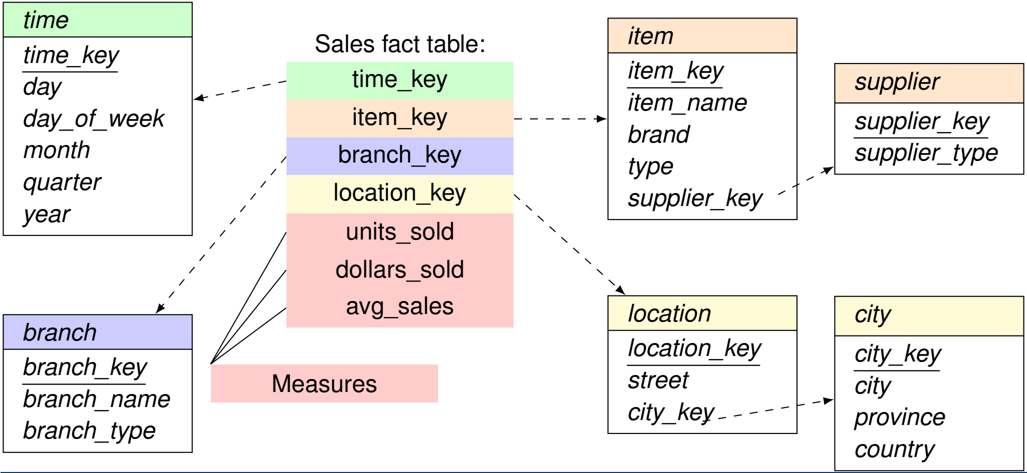
Conceptual Modeling of Data Warehouses

1. **Star schema:**.
   - A fact table in the middle connected to a set of dimension tables.
2. **Snowflake schema:**.
   - A refinement of the star schema where some dimensional hierarchies are **normalized** into a set of smaller dimension tables, forming a shape similar to a snowflake.
   - I. e. dimension tables of a star schema are split into multiple (dimension) tables along their respective granularity level, but not split/normalized for every granularity.
3. **Fact constellations:**.
   - Multiple fact tables sharing dimension tables, viewed as a collection of stars, therefore called **galaxy schema** or fact constellation.

