Data Mining

Session 4 – Main Theme
Data Warehousing and OLAP

Dr. Jean-Claude Franchitti

New York University
Computer Science Department
Courant Institute of Mathematical Sciences

Adapted from course textbook resources
Data Mining Concepts and Techniques (2nd Edition)
Jiawei Han and Micheline Kamber

Agenda

1. **Session Overview**
2. **Data Warehousing and OLAP**
3. **Summary and Conclusion**

What is the class about?

- Course description and syllabus:
  - » http://www.nyu.edu/classes/jcf/g22.3033-002/
  - » http://www.cs.nyu.edu/courses/spring10/G22.3033-002/index.html

- Textbooks:
  - » ***Data Mining: Concepts and Techniques (2nd Edition)***

Jiawei Han, Micheline Kamber
Morgan Kaufmann
ISBN-10: 1-55860-901-6, ISBN-13: 978-1-55860-901-3, (2006)

  - » ***Microsoft SQL Server 2008 Analysis Services Step by Step***

Scott Cameron
Microsoft Press
ISBN-10: 0-73562-620-0, ISBN-13: 978-0-73562-620-31 1st Edition (04/15/09)

Session Agenda

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining
- Data Generalization and Concept Description
- Summary

Icons / Metaphors


Information


Common Realization


Knowledge/Competency Pattern


Governance


Alignment


Solution Approach

Agenda

1 **Session Overview**

2 **Data Warehousing and OLAP**

3 **Summary and Conclusion**

Data Warehousing and OLAP - Sub-Topics

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining
- Data generalization and concept description

What is Data Warehouse?

- Defined in many different ways, but not rigorously.
  - A decision support database that is maintained separately from the organization's operational database
  - Support information processing by providing a solid platform of consolidated, historical data for analysis.
- "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, on-line 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.

Data Warehouse—Time Variant

- The time horizon for the data warehouse is significantly longer than that of operational systems
  - » Operational database: current value data
  - » Data warehouse data: 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
  - » But the key of operational data may or may not contain "time element"

Data Warehouse—Nonvolatile

- A physically separate store of data transformed from the operational environment
- Operational update of data does not occur in the data warehouse environment
  - » Does not require transaction processing, recovery, and concurrency control mechanisms
  - » Requires only two operations in data accessing:
    - *initial loading of data* and *access of data*

Data Warehouse vs. Heterogeneous DBMS

- Traditional heterogeneous DB integration: A query driven approach
  - » Build wrappers/mediators on top of heterogeneous databases
  - » When a query is posed to a client site, a meta-dictionary is used to translate the query into queries appropriate for individual heterogeneous sites involved, and the results are integrated into a global answer set
  - » Complex information filtering, compete for resources
- Data warehouse: update-driven, high performance
  - » Information from heterogeneous sources is integrated in advance and stored in warehouses for direct query and analysis

Data Warehouse vs. Operational DBMS

- OLTP (on-line transaction processing)
  - » Major task of traditional relational DBMS
  - » Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
- OLAP (on-line analytical processing)
  - » Major task of data warehouse system
  - » Data analysis and decision making
- Distinct features (OLTP vs. OLAP):
  - » User and system orientation: customer vs. market
  - » Data contents: current, detailed vs. historical, consolidated
  - » Database design: ER + application vs. star + subject
  - » View: current, local vs. evolutionary, integrated
  - » Access patterns: update vs. read-only but complex queries

OLTP vs. OLAP

	OLTP	OLAP
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc
access	read/write index/hash on prim. key	lots of scans
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response

Why 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 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

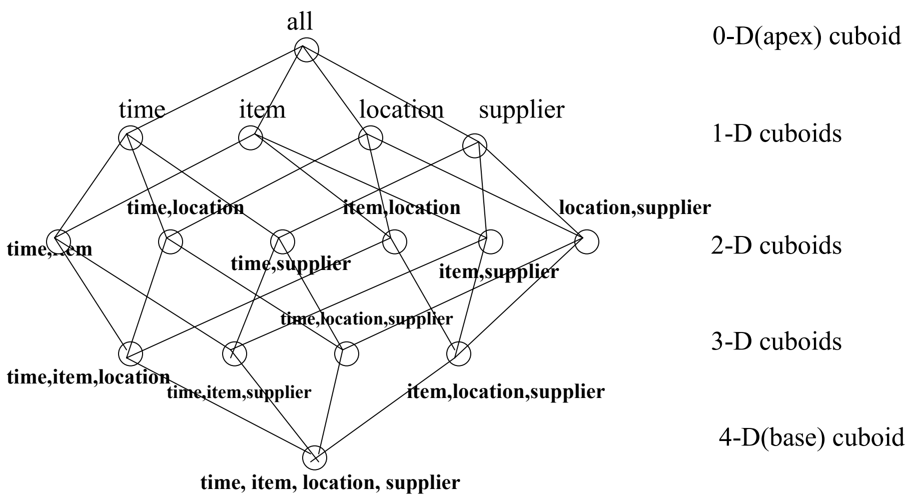
Data Warehousing and OLAP - Sub-Topics

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining
- Data generalization and concept description

From Tables and Spreadsheets to Data Cubes

- A data warehouse is based on a multidimensional data model which views data in the form of a data cube
- A data cube, such as sales, allows data 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 keys to each of the related dimension tables
- In data warehousing literature, an n-D base cube is called a base cuboid. The top most 0-D cuboid, which holds the highest-level of summarization, is called the apex cuboid. The lattice of cuboids forms a data cube.

