MTAT.03.183
Data Mining

Week 7: Online Analytical Processing and Data Warehouses
Marlon Dumas
marlon.dumas ät ut . ee

Acknowledgment
• This slide deck is a “mashup” of the following publicly available slide decks:
  - http://www.postech.ac.kr/~swhwang/grass/DataCube.ppt
  shani/DataCube/CubeNotesKerschberg.ppt
  - http://ohr.gsfc.nasa.gov/wfstatistics/Data_Cube_Traini
  ng.ppt
  - http://www.cs.uiuc.edu/homes/hanj/bk2/03.ppt

Outline
• The “data cube” abstraction
• Multidimensional data models
• Data warehouses

Typical Data Analysis Process
• Formulate a query to extract relevant information
• Extract aggregated data from the database
• Visualize the result to look for patterns.
• Analyze the result and formulate new queries.
• Online Analytical Processing (OLAP) is about supporting such processes
• OLAP characteristics: No updates, lots of aggregation, need to visualize and to interact
• Let’s first talk about aggregation…

Relational Aggregation Operators
• SQL has several aggregate operators:
  - SUM(), MIN(), MAX(), COUNT(), AVG()
• The basic idea is:
  - Combine all values in a column into a single scalar value
• Syntax
  - SELECT AVG(T emp) FROM Weather;

The Relational GROUP BY Operator
• GROUP BY allows aggregates over table sub-groups
  - SELECT Time, Altitude, AVG(T emp) FROM Weather GROUP BY Time, Altitude;

<table>
<thead>
<tr>
<th>Time</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Altitude (m)</th>
<th>Temp</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/9/5:1500</td>
<td>20</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/9/5:1500</td>
<td>20</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/9/5:1500</td>
<td>19</td>
<td>17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/9/9:1500</td>
<td>50</td>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07/9/9:1500</td>
<td>50</td>
<td>21</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Limitations of the GROUP BY

- Group-by is one-dimensional: one group per combination of the selected attribute values
- Does not give sub-totals

<table>
<thead>
<tr>
<th>Model</th>
<th>Year</th>
<th>Color</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>Black</td>
<td>50</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>Black</td>
<td>85</td>
</tr>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>White</td>
<td>40</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>White</td>
<td>115</td>
</tr>
</tbody>
</table>

1. Calculate total sales per year
2. Compute total sales per year and per color
3. Calculate sales per year, per color and per model

Grouping with Sub-Totals (Pivot table)

- Sales by Model by Year by Color

<table>
<thead>
<tr>
<th>Sales by Model by Year by Color</th>
<th>Year</th>
<th>Color</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chevy 1994 Black</td>
<td>50</td>
<td>Black</td>
<td></td>
</tr>
<tr>
<td>Chevy 1995 Black</td>
<td>85</td>
<td>Black</td>
<td></td>
</tr>
<tr>
<td>Chevy 1994 White</td>
<td>40</td>
<td>White</td>
<td></td>
</tr>
<tr>
<td>Chevy 1995 White</td>
<td>115</td>
<td>White</td>
<td></td>
</tr>
</tbody>
</table>

Note that sub-totals by color are missing, if added it becomes a cross-tabulation

Grouping with Sub-Totals (Relational version)

Sub-totals by color are still missing...

SQL Query

```sql
SELECT 'ALL', 'ALL', 'ALL', SUM(Sales)
FROM Sales
GROUP BY Model = 'Cherry'
UNION
SELECT Model, ALL, ALL, SUM(Sales)
FROM Sales
GROUP BY Year
UNION
SELECT Model, Year, ALL, SUM(Sales)
FROM Sales
GROUP BY Color
```

Adding the colors...

```sql
SELECT 'ALL', 'ALL', 'ALL', SUM(Sales)
FROM Sales
GROUP BY Model = 'Cherry'
UNION
SELECT Model, Year, ALL, SUM(Sales)
FROM Sales
GROUP BY Color
```
Cube: Each Attribute is a Dimension

- N-dimensional Aggregate (sum(), max(),...)
  - Fits relational model exactly:
    - \( a_1, a_2, ..., a_n, f() \)
  - Super-aggregate over \( N-1 \) Dimensional sub-cubes
    - \( \text{ALL, } a_2, ..., a_n, f() \)
    - \( ... \)
    - \( a_1, a_2, ..., \text{ALL, ALL, } f() \)
    - This is the \( N-1 \) Dimensional cross-tab.
- Super-aggregate over \( N-2 \) Dimensional sub-cubes
  - \( \text{ALL, ALL, } a_3, ..., a_n, f() \)
  - \( ... \)
  - \( \text{a_1, a_2, ..., ALL, ALL, f() \)  

The Data Cube Concept

CUBE Operator

Possible syntax

- Proposed syntax example:
  - SELECT Model, Make, Year, SUM(Sales) FROM Sales WHERE Model IN ("Chevy", "Ford") AND Year BETWEEN 1990 AND 1994 GROUP BY CUBE Model, Make, Year HAVING SUM(Sales) > 0;
  - Note: GROUP BY operator repeats aggregate list
    - in select list
    - in group by list

Sub-cube Derivation

Dimension collapse, * denotes ALL
Rollup Operator

ROLLUP Operator: special case of CUBE Operator
Return “Sales Roll Up by Store by Quarter” in 1994:
```
SELECT Store, quarter, SUM(Sales)
FROM Sales
WHERE nation="Korea" AND Year=1994
GROUP BY ROLLUP Store, Quarter(Date) AS quarter;
```

Cube Operator Example

```
<table>
<thead>
<tr>
<th>Model Year</th>
<th>Color</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chevy</td>
<td>1990</td>
<td>red</td>
</tr>
<tr>
<td></td>
<td></td>
<td>white</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue</td>
</tr>
<tr>
<td>Chevy</td>
<td>1991</td>
<td>red</td>
</tr>
<tr>
<td></td>
<td></td>
<td>white</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue</td>
</tr>
<tr>
<td>Chevy</td>
<td>1992</td>
<td>red</td>
</tr>
<tr>
<td></td>
<td></td>
<td>white</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue</td>
</tr>
<tr>
<td>Ford</td>
<td>1990</td>
<td>red</td>
</tr>
<tr>
<td></td>
<td></td>
<td>white</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue</td>
</tr>
<tr>
<td>Ford</td>
<td>1991</td>
<td>red</td>
</tr>
<tr>
<td></td>
<td></td>
<td>white</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue</td>
</tr>
<tr>
<td>Ford</td>
<td>1992</td>
<td>red</td>
</tr>
<tr>
<td></td>
<td></td>
<td>white</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue</td>
</tr>
<tr>
<td>Chevy</td>
<td>1992</td>
<td>red</td>
</tr>
<tr>
<td></td>
<td></td>
<td>white</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue</td>
</tr>
<tr>
<td>Ford</td>
<td>1990</td>
<td>red</td>
</tr>
<tr>
<td></td>
<td></td>
<td>white</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue</td>
</tr>
<tr>
<td>Ford</td>
<td>1991</td>
<td>red</td>
</tr>
<tr>
<td></td>
<td></td>
<td>white</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue</td>
</tr>
<tr>
<td>Ford</td>
<td>1992</td>
<td>red</td>
</tr>
<tr>
<td></td>
<td></td>
<td>white</td>
</tr>
<tr>
<td></td>
<td></td>
<td>blue</td>
</tr>
</tbody>
</table>
```

Summary

- Problems with GROUP BY
  - GROUP BY cannot directly construct
    - Pivot tables / roll-up reports
    - Cross-Tabs
  - CUBE Operator
    - Generalizes GROUP BY and Roll-Up and Cross-Tabs!!

Now let’s have a look at one...

- NASA Workforce cubes
  [http://nasapeople.nasa.gov/workforce](http://nasapeople.nasa.gov/workforce)
- Btell demo reports
  - [http://www.btell.de](http://www.btell.de)
  - Follow the “demo” link and start a demo, then go to reports

Multidimensional Data

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

Multidimensional Data Modeling

- Key concepts
  - Dimensions: entities along which data can be aggregated (e.g. Car model, time, color)
  - Measures: scalar values to be aggregated (e.g. number of sales)
- Two types of tables
  - Dimension tables
  - Fact table: stores measures and references to dimensions
**OLTP vs. OLAP**

- **OLTP** – Online Transaction Processing
  - Traditional database technology
  - Many small transactions (point queries: UPDATE or INSERT)
  - Avoid redundancy, normalize schemas
  - Access to consistent, up-to-date database
- **OLTP Examples:**
  - Flight reservation
  - Banking and financial transactions
  - Order Management, Procurement, ...
- **Extremely fast response times...**

**OLTP vs. OLAP**

- **OLAP** – Online Analytical Processing
  - Big aggregate queries, no Updates
  - Redundancy a necessity (Materialized Views, special-purpose indexes, de-normalized schemas)
  - Periodic refresh of data (daily or weekly)
- **OLAP Examples**
  - Decision support (sales per employee)
  - Marketing (purchases per customer)
  - Biomedical databases
- **Goal:** Response Time of seconds / few minutes

**OLTP vs. OLAP (Water and Oil)**

- Lock Conflicts: OLAP blocks OLTP
- Database design:
  - OLTP normalized, OLAP de-normalized
- Tuning, Optimization
  - OLTP: inter-query parallelism, heuristic optimization
  - OLAP: intra-query parallelism, full-fledged optimization
- Freshness of Data:
  - OLTP: serializability
  - OLAP: reproducibility
- Integrity:
  - OLTP: ACID
  - OLAP: Sampling, Confidence Intervals

**Solution: Data Warehouse**

- Special Sandbox for OLAP
- Data input using OLTP systems
- Data Warehouse aggregates and replicates data (special schema)
- New Data is periodically uploaded to Warehouse
**DW Architecture**

- **OLTP**
  - OLTP Applications
  - DB1
  - DB2
  - DB3

- **OLAP**
  - GUI, Spreadsheets
  - Data Warehouse

**DW Products and Tools**

- Oracle 11g, IBM DB2, Microsoft SQL Server, ...
  - All provide OLAP extensions
- SAP Business Information Warehouse
  - ERP vendors
- MicroStrategy, Cognos (now IBM)
  - Specialized vendors
  - Kind of Web-based EXCEL
- Niche Players (e.g., Btell)
  - Vertical application domain

**Reference (highly recommended)**

- [http://citeseer.ist.psu.edu/old/392672.html](http://citeseer.ist.psu.edu/old/392672.html)
- Data Warehousing chapter of Jianwei Han’s textbook (chapter 3)

**Homework**

- Exercises 1 and 4 at:
  - [http://www.systems.ethz.ch/education/courses/fs09/data-warehousing/ex2.pdf](http://www.systems.ethz.ch/education/courses/fs09/data-warehousing/ex2.pdf)
- Multidimensional data modeling exercise in course Wiki pages