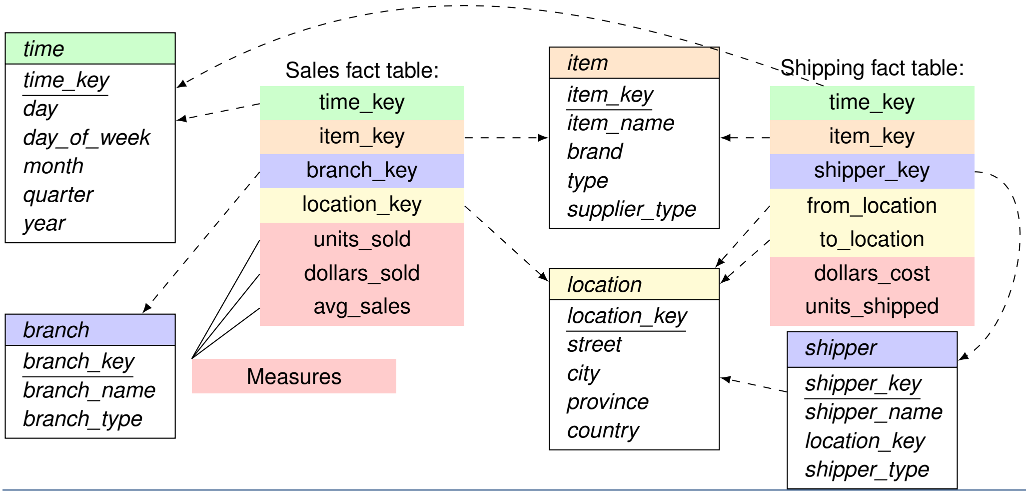
Example of a Star Schema



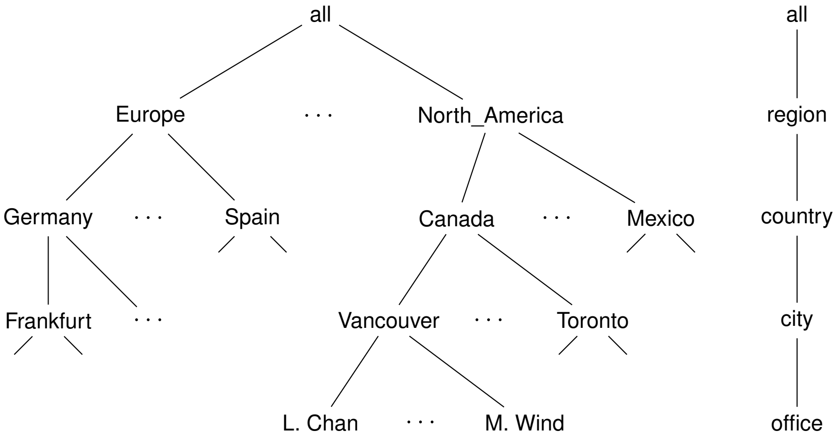
Example of Snowflake Schema



Example of Fact Constellation



A Concept Hierarchy: Dimension (Location)



Data-Cube Measures

Data-Cube Measure

A *data-cube measure*, also called a *fact*, is a numeric function that can be evaluated at each point in the data cube space.

Three Categories:

1. **Distributive:**
   - If the result derived by applying the function to the n aggregate values obtained for n partitions of the dataset is the same as that derived by applying the function on all the data without partitioning.
     E.g. `COUNT`, `SUM`, `MIN`, `MAX`.
2. **Algebraic:**
   - If it can be computed by an algebraic function with M arguments, each of which is obtained by applying a distributive aggregate function.
     E.g. `AVG`, $\mathtt{MIN}_N$, `STD`.
3. **Holistic:**
   - If there is no constant bound on the storage size needed to describe a subaggregate.
     E.g. `MEDIAN`, `MODE`, `RANK`.

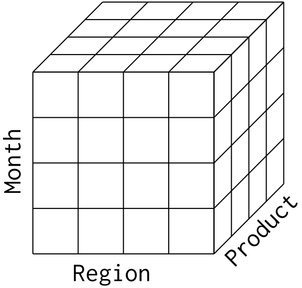
Aggregation Type

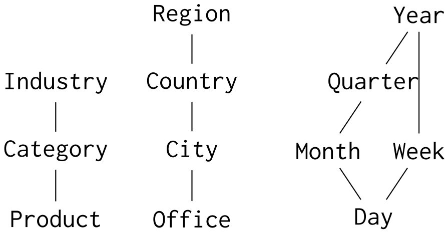
- **Non-trivial property.**
  - Next to name and value range.
- **Defines the set of aggregation operations that can be executed on a measure (a fact).**
- **STOCK:** Measure at a specific point in time.
  - Aggregated as desired.
  - E.g. sales turnover, quantity of an item ordered per day.
- **FLOW:** Measure over a period of time.
  - Aggregated as desired, but temporal aggregation not permitted.
  - E.g. total stock and total inventory. Yet, summarization of article stock over multiple days makes no sense!
- **VPU (Value per Unit):** Measures that cannot be summed.
  - E.g. unit price, tax rates, exchange rates.
- **Always applicable: `MIN`, `MAX` and `AVG`.**

