Data Warehousing, OLAP, and Data Mining

Marek Rychly
mrychly@strathmore.edu

Strathmore University,
@iLabAfrica

&

Brno University of Technology,
Faculty of Information Technology

Advanced Databases and Enterprise Systems
21 October 2015
1 **Data warehousing**
   - Data Warehouses and RDBMS/OLTP Systems
   - Architecture of Data Warehouses
   - Design of Data Warehouses

2 **Online Analytical Processing (OLAP)**
   - OLAP Operations
   - OLAP Tools

3 **Data Mining**
   - Data Mining Techniques
Relational DBM/OLTP systems focus on current (valid) data. (OLTP = Online Transaction Processing, i.e. for transaction-oriented applications)

However, for analysis and decision, historical data are needed. (analysis to detect and describe trends, correlations, etc.; to support decisions)

Decision-makers need data warehouses and analysis tools: Online Analytical Processing (OLAP) and data mining tools (the warehouse holds data that is historical, detailed, and summarized to various levels and rarely subject to change other than being supplemented with new data)

OLTP is generally regarded as unsuitable for data warehousing. (OLTP systems are designed to maximize the transaction processing capacity)

In a company, there are usually several OLTP systems for different business processes, but just a single data warehouse
  • to answer queries that are ad-hoc, unstructured, and heuristic;
  • to process few of transactions that are unpredictable in nature.
Data warehousing is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of the decision-making process.

- **Subject-oriented** – focused on the major subjects of the enterprise rather than the major application areas. (on customers and products, rather than on customer invoicing and stock control)

- **Integrated** – merging different source data from different enterprise-wide applications systems. (to present decision-support data rather than application-oriented data)

- **Time-variant** – storing data accurate and valid at some point in time or over some time interval. (usually storing a series of snapshots of source application data)

- **Non-volatile** – does not update data in real time but refreshes from operational systems on a regular basis. (continually absorbs new data, incrementally integrating it with the previous data)
OLTP and Data Warehousing Systems

OLTP systems
- Holds current data
- Stores detailed data
- Data is dynamic
- Repetitive processing
- High level of trans. throughput
- Predictable pattern of usage
- Transaction-driven
- Application-oriented
- Supports day-to-day decisions
- Serves large number of clerical/operational users

Data warehousing systems
- Holds historical data
- Stores detailed, lightly, and highly summarized data
- Data is largely static
- Ad-hoc, unstructured, and heuristic processing
- Medium to low level of trans. throughput
- Unpredictable pattern of usage
- Analysis driven
- Subject-oriented
- Supports strategic decisions
- Serves relatively low number of managerial users
Problems of Data Warehousing

- **Hidden problems with source systems**
  (problems in the extract-transform-load/ETL process are difficult to detect)

- **Underestimation of resources for data loading and storage**
  (the ETL process can take a long time and a large amounts of disk space)

- **Required data not captured**
  (ad-hoc queries may require data that are not available in a warehouse)

- **Increased end-user demands**
  (users are requesting answers to more and more complex queries)

- **Data ownership**
  (sensitive data may be made accessible after loading into a warehouse)

- **High maintenance**
  (modifications in the source systems may affect the data warehouse)

- **Long-duration projects**
  (building of a large warehouse can take up to three years)

- **Complexity of integration**
  (integration of the input data as well as tools for ETL is quite difficult)
Architecture and Information Flows of DW

Architecture Components of Data Warehouse

- **Operational Data Store (ODS)**
  (a repository of integrated operational data to be moved into the warehouse)

- **Load manager**
  (extracts data from data sources or ODS and loads data into the warehouse)

- **Warehouse manager**
  (ensures data consistency, transformation, and merging, sets indexes and views, generation of de-normalizations and aggregations, backing-up and archiving data)

- **Query manager**
  (manages user queries using vendor end-user data access tools, data warehouse monitoring tools, database facilities, and custom-built programs)

- **End-user access tools**
  (tools for reporting, query, OLAP and data mining tools)

- **Data mart**
  (a warehouse part for one business obj. of a particular dept. or business function)
Data Flows in Data Warehouse

- **Inflow** – extraction, cleansing, and loading of the source data.
  (from operational data store to a warehouse by the load manager)

- **Upflow** – summarizing, packaging, and distribution of stored data.
  (the evaluation of detailed data in the warehouse manager to get lightly or highly
  summarized/aggregated data that can speed up the performance of queries)

- **Metaflow** – synchronizing data and their meta-data
  (inside the warehouse manager to describe stored data and their I/O flows)

- **Downflow** – archiving and backing-up the data in the warehouse.
  (from the warehouse manager to a backup storage)

- **Outflow** – answering ad-hoc queries or delivering reports.
  (from the warehouse manager to end-users by the end-user access tools)
  - to query tools and reporting – ad-hoc queries and regular reports
  - to app. development tools – in-house decision support systems
  - to Executive Information System (EIS) tools – data overview
  - to Online Analytical Proc. (OLAP) – retrospective multi-dim. queries
  - to data mining tools – to create predictive models
A relational DBMS suitable for data warehousing must deal with

- **load performance** – loading source data in narrow time windows
- **load processing** – loading is a single, seamless unit of work
  (conv. data, filter, reformat, check integrity, store phy., index, and update meta-data)
- **data quality management** – data loaded from “dirty” sources
- **query performance** – answer large and complex ad-hoc queries
- **tera-byte scalability** – no arch. limitations to the size of db.
- **mass user scalability** – many concurrent users accessing db.
- **networked data warehouse** – large networks of warehouses
- **warehouse administration** – setting limits, accounting, optim.
- **integrated dimensional analysis** – support multi-dim. views
- **advanced query functionality** – advanced analytical ops.
### Dimensionality Modelling and Schemas

#### Definition (Dimensionality modelling)

Dimensionality modelling present the data in a standard, intuitive form that allows for high-performance access.

#### Definition (Star schema)

Star schema has a fact table containing factual data in the centre, surrounded by dimension denormalized tables with reference data.

#### Definition (Snowflake/Starflake schema)

Snowflake is a variant of the star schema with normalized dimension tables. Starflake schema is a mixture of star and snowflake schemas.

- Denormalized tables in a star schema are breaking normal forms. (trans. func. deps. required to fit all data of a fact dimension into one dim. table)
- Neither stars nor snow/starflakes cannot share a table between dims. (dimensions are interconnected only via a fact table in the centre, a star topology)
Example of a Snowflake Schema

(adopted from “Database Data Warehousing Guide, Oracle”)

Note: In the case of a star schema, data in tables “suppliers” and “countries” would be merged into denormalized tables “products” and “customers”, respectively.
Denormalized Dimensional Tables

- **Efficiency**
  (more efficient access by report writers and query tools, no additional joins)

- **Ability to handle changing requirements**
  (the fact table can be accessed from all dimensions, support ad-hoc user queries)

- **Extensibility**
  (unrestricted adding of new fact tables, new dimensions, new dim. attributes, etc.)

- **Ability to model common business situations**
  (easy to query facts in report writers, query tools, and other user interfaces)

- **Predictable query processing (analytical operations):**
  - drill down
    (adding more dimension attributes from within a single star schema)
  - drill across
    (linking separate fact tables together via the shared/conformed dimensions)
  - ...and others (see OLAP)
Design Methodology for Data Warehouses (1)

1. choose the business process (define a mart)
   (identify a discrete BP that will be described by facts; e.g., a property sale process)

2. choose the grain
   (decide exactly what are individual fact instances; e.g., individual sales)

3. identify and conform the dimensions
   (set the context for asking questions about the facts; conform exactly the same
dimension tables in multiple marts, i.e., merge them into one table and share it)

4. choose the facts
   (check facts in the fact tables to be numeric and additive\(^1\); remove non-additive fact
(e.g., rates computed from the other facts in the table) and the fact with different
granularity from the other facts in the table (e.g., aggregated facts for time periods);
replace non-numeric PKs by numeric surrogate PKs in dimension tables)