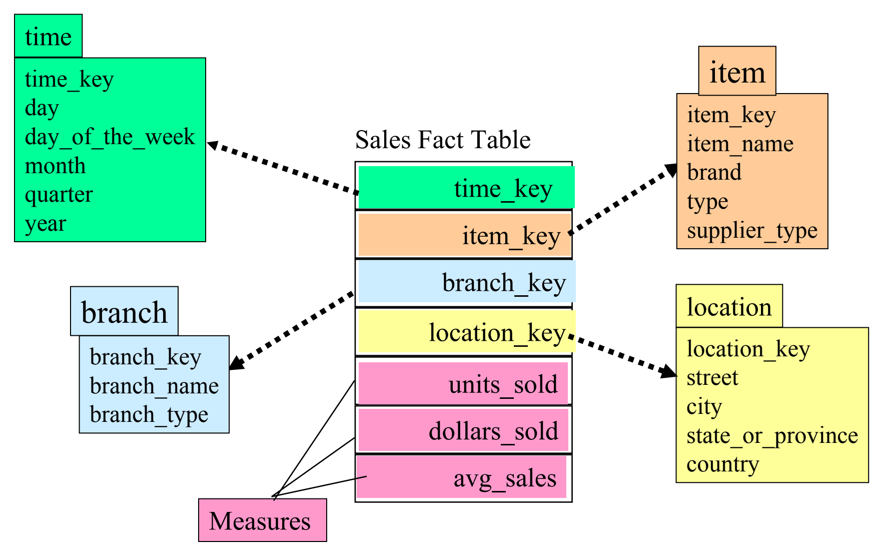
Cube: A Lattice of Cuboids



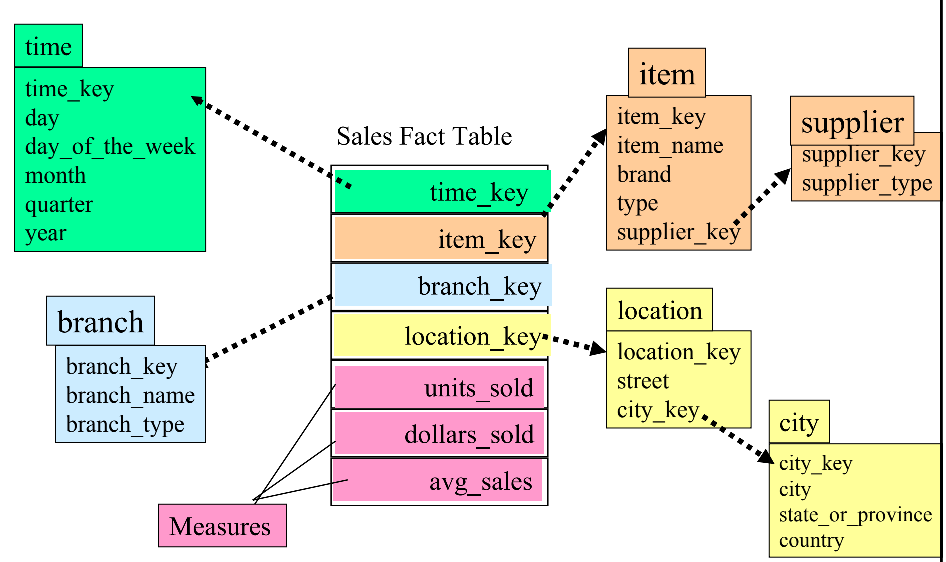
Conceptual Modeling of Data Warehouses

- Modeling data warehouses: dimensions & measures
  - » Star schema: A fact table in the middle connected to a set of dimension tables
  - » Snowflake schema: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  - » Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

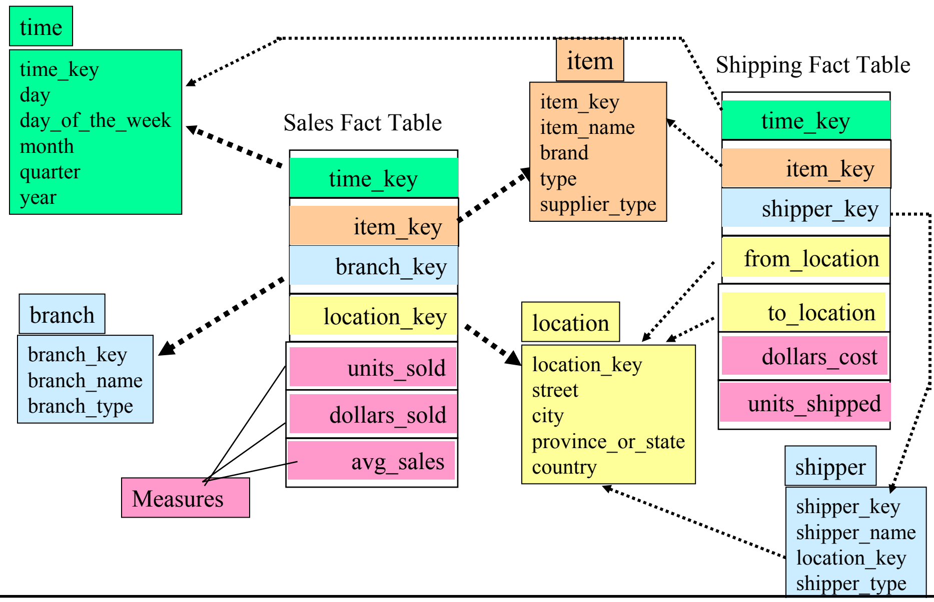
Example of Star Schema



Example of Snowflake Schema



Example of Fact Constellation



Cube Definition Syntax (BNF) in DMQL

- Cube Definition (Fact Table)

  define cube <cube_name> [<dimension_list>]: <measure_list>

- Dimension Definition (Dimension Table)

  define dimension <dimension_name> as (<attribute_or_subdimension_list>)