Aggregation Type

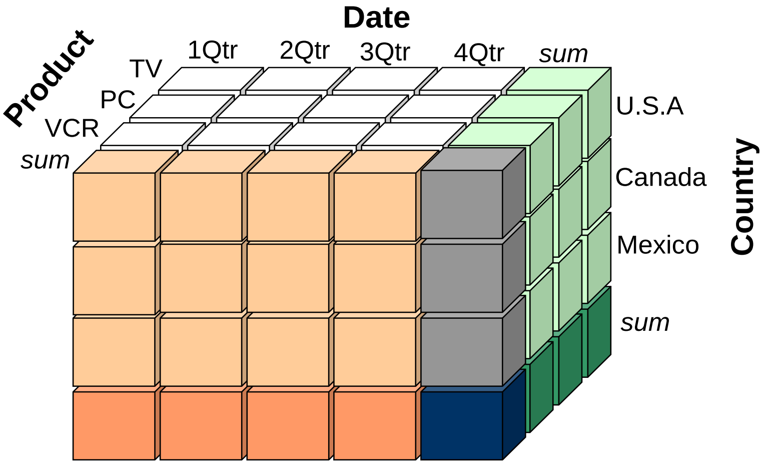
Sales volume as a function of product, month, and region.

- Dimensions: `Product`, `Location`, `Time`.
- Hierarchical summarization paths.

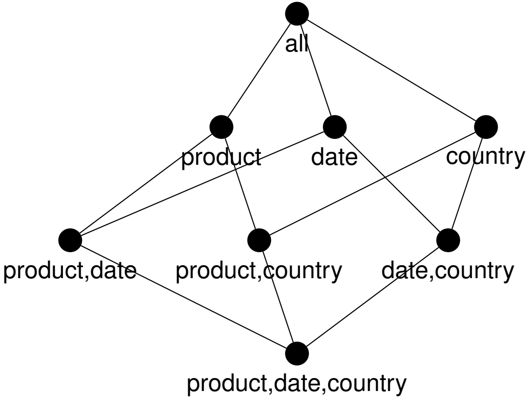




Data Cube Sample



Cuboids Corresponding to the Cube



0-dimensional (apex) cuboid

1-dimensional cuboid

2-dimensional cuboid

3-dimensional (base) cuboid

Typical OLAP Operations

- **Roll up (drill up): summarize data.**
  - By climbing up hierarchy or by dimension reduction.
- **Drill down (roll down): reverse of roll up.**
  - From higher-level summary to lower-level summary or detailed data, or introducing new dimensions.
- **Slice and dice: project and select.**
- **Pivot (rotate):**
  - Reorient the cube, visualization, 3D to series of 2D planes.
- **Other operations:**
  - **Drill across**: involving (across) more than one fact table.
  - **Drill through**: through the bottom level of the cube
    to its back-end relational tables (using SQL).

Data Warehouse Design and Usage

Design of Data Warehouse: A Business-analysis Framework

Four views regarding the design of a data warehouse:

- **Top-down view:**
  - Allows selection of the relevant information necessary for the data warehouse.
- **Data-source view:**
  - Exposes the information being captured, stored, and managed by operational systems.
- **Data warehouse view:**
  - Consists of fact tables and dimension tables.
- **Business-query view:**
  - Sees the perspectives of data in the warehouse from the view of the end-user.

