POLARIS
A System for Query, Analysis and Visualization of Multi-dimensional Relational Databases

By Chris Stolte and Pat Hanrahan, Stanford University

Presenter: Ganesh Viswanathan
CIS6930: Data Science: Large-scale Advanced Data Analysis
University of Florida
September 15, 2011

Area: Data visualization & Interface design
Outline

- Motivation
- Existing tools: charts, pivot tables, etc.
- Polaris
  - Why use Relational DB approach?
  - Design goals and features
  - Interface – visual specification
  - Formalism – table algebra
  - Types of graphics, and Query handling
- Demo – Tableau software
- Related work
- Discussion
Motivation

- Large multi-dimensional databases have become very common
  - corporate data warehouses
    - Amazon, Walmart, ...
  - scientific projects:
    - Human Genome Project
    - Sloan Digital Sky Survey
- A major challenge for these huge databases is to extract meaning from the data they contain such as:
  - to discover structure,
  - to find patterns, and
  - to derive causal relationship.
- Need tools for exploration and analysis of these databases
Existing Tools: Charts

- typically provide a “gallery” of charts
- hard to iteratively explore
- simple charts can display few dimensions
Existing tools: Pivot Tables

- common interface to data warehouses
- simple interface based on drag-and-drop
- generate text tables from databases:

<table>
<thead>
<tr>
<th>Region</th>
<th>East</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sum of Sales</strong></td>
<td><strong>Product</strong></td>
</tr>
<tr>
<td><strong>Month</strong></td>
<td><strong>Salesperson</strong></td>
</tr>
<tr>
<td>May</td>
<td>Buchanan</td>
</tr>
<tr>
<td></td>
<td>Davolio</td>
</tr>
<tr>
<td>May Total</td>
<td></td>
</tr>
<tr>
<td>Jun</td>
<td>Buchanan</td>
</tr>
<tr>
<td></td>
<td>Davolio</td>
</tr>
<tr>
<td>Jun Total</td>
<td></td>
</tr>
<tr>
<td>Grand Total</td>
<td></td>
</tr>
</tbody>
</table>
Pivot Tables

- Multi-dimensional databases are often treated as n-dimensional data cubes.

- Pivot Tables allow rotation of multi-dimensional datasets, allowing different dimensions to assume the rows and columns of the table, with the remaining dimensions being aggregated within the table.
# Pivot Table Example: Baseball data