- Special Case (Shared Dimension Tables)
  - First time as "cube definition"
  - define dimension <dimension_name> as <dimension_name_first_time> in cube <cube_name_first_time>

Defining Star Schema in DMQL

define cube sales_star [time, item, branch, location]:

dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)

define dimension time as (time_key, day, day_of_week, month, quarter, year)

define dimension item as (item_key, item_name, brand, type, supplier_type)

define dimension branch as (branch_key, branch_name, branch_type)

define dimension location as (location_key, street, city, province_or_state, country)

Defining Snowflake Schema in DMQL

define cube sales_snowflake [time, item, branch, location]:

dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)

define dimension time as (time_key, day, day_of_week, month, quarter, year)

define dimension item as (item_key, item_name, brand, type, supplier(supplier_key, supplier_type))

define dimension branch as (branch_key, branch_name, branch_type)

define dimension location as (location_key, street, city(city_key, province_or_state, country))

Defining Fact Constellation in DMQL

define cube sales [time, item, branch, location]:

dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)

define dimension time as (time_key, day, day_of_week, month, quarter, year)

define dimension item as (item_key, item_name, brand, type, supplier_type)

define dimension branch as (branch_key, branch_name, branch_type)

define dimension location as (location_key, street, city, province_or_state, country)

define cube shipping [time, item, shipper, from_location, to_location]:

dollar_cost = sum(cost_in_dollars), unit_shipped = count(*)

define dimension time as time in cube sales

define dimension item as item in cube sales

define dimension shipper as (shipper_key, shipper_name, location as location in cube sales, shipper_type)

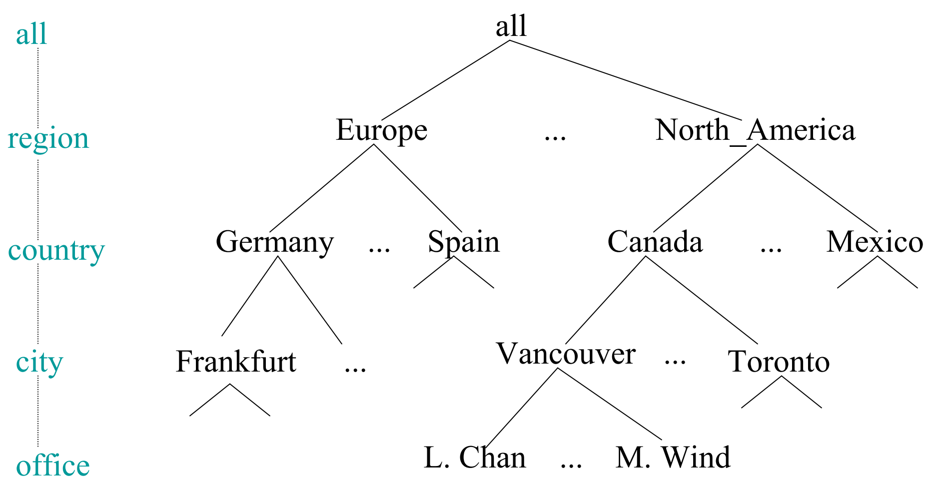
define dimension from_location as location in cube sales

define dimension to_location as location in cube sales

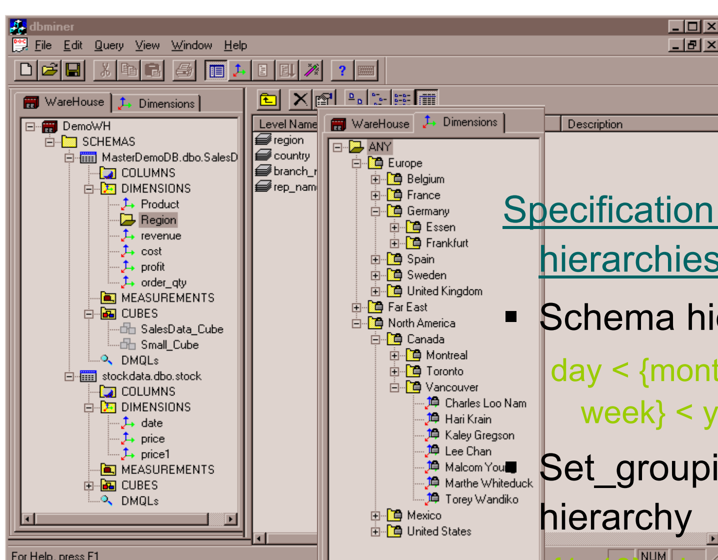
Measures of Data Cube: Three Categories

- Distributive: if the result derived by applying the function to *n* aggregate values is the same as that derived by applying the function on all the data without partitioning
  - E.g., count(), sum(), min(), max()
- Algebraic: if it can be computed by an algebraic function with *M* arguments (where *M* is a bounded integer), each of which is obtained by applying a distributive aggregate function
  - E.g., avg(), min_N(), standard_deviation()
- Holistic: if there is no constant bound on the storage size needed to describe a subaggregate.
  - E.g., median(), mode(), rank()

A Concept Hierarchy: Dimension (location)



View of Warehouses and Hierarchies



Specification of hierarchies

- Schema hierarchy

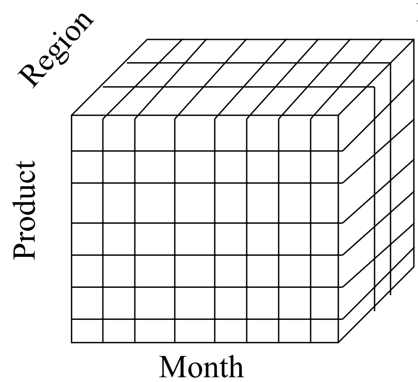
  day < {month < quarter; week} < year

- Set_grouping hierarchy

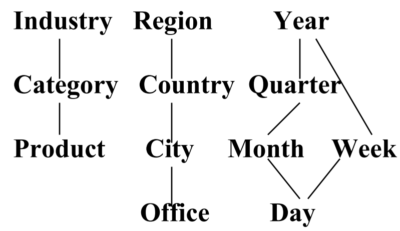
  {1..10} < inexpensive

Multidimensional Data

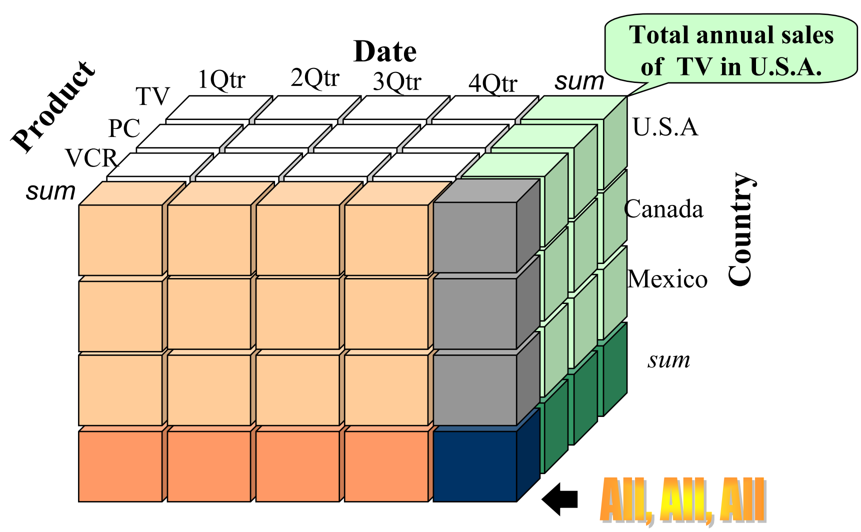
- Sales volume as a function of product, month, and region



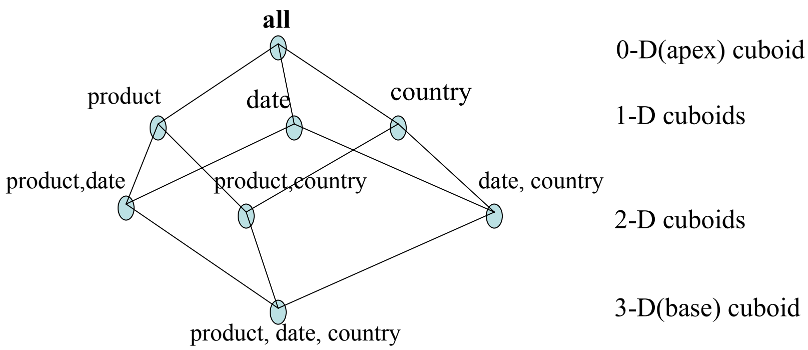
Dimensions: Product, Location, Time
Hierarchical summarization paths



