Information Systems

WS 2005, JKU Linz
Course 8: On-Line Analytical Processing and Search Engines

Gábor Bodnár

URL: `http://www.risc.uni-linz.ac.at/education/courses/ws2005/is/`

Overview

- Data Warehouses
- On-Line Analytical Processing
- Search Engines

Data Warehouses

A *data warehouse* is a collection of integrated, nonvolatile enterprise data, which is often restructured, pre-analyzed, and reflect evolutionary processes of the enterprise.

The data is usually collected throughout a long period of time, typically from the operational databases of the enterprise.

Practically it is a huge database for analytical purposes.

Data Marts

It is a special-purpose, integrated and time variant database, which can be volatile, supporting only a restricted subset of queries.

Properties of Data Warehouses

- The DBMS should be of high performance.
- The warehouse databases can be less normalized than the operational databases.
- When data is loaded into the warehouse, additional error correction might also be applied.
- A disadvantage of data warehouses is their cost.

The queries occurring in the context of data warehouses can be much more complex and sophisticated than the ones on the operational databases.

Loading the Data Warehouse

- Extract: often from the working operational database "in parallel" and on the physical level.
- Cleansing: e.g. filling up missing data with defaults, correcting typos, etc.
- Transformation and Consolidation: splitting/combining the extracted data to conform the scheme of the data warehouse; doing "time synchronization".
- Load: moving data into the warehouse, checking integrity, building indices/hashes.

Design Principles for Data Warehouses

- Logical design: the same rules apply as for OLTP applications; meaning of the data can be encoded by integrity constraints.
- Physical design: partitioning, indexing/hashing, controlled redundancy (prejoins, replication, derived data).
- Derived data: calculated columns, summary tables (updated via triggers).

Remark: The prejoins should not appear on the logical level.

On-Line Analytical Processing (OLAP)

On-line analytical processing is a prominent application of data warehouses (another one is data mining).

The characterizing features of OLAP

- Multidimensional view of data.
- Data aggregation with respect to many groupings.

The attributes spanning the multidimensional array are called *independent* and the ones in the array are called *dependent*.

Multidimensional View of Data

Requirements:

- Pivoting: ability to swap roles of an independent and a dependent variable.
- Drill up/drill down: moving up/down in a hierarchy of resolutions of an attribute.

Advantages of multidimensional data representation:

- Conceptually better representation of data.
- Fast responses to aggregation queries.

Relational OLAP

Disadvantages of multidimensional data representation:

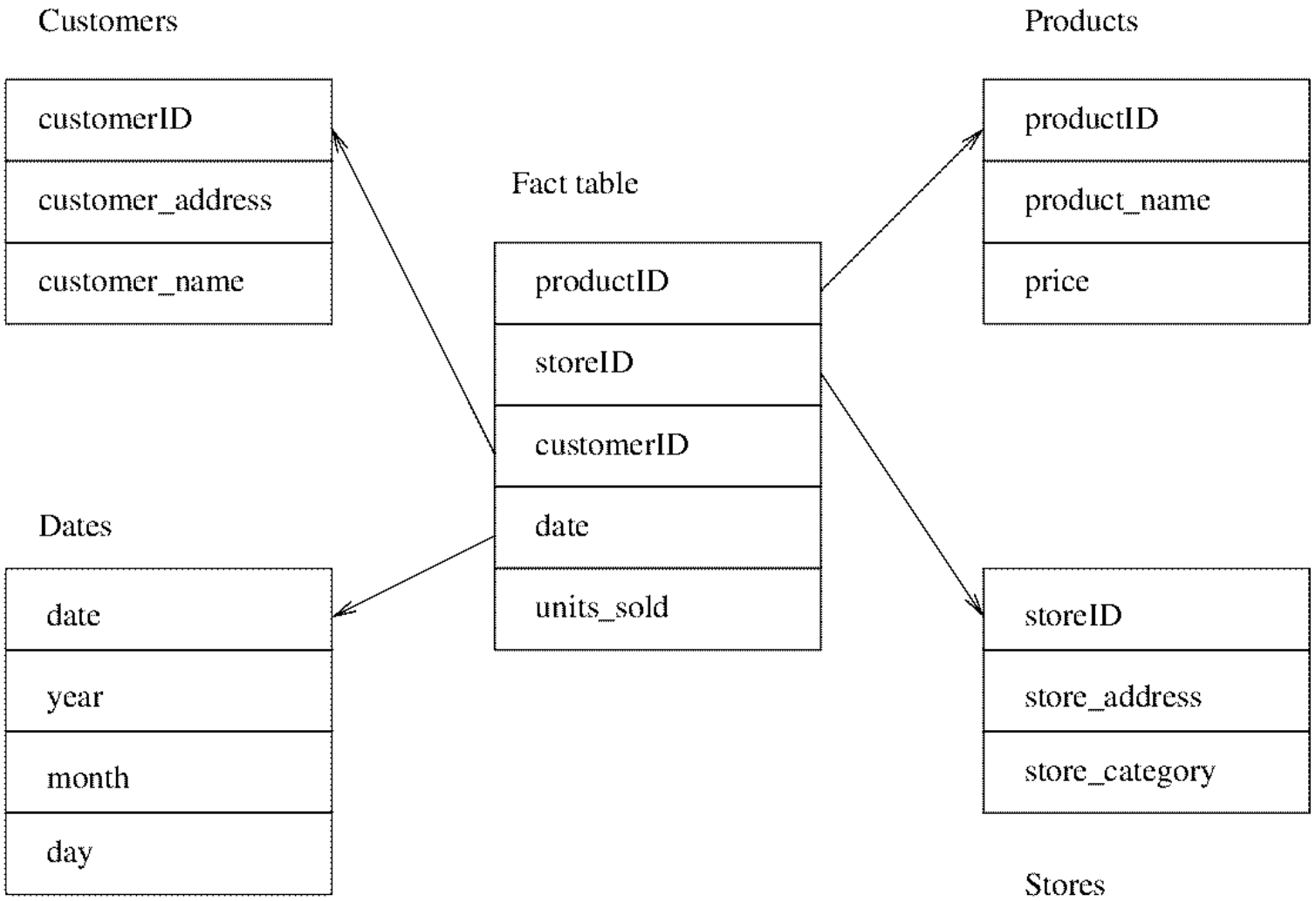
- Data explosion (alleviated by sparse representation techniques)
- The technology supporting them is not as matured as RDBMSes.

Therefore multidimensional databases are often represented by relational databases, for instance using the *star scheme*.

At the corners of the star are the *dimension tables*. At the center is the *fact table*.

Hierarchies in the domains of the independent attributes can be encoded in the design by decomposition of tables, obtaining a *snowflake scheme*.

Star Scheme


Customers
customerID
customer_address
customer_name
Products
productID
product_name
price
Fact table
productID
storeID
customerID
date
units_sold
Dates
date
year
month
day
storeID
store_address
store_category
Stores

SQL Queries for OLAP

Since SQL 1999 the GROUP BY clause can specify grouping schemes.

- GROUP BY GROUPING SETS ((col_name [,col_name]*)
  [, (col_name [,col_name]*)]*)

  Providing the sets of attributes to group on.

- GROUP BY ROLLUP (col_name [,col_name]*)

  Group on all the initial sublists of the given one.

- GROUP BY CUBE (col_name [,col_name]*)

  Group on all possible subsets of the given one.

Example

```
SELECT sid, cid, AVG(grade) AS avg_grade
    FROM Results
    GROUP BY GROUPING SETS ((sid), (cid))
```

Result

sid	cid	grade
0251563	327456	1
0251563	327564	2
0245654	327456	1

sid	cid	avg_grade
0251563	NULL	1.5
0245654	NULL	1.0
NULL	327456	1.0
NULL	327564	2.0

Summary on DW and OLAP

- A *data warehouse* is a collection of integrated, nonvolatile enterprise data, which is often restructured, pre-analyzed, and reflect evolutionary processes of the enterprise.
- Stages of loading DWs, design principles for DWs.
- OLAP, multidimensional view of data.
- ROLAP, start- and snowflake schemes.
- SQL queries for OLAP applications.