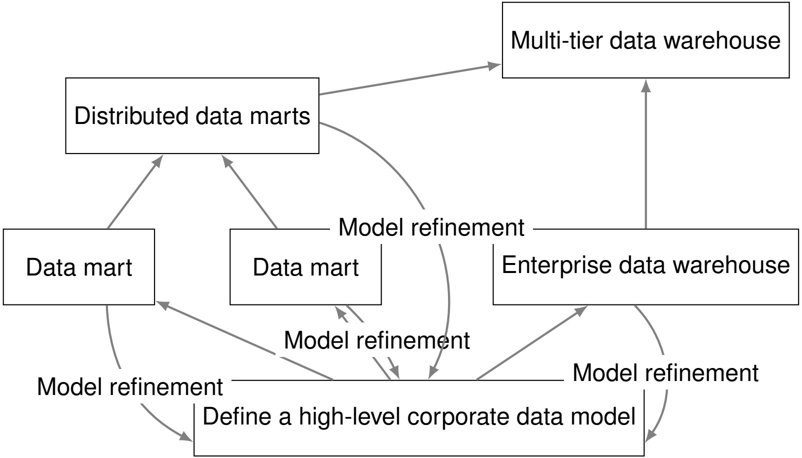
Data Warehouse Design Process

- **Top-down, bottom-up approaches or a combination of both:**
  - **Top-down:** starts with overall design and planning (mature).
  - **Bottom-up:** starts with experiments and prototypes (rapid).
- **From software-engineering point of view:**
  - **Waterfall:** structured and systematic analysis at each step before proceeding to the next.
  - **Spiral:** rapid generation of increasingly functional systems, short turn-around time.
- **Typical Data warehouse design process:**
  1. Choose a **business process** to model, e.g., orders, invoices, etc.
  2. Choose a **grain** (atomic level of data) of the business process.
  3. Choose **dimensions** that will apply to each fact-table record.
  4. Choose a **measure** that will populate each fact-table record.

DWH Construction is No Easy Feat

Construction of a data warehouse is a difficult long-term task. It is absolutely necessary that its implementation scope is clearly defined at the beginning. Goals and tasks should be *SMART* (specific, measurable, achievable, relevant, and time-related).

Data Warehouse Development: A Recommended Approach



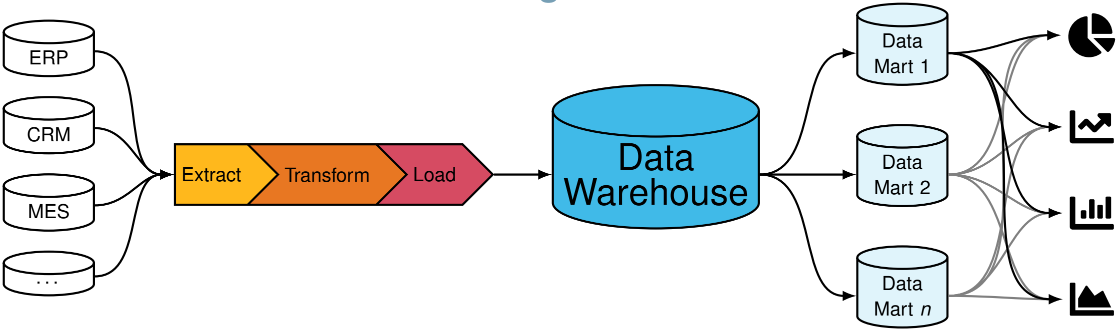
Data Warehouse and Data Mining

From Online Analytical Processing (OLAP) To Online Analytical Mining (OLAM)

Why online analytical mining?

- DW contains integrated, consistent, cleaned data.
- Available information-processing structure surrounding data warehouses.
  - ODBC, OLEDB, Web access, service facilities, reporting, and OLAP tools.
- OLAP-based exploratory data analysis.
  - Mining with drilling, dicing, pivoting, etc.
- Online selection of data-mining functions.
  - Integration and swapping of multiple mining functions, algorithms, and tasks.

Data Warehouse and Data Mining



- Data mining algorithms in transformation step: E.g. integrate articles from two systems that have different article group hierarchy. Goal: Map one article group hierarchy to the existing article group hierarchy.
- Frequent pattern mining and clustering in reporting: E.g. affinity analysis, revenue prediction, cluster customers and use this insight for a new marketing campaign.

Summary

Summary

- **Data warehousing: multi-dimensional model of data.**
  - A data cube consists of dimensions and measures.
  - Star schema, snowflake schema, fact constellations.
  - OLAP operations: drilling, rolling, slicing, dicing and pivoting.
- **Data warehouse architecture, design, and usage.**
  - Multi-tiered architecture.
  - Business-analysis design framework.
  - Information processing, analytical processing, data mining, OLAM (Online Analytical Mining).

Interested in More Information?

