What is a Data Warehouse

- A data warehouse:
  - archives information gathered from multiple sources, and
  - stores it under a unified schema, at a single site.

- Important for large businesses
  - They generate data from multiple divisions, possibly at multiple sites
Loading the Data Warehouse

Source Systems (OLTP) \rightarrow Data Staging Area \rightarrow Data Warehouse

- Data is periodically extracted
- Data is cleansed and transformed
- Users query the data warehouse

Data Analysis and OLAP

- Aggregate functions summarize large volumes of data
- **Online Analytical Processing (OLAP)**
  - Interactive analysis of data, allowing data to be summarized and viewed in different ways in an *online* fashion (with negligible delay)
Multidimensional Data

- Data that can be modeled as dimension attributes and measure attributes are called **multidimensional data**.
  
  - Given a relation used for data analysis, we can identify **measure attributes**, those that measure some value, and can be aggregated upon.
    - E.g., the attribute *number* of the *sales* relation is a measure attribute, since it measures the number of units sold.
  
  - Some of the other attributes of the relation are identified as **dimension attributes**, since they define the dimensions on which measure attributes, and summaries of measure attributes, are viewed.

Cross-tab

- A cross-tab is a table where
  
  - values for one of the dimension attributes form the row headers, values for another dimension attribute form the column headers
    - Other dimension attributes are listed on top
  
  - Values in individual cells are (aggregates of) the values of the dimension attributes that specify the cell.
Cross-tab example

<table>
<thead>
<tr>
<th>item-name</th>
<th>color</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dark</td>
<td>pastel</td>
<td>white</td>
<td>Total</td>
</tr>
<tr>
<td>skirt</td>
<td>8</td>
<td>35</td>
<td>10</td>
<td>53</td>
</tr>
<tr>
<td>dress</td>
<td>20</td>
<td>10</td>
<td>5</td>
<td>35</td>
</tr>
<tr>
<td>shirt</td>
<td>14</td>
<td>7</td>
<td>28</td>
<td>49</td>
</tr>
<tr>
<td>pant</td>
<td>20</td>
<td>2</td>
<td>5</td>
<td>27</td>
</tr>
<tr>
<td>Total</td>
<td>62</td>
<td>54</td>
<td>48</td>
<td>164</td>
</tr>
</tbody>
</table>

Relational Representation of cross-tabs

- Crosstabs can be represented as relations
- The value all is used to represent aggregates
- The SQL:1999 standard actually uses null values in place of all
- More on this later....
Cross-tabs of >2 dims?

- Generalization of cross-tab for more than two dimensions is a data cube
- 3-dimensional data cubes are “easy” to visualize
- Crosstab can be used as two-dimensional views of any n-dimensional data cube

Data Cube Example

![Data Cube Example](image)
Data Cube

- Axes of the cube represent attributes of the data records
  - e.g. color, month, state
  - Called dimensions

- Cells hold aggregated measurements
  - e.g. total $ sales, number of autos sold
  - Called facts

- Real data cubes have >> 3 dimensions

Slicing and Dicing

- Slicing: Red, Blue, Gray
- Dicing: Jul, Aug, Sep, PA, OH, MD
- Total: Red, Blue

Alexandros Labrinidis, Univ. of Pittsburgh
Querying the Data Cube

- Cross-tabulation
  - "Cross-tab" for short
  - Report data grouped by 2 dimensions
  - Aggregate across other dimensions
  - Include subtotals

- Operations on a cross-tab
  - Roll up (further aggregation)
  - Drill down (less aggregation)

<table>
<thead>
<tr>
<th></th>
<th>PA</th>
<th>OH</th>
<th>MD</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul</td>
<td>45</td>
<td>33</td>
<td>30</td>
<td>108</td>
</tr>
<tr>
<td>Aug</td>
<td>50</td>
<td>36</td>
<td>42</td>
<td>128</td>
</tr>
<tr>
<td>Sep</td>
<td>38</td>
<td>31</td>
<td>40</td>
<td>109</td>
</tr>
<tr>
<td>Total</td>
<td>133</td>
<td>100</td>
<td>112</td>
<td>345</td>
</tr>
</tbody>
</table>

Number of Autos Sold

Roll Up and Drill Down

- Roll up by Month
- Drill down by Color
Full Data Cube with Subtotals

- Pre-computation of aggregates → fast answers to OLAP queries
- Ideally, pre-compute all $2^n$ types of subtotals
- Otherwise, perform aggregation as needed
- Coarser-grained totals can be computed from finer-grained totals
  - But not the other way around

Data Cube Lattice
Hierarchies on Dimensions

- **Hierarchy** on dimension attributes: lets dimensions to be viewed at different levels of detail
  - E.g. the dimension DateTime can be used to aggregate by hour of day, date, day of week, month, quarter or year

---

Cross Tabs With Hierarchies

- Crosstabs can be easily extended to deal with hierarchies
  - Can drill down or roll up on a hierarchy

<table>
<thead>
<tr>
<th>category</th>
<th>item-name</th>
<th>dark</th>
<th>pastel</th>
<th>white</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>womenswear</td>
<td>skirt</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>dress</td>
<td>20</td>
<td>20</td>
<td>5</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>subtotal</td>
<td>28</td>
<td>28</td>
<td>15</td>
<td>88</td>
</tr>
<tr>
<td>menswear</td>
<td>pants</td>
<td>14</td>
<td>14</td>
<td>28</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>shirt</td>
<td>20</td>
<td>20</td>
<td>5</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>subtotal</td>
<td>34</td>
<td>34</td>
<td>33</td>
<td>76</td>
</tr>
<tr>
<td>total</td>
<td></td>
<td>62</td>
<td>62</td>
<td>48</td>
<td>164</td>
</tr>
</tbody>
</table>
MOLAP vs. ROLAP

- MOLAP = Multidimensional OLAP
- Store data cube as multidimensional array
- (Usually) pre-compute all aggregates
- Advantages:
  - Very efficient data access → fast answers
- Disadvantages:
  - Doesn’t scale to large numbers of dimensions
  - Requires special-purpose data store

Sparsity

- Imagine a data warehouse for Giant Eagle.

- Suppose dimensions are: Customer, Product, Store, Day
  - If there are 100,000 customers, 10,000 products, 1,000 stores, and 1,000 days...
  - ...data cube has 1,000,000,000,000,000 cells!

- Fortunately, most cells are empty.
  - A given store doesn’t sell every product on every day.
  - A given customer has never visited most of the stores.
  - A given customer has never purchased most products.

- Multi-dimensional arrays are not an efficient way to store sparse data.
MOLAP vs. ROLAP

- ROLAP = Relational OLAP
  - Store data cube in relational database
  - Express queries in SQL

- Advantages:
  - Scales well to high dimensionality
  - Scales well to large data sets
  - Sparsity is not a problem
  - Uses well-known, mature technology

- Disadvantages:
  - Query performance is slower than MOLAP
  - Need to construct explicit indexes

Creating a Cross-tab with SQL

```
SELECT state, month, SUM(quantity)
FROM sales
GROUP BY state, month
WHERE color = 'Red'
```
What about the totals?

- SQL aggregation query with GROUP BY does not produce subtotals, totals
- Our cross-tab report is incomplete.

<table>
<thead>
<tr>
<th></th>
<th>CA</th>
<th>OR</th>
<th>WA</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul</td>
<td>45</td>
<td>33</td>
<td>30</td>
<td>?</td>
</tr>
<tr>
<td>Aug</td>
<td>50</td>
<td>36</td>
<td>42</td>
<td>?</td>
</tr>
<tr>
<td>Sep</td>
<td>38</td>
<td>31</td>
<td>40</td>
<td>?</td>
</tr>
<tr>
<td>Total</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

One solution: a big UNION ALL

```
Original Query
SELECT state, month, SUM(quantity)
FROM sales
GROUP BY state, month
WHERE color = 'Red'
UNION ALL
SELECT state, "ALL", SUM(quantity)
FROM sales
GROUP BY state
WHERE color = 'Red'
UNION ALL
SELECT "ALL", month, SUM(quantity)
FROM sales
GROUP BY month
WHERE color = 'Red'
UNION ALL
SELECT "ALL", "ALL", SUM(quantity)
FROM sales
WHERE color = 'Red'
```
A better solution

- "UNION ALL" solution gets cumbersome with more than 2 grouping attributes
- n grouping attributes → $2^n$ parts in the union
- OLAP extensions added to SQL 99 are more convenient
  - CUBE, ROLLUP

```
SELECT state, month, SUM(quantity)
FROM sales
GROUP BY CUBE(state, month)
WHERE color = 'Red'
```

Results of the CUBE query

<table>
<thead>
<tr>
<th>State</th>
<th>Month</th>
<th>SUM(quantity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>Jul</td>
<td>45</td>
</tr>
<tr>
<td>CA</td>
<td>Aug</td>
<td>50</td>
</tr>
<tr>
<td>CA</td>
<td>Sep</td>
<td>38</td>
</tr>
<tr>
<td>CA</td>
<td>NULL</td>
<td>133</td>
</tr>
<tr>
<td>OR</td>
<td>Jul</td>
<td>33</td>
</tr>
<tr>
<td>OR</td>
<td>Aug</td>
<td>36</td>
</tr>
<tr>
<td>OR</td>
<td>Sep</td>
<td>31</td>
</tr>
<tr>
<td>OR</td>
<td>NULL</td>
<td>100</td>
</tr>
<tr>
<td>WA</td>
<td>Jul</td>
<td>30</td>
</tr>
<tr>
<td>WA</td>
<td>Aug</td>
<td>42</td>
</tr>
<tr>
<td>WA</td>
<td>Sep</td>
<td>40</td>
</tr>
<tr>
<td>WA</td>
<td>NULL</td>
<td>112</td>
</tr>
<tr>
<td>NULL</td>
<td>Jul</td>
<td>108</td>
</tr>
<tr>
<td>NULL</td>
<td>Aug</td>
<td>128</td>
</tr>
<tr>
<td>NULL</td>
<td>Sep</td>
<td>109</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>345</td>
</tr>
</tbody>
</table>

Notice the use of NULL for totals

Subtotals at all levels
ROLLUP vs. CUBE

- CUBE computes entire lattice
- ROLLUP computes one path through lattice
  - Order of GROUP BY list matters
  - Groups by all prefixes of the GROUP BY list

GROUP BY ROLLUP(A,B,C)
- A,B,C
- (A,B) subtotals
- (A) subtotals
- Total

GROUP BY CUBE(A,B,C)
- A,B,C
- Subtotals for the following:
  - (A,B), (A,C), (B,C),
  - (A), (B), (C)
- Total

ROLLUP example

```
SELECT color, month, state, SUM(quantity)
FROM sales
GROUP BY ROLLUP(color,month,state)
```
**Star Schema**

- **Fact table**
- **Dimension tables**

**Dimension Tables**

- Each one corresponds to a real-world object or concept.
  - Examples: Customer, Product, Date, Employee, Region, Store, Promotion, Vendor, Partner, Account, Department

- Properties of dimension tables:
  - Contain many descriptive columns
  - Dimension tables are wide (dozens of columns)
  - Generally don't have too many rows
    - At least in comparison to the fact tables
    - Usually < 1 million rows
  - Contents are relatively static
    - Almost like a lookup table

- Uses of dimension tables:
  - Filters are based on dimension attributes
  - Grouping columns are dimension attributes
  - Fact tables are referenced through dimensions
Fact Tables

- Each fact table contains measurements about a process of interest.
- Each fact row contains two things:
  - Numerical measure columns
  - Foreign keys to dimension tables
  - That’s all!

- Properties of fact tables:
  - Very big
    - Often millions or billions of rows
  - Narrow
    - Small number of columns
  - Changes often
    - New events in the world → new rows in the fact table
    - Typically append-only

- Uses of fact tables:
  - Measurements are aggregated fact columns.