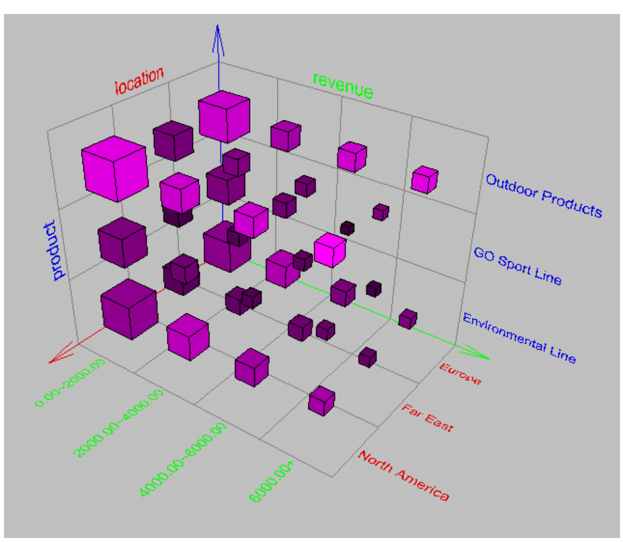
A Sample Data Cube



Cuboids Corresponding to the Cube



Browsing a Data Cube

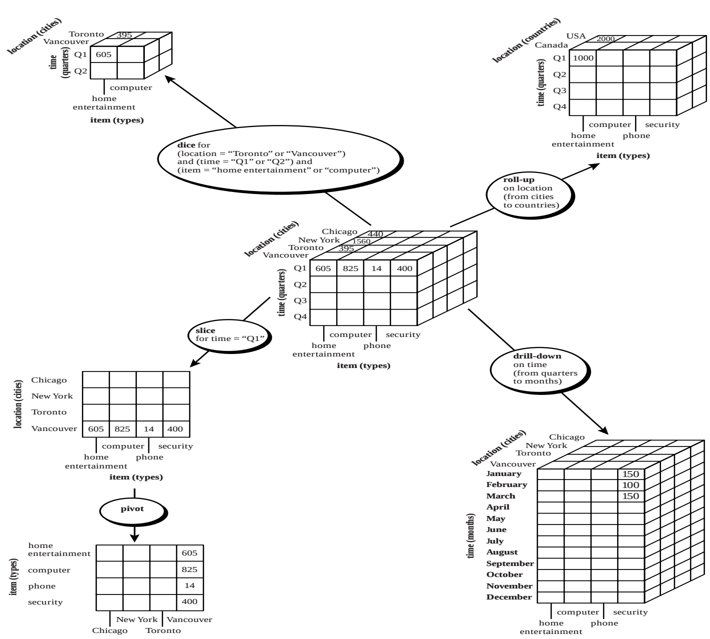


- Visualization
- OLAP capabilities
- Interactive manipulation

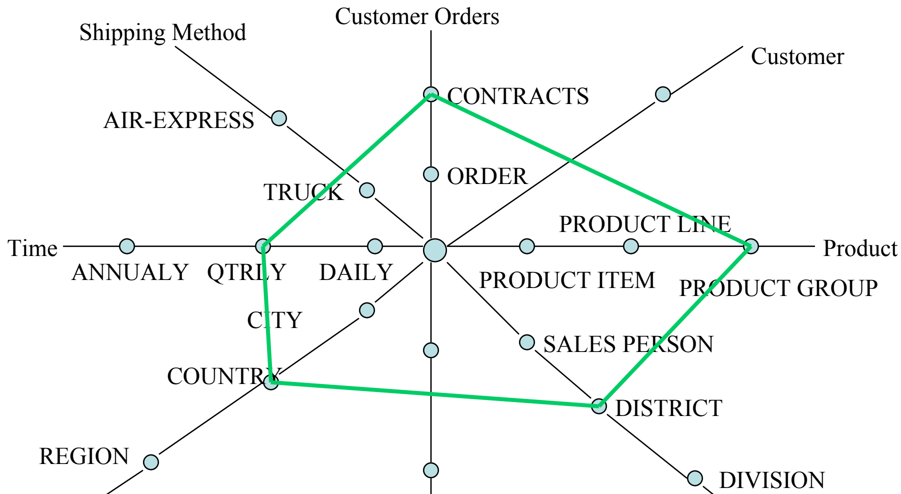
Typical OLAP Operations (1/2)

- 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)*

Typical OLAP Operations (2/2)



A Star-Net Query Model



Each circle is called a footprint

Data Warehousing and OLAP - Sub-Topics

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining
- Data generalization and concept description

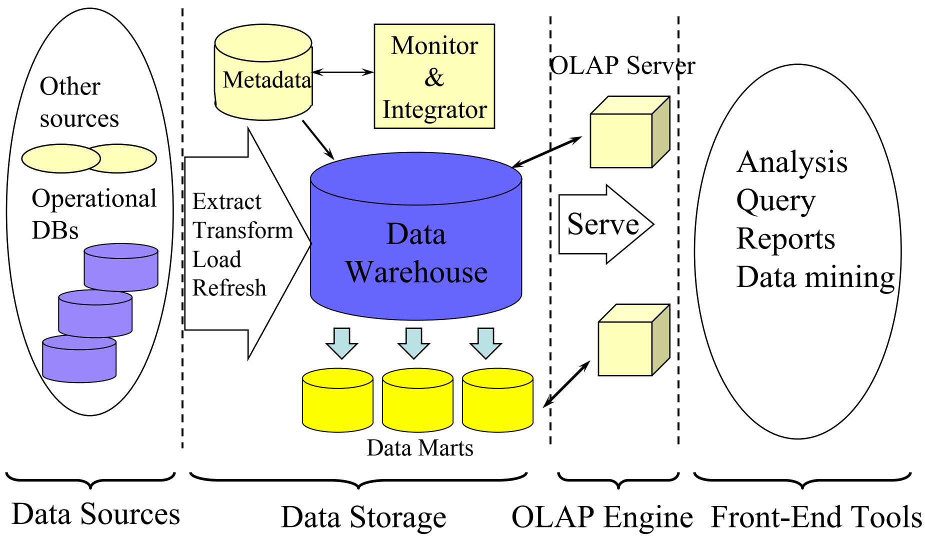
Design of Data Warehouse: A Business Analysis Framework

