An Overview of Data Warehousing and OLAP Technology

Slides modified by Marie (original: Otto Bian)
Discussion: Juntong
Motivation

• **Data is used to make decisions**
  - Rapid grow of data, operational data and facts
  - Data is usually in different databases and in different physical places.

• Need for accessible, precise and comprehensive data
• Fast access regardless of the size of data
• Call for historical analysis of data
• **Goal:** provide support for decision-making rather than processing daily transactions
Decision Support

- Computerized information systems that support decision-making
- Need for historical, summarized and consolidated data from heterogeneous sources
- **Goal:** Support knowledge workers with decision making
  - Traditional DBMSs targeted for OLTP (on-line transaction processing) not suitable for this
Data Warehouse

“subject-oriented, integrated, time-varying, non-volatile collection of data that is used primarily in organizational decision making.”
<table>
<thead>
<tr>
<th></th>
<th>OLTP</th>
<th>OLAP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Users</strong></td>
<td>Clerk, IT professional</td>
<td>Knowledge worker</td>
</tr>
<tr>
<td><strong>Function</strong></td>
<td>Day to day operations</td>
<td>Decision support</td>
</tr>
<tr>
<td><strong>DB Design</strong></td>
<td>Application-oriented</td>
<td>Subject-oriented</td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td>Current, up-to-date detailed.</td>
<td>Historical, summarized, multidimensional, …</td>
</tr>
<tr>
<td><strong>Usage</strong></td>
<td>Repetitive</td>
<td>Ad-hoc</td>
</tr>
<tr>
<td><strong>Access</strong></td>
<td>Read/write</td>
<td>Lots of scans</td>
</tr>
<tr>
<td><strong>Unit of work</strong></td>
<td>Short, simple transaction</td>
<td>Complex query</td>
</tr>
<tr>
<td># rec accessed</td>
<td>Tens</td>
<td>Millions</td>
</tr>
<tr>
<td># users</td>
<td>Thousands</td>
<td>Hundreds</td>
</tr>
<tr>
<td>DB size</td>
<td>100 MB- GB</td>
<td>100 GB-TB</td>
</tr>
<tr>
<td>Metric</td>
<td>Transaction throughput</td>
<td>Query throughput</td>
</tr>
</tbody>
</table>
Discussion (in pairs)

Now that we have discussed the differences between OLTP and OLAP.

- What are some real-world use cases of OLTP and OLAP?
- Which types of businesses require more out of OLAP vs OLTP?
Figure 1. Data Warehousing Architecture
Multidimensional data

Figure 2. Multidimensional data
OLAP Architecture

Figure 1. Data Warehousing Architecture
Multidimensionality is core to facilitate complex analyses and visualizations.

**Star Schema**

- **Fact table** has a pointer to each of the dimensions (acts as a multidimensional coordinate with numerical measures).
- **Dimension tables** store attributes of the dimension.
- Fact table connects to all dimension tables with a multiple join.

---

**Figure 3. A Star Schema.**
Materialized Views

• Database object that contains the results of a query

• Challenges
  • Understanding which views to materialize.
  • Understand how to use such views to answer queries.
  • Efficiently updating materialized views during load and refresh.

• Choice can depend on workload characteristics, update costs, storage requirements.
Metadata Requirements

• Reflection upon the use of data within the warehouse

• Administrative metadata
  • E.g. description of the source database

• Business metadata
  • E.g. business terms and definitions

• Operational metadata
  • Monitoring information

Figure 1. Data Warehousing Architecture
Summary

• Data Warehouse for decision support
• Focuses on complex querying over large volume of data (OLAP)
• N-dimensional rather than relational table (Cube)
• Importance of metadata managing
• Possible to analyze trends (historical analysis)
• Challenging to develop efficient query processing (Materialized Views)
Discussion (in groups of ~4)

• How does the heterogeneity in data warehouses differ from the topics that we've discussed in data integration?

• What are some applications that you would use data integration for? What about a data warehouse?
  • Can you think of any applications for which both would be a good solution?
Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals

Slides modified by Marie (original: Jim Cao)
Discussion: Juntong
Data Analysis Applications

• Looking for anomalies, unusual patterns
• 4 steps:
  • Formulating a Query
  • Extracting Aggregated Data
  • Visualizing the Results
  • Analyzing the Results
• Goal: categorization of data values and trends, statistical information, contrast one category with another
Discussion (in pairs)

This paper is a technical report from Microsoft.

• What is the purpose (or motivations) of such technical reports from the industry?

• Who are the targeted audience?

• What are some potential takeaways from this type of paper?
Dimensionality Reduction

• Dimensionality reduction in data visualization for better comprehensibility
• Represent N-dimension data in 2- or 3-D

• Example – Car Sale:
  • Many different information: date of sale, sales company, color of car, model of car, year of car, etc.
  • Analyze subset of these attributes (e.g. color, model)
    → More manageable, focused dataset
Relational Representation

• 2D flat files model an N-dimensional problem as a relation with N-attribute domains.
However, consider...

- Data aggregated at a coarse level and then finer levels
  - **Rolling-up:** Going up the levels
  - **Drilling-down:** Going down the levels

- Table 3.a: Aggregated data at 3 distinct levels with subtotals
  - Model then Year then Color
- Sales rolled up by using totals and subtotals

- **Problems:**
  - Not relational data – the empty cells (NULL values) cannot form a key.
  - Exponential increase in number of aggregation columns when rolling up

- **Challenge with SQL:**
  - GROUP BY operator does not allow a direct construction of histograms
  - Possible inelegant solution: Multiple separate queries with different GROUP BY clauses

### Table 3.a: Sales Roll Up by Model by Year by Color

<table>
<thead>
<tr>
<th>Model</th>
<th>Year</th>
<th>Color</th>
<th>Sales by Model by Year by Color</th>
<th>Sales by Model by Year</th>
<th>Sales by Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>black</td>
<td>50</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>white</td>
<td>40</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1995</td>
<td>black</td>
<td>85</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>white</td>
<td>115</td>
<td>200</td>
<td>290</td>
</tr>
</tbody>
</table>

Not relational
Date’s Alternative

• Displays aggregated data without additional columns for each roll-up level
  → Combines the data different levels

• Problems:
  • Enormous number of domains
  • Naming problems
  • Very long names
  • Same SQL challenges

Table 3.b: Sales Roll-Up by Model by Year by Color as recommended by Chris Date [Date1].

<table>
<thead>
<tr>
<th>Model</th>
<th>Year</th>
<th>Color</th>
<th>Sales</th>
<th>Sales by Model by Year</th>
<th>Sales by Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>black</td>
<td>50</td>
<td>90</td>
<td>290</td>
</tr>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>white</td>
<td>40</td>
<td>90</td>
<td>290</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>black</td>
<td>85</td>
<td>200</td>
<td>290</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>white</td>
<td>115</td>
<td>200</td>
<td>290</td>
</tr>
</tbody>
</table>
Pivot Table

- Excel pivot table alternative to the traditional roll-up method (Table 3.a)
  - Transposes spreadsheet
  - Aggregate cells based on cell values

- Problem:
  - Size of the Pivot Table
  - N and M values → pivot table has N x M values
    - Many columns
    - Obtuse column

Table 4: An Excel pivot table representation of Table 3 with Ford sales data included.

<table>
<thead>
<tr>
<th>Model</th>
<th>black</th>
<th>white</th>
<th>1994 Total</th>
<th>1995 Total</th>
<th>Grand Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chevy</td>
<td>50</td>
<td>40</td>
<td>90</td>
<td>85</td>
<td>115</td>
</tr>
<tr>
<td>Ford</td>
<td>50</td>
<td>10</td>
<td>60</td>
<td>85</td>
<td>75</td>
</tr>
<tr>
<td>Grand Total</td>
<td>100</td>
<td>50</td>
<td>150</td>
<td>170</td>
<td>190</td>
</tr>
</tbody>
</table>
Data CUBE

• N-dimensional generalization of simple aggregate functions
• Data cube operator builds a table containing all these aggregate values
  • 0-D data cube: a point.
  • 1-D data cube: a line & a point.
  • 2-D data cube: a cross tabulation, a plane, two lines, and a point.
  • 3-D data cube: a cube with three intersecting 2D cross tabs

• Data Cube vs. SQL:
  • Simultaneous aggregation across multiple dimensions possible
  • More complex and flexible analyzes possible
The CUBE operator

- CUBE = relational operator
  - GROUP BY and ROLL UP: degenerate forms of the operator

- Power Set of Aggregation Columns
  - All possible subsets of grouping columns

- How does a CUBE operator work?
  - First Aggregate
  - Second Aggregate with UNIONS
  - Possible super-aggregate Values
  - Super-aggregates through ROLLUP

- Super-aggregate → aggregated data that summarizes multiple levels of data hierarchies

```
SELECT Model, Year, Color, SUM(Sales) AS Sales
FROM Sales
WHERE Model in ['Ford', 'Chevy']
  AND Year BETWEEN 1994 AND 1995
GROUP BY CUBE Model, Year, Color;
```
```sql
SELECT Model, Year, Color, SUM(Sales) AS Sales
FROM Sales
WHERE Model in ['Ford', 'Chevy']
AND Year BETWEEN 1994 AND 1995
GROUP BY CUBE Model, Year, Color;
```

Figure 4: A 3D data cube (right) built from the table at the left by the **CUBE** statement at the top of the figure.
Discussion (in groups of 4)

The abstract mentions that “many of the features are being added to the SQL standard”.

• Does this strike you as a big or a small change to SQL?

• How do we decide if a new feature should be added to the standard? What are the deciding factors?
Data Cubes - Summary

• Generalizes and unifies aggregates, group by, histograms, roll-ups and drill-downs and cross tabs.

• Based on a relational representation of aggregate data

• Easy to compute for a wide class of functions

• Flexible and dynamic form of data analysis