Information Retrieval Systems

The purpose of IR systems is to identify and rank documents from a large class (e.g. documents on the WWW) that contain certain (boolean combination of) keywords. At the heart of an IR system is a database, which is an (inverted) index.

Important stages of building and running an IR system:

- Data acquisition
- Document processing
- Query processing
- Result ranking

Data Acquisition

Human powered:

- authors submit their web-pages URLs.
- editors collect information from the web.

Automated:

- Autonomous software agents collect information from the web.

The software agents submit documents they find on the web to the document processing subsystem.

They may act also as an initial filter on the documents (e.g. extracting keywords from PDF, PS, PPT, etc. files).

Software Agents

An agent downloads web-pages (only the text part) or PDF/PS/etc. documents and does quick analysis on them: e.g. locate further URLs, search for keywords that can be indexed, etc.

Usually they use some kind of recursive algorithm; they control the depth of search and avoid circular trips.
Typical applications:

- indexing web pages,
- finding dead links on web pages,
- accumulating popularity information of web sites,
- check update/growth of web-sites, etc.

Document Processing

Document processing phases:

- Preprocessing: transforming the input document into some standard internal format; breaking down the document into retrievable units.
- Tokenization: identifying indexable elements (keywords, phrases, etc.).
- Stop word elimination (a, an, the, is, are, for, of, etc.).
- Term stemming: produce word roots from the keyword candidates of the document.

Document Processing

Document processing phases continued:

- Extracting indexed terms, frequencies, named entities, categories.
- Weight assignments to indexed terms: usually $TF \cdot \log(N/IDF)$ (with possibly length normalization) where N is the number of documents in the database.
- Updating the database with the so obtained keywords/phrases and the assigned weights.

Query Processing

Many phases are analogous to ones in document processing.

- Query tokenization: usually the separators are whitespaces of punctuation marks.
- Control keyword detection: AND, OR, NEAR, etc.
- (The query can possibly be processed at this point.)
- Stop word removal and term stemming.

Query Processing

Query processing phases continued:

- (The query can possibly be processed at this point.)
- Query expansion: adding synonyms, generalization and specialization of terms.
- Query term weighing (e.g. first terms in the query are usually more significant).
- Query processing.

Result Ranking

The task is to assign scores to the query results that reflect their relevance for the given query.

They usually regard:

- presence/absence, weights, proximity, location of keywords in the result document,
- size, age, metadata of the result document,
- links pointing outward from and links point to the result document.

Ranking w.r.t. the last aspect is usually done by the HITS (Hyperlink Induced Topic Search) algorithm.

The HITS Algorithm

The basic principles are

- If a document A has a link to document B then the author of A considers B as a valuable information source.
- If A points to a lot of "good" documents then the opinion of the author of A is more valuable.

A document which points to many others is a good *hub*, and a document to which many others point is a good *authority*.

The HITS Algorithm

1. Fixing a root set of relevant documents for a query.

2. Extend the root set with all documents from the database that point to a document in the root set and to which documents of the root set point.

3. For each document P set $H_P = 1$, $A_P = 1$.

4. Set for each P: $H'_P = \sum_{P \to S} A_S$, $A'_P = \sum_{S \to P} H_S$.

5. Let $H = \max_P H'_P$, $A = \max_P A'_P$, and set $H_P = H'_P/H$, $A_P = A'_P/A$ for each P.

6. Repeat from step 4. until some convergence threshold is reached.

Summary of IR systems

- Data acquisition (software agents)
- Document processing
- Query processing
- Result ranking (HITS algorithm)