- 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 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, quick turn around
- Typical data warehouse design process
  - » Choose a business process to model, e.g., orders, invoices, etc.
  - » Choose the *grain* (*atomic level of data*) of the business process
  - » Choose the dimensions that will apply to each fact table record
  - » Choose the measure that will populate each fact table record

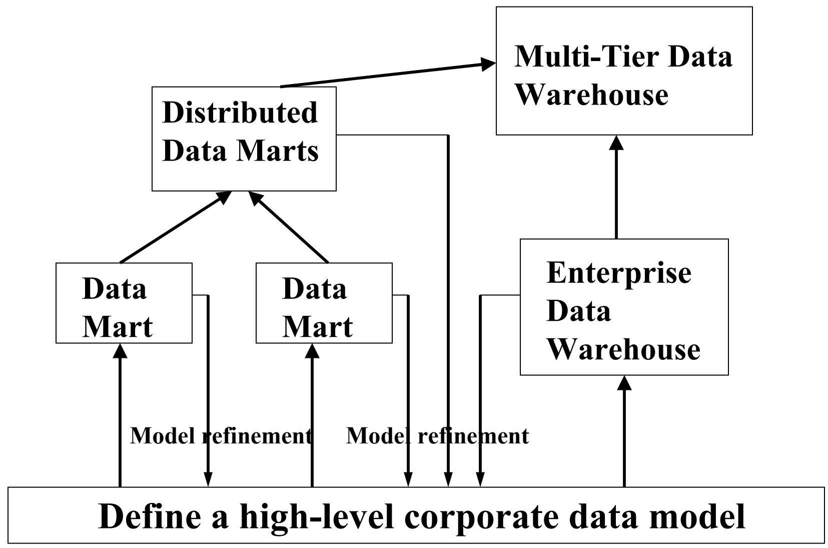
Data Warehouse: A Multi-Tiered Architecture



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 groups 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

Data Warehouse Development: A Recommended Approach



Data Warehouse Back-End Tools and Utilities

- Data extraction
  - » get data from multiple, heterogeneous, and external sources
- Data cleaning
  - » detect errors in the data and rectify them when possible
- Data transformation
  - » convert data from legacy or host format to warehouse format
- Load
  - » sort, summarize, consolidate, compute views, check integrity, and build indicies and partitions
- Refresh
  - » propagate the updates from the data sources to the warehouse

Metadata Repository

- Meta data is the data defining warehouse objects. It stores:
- Description of the structure of the data warehouse
  - schema, view, dimensions, hierarchies, derived data defn, data mart locations and contents
- Operational meta-data
  - 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)
- The algorithms used for summarization
- The mapping from operational environment to the 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

OLAP Server Architectures

- Relational OLAP (ROLAP)
  - Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middle ware
  - 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 SQLServer)
  - 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 Warehousing and OLAP - Sub-Topics

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining
- Data generalization and concept description

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 levels?

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

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

Cube Operation

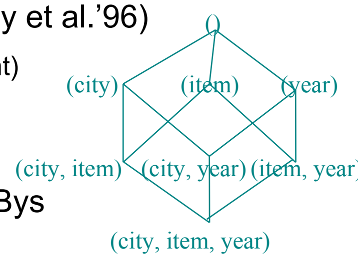
- Cube definition and computation in DMQL

    define cube sales[item, city, year]: sum(sales_in_dollars)

    compute cube sales

- Transform it into a SQL-like language (with a new operator cube by, introduced by Gray et al.'96)

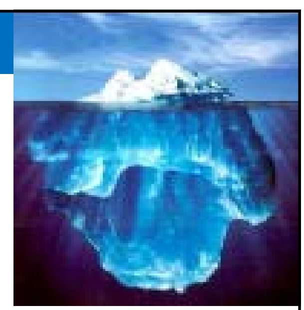
```
SELECT item, city, year, SUM (amount)
FROM SALES
CUBE BY item, city, year
```



- Need compute the following Group-Bys

    (date, product, customer),
    (date,product),(date, customer), (product, customer),
    (date), (product), (customer)
    ()

Iceberg Cube



- Computing only the cuboid cells whose count or other aggregates satisfying the condition like

    HAVING COUNT(*) >= *minsup*

- Motivation
    - Only a small portion of cube cells may be "above the water'' in a sparse cube
    - Only calculate "interesting" cells—data above certain threshold
    - Avoid explosive growth of the cube
        - Suppose 100 dimensions, only 1 base cell. How many aggregate cells if count >= 1? What about count >= 2?

Indexing OLAP Data: Bitmap Index

- Index on a particular column
- Each value in the column has a bit vector: bit-op is fast
- The length of the bit vector: # of records in the base table
- The *i*-th bit is set if the *i*-th row of the base table has the value for the indexed column
- not suitable for high cardinality domains

Base table

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

Index on Region

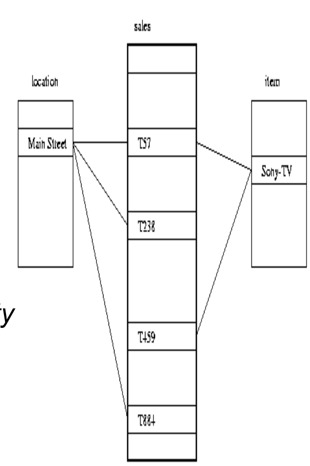
RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