\(^1\)fully additive facts can be summed across any of the dimensions associated with
their fact table, semi-additive can be summed across some dimensions, but not all
Design Methodology for Data Warehouses (2)

5. store pre-calculations in the fact table
   (identify/utilize opportunities to use pre-calcs. in the fact table to simplify queries)

6. round out the dimension tables
   (add text descriptions to the dimensions to be intuitive and understandable)

7. choose the duration of the database
   (decide how far back in time the facts table goes)

8. track slowly changing dimensions
   (identify slowly changing dimensions/attributes\(^2\) in attributes and move them into standalone dimension tables, e.g., “buyer” dimension table has “address” attribute which can change over time as the buyer was moving, so the “address” has to be replaced by a FK of a new dimension “address” \(\Rightarrow\) will result into a snowflake)

9. decide query priorities and modes
   (consider physical design issues to optimize queries, utilize pre-stored summaries or aggregations, tune up indexing performance, set up administration and security)

\(^2\)slowly changing dimensions are attributes with values varying over time without controlling of their change and managing historical values, they are difficult to identify
Example on Data Warehouse Design (1)
an initial ER model with identified discrete business processes main entities

Example on Data Warehouse Design (2)

one of discrete business processes main entities with its context

Example on Data Warehouse Design (3)

an initial start schema with two fact tables and four conformed dimension tables

Example on Data Warehouse Design (4)
a part of the start schema (a mart) with a badly structured fact table

Example on Data Warehouse Design (5)

a part of the start schema (a mart) with a correctly structured fact table

Example on Data Warehouse Design (6)
the resulting dimensional model (the overall fact constellation)

Online Analytical Processing (OLAP)

Definition (Online Analytical Processing, OLAP)

OLAP is the process of the dynamic synthesis, analysis, and consolidation of large volumes of multi-dimensional data.

- OLAP applications must provide users with
  - multi-dimensional views of data
  - support for complex calculations
  - time intelligence

- multi-dimensional views of data enable users to analyse data across any dimension at any level of aggregation with equal functionality and ease
**Consolidation** (roll-up) involves the aggregation of data, across a particular domain or by complex expr. involving interrelated data. (e.g., branch offices rolled up to cities, and cities rolled up to countries)

**Drill-down** is the reverse of consolidation and involves displaying the detailed data that comprises the consolidated data. (e.g., countries drilled down to cities, cities drilled down to branches)

**Slicing and dicing** (pivoting) refers to the ability to look at the data from different viewpoints, often performed along a time axis in order to analyse trends and find patterns. (e.g., one slice of the revenue data may display all revenue generated per type of property within cities, another revenue generated by branch office within each city)
OLAP Roll Up/Drill Down Operations

(adopted from “Web-enabled OLAP Tutorial”)

Marek Rychly
Data Warehousing, OLAP, and Data Mining — ADES, 21 October 2015
OLAP Dice/Slice Operations

(adapted from “Web-enabled OLAP Tutorial”)
Data Warehouse Architecture

Information Sources

- Semistructured Sources
- Operational DB’s

extract
transform
load
refresh
e tc.

Data Warehouse
Server
(Tier 1)

Data Warehouse

OLAP Servers
(Tier 2)
e.g., MOLAP

serve

Data Marts

OLAP

Clients
(Tier 3)

serve

Query/Reporting
e.g., ROLAP

serve

Data Mining

(adopted from “Pranav Joshi: OLAP Servers”)
OLAP Tools and Data Warehouses

- Multi-dimensional OLAP (MOLAP) – data to analyse are stored in a DW where analyses are performed on request by OLAP tools.

- Relational OLAP (ROLAP) – a DW acts a meta-layer retrieving the data for analysis from a RDBMS as requested by OLAP tools. (does not create a static multi-dimensional data structure of a typical warehouse)

- Hybrid OLAP (HOLAP) – OLAP tools utilize and query data from both RDBMS and DW as necessary (it is most efficient and easy to implement, combine power of both approaches)

- Desktop OLAP (DOLAP) – data from RDBMS or MOLAP are deployed to individual client PCs that perform OLAP analysis (data can be pushed or pulled on demand from RDBMS and MOLAP)

OLAP tools have to support “dynamic sparse matrix handling”. (efficient storage and processing of models that can easily comprise millions of cell references, many of which may have no appropriate data at any one point in time)
OLAP Architectures

(adopted from “Pranav Joshi: OLAP Servers”)
## OLAP Architectures – Advantages and Disadvantages

<table>
<thead>
<tr>
<th>Type</th>
<th>Strength</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOLAP</td>
<td>- Fast</td>
<td>- Size limit (Overall &amp; within Dimension)</td>
</tr>
<tr>
<td></td>
<td>- Flexible Querying (within confine of Cube)</td>
<td>- &quot;Proprietary&quot; Database</td>
</tr>
<tr>
<td></td>
<td>- Write=Back capabilitie</td>
<td>- Cube Management</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Drill-thru/around limitations</td>
</tr>
<tr>
<td>ROLAP</td>
<td>- Scalable Data Volumes</td>
<td>- Admin overhead for Query performance (Aggregate Management)</td>
</tr>
<tr>
<td></td>
<td>- Scalable number of User</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Query Management Layer provide OLAP &quot;Open&quot; Database</td>
<td></td>
</tr>
<tr>
<td>HOLAP</td>
<td>- Theoretically best of both worlds</td>
<td>- Expense and Admin overhead for multiple products</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- plus Risks listed above</td>
</tr>
</tbody>
</table>

(adopted from “Ardijan Abu Hanifah: Apakah OLAP, MOLAP, ROLAP dan HOLAP itu?”)
Definition (Data mining)

Data mining is the process of extracting valid, previously unknown, comprehensible, and actionable information from large databases and using it to make crucial business decisions.

- finding hidden\&unexpected patterns and relationships in data sets
- starts by developing an optimal representation of the structure of sample data, during which time knowledge is acquired
- this knowledge is then extended to larger sets of data (the assumption that the larger data set has a structure similar to the sample data)
**Data Mining Techniques (1)**

- **Predictive modeling** uses “supervised learning” for construction of a model based on training data.
  - Classification – the model is a decision tree or a neural network (nodes represent questions/tests and edges represent answers/values)
  - Value prediction – the model is a (non-)linear regression function (attempts to fit a straight/curved line through a plot of linear/non-linear data)

- **Database segmentation/clustering** uses “unsupervised learning” to split real data into an unknown number of segments or clusters with high internal homogeneity and external heterogeneity (i.e., homogeneous segments of similar records sharing a number of properties)
**Data Mining Techniques (2)**

- **Link analysis** to find relationships between data
  - Association discovery – finding and applying association rules
    (finds items that imply the presence of other items in the same event)
  - Sequential pattern discovery
    (finds patterns between events such that the presence of one set of items is followed by another set of items in a database of events over a period of time)
  - Similar time sequence discovery
    (discovers of links between two sets of data that are time-dependent)

- **Deviation detection** to express deviation from some previously known expectation and norm
  - by statistical analysis (e.g. by regression)
  - by (graphical) visualization of its results
Example on Classification by a Decision Tree

Customer renting property > 2 years?

No
Rent property

Yes
Customer age > 25 years?

No
Rent property

Yes
Buy property

Example on Classification by a Neural Network

Example of Database Segmentation/Clustering

Example of Deviation Detection by Visualization

DWs store subject-oriented data at various levels of aggregation.

To store and efficiently process data, DWs use a star schema.

Data stored in the warehouses can be processed by OLAP tools. (typically by roll-up/drill-down and slice/dice operations)

Data mining is used to find hidden knowledge on the data. (to identify patterns and relationships, to predict future, rather than analyse past)

In the next lecture:

- Post-relational, Non-relational, and NoSQL Databases (multimedia, spatial, and object dbs., distributed db. issues, NoSQL dbs., etc.)
Thank you for your attention!

Marek Rychly
<mrychly@strathmore.edu>