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](http://www.postech.ac.kr/~swhwang/grass/DataCube.ppt)
  - [http://www.cs.uiuc.edu/homes/hanj/bk2/03.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(Temp) FROM Weather;
The Relational GROUP BY Operator

- GROUP BY allows aggregates over table sub-groups
  
  ```
  SELECT Time, Altitude, AVG(Temp)
  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>...</td>
<td>...</td>
<td>20</td>
<td>24</td>
</tr>
<tr>
<td>07/9/5:1500</td>
<td>...</td>
<td>...</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td>07/9/5:1500</td>
<td>...</td>
<td>...</td>
<td>100</td>
<td>17</td>
</tr>
<tr>
<td>07/9/9:1500</td>
<td>...</td>
<td>...</td>
<td>50</td>
<td>19</td>
</tr>
<tr>
<td>07/9/9:1500</td>
<td>...</td>
<td>...</td>
<td>50</td>
<td>21</td>
</tr>
<tr>
<td>07/9/5:1500</td>
<td>...</td>
<td>...</td>
<td>20</td>
<td>23</td>
</tr>
<tr>
<td>07/9/5:1500</td>
<td>...</td>
<td>...</td>
<td>100</td>
<td>17</td>
</tr>
<tr>
<td>07/9/9:1500</td>
<td>...</td>
<td>...</td>
<td>50</td>
<td>20</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></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Chevy</td>
<td>50</td>
<td>40</td>
<td>90</td>
<td>85</td>
<td>115</td>
<td>200</td>
<td>290</td>
</tr>
<tr>
<td>Ford</td>
<td>50</td>
<td>10</td>
<td>60</td>
<td>85</td>
<td>75</td>
<td>160</td>
<td>220</td>
</tr>
<tr>
<td>Grand total</td>
<td>100</td>
<td>50</td>
<td>150</td>
<td>170</td>
<td>190</td>
<td>360</td>
<td>510</td>
</tr>
</tbody>
</table>

- Note that sub-totals by color are missing, if added it becomes a cross-tabulation
**Grouping with sub-totals (cross-tab)**

<table>
<thead>
<tr>
<th>Chevy</th>
<th>1994</th>
<th>1995</th>
<th>Total (ALL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black</td>
<td>50</td>
<td>85</td>
<td>135</td>
</tr>
<tr>
<td>White</td>
<td>40</td>
<td>115</td>
<td>155</td>
</tr>
<tr>
<td>Total (ALL)</td>
<td>90</td>
<td>200</td>
<td>290</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Year</th>
<th>Color</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>Black</td>
<td>50</td>
</tr>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>White</td>
<td>40</td>
</tr>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>ALL</td>
<td>90</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>Black</td>
<td>85</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>White</td>
<td>115</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>ALL</td>
<td>200</td>
</tr>
<tr>
<td>Chevy</td>
<td>ALL</td>
<td>ALL</td>
<td>290</td>
</tr>
<tr>
<td>Chevy</td>
<td>ALL</td>
<td>Black</td>
<td>135</td>
</tr>
<tr>
<td>Chevy</td>
<td>ALL</td>
<td>White</td>
<td>155</td>
</tr>
</tbody>
</table>
Grouping with Sub-Totals (Relational version)

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

Sub-totals by color are still missing…

<table>
<thead>
<tr>
<th>Model</th>
<th>Year</th>
<th>Color</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>Black</td>
<td>50</td>
</tr>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>White</td>
<td>40</td>
</tr>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>ALL</td>
<td>90</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>Black</td>
<td>85</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>White</td>
<td>115</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>ALL</td>
<td>200</td>
</tr>
<tr>
<td>Chevy</td>
<td>ALL</td>
<td>ALL</td>
<td>290</td>
</tr>
</tbody>
</table>
SQL Query

```
SELECT 'ALL', 'ALL', 'ALL', SUM(Sales)
FROM Sales
WHERE Model = 'Chevy'
UNION
SELECT Model, 'ALL', 'ALL', SUM(Sales)
FROM Sales
WHERE Model = 'Chevy'
    GROUP BY Model
UNION
SELECT Model, Year, 'ALL', SUM(Sales)
    FROM Sales
WHERE Model = 'Chevy'
    GROUP BY Model, Year
UNION
SELECT Model, Year, Color, SUM(Sales)
    FROM Sales
WHERE Model = 'Chevy'
    GROUP BY Model, Year, Color;
```
Adding the colors...

<table>
<thead>
<tr>
<th>Model</th>
<th>Year</th>
<th>Color</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>Black</td>
<td>50</td>
</tr>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>White</td>
<td>40</td>
</tr>
<tr>
<td>Chevy</td>
<td>1994</td>
<td>ALL</td>
<td>90</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>Black</td>
<td>85</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>White</td>
<td>115</td>
</tr>
<tr>
<td>Chevy</td>
<td>1995</td>
<td>ALL</td>
<td>200</td>
</tr>
<tr>
<td>Chevy</td>
<td>ALL</td>
<td>ALL</td>
<td>290</td>
</tr>
</tbody>
</table>

*Table 5b.* Sales summary rows missing form Table 5a to convert the roll-up into a cube.

```sql
SELECT 'ALL', 'ALL', 'ALL', SUM(Sales)
FROM Sales
WHERE Model = 'Chevy'
UNION
SELECT Model, 'ALL', 'ALL', SUM(Sales)
FROM Sales
WHERE Model = 'Chevy'
GROUP BY Model
UNION
SELECT Model, Year, 'ALL', SUM(Sales)
FROM Sales
WHERE Model = 'Chevy'
GROUP BY Model, Year
UNION
SELECT Model, Year, Color, SUM(Sales)
FROM Sales
WHERE Model = 'Chevy'
GROUP BY Model, Year, Color;
```
CUBE and Roll Up Operators

The Data Cube and The Sub-Space Aggregates

Aggregate
Sum

Group By (with total)
By Color
RED WHITE BLUE
Sum

Cross Tab
Chevy Ford
By Color
RED WHITE BLUE
By Make
Sum

By Year
By Make & Year
By Color & Year
By Make & Color
Sum
The Cube

- An Example of 3D Data Cube
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
  - $\text{ALL, } a_2, ... , a_N$, $f()$
  - $a_3, \text{ALL, } a_3, ... , a_N$, $f()$
  - ...
  - $a_1, a_2, ... , \text{ALL}$, $f()$
  - This is the $N$-1 Dimensional cross-tab.
- Super-aggregate over $N$-2 Dimensional sub-cubes
  - $\text{ALL, ALL, } a_3, ... , a_N$, $f()$
  - ...
  - $a_1, a_2, ... , \text{ALL, ALL}$, $f()$
The Data Cube Concept
Sub-cube Derivation

Dimension collapse, * denotes ALL
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
Rollup Operator

- **ROLLUP Operator**: special case of CUBE Operator

Return “Sales Roll Up by Store by Quarter” in 1994:

```sql
SELECT Store, quarter, SUM(Sales)
FROM Sales
WHERE nation="Korea" AND Year = 1994
GROUP BY ROLLUP Store, Quarter(Date) AS quarter;
```
### Cube Operator Example

**SALES**

<table>
<thead>
<tr>
<th>Model</th>
<th>Year</th>
<th>Color</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chevy</td>
<td>1990</td>
<td>red</td>
<td>5</td>
</tr>
<tr>
<td>Chevy</td>
<td>1990</td>
<td>white</td>
<td>87</td>
</tr>
<tr>
<td>Chevy</td>
<td>1990</td>
<td>blue</td>
<td>62</td>
</tr>
<tr>
<td>Chevy</td>
<td>1991</td>
<td>red</td>
<td>54</td>
</tr>
<tr>
<td>Chevy</td>
<td>1991</td>
<td>white</td>
<td>95</td>
</tr>
<tr>
<td>Chevy</td>
<td>1991</td>
<td>blue</td>
<td>49</td>
</tr>
<tr>
<td>Chevy</td>
<td>1992</td>
<td>red</td>
<td>31</td>
</tr>
<tr>
<td>Chevy</td>
<td>1992</td>
<td>white</td>
<td>54</td>
</tr>
<tr>
<td>Chevy</td>
<td>1992</td>
<td>blue</td>
<td>71</td>
</tr>
<tr>
<td>Ford</td>
<td>1990</td>
<td>red</td>
<td>64</td>
</tr>
<tr>
<td>Ford</td>
<td>1990</td>
<td>white</td>
<td>62</td>
</tr>
<tr>
<td>Ford</td>
<td>1990</td>
<td>blue</td>
<td>63</td>
</tr>
<tr>
<td>Ford</td>
<td>1991</td>
<td>red</td>
<td>52</td>
</tr>
<tr>
<td>Ford</td>
<td>1991</td>
<td>white</td>
<td>9</td>
</tr>
<tr>
<td>Ford</td>
<td>1991</td>
<td>blue</td>
<td>55</td>
</tr>
<tr>
<td>Ford</td>
<td>1992</td>
<td>red</td>
<td>27</td>
</tr>
<tr>
<td>Ford</td>
<td>1992</td>
<td>white</td>
<td>62</td>
</tr>
<tr>
<td>Ford</td>
<td>1992</td>
<td>blue</td>
<td>39</td>
</tr>
</tbody>
</table>

**DATA CUBE**

<table>
<thead>
<tr>
<th>Model</th>
<th>Year</th>
<th>Color</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>ALL</td>
<td>ALL</td>
<td>942</td>
</tr>
<tr>
<td>chevy</td>
<td>ALL</td>
<td>ALL</td>
<td>510</td>
</tr>
<tr>
<td>ford</td>
<td>ALL</td>
<td>ALL</td>
<td>432</td>
</tr>
<tr>
<td>ALL</td>
<td>1990</td>
<td>ALL</td>
<td>343</td>
</tr>
<tr>
<td>ALL</td>
<td>1991</td>
<td>ALL</td>
<td>314</td>
</tr>
<tr>
<td>ALL</td>
<td>1992</td>
<td>ALL</td>
<td>285</td>
</tr>
<tr>
<td>ALL</td>
<td>ALL</td>
<td>red</td>
<td>165</td>
</tr>
<tr>
<td>ALL</td>
<td>ALL</td>
<td>white</td>
<td>273</td>
</tr>
<tr>
<td>ALL</td>
<td>ALL</td>
<td>blue</td>
<td>339</td>
</tr>
<tr>
<td>ALL</td>
<td>ALL</td>
<td>ALL</td>
<td>154</td>
</tr>
<tr>
<td>ALL</td>
<td>1990</td>
<td>ALL</td>
<td>199</td>
</tr>
<tr>
<td>ALL</td>
<td>1991</td>
<td>ALL</td>
<td>157</td>
</tr>
<tr>
<td>ALL</td>
<td>1992</td>
<td>ALL</td>
<td>189</td>
</tr>
<tr>
<td>ALL</td>
<td>1990</td>
<td>ALL</td>
<td>116</td>
</tr>
<tr>
<td>ALL</td>
<td>1991</td>
<td>ALL</td>
<td>128</td>
</tr>
<tr>
<td>chevy</td>
<td>ALL</td>
<td>red</td>
<td>91</td>
</tr>
<tr>
<td>chevy</td>
<td>ALL</td>
<td>white</td>
<td>236</td>
</tr>
<tr>
<td>chevy</td>
<td>ALL</td>
<td>blue</td>
<td>183</td>
</tr>
<tr>
<td>ford</td>
<td>ALL</td>
<td>red</td>
<td>144</td>
</tr>
<tr>
<td>ford</td>
<td>ALL</td>
<td>white</td>
<td>133</td>
</tr>
<tr>
<td>ford</td>
<td>ALL</td>
<td>blue</td>
<td>156</td>
</tr>
<tr>
<td>ALL</td>
<td>1990</td>
<td>red</td>
<td>69</td>
</tr>
<tr>
<td>ALL</td>
<td>1990</td>
<td>white</td>
<td>149</td>
</tr>
<tr>
<td>ALL</td>
<td>1990</td>
<td>blue</td>
<td>125</td>
</tr>
<tr>
<td>ALL</td>
<td>1991</td>
<td>red</td>
<td>107</td>
</tr>
<tr>
<td>ALL</td>
<td>1991</td>
<td>white</td>
<td>104</td>
</tr>
<tr>
<td>ALL</td>
<td>1991</td>
<td>blue</td>
<td>104</td>
</tr>
<tr>
<td>ALL</td>
<td>1992</td>
<td>red</td>
<td>59</td>
</tr>
<tr>
<td>ALL</td>
<td>1992</td>
<td>white</td>
<td>116</td>
</tr>
<tr>
<td>ALL</td>
<td>1992</td>
<td>blue</td>
<td>110</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

Dimensions: Product, Location, Time
Hierarchical summarization paths

- Industry
- Region
- Year
- Category
- Country
- Quarter
- Product
- City
- Month
- Week
- Office
- Day
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
Star Schema

Sales Fact Table
- time_key
- item_key
- branch_key
- location_key
- units_sold
- dollars_sold
- avg_sales

Time
- time_key
- day
- day_of_the_week
- month
- quarter
- year

Branch
- branch_key
- branch_name
- branch_type

Item
- item_key
- item_name
- brand_type
- supplier_type

Location
- location_key
- street
- city
- state_or_province
- country

Measure
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...

Carsten Binnig, ETH Zürich
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

Extract Transform Load (ETL)
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:

• Multidimensional data modeling exercise in course Wiki pages