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: JI(R-id, S-id) where R (R-id, …) ⊳⊲ S (S-id, …)
- Traditional indices map the values to a list of record ids
  - » It materializes relational join in JI file and speeds up relational join
- In data warehouses, join index relates the values of the dimensions of a start schema to rows in the fact table.
  - » E.g. fact table: *Sales* and two dimensions *city* and *product*
    - A join index on *city* maintains for each distinct city a list of R-IDs of the tuples recording the Sales in the city
  - » Join indices can span multiple dimensions



Efficient Processing 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 op.
  - 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 structs in MOLAP

Data Warehousing and OLAP - Sub-Topics

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining
- Data generalization and concept description

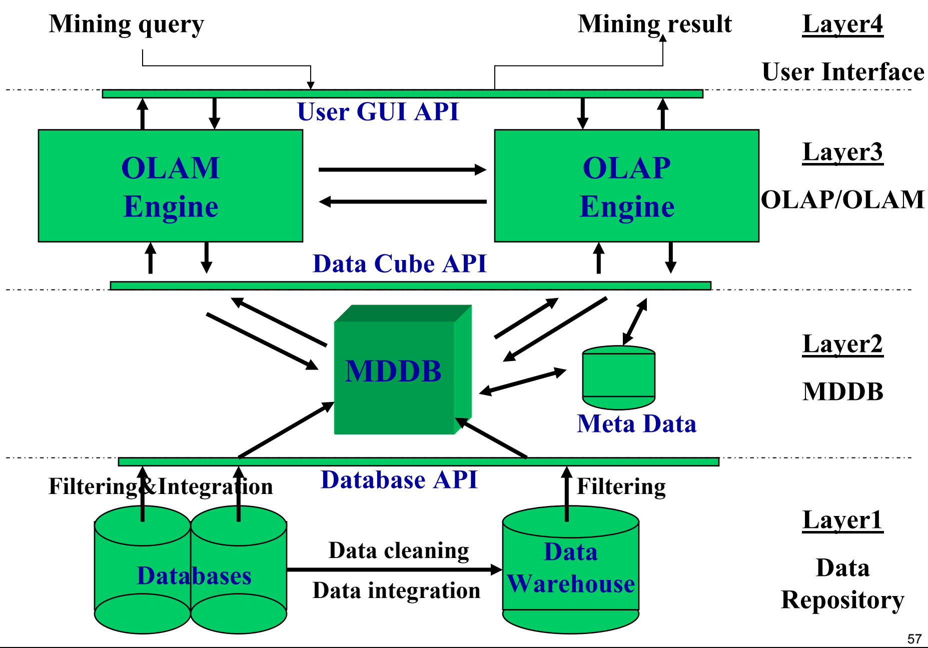
Data Warehouse Usage

- Three kinds of data warehouse applications
  - » Information processing
    - supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs
  - » Analytical processing
    - multidimensional analysis of data warehouse data
    - supports basic OLAP operations, slice-dice, drilling, pivoting
  - » Data mining
    - knowledge discovery from hidden patterns
    - supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools

From OLAP to On Line Analytical Mining (OLAM)

- Why online analytical mining?
  - » High quality of data in data warehouses
    - DW contains integrated, consistent, cleaned data
  - » Available information processing structure surrounding data warehouses
    - ODBC, OLEDB, Web accessing, service facilities, reporting and OLAP tools
  - » OLAP-based exploratory data analysis
    - Mining with drilling, dicing, pivoting, etc.
  - » On-line selection of data mining functions
    - Integration and swapping of multiple mining functions, algorithms, and tasks

An OLAM System Architecture



Data Warehousing and OLAP - Sub-Topics

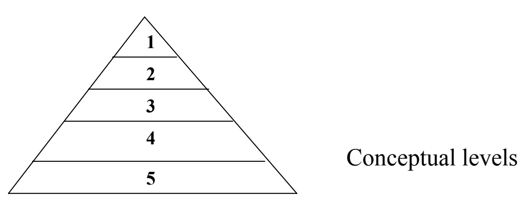
- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining
- Data generalization and concept description

What is Concept Description?

- Descriptive vs. predictive data mining
  - » Descriptive mining: describes concepts or task-relevant data sets in concise, summarative, informative, discriminative forms
  - » Predictive mining: Based on data and analysis, constructs models for the database, and predicts the trend and properties of unknown data
- Concept description:
  - » Characterization: provides a concise and succinct summarization of the given collection of data
  - » Comparison: provides descriptions comparing two or more collections of data

Data Generalization and Summarization-based Characterization

- Data generalization
  - » A process which abstracts a large set of task-relevant data in a database from a low conceptual levels to higher ones.



  - » Approaches:
    - Data cube approach(OLAP approach)
    - Attribute-oriented induction approach

Attribute-Oriented Induction

- Proposed in 1989 (KDD '89 workshop)
- Not confined to categorical data nor particular measures
- How it is 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
  - » Interactive presentation with users

Basic Principles of Attribute-Oriented Induction

- Data focusing: task-relevant data, including dimensions, and 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.

Example

- DMQL: Describe general characteristics of graduate students in the Big-University database

```
use Big_University_DB
mine characteristics as "Science_Students"
in relevance to name, gender, major, birth_place,
   birth_date, residence, phone#, gpa
from student
where status in "graduate"
```

- Corresponding SQL statement:

```
Select name, gender, major, birth_place, birth_date,
   residence, phone#, gpa
from student
where status in {"Msc", "MBA", "PhD" }
```

Class Characterization: An Example

Initial Relation

Name	Gender	Major	Birth-Place	Birth_date	Residence	Phone #	GPA
Jim Woodman	M	CS	Vancouver,BC, Canada	8-12-76	3511 Main St., Richmond	687-4598	3.67
Scott Lachance	M	CS	Montreal, Que, Canada	28-7-75	345 1st Ave., Richmond	253-9106	3.70
Laura Lee	F	Physics	Seattle, WA, USA	25-8-70	125 Austin Ave., Burnaby	420-5232	3.83
…	…	…	…	…	…	…	…
Removed	Retained	Sci,Eng, Bus	Country	Age range	City	Removed	Excl, VG,..

Prime Generalized Relation

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
…	…	…	…	…	…	…

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

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.,

$$grad(x) \land male(x) \Rightarrow$$
$$birth\_region(x) = "Canada"[t:53\%] \lor birth\_region(x) = "foreign"[t:47\%].$$

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
  - Compare tuples with the same high level descriptions
  - Present for every tuple its description and two measures
    - support - distribution within single class
    - comparison - distribution between classes
  - Highlight the tuples with strong discriminant features
- Relevance Analysis:
  - Find attributes (features) which best distinguish different classes

Quantitative Discriminant Rules

- Cj = target class
- q_a = a generalized tuple covers some tuples of class
  - but can also cover some tuples of contrasting class
- d-weight
  - range: [0, 1]

$$d-weight = \frac{count(q_a \in C_j)}{\sum_{i=1}^{m} count(q_a \in C_i)}$$

- quantitative discriminant rule form

$$\forall X,\ target\_class(X) \Leftarrow condition(X) \quad [d : d\_weight]$$

Example: Quantitative Discriminant Rule

Status	Birth_country	Age_range	Gpa	Count
Graduate	Canada	25-30	Good	90
Undergraduate	Canada	25-30	Good	210

Count distribution between graduate and undergraduate students for a generalized tuple

- Quantitative discriminant rule

$$\forall X,\ graduate\_student(X) \Leftarrow$$
$$birth\_country(X) = "Canada" \wedge age\_range(X) = "25-30" \wedge gpa(X) = "good"\ \ [d : 30\%]$$

  - where 90/(90 + 210) = 30%

Class Description

- Quantitative characteristic rule

$$\forall\ X,\ target\_class(X) \Rightarrow condition(X)\ \ [t : t\_weight]$$

  - necessary

- Quantitative discriminant rule

$$\forall\ X,\ target\_class(X) \Leftarrow condition(X)\ \ [d : d\_weight]$$

  - sufficient

- Quantitative description rule

$$\forall\ X,\ target\_class(X) \Leftrightarrow$$
$$condition_1(X)\,[t : w_1, d : w'_1] \vee \ldots \vee condition_n(X)\,[t : w_n, d : w'_n]$$

  - necessary and sufficient

Example: Quantitative Description Rule

Location/item		TV			Computer			Both_items	
	Count	*t-wt*	*d-wt*	*Count*	*t-wt*	*d-wt*	*Count*	*t-wt*	*d-wt*
Europe	80	25%	40%	240	75%	30%	320	100%	32%
N_Am	120	17.65%	60%	560	82.35%	70%	680	100%	68%
Both_ regions	200	20%	100%	800	80%	100%	1000	100%	100%

Crosstab showing associated t-weight, d-weight values and total number (in thousands) of TVs and computers sold at AllElectronics in 1998

- Quantitative description rule for target class *Europe*

$$\forall X, Europe(X) \Leftrightarrow$$
$$(item(X) = "TV")[\mathrm{t}:25\%, \mathrm{d}:40\%] \vee (item(X) = "computer")[\mathrm{t}:75\%, \mathrm{d}:30\%]$$

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

Agenda

1 Session Overview

2 Data Warehousing and OLAP

3 Summary and Conclusion

Summary

- Data generalization: Attribute-oriented induction
- Data warehousing: A multi-dimensional model of a data warehouse
  - Star schema, snowflake schema, fact constellations
  - A data cube consists of dimensions & measures
- OLAP operations: drilling, rolling, slicing, dicing and pivoting
- Data warehouse architecture
- OLAP servers: ROLAP, MOLAP, HOLAP
- Efficient computation of data cubes
  - Partial vs. full vs. no materialization
  - Indexing OALP data: Bitmap index and join index
  - OLAP query processing
- From OLAP to OLAM (on-line analytical mining)

References (1/2)

- 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.
- 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

References (2/2)

- C. Imhoff, N. Galemmo, 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 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
- A. Shoshani. OLAP and statistical databases: Similarities and differences. PODS'00.
- S. Sarawagi and M. Stonebraker. Efficient organization of large multidimensional arrays. ICDE'94
- OLAP council. MDAPI specification version 2.0. In http://www.olapcouncil.org/research/apily.htm, 1998
- E. Thomsen. OLAP Solutions: Building Multidimensional Information Systems. John Wiley, 1997
- P. Valduriez. Join indices. ACM Trans. Database Systems, 12:218-246, 1987.
- J. Widom. Research problems in data warehousing. CIKM'95.

Assignments & Readings

- Readings

  » Chapter 3
- Assignment #3
  » TBA

Next Session: Characterization