Our appendix in this document covers:

- **Implementation: efficient computation of data cubes.**
  - Partial vs. full vs. no materialization.
  - Indexing OLAP data: Bitmap index and join index.
  - OLAP query processing.
  - OLAP servers: ROLAP, MOLAP, HOLAP.
- **Data generalization: attribute-oriented induction.**

Additionally, check out these books:

- R. Kimball and J. Caserta, *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. Wiley, 2004, ISBN: 978-0764567575
- W. H. Inmon, *Building the Data Warehouse*. Wiley, 2005, 4th edition, ISBN: 978-076459-944-6
- In German: A. Bauer and H. Günzel, *Data Warehouse Systeme – Architektur, Entwicklung, Anwendung*. dpunkt.verlag GmbH, 2004, 4th edition, ISBN: 978-3-89864-785-4

Any questions about this chapter?

Ask them now or ask them later in our forum:

StudOn Forum
`https://www.studon.fau.de/frm5699567.html`

Appendix

Data Warehouse Implementation

Efficient Data-Cube Computation

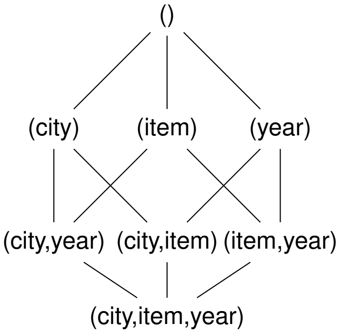
- **Data cube can be viewed as a lattice of cuboids.**
- The bottom-most cuboid is the base cuboid.
- The top-most cuboid (apex) contains only one cell.
- How many cuboids in an n-dimensional cube with L_i levels associated with dimension i?

$$T = \prod_{i=1}^{n} (L_i + 1). \tag{1}$$

- **Materialization of data cube.**
  - Materialize each (cuboid) (full materialization),
    none (no materialization), or some (partial materialization).
  - Selection of cuboids to materialize based on size, sharing, access frequency, etc.

The "Compute Cube" Operator

- **Cube definition and computation in DMQL:**
```
DEFINE CUBE sales [item, city, year]:
SUM (sales_in_dollars);
COMPUTE CUBE sales;
```
- **Transform it into an SQL-like language:**
with a new operator `CUBE BY` (Gray et al. 96).
```
SELECT item, city, year, SUM (amount)
FROM sales
CUBE BY item, city, year;
```
- **Need to compute the following `GROUP BY`s:**
```
(city, item, year),
(city, item), (city, year),
(item, year),
(city), (item), (year)
()
```



Indexing OLAP Data: Bitmap Index

- **Index on a particular column.**
- **Each value in the column has a bit vector: bit-op is fast.**
- **Length of bit vector: $\#$ of records in base table.**
- ***i*-th bit set, if *i*-th row of base table has value of bit vector.**
- **Not suitable for high-cardinality domains:**
  - A bit compression technique called Word-Aligned Hybrid (WAH) makes it work for high-cardinality domain as well [Wu et al., TODS'06].

Base table

Cust	Region	Type
$C1$	Asia	Retail
$C2$	Europe	Dealer
$C3$	Asia	Dealer
$C4$	America	Retail
$C5$	Europe	Dealer

Index on region

RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

Index on type

RecID	Retail	Dealer
1	1	0
2	0	1
3	0	1
4	1	0
5	0	1

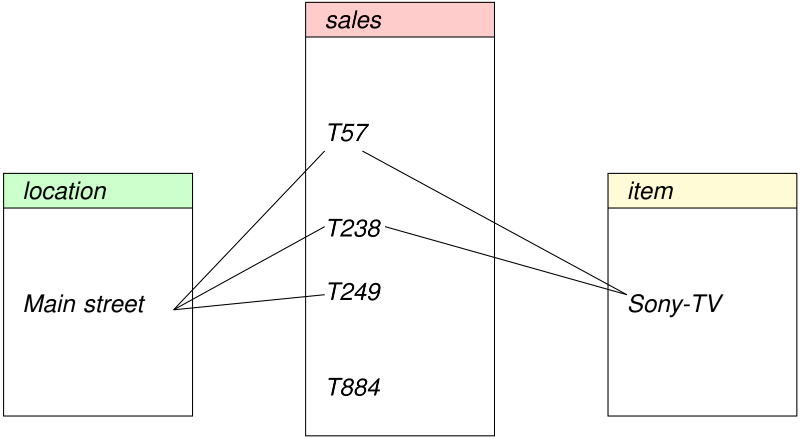
Indexing OLAP Data: Join Indices

- **Join index:**

$$\mathsf{JI}(\texttt{R-id}, \texttt{S-id}) \quad \text{where} \quad \mathsf{R}(\texttt{R-id}, \ldots) \bowtie \mathsf{S}(\texttt{S-id}, \ldots). \tag{2}$$

- **Traditional indices map the values to a list of record ids.**
  - Materializes relational join in JI-file and speeds it up.
- **In data warehouses, join index relates the values of the dimensions of a star schema to rows in the fact table.**
  - E.g. fact table: Sales and two dimensions location and item.
    - A join index on location maintains for each distinct location a list of `R-ids` of the tuples recording the sales in that location.
  - Join indices can span multiple dimensions.

Indexing OLAP data: Join Indices (Example)



Efficient Processing of OLAP Queries

- **Determine which operations should be performed on the available cuboids.**
  - Transform drill, roll, etc. into corresponding SQL and/or OLAP operations.
    E.g. `dice` = `selection` + `projection`.
- **Determine which materialized cuboid(s) should be selected for OLAP operation.**
  - Let the query to be processed be on `{brand, province_or_state}` with the condition `"year = 2004"`, and there are 4 materialized cuboids available:
    1) `year, item_name, city`.
    2) `year, brand, country`.
    3) `year, brand, province_or_state`.
    4) `item_name, province_or_state where year = 2004`.
  - Which should be selected to process the query?
- **Explore indexing structures and compressed vs. dense-array structures in MOLAP.**

OLAP Server Architectures

- **Relational OLAP (ROLAP).**
  - Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middleware.
  - Include optimization of DBMS backend, implementation of aggregation navigation logic, and additional tools and services.
  - Greater scalability.
- **Multidimensional OLAP (MOLAP).**
  - Sparse array-based multidimensional storage engine.
  - Fast indexing to pre-computed summarized data.
- **Hybrid OLAP (HOLAP) (e.g., Microsoft SQL-Server).**
  - Flexibility, e.g., low level: relational, high-level: array.
- **Specialized SQL servers (e.g., Redbricks).**
  - Specialized support for SQL queries over star/snowflake schemas.

Data Generalization by Attribute-Oriented Induction

Data Generalization

- **Summarize data:**
  - **By replacing relatively low-level values**
    e.g. numerical values for the attribute `age`
    with higher-level concepts
    e.g. `young`, `middle-aged` and `senior`.
  - **By reducing the number of dimensions**
    e.g. removing `birth_date` and `telephone_number`
    when summarizing the behavior of a group of students.
  - Describe concepts in concise and succinct terms at generalized (rather than low) levels of abstractions:
    - Facilitates users in examining the general behavior of the data.
    - Makes dimensions of a data cube easier to grasp.

Attribute-Oriented Induction

- **Proposed in 1989** (KDD'89 workshop).
- **Not confined to categorical data nor to particular measures.**
- **How is it done?**
  - Collect the **task-relevant data** (initial relation) using a relational database query.
  - Perform **generalization** by attribute removal or attribute generalization.
  - Apply **aggregation** by merging identical, generalized tuples and accumulating their respective counts.
  - Interaction with users for knowledge presentation.

Attribute-Oriented Induction: An Example

- **Example:** Describe general characteristics of graduate students in a university database.
- **Step 1:** Fetch relevant set of data using an SQL statement, e.g.

```
SELECT name, gender, major, birth_place, birth_date, residence, phone#, gpa
FROM student
WHERE student_status IN "Msc", "MBA", "PhD";
```

- **Step 2:** Perform attribute-oriented induction.
- **Step 3:** Present results in generalized-relation, cross-tab, or rule forms.

Class Characterization: An Initial Relation (I)

Name	Gender	Major	Birth place	Birth date	Residence	Phone number	GPA
Jim	M	CS	Vancouver, BC, Canada	08-21-76	3511 Main St., Richmond	687-4598	3.67
Scott Lachance	M	CS	Montreal, Que, Canada	28-07-75	345 1st Ave., Richmond	253-9106	3.70
Laura Lee	F	Physics	Seattle, WA, USA	25-08-70	125 Austin Ave., Burnaby	420-5232	3.83
Removed	Retained	Sci, Eng, Bus	Canada, Foreign	Age range	City	Removed	Excl, Vg,. . .

Class Characterization: Prime Generalized Relation (II)

Gender	Major	Birth region	Age range	Residence	GPA	Count
M	Science	Canada	20-25	Richmond	Very good	16
F	Science	Foreign	25-30	Burnaby	Excellent	22
...	...	...	...	...	...	...

Class Characterization: An Example (III)

Cross-table of birth region and gender:

	Canada	Foreign	Total
M	16	14	30
F	10	22	32
Total	26	36	62

Basic Principles of Attribute-Oriented Induction

- **Data focusing:**
  - Task-relevant data, including dimensions.
  - The result is the **initial relation**.
- **Attribute removal:**
  - Remove attribute A, if there is a large set of distinct values for A,
    but (1) there is no generalization operator on A,
    or (2) A's higher-level concepts are expressed in terms of other attributes.
- **Attribute generalization:**
  - If there is a large set of distinct values for A,
    and there exists a **set of generalization operators** on A,
    then select an operator and generalize A.
- **Attribute-threshold control:**
  - Typical 2-8, specified/default.
- **Generalized-relation-threshold control:**
  - Control the final relation/rule size.

Attribute-Oriented Induction: Basic Algorithm

- **InitialRel:**
  - Query processing of task-relevant data, deriving the initial relation.
- **PreGen:**
  - Based on the analysis of the number of distinct values in each attribute, determine generalization plan for each attribute: removal? Or how high to generalize?
- **PrimeGen:**
  - Based on the PreGen plan, perform generalization to the right level to derive a "prime generalized relation", accumulating the counts.
- **Presentation:**
  - User interaction:
    1. Adjust levels by drilling.
    2. Pivoting.
    3. Mapping into rules, cross tabs, visualization presentations.

Presentation of Generalized Results

- **Generalized relation:**
  - Relations where some or all attributes are generalized, with counts or other aggregation values accumulated.
- **Cross tabulation:**
  - Mapping results into cross-tabulation form (similar to contingency tables).
  - Visualization techniques: pie charts, bar charts, curves, cubes, and other visual forms.
- **Quantitative characteristic rules:**
  - Mapping generalized results into characteristic rules with quantitative information associated with it, e.g.

$$\text{grad}(x) \wedge \text{male}(x) \implies \tag{3}$$

$$\text{birth\_region}(x) = \text{"Canada"}[t : 53\%] \vee \tag{4}$$

$$\text{birth\_region}(x) = \text{"foreign"}[t : 47\%]. \tag{5}$$

Mining-Class Comparisons

- **Comparison: Comparing two or more classes.**
- **Method:**
  - Partition the set of relevant data into the **target class** and the **contrasting class(es)**.
  - Generalize both classes to the same high-level concepts (i.e. AOI).
    - Including aggregation.
  - Compare tuples with the same high-level concepts.
  - Present for each tuple its description and two measures.
    - Support – distribution within single class (counts, percentage).
    - Comparison – distribution between classes.
  - Highlight the tuples with strong discriminant features.
- **Relevance Analysis:**
  - Find attributes (features) which best distinguish different classes.

Attribute-Oriented Induction vs. Cube-based OLAP

- **Similarity:**
  - Data generalization.
  - Presentation of data summarization at multiple levels of abstraction.
  - Interactive drilling, pivoting, slicing and dicing.
- **Differences:**
  - OLAP has systematic preprocessing, query independent, and can drill down to rather low level.
  - AOI has automated desired-level allocation and may perform dimension-relevance analysis/ranking when there are many relevant dimensions.
  - AOI works on data which are not in relational forms.