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Baseball</td>
<td>Name</td>
<td>At Bats</td>
<td>Hits</td>
<td>Home Runs</td>
<td>Runs</td>
<td>Rbi</td>
<td>Walks</td>
<td>Years In Majors</td>
<td>Career At Bats</td>
<td>Career Hits</td>
</tr>
<tr>
<td>2</td>
<td>Cliff Johnson</td>
<td>Cliff Johnson</td>
<td>19</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>41</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>Doug Baker</td>
<td>Doug Baker</td>
<td>24</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>159</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>Ricky Jones</td>
<td>Ricky Jones</td>
<td>33</td>
<td>6</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>1</td>
<td>33</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>Bill Russell</td>
<td>Bill Russell</td>
<td>216</td>
<td>54</td>
<td>0</td>
<td>21</td>
<td>18</td>
<td>15</td>
<td>18</td>
<td>7318</td>
<td>1926</td>
</tr>
<tr>
<td>6</td>
<td>Julio Cruz</td>
<td>Julio Cruz</td>
<td>209</td>
<td>45</td>
<td>0</td>
<td>38</td>
<td>19</td>
<td>42</td>
<td>10</td>
<td>3859</td>
<td>916</td>
</tr>
<tr>
<td>7</td>
<td>Tony Armas</td>
<td>Tony Armas</td>
<td>16</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>28</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>Luis Rivera</td>
<td>Luis Rivera</td>
<td>166</td>
<td>34</td>
<td>0</td>
<td>20</td>
<td>13</td>
<td>17</td>
<td>1</td>
<td>166</td>
<td>34</td>
</tr>
<tr>
<td>9</td>
<td>Bobby Meacham</td>
<td>Bobby Meacham</td>
<td>161</td>
<td>36</td>
<td>0</td>
<td>19</td>
<td>10</td>
<td>17</td>
<td>4</td>
<td>1053</td>
<td>244</td>
</tr>
<tr>
<td>10</td>
<td>Kurt Stillwell</td>
<td>Kurt Stillwell</td>
<td>279</td>
<td>64</td>
<td>0</td>
<td>31</td>
<td>26</td>
<td>30</td>
<td>1</td>
<td>279</td>
<td>64</td>
</tr>
<tr>
<td>11</td>
<td>Argenis Salazar</td>
<td>Argenis Salazar</td>
<td>298</td>
<td>73</td>
<td>0</td>
<td>24</td>
<td>24</td>
<td>7</td>
<td>3</td>
<td>509</td>
<td>108</td>
</tr>
<tr>
<td>12</td>
<td>Rafael Belliard</td>
<td>Rafael Belliard</td>
<td>309</td>
<td>72</td>
<td>0</td>
<td>33</td>
<td>31</td>
<td>26</td>
<td>5</td>
<td>354</td>
<td>82</td>
</tr>
<tr>
<td>13</td>
<td>Steve Jeltz</td>
<td>Steve Jeltz</td>
<td>439</td>
<td>96</td>
<td>0</td>
<td>44</td>
<td>36</td>
<td>65</td>
<td>4</td>
<td>711</td>
<td>148</td>
</tr>
<tr>
<td>14</td>
<td>Vince Coleman</td>
<td>Vince Coleman</td>
<td>600</td>
<td>139</td>
<td>0</td>
<td>94</td>
<td>29</td>
<td>60</td>
<td>2</td>
<td>1236</td>
<td>309</td>
</tr>
<tr>
<td>15</td>
<td>Curt Wilkerson</td>
<td>Curt Wilkerson</td>
<td>236</td>
<td>56</td>
<td>0</td>
<td>27</td>
<td>15</td>
<td>11</td>
<td>4</td>
<td>1115</td>
<td>270</td>
</tr>
<tr>
<td>16</td>
<td>Alan Wiggins</td>
<td>Alan Wiggins</td>
<td>239</td>
<td>60</td>
<td>0</td>
<td>30</td>
<td>11</td>
<td>22</td>
<td>6</td>
<td>1941</td>
<td>510</td>
</tr>
<tr>
<td>17</td>
<td>Pete Rose</td>
<td>Pete Rose</td>
<td>237</td>
<td>52</td>
<td>0</td>
<td>15</td>
<td>25</td>
<td>30</td>
<td>24</td>
<td>14053</td>
<td>4256</td>
</tr>
<tr>
<td>18</td>
<td>Ozzie Smith</td>
<td>Ozzie Smith</td>
<td>514</td>
<td>144</td>
<td>0</td>
<td>67</td>
<td>54</td>
<td>79</td>
<td>9</td>
<td>4739</td>
<td>1169</td>
</tr>
<tr>
<td>19</td>
<td>Dave Collins</td>
<td>Dave Collins</td>
<td>419</td>
<td>113</td>
<td>1</td>
<td>44</td>
<td>27</td>
<td>44</td>
<td>12</td>
<td>4484</td>
<td>1231</td>
</tr>
<tr>
<td>20</td>
<td>Rudy Law</td>
<td>Rudy Law</td>
<td>307</td>
<td>80</td>
<td>1</td>
<td>42</td>
<td>36</td>
<td>29</td>
<td>7</td>
<td>2421</td>
<td>656</td>
</tr>
<tr>
<td>21</td>
<td>Juan Bonilla</td>
<td>Juan Bonilla</td>
<td>284</td>
<td>69</td>
<td>1</td>
<td>33</td>
<td>18</td>
<td>25</td>
<td>5</td>
<td>1407</td>
<td>361</td>
</tr>
<tr>
<td>22</td>
<td>Tom O'Malley</td>
<td>Tom O'Malley</td>
<td>181</td>
<td>46</td>
<td>1</td>
<td>19</td>
<td>18</td>
<td>17</td>
<td>5</td>
<td>937</td>
<td>238</td>
</tr>
<tr>
<td>23</td>
<td>Bip Roberts</td>
<td>Bip Roberts</td>
<td>241</td>
<td>61</td>
<td>1</td>
<td>34</td>
<td>12</td>
<td>14</td>
<td>1</td>
<td>241</td>
<td>61</td>
</tr>
<tr>
<td>24</td>
<td>Bob Boone</td>
<td>Bob Boone</td>
<td>22</td>
<td>10</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>84</td>
<td>26</td>
</tr>
<tr>
<td>25</td>
<td>Andy Allanson</td>
<td>Andy Allanson</td>
<td>293</td>
<td>66</td>
<td>1</td>
<td>30</td>
<td>29</td>
<td>14</td>
<td>1</td>
<td>293</td>
<td>66</td>
</tr>
<tr>
<td>26</td>
<td>Billy Jo Robidoux</td>
<td>Billy Jo Robidoux</td>
<td>181</td>
<td>41</td>
<td>1</td>
<td>15</td>
<td>21</td>
<td>33</td>
<td>2</td>
<td>232</td>
<td>50</td>
</tr>
<tr>
<td>27</td>
<td>Al Newman</td>
<td>Al Newman</td>
<td>185</td>
<td>37</td>
<td>1</td>
<td>23</td>
<td>8</td>
<td>21</td>
<td>2</td>
<td>214</td>
<td>42</td>
</tr>
<tr>
<td>28</td>
<td>Harold Reynolds</td>
<td>Harold Reynolds</td>
<td>445</td>
<td>99</td>
<td>1</td>
<td>46</td>
<td>24</td>
<td>29</td>
<td>4</td>
<td>618</td>
<td>129</td>
</tr>
<tr>
<td>29</td>
<td>Terry Pendleton</td>
<td>Terry Pendleton</td>
<td>578</td>
<td>138</td>
<td>1</td>
<td>56</td>
<td>59</td>
<td>34</td>
<td>3</td>
<td>1399</td>
<td>357</td>
</tr>
<tr>
<td>30</td>
<td>Dave Anderson</td>
<td>Dave Anderson</td>
<td>216</td>
<td>53</td>
<td>1</td>
<td>31</td>
<td>15</td>
<td>22</td>
<td>4</td>
<td>926</td>
<td>210</td>
</tr>
<tr>
<td>31</td>
<td>Tom Foley</td>
<td>Tom Foley</td>
<td>263</td>
<td>70</td>
<td>1</td>
<td>26</td>
<td>23</td>
<td>30</td>
<td>4</td>
<td>888</td>
<td>220</td>
</tr>
</tbody>
</table>
## Pivot Table Example: Baseball data

