Decision Support: Warehousing, OLAP, and Data Mining

Module 9, Lecture 5

Introduction

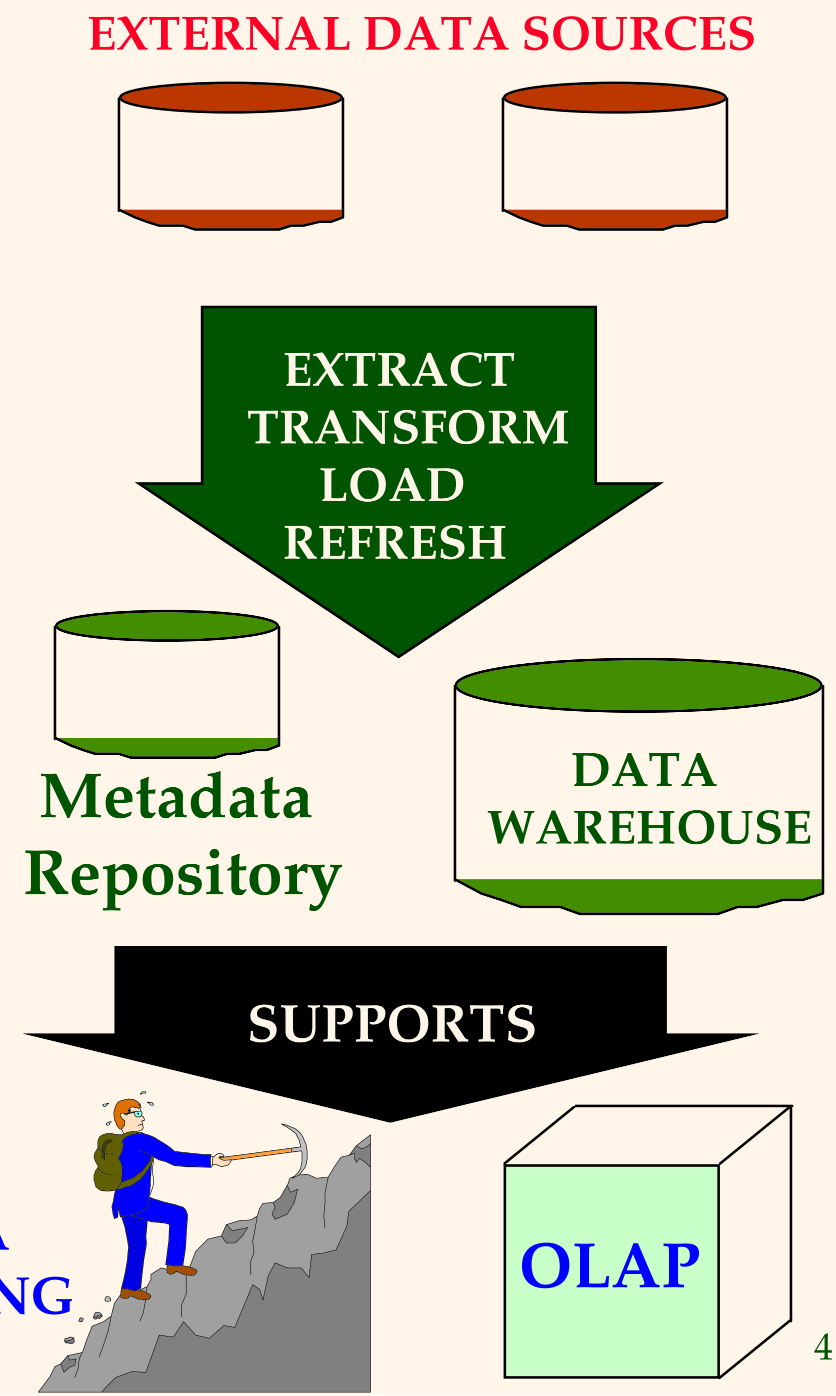
- Increasingly, organizations are analyzing current and historical data to identify useful patterns and support business strategies.
- Emphasis is on complex, interactive, exploratory analysis of very large datasets created by integrating data from across all parts of an enterprise; data is fairly static.
  - Contrast such **On-Line Analytic Processing (OLAP)** with traditional **On-line Transaction Processing (OLTP):** mostly long queries, instead of short update Xacts.

Three Complementary Trends

- Data Warehousing: Consolidate data from many sources in one large repository.
  - Loading, periodic synchronization of replicas.
  - Semantic integration.
- OLAP: Complex SQL queries, business-oriented queries based on spreadsheet-style operations and "multidimensional" view of data.
- Data Mining: Exploratory analysis; essentially, fishing for interesting trends and anomalies.

Data Warehousing

- Integrated data spanning long time periods, often augmented with summary information.
- Several gigabytes to terabytes common.
- Interactive response times expected for complex queries; ad-hoc updates uncommon.



Warehousing Issues

- Semantic Integration: When getting data from multiple sources, must eliminate mismatches, e.g., different currencies, schemas.
- Heterogeneous Sources: Must access data from a variety of source formats and repositories.
  - Replication capabilities can be exploited here.
- Load, Refresh, Purge: Must load data, periodically refresh it, and purge too-old data.
- Metadata Management: Must keep track of source, loading time, and other information for all data in the warehouse.

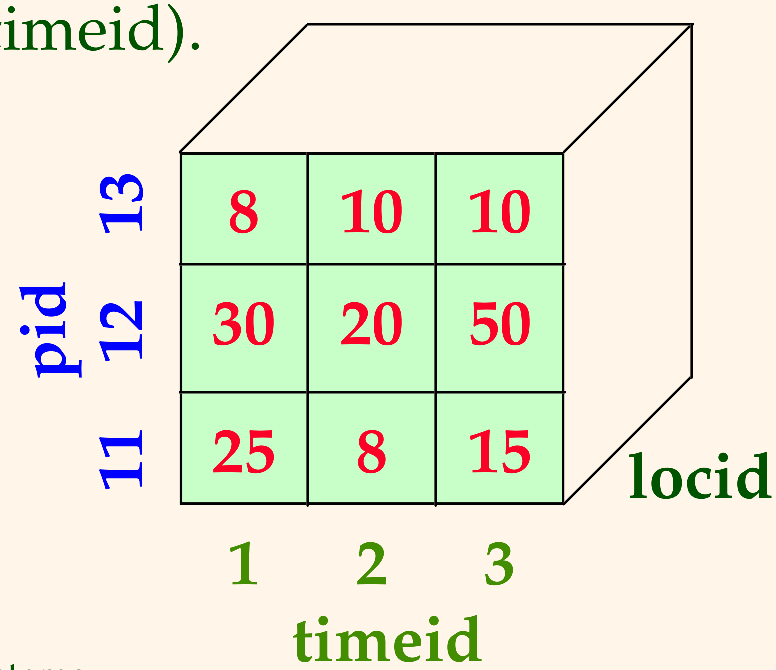
Multidimensional Data Model

- Collection of numeric measures, which depend on a set of dimensions.
  - E.g., measure **Sales**, dimensions **Product** (key: pid), **Location** (locid), and **Time** (timeid).

pid	timeid	locid	sales
11	1	1	25
11	2	1	8
11	3	1	15
12	1	1	30
12	2	1	20
12	3	1	50
13	1	1	8
13	2	1	10
13	3	1	10
11	1	2	35

• • •

Slice locid=1 is shown:

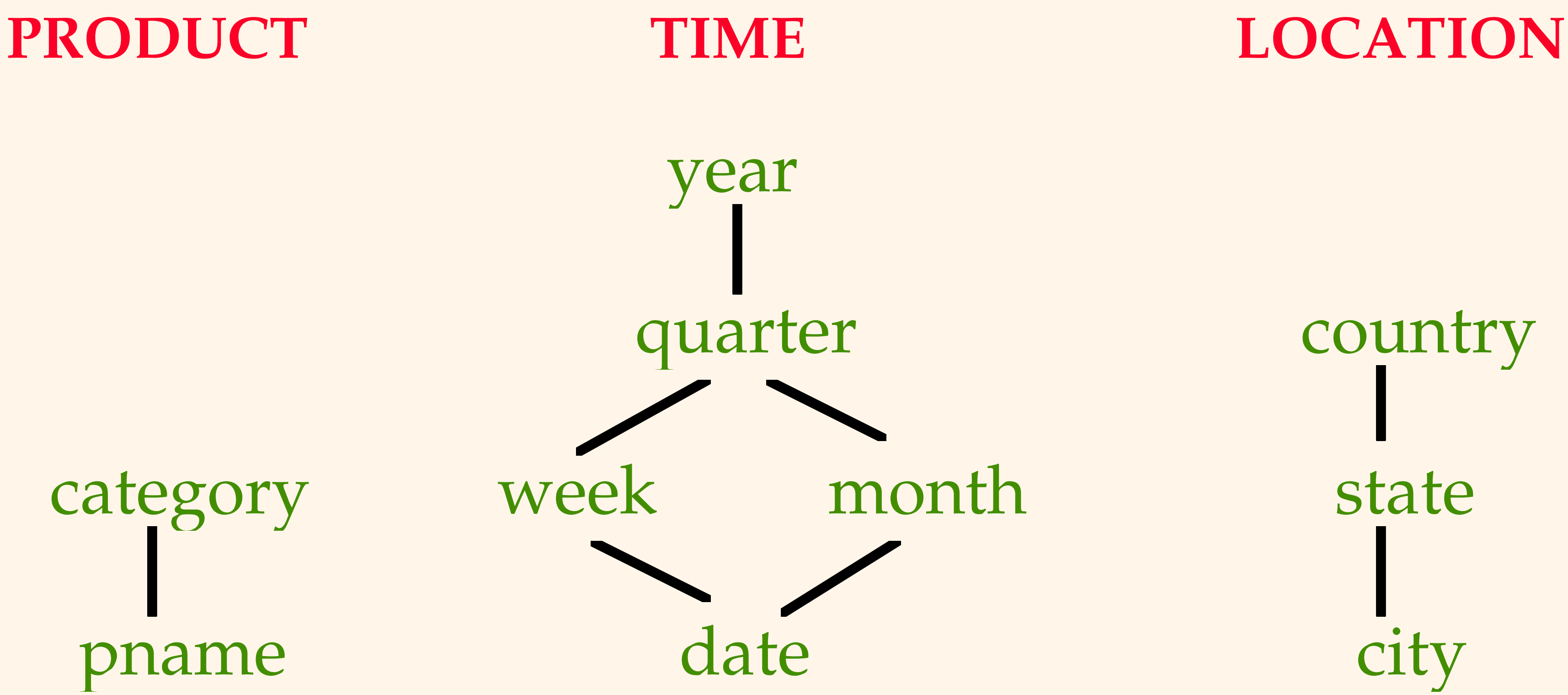


MOLAP vs ROLAP

- Multidimensional data can be stored physically in a (disk-resident, persistent) array; called MOLAP systems. Alternatively, can store as a relation; called ROLAP systems.
- The main relation, which relates dimensions to a measure, is called the fact table. Each dimension can have additional attributes and an associated dimension table.
  - E.g., **Products(pid, pname, category, price)**
  - Fact tables are *much* larger than dimensional tables.

Dimension Hierarchies

- For each dimension, the set of values can be organized in a hierarchy:



OLAP Queries

- Influenced by SQL and by spreadsheets.
- A common operation is to aggregate a measure over one or more dimensions.
  - Find total sales.
  - Find total sales for each city, or for each state.
  - Find top five products ranked by total sales.
- Roll-up: Aggregating at different levels of a dimension hierarchy.
  - E.g., Given total sales by city, we can roll-up to get sales by state.

OLAP Queries

- Drill-down: The inverse of roll-up.
  - E.g., Given total sales by state, can drill-down to get total sales by city.
  - E.g., Can also drill-down on different dimension to get total sales by product for each state.
- Pivoting: Aggregation on selected dimensions.
  - E.g., Pivoting on Location and Time yields this **cross-tabulation**:

	WI	CA	Total
1995	63	81	144
1996	38	107	145
1997	75	35	110
Total	176	223	339

- Slicing and Dicing: Equality and range selections on one or more dimensions.

Comparison with SQL Queries

- The cross-tabulation obtained by pivoting can also be computed using a collection of SQLqueries:

```
SELECT SUM(S.sales)
FROM    Sales S, Times T, Locations L
WHERE  S.timeid=T.timeid AND S.timeid=L.timeid
GROUP BY T.year, L.state
```

```
SELECT SUM(S.sales)
FROM    Sales S, Times T
WHERE  S.timeid=T.timeid
GROUP BY T.year
```

```
SELECT SUM(S.sales)
FROM    Sales S, Location L
WHERE  S.timeid=L.timeid
GROUP BY L.state
```

The CUBE Operator

- Generalizing the previous example, if there are k dimensions, we have 2^k possible SQL GROUP BY queries that can be generated through pivoting on a subset of dimensions.
- CUBE pid, locid, timeid BY SUM Sales
  - Equivalent to rolling up Sales on all eight subsets of the set {pid, locid, timeid}; each roll-up corresponds to an SQL query of the form:

Lots of recent work on optimizing the CUBE operator!

```
SELECT SUM(S.sales)
FROM   Sales S
GROUP BY grouping-list
```

Design and Implementation Issues

- Fact table typically in BCNF; dimension tables not normalized. (Intuition?)
- New indexing techniques: Bit Map indexes, Join indexes, array representations, compression, precomputation of aggregations, etc.
- E.g., Bit Map index:

Bit-vector: 1 bit for each possible value.

Many queries can be answered using bit-vector ops!

sex

10
10
01
10

custid name sex rating

custid	name	sex	rating
112	Joe	M	3
115	Ram	M	5
119	Sue	F	5
112	Woo	M	4

rating

00100
00001
00001
00010

Data Mining

- This is a broad term applied to several kinds of exploratory analysis.
  - E.g., Association rules, sequential patterns, classification, clustering, Bayesian networks for inferring causality, sequence similarity, visualization.
- Related to work in Machine Learning, Math Programming and Statistics.
- New emphasis: Very large datasets.

Association Rules

- Given a collection of customer purchases (or "transactions") of the form:

transid	custid	date	item	price	qty

we want to identify rules of the form:

{pen} => {ink}

"A transaction in which a pen was bought is
is also likely to have involved an ink purchase."

Association Rules

LHS => RHS

- Support: % of transactions containing all items in LHS and RHS of rule.
- Confidence: Consider transactions containing all LHS item. Confidence is the % of these transactions that also contain all RHS items.
- Should be used for prediction with great caution; rules do not really reflect ***causality***.
  - If users buy pencils whenever they buy pens, we have the rule {pencil} => {ink}, but there is no causal link; offering a promotion on pencils in order to stimulate sales of ink will be a failure!

Efficient Computation of Assoc. Rules

- Goal: To efficiently find all rules that satisfy a user-specified support and confidence level.
- Idea: To rely on the "a priori" property that if an itemset has minimum support (I.e., is a frequent itemset) then every subset of this itemset also has minimum support.
- Iteratively: Find all 1-item frequent itemsets; then all 2-item frequent itemsets, etc.
  - In each iteration k, only consider itemsets that contain some k-1 frequent itemset.
- Given all frequent itemsets, finding rules is easy.

Summary

- Decision support is an emerging, rapidly growing subarea of databases.
- Involves the creation of large, consolidated data repositories called data warehouses.
- Warehouses exploited using sophisticated analysis techniques: complex SQL queries, OLAP "multidimensional" queries (influenced by both SQL and spreadsheets), and exploratory analysis (data mining).