

# Chapter IV: OLAP

## Knowledge Discovery in Databases

Luciano Melodia M.A.

Evolutionary Data Management, Friedrich-Alexander University Erlangen-Nürnberg

Summer semester 2021



## Chapter IV: Data warehousing and online analytical processing

### **Data warehouse: basic concepts.**

Data-warehouse modeling: data cube and OLAP.

Data-warehouse design and usage.

Data-warehouse Implementation.

Data generalization by attribute-oriented induction.

Summary.

## What is a data warehouse?

**Defined in many different ways, but not rigorously:**

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.

**Famous:**

*A data warehouse is a **subject-oriented, integrated, time-variant, and nonvolatile** collection of data in support of management's decision-making process.*

– W. H. Inmon.

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

## 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, etc.

When data is moved to the warehouse, it is converted.

ETL – Extraction, Transformation, Loading, see below.

## 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:

- Initial loading of data.

- Refresh (update, often periodically, e.g. over night).

- Access of data.

## OLTP vs. OLAP

	OLTP	OLAP
<b>users</b>	clerk, IT professional	knowledge worker
<b>function</b>	day-to-day operations	decision support
<b>DB design</b>	application-oriented	decision support
<b>data</b>	current, up-to-date; detailed, flat relational; isolated	historical; summarized, multidimensional, integrated, consolidated
<b>usage</b>	repetitive	ad-hoc
<b>access</b>	read/write; index/hash on primary key	lots of scans
<b>unit of work</b>	short, simple transaction	complex query
<b>#-records accessed</b>	10	$10^6$
<b>#-users</b>	1000	100
<b>DB size</b>	100 MB to GB	100 GB to TB
<b>quantification</b>	transaction throughput	query throughput, response



## Why a separate data warehouse?

### High performance for both systems:

**DBMS:** tuned for OLTP; Access methods, indexing concurrency control, recovery.

**Warehouse:** tuned for OLAP; Complex OLAP queries, multidimensional view, consolidation.

### Different functions and different data:

Missing data:

Decision support (DS) requires **historical data**  
which operational DBs do not typically maintain.

Data consolidation:

DS requires **consolidation** (aggregation, summarization)  
of data from heterogeneous sources.

Data quality:

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

### Enterprise Warehouse:

Collects all of the information about subjects spanning the entire organization.

### Data mart:

A **subset** of corporate-wide data that is of value to a **specific group of users**.  
Its scope is confined to specific, selected groups, such as marketing data mart.  
Independent vs. dependent (directly from warehouse) data mart.

### Virtual warehouse:

A set of **views** over operational databases.  
Only some of the possible summary views may be materialized.

## Extraction, transformation, and loading (ETL)

### Extraction:

Get data from multiple, heterogeneous, and external sources.

### Cleaning:

Detect errors in the data and rectify them if possible.

### Transformation:

Convert data from legacy or host format to warehouse format.

### Loading:

Sort, summarize, consolidate, compute views, check integrity, and build indexes and partitions.

### Refresh:

Propagate only the updates from the data sources to the warehouse.

## Metadata repository

**Metadata: the data defining data-warehouse objects.**

**Description of the structure of the data warehouse:**

Schema, view, dimensions, hierarchies, derived-data definition, data-mart locations and contents.

**Operational metadata:**

**Data lineage** (history of migrated data and transformation path).

Currency of data (active, archived, or purged).

Monitoring information (warehouse-usage statistics, error reports, audit trails).

**Algorithms used for summarization.**

**Mapping from operational environment to data warehouse.**

**Data related to system performance:**

Warehouse schema, view and derived-data definitions.

**Business data:**

Business terms and definitions, ownership of data, charging policies.

## Chapter IV: Data warehousing and online analytical processing

Data warehouse: basic concepts.

**Data-warehouse modeling: data cube and OLAP.**

Data-warehouse design and usage.

Data-warehouse Implementation.

Data generalization by attribute-oriented induction.

Summary.

## From tables and spreadsheets to data cubes

Data warehouse: basic concepts.

Based on a **multidimensional data model** which views data in the form of a **data cube**.

### Data cube.

Allows data (here: sales) to be modeled and viewed in multiple dimensions.

**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.

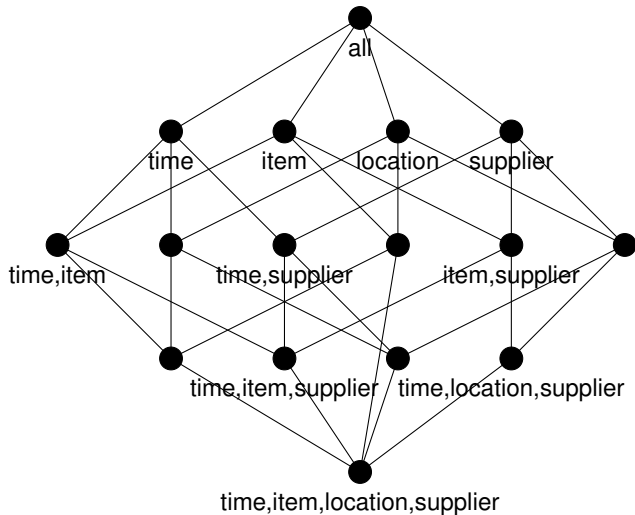
### Top most 0-dimensional cuboid.

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

### Star schema:.

A fact table in the middle connected to a set of dimension tables.

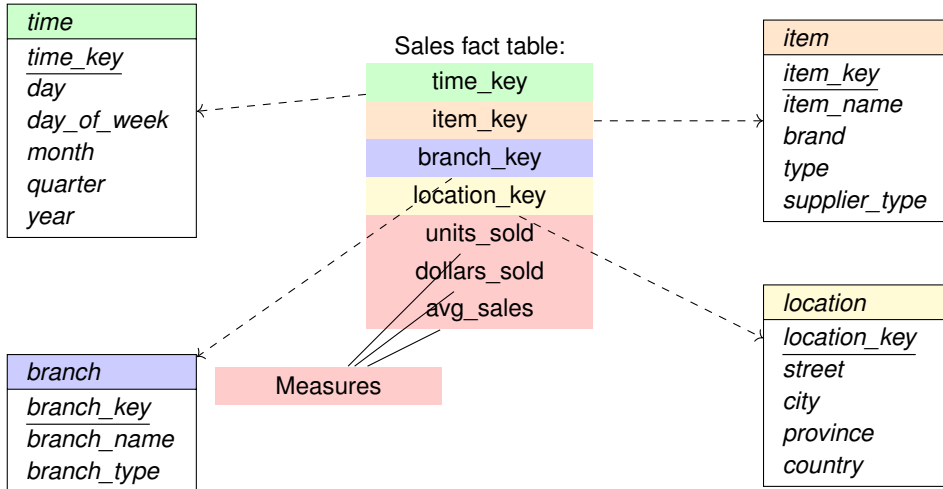
### Snowflake schema:.

A refinement of the star schema where some dimensional hierarchy is **normalized** into a set of smaller dimension tables, forming a shape similar to a snowflake.

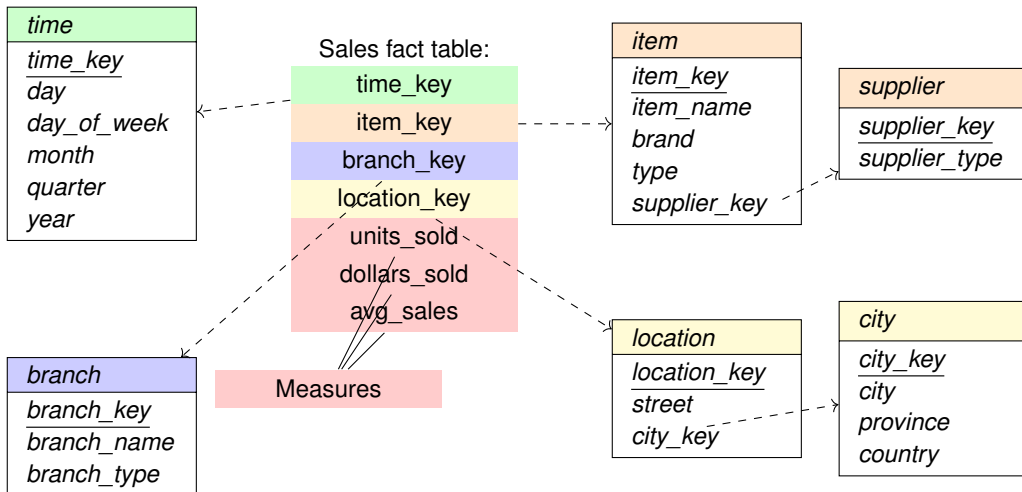
### Fact constellations:.

Multiple fact tables sharing dimension tables, viewed as a collection of stars, therefore called **galaxy schema** or fact constellation.

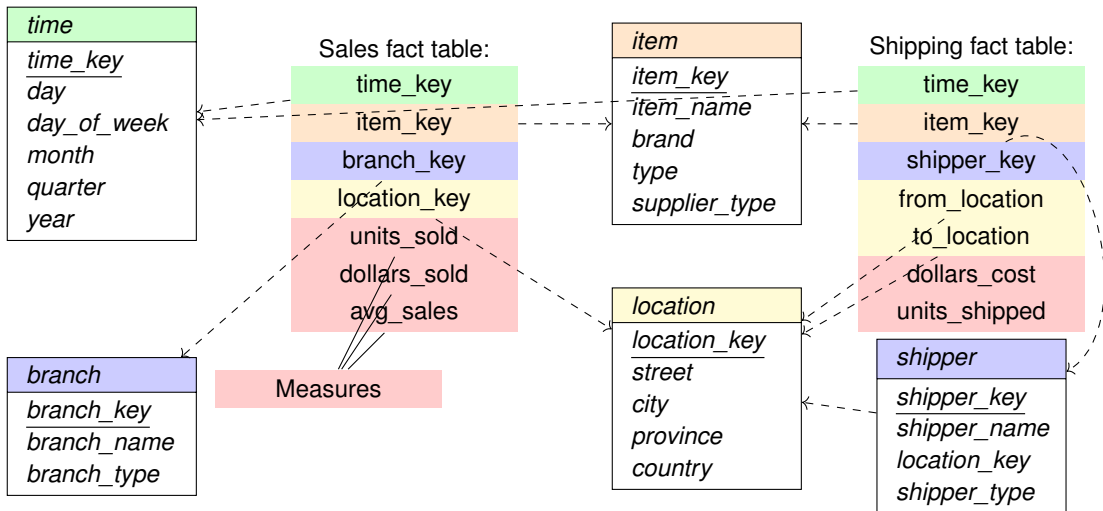
## Example of star schema



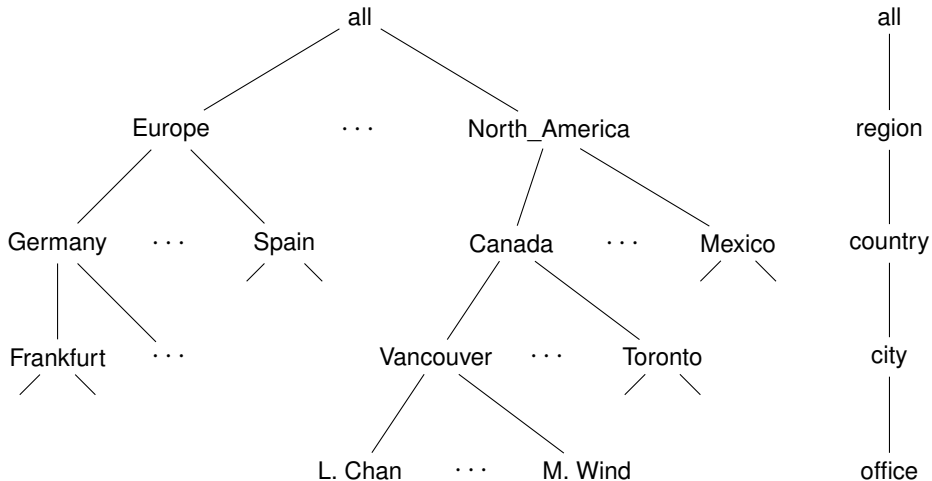
## Example of snowflake schema



## Example of fact constellation



## A concept hierarchy: dimension (location)



## Data-cube measures: three categories

### 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.

### Functional:

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,  $\text{MIN}_N$ , STD.

### 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).**

### FLOW:

Any aggregation.

E.g. sales turnover.

### STOCK:

No temporal aggregation.

E.g. stock, inventory.

### VPU (Value per Unit:

No summarization.

E.g. price, tax, in general factors.

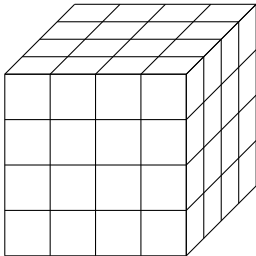
**(Always applicable: MIN, MAX and AVG).**

## Aggregation type

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

Dimensions: Product, Location, Time.

Hierarchical summarization paths.





## Chapter IV: Data warehousing and online analytical processing

Data warehouse: basic concepts.

Data-warehouse modeling: data cube and OLAP.

Data-warehouse design and usage.

Data-warehouse Implementation.

**Data generalization by attribute-oriented induction.**

Summary.

## 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	Country	Age range	City	Removed	Excl, Vg,...

## Class characterization: prime generalized relation (II)

Gender	Major	Birth re- gion	Age range	Residence	GPA	Count
M	Science	Canada	20-35	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 result into characteristic rules with quantitative information associated with it, e.g.

$$\text{grad}(x) \wedge \text{male}(x) \implies \text{birth\_region}(x) \quad (1)$$

$$= \text{"Canada"}[t : 53\%] \vee \text{birth\_region}(x) \quad (2)$$

$$= \text{"foreign"}[t : 47\%]. \quad (3)$$

## 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.

## Concept description 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.

## Chapter IV: Data warehousing and online analytical processing

Data warehouse: basic concepts.

Data-warehouse modeling: data cube and OLAP.

Data-warehouse design and usage.

Data-warehouse Implementation.

Data generalization by attribute-oriented induction.

**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).

### **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.**

## References (I)

S. Agarwal, R. Agrawal, P. M. Deshpande, A. Gupta, J. F. Naughton, R. Ramakrishnan, and S. Sarawagi: On the computation of multidimensional aggregates. VLDB'96.

D. Agrawal, A. E. Abbadi, A. Singh, and T. Yurek: Efficient view maintenance in data warehouses. SIGMOD'97.

R. Agrawal, A. Gupta, and S. Sarawagi: Modeling multidimensional databases. ICDE'97.

S. Chaudhuri and U. Dayal: An overview of data warehousing and OLAP technology. ACM SIGMOD Record, 26:65-74, 1997-

E. F. Codd, S. B. Codd, and C. T. Salley: Beyond decision support. Computer World, 27, July 1993.

J. Gray, et al. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. Data Mining and Knowledge Discovery, 1:29-54, 1997.

A. Gupta and I. S. Mumick. Materialized Views: Techniques, Implementations, and Applications. MIT Press, 1999.

## References (II)

- J. Han: Towards on-line analytical mining in large databases. ACM SIGMOD Record, 27:97-107, 1998.
- V. Harinarayan, A. Rajaraman, and J. D. Ullman: Implementing data cubes efficiently. SIGMOD'96.
- J. Hellerstein, P. Haas, and H. Wang: Online aggregation. SIGMOD'97.
- C. Imhoff, N. Galemme, and J. G. Geiger: Mastering Data Warehouse Design: Relational and Dimensional Techniques. John Wiley, 2003.
- W. H. Inmon: Building the Data Warehouse. John Wiley, 1996.
- R. Kimball and M. Ross: The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. 2ed. John Wiley, 2002.
- P. O'Neil and G. Graefe: Multi-table joins through bitmapped join indices. ACM SIGMOD Record, 24:8–11, Sept. 1995.
- P. O'Neil and D. Quass: Improved query performance with variant indexes. SIGMOD'97.
- Microsoft. OLEDB for OLAP programmer's reference version 1.0. In <http://www.microsoft.com/data/oledb/olap>, 1998.



Thank you for your attention.  
**Any questions about the fourth chapter?**

Ask them now, or again, drop me a line:  
✉ [luciano.melodia@fau.de](mailto:luciano.melodia@fau.de).