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>League</td>
<td>(All)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>Sum of Walks</td>
<td></td>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Sum of Errors</td>
<td></td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>Sum of Assists</td>
<td></td>
<td>58</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>23</td>
<td>Sum of Walks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Sum of Errors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Sum of Assists</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>32</td>
<td>Sum of Walks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Sum of Errors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>Sum of Assists</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1B</td>
<td>Sum of Walks</td>
<td></td>
<td>30</td>
<td>52</td>
<td>78</td>
<td>40</td>
<td>57</td>
<td>96</td>
<td>102</td>
<td>29</td>
<td>91</td>
<td>64</td>
<td>43</td>
<td>51</td>
<td>74</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>Sum of Errors</td>
<td></td>
<td>4</td>
<td>8</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>12</td>
<td>18</td>
<td>9</td>
<td>2</td>
<td>11</td>
<td>18</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>Sum of Assists</td>
<td></td>
<td>40</td>
<td>102</td>
<td>88</td>
<td>157</td>
<td>139</td>
<td>137</td>
<td>102</td>
<td>84</td>
<td>108</td>
<td>111</td>
<td>98</td>
<td>119</td>
<td>90</td>
</tr>
<tr>
<td>17</td>
<td>1O</td>
<td>Sum of Walks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>Sum of Errors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td>Sum of Assists</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>2B</td>
<td>Sum of Walks</td>
<td></td>
<td>58</td>
<td>66</td>
<td>47</td>
<td>65</td>
<td>55</td>
<td>88</td>
<td>52</td>
<td>53</td>
<td>63</td>
<td>81</td>
<td>43</td>
<td>59</td>
<td>26</td>
</tr>
<tr>
<td>21</td>
<td></td>
<td>Sum of Errors</td>
<td></td>
<td>10</td>
<td>19</td>
<td>11</td>
<td>14</td>
<td>14</td>
<td>10</td>
<td>19</td>
<td>17</td>
<td>11</td>
<td>16</td>
<td>10</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>22</td>
<td></td>
<td>Sum of Assists</td>
<td></td>
<td>411</td>
<td>487</td>
<td>291</td>
<td>450</td>
<td>478</td>
<td>697</td>
<td>475</td>
<td>442</td>
<td>421</td>
<td>329</td>
<td>439</td>
<td>432</td>
<td>347</td>
</tr>
<tr>
<td>23</td>
<td>2S</td>
<td>Sum of Walks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>Sum of Errors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>Sum of Assists</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>3B</td>
<td>Sum of Walks</td>
<td></td>
<td>44</td>
<td>83</td>
<td>32</td>
<td>105</td>
<td>71</td>
<td>117</td>
<td>73</td>
<td>56</td>
<td>45</td>
<td>66</td>
<td>80</td>
<td>30</td>
<td>72</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td>Sum of Errors</td>
<td></td>
<td>16</td>
<td>8</td>
<td>24</td>
<td>19</td>
<td>14</td>
<td>26</td>
<td>10</td>
<td>25</td>
<td>23</td>
<td>32</td>
<td>16</td>
<td>24</td>
<td>41</td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>Sum of Assists</td>
<td></td>
<td>270</td>
<td>258</td>
<td>213</td>
<td>267</td>
<td>272</td>
<td>375</td>
<td>290</td>
<td>292</td>
<td>242</td>
<td>297</td>
<td>218</td>
<td>170</td>
<td>292</td>
</tr>
</tbody>
</table>
### Pivot Table Example: Baseball data

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Sum of Walks</th>
<th>Sum of Errors</th>
<th>Sum of Assists</th>
<th>Sum of Walks</th>
<th>Sum of Errors</th>
<th>Sum of Assists</th>
<th>Sum of Walks</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>League</td>
<td></td>
<td>(All)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Team</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Mon.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Atl.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>Bal.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Position</td>
<td>Sum of Walks</td>
<td>Sum of Errors</td>
<td>Sum of Assists</td>
<td>Sum of Walks</td>
<td>Sum of Errors</td>
<td>Sum of Assists</td>
<td>Sum of Walks</td>
</tr>
<tr>
<td>2</td>
<td>1B</td>
<td>30</td>
<td>4</td>
<td>40</td>
<td>52</td>
<td>8</td>
<td>102</td>
<td>78</td>
</tr>
<tr>
<td>3</td>
<td>2B</td>
<td>58</td>
<td>10</td>
<td>411</td>
<td>66</td>
<td>19</td>
<td>487</td>
<td>47</td>
</tr>
<tr>
<td>4</td>
<td>2S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3B</td>
<td>44</td>
<td>16</td>
<td>270</td>
<td>83</td>
<td>8</td>
<td>258</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>3O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>C</td>
<td>41</td>
<td>11</td>
<td>73</td>
<td>63</td>
<td>13</td>
<td>93</td>
<td>45</td>
</tr>
<tr>
<td>8</td>
<td>CD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>CF</td>
<td>57</td>
<td>8</td>
<td>12</td>
<td>75</td>
<td>6</td>
<td>6</td>
<td>53</td>
</tr>
<tr>
<td>10</td>
<td>CS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>DH</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>12</td>
<td>DO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>13</td>
<td>LF</td>
<td>78</td>
<td>6</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>49</td>
</tr>
<tr>
<td>14</td>
<td>O1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>OD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>OF</td>
<td>18</td>
<td>2</td>
<td>2</td>
<td>43</td>
<td>4</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>17</td>
<td>OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>RF</td>
<td>37</td>
<td>3</td>
<td>11</td>
<td>21</td>
<td>5</td>
<td>8</td>
<td>37</td>
</tr>
<tr>
<td>19</td>
<td>S3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>SS</td>
<td>42</td>
<td>24</td>
<td>341</td>
<td>12</td>
<td>8</td>
<td>19</td>
<td>290</td>
</tr>
<tr>
<td>21</td>
<td>UT</td>
<td>30</td>
<td>4</td>
<td>147</td>
<td>12</td>
<td>6</td>
<td>18</td>
<td>58</td>
</tr>
<tr>
<td>22</td>
<td>Grand Total</td>
<td>457</td>
<td>94</td>
<td>1378</td>
<td>444</td>
<td>117</td>
<td>1639</td>
<td>530</td>
</tr>
</tbody>
</table>
Pivot Table Example: Baseball data

<p>| | | | | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td>O</td>
<td>P</td>
<td>Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Position</td>
<td>SS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Team</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>Sum of Walks</td>
<td>70</td>
<td>42</td>
<td>48</td>
<td>12</td>
<td>32</td>
<td>59</td>
<td>22</td>
<td>54</td>
<td>30</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>A</td>
<td>Sum of Errors</td>
<td>13</td>
<td>25</td>
<td>18</td>
<td>22</td>
<td>18</td>
<td>22</td>
<td>25</td>
<td>20</td>
<td>26</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>A</td>
<td>Sum of Assists</td>
<td>482</td>
<td>282</td>
<td>389</td>
<td>459</td>
<td>374</td>
<td>445</td>
<td>460</td>
<td>327</td>
<td>377</td>
<td>149</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>N</td>
<td>Sum of Walks</td>
<td>42</td>
<td>8</td>
<td>70</td>
<td>42</td>
<td>48</td>
<td>33</td>
<td>30</td>
<td>32</td>
<td>59</td>
<td>41</td>
<td>30</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>N</td>
<td>Sum of Errors</td>
<td>24</td>
<td>19</td>
<td>13</td>
<td>25</td>
<td>18</td>
<td>54</td>
<td>16</td>
<td>18</td>
<td>22</td>
<td>17</td>
<td>25</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>N</td>
<td>Sum of Assists</td>
<td>341</td>
<td>290</td>
<td>465</td>
<td>205</td>
<td>416</td>
<td>317</td>
<td>369</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Total Sum of Walks</td>
<td>42</td>
<td>8</td>
<td>70</td>
<td>42</td>
<td>48</td>
<td>33</td>
<td>30</td>
<td>32</td>
<td>59</td>
<td>41</td>
<td>22</td>
<td>30</td>
<td>54</td>
<td>30</td>
<td>53</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Total Sum of Errors</td>
<td>24</td>
<td>19</td>
<td>13</td>
<td>25</td>
<td>18</td>
<td>54</td>
<td>16</td>
<td>18</td>
<td>22</td>
<td>17</td>
<td>25</td>
<td>25</td>
<td>20</td>
<td>26</td>
<td>28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Total Sum of Assists</td>
<td>341</td>
<td>290</td>
<td>482</td>
<td>262</td>
<td>389</td>
<td>924</td>
<td>205</td>
<td>374</td>
<td>445</td>
<td>416</td>
<td>460</td>
<td>317</td>
<td>327</td>
<td>377</td>
<td>518</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Relational databases

- Each row in table = basic entity (tuple)
- Each column represents a field
- Fields can be nominal, ordinal, or quantitative
Relational Data Schema

- Structural description of data sets
- Primitives: attributes, tuples and relations
Motivation

- Relational data schema enables flexible database design
- No corresponding flexible ways to construct effective UI and visualization
  - unique data schema
  - unique visualization/coordination
  - database keeps changing
  - different views for same data
### Mismatch in design capabilities

<table>
<thead>
<tr>
<th></th>
<th>Relational Databases</th>
<th>Traditional Visualization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Design Goal</strong></td>
<td>Data design</td>
<td>Visualization design</td>
</tr>
<tr>
<td><strong>Design Method</strong></td>
<td>Data schema</td>
<td>Program code</td>
</tr>
<tr>
<td><strong>Designer</strong></td>
<td>Data owner</td>
<td>Programmer only</td>
</tr>
<tr>
<td><strong>Design Change</strong></td>
<td>Rapid, dynamic</td>
<td>Slow, static</td>
</tr>
<tr>
<td><strong>Adaptability</strong></td>
<td>Flexible</td>
<td>Brittle</td>
</tr>
</tbody>
</table>
Requirements on UI for Analysis and Exploration

- **Data dense displays**: display both many tuples & many dimensions

- **Multiple display types**: different displays suited to different tasks

- **Exploratory interfaces**: rapidly change data transformations and views
Polaris

- Polaris is an interface for the exploration of multi-dimensional databases that extends the pivot table interface to directly generate a rich, expressive set of graphical displays.
Polaris Design Goals

- Generate rich table-based graphical displays rather than tables of text
- Single conceptual model for both graphs and tables
- Preserve ability to rapidly construct displays
- Interactive analysis and exploration versus static visualization
- Simple, consistent interface
- Ease analysis and exploration:
  - Want to extract meaning from data
  - Process of hypothesis, experiment, and discovery
  - Path of exploration is unpredictable
Features of Polaris

- Builds tables using an algebraic formalism involving the fields of the database.
- Each table consists of layers and panes, and each pane may be a different graphic.
- An interface for constructing visual specifications of table-based graphical displays.
- The state of the interface can be interpreted as a visual specification of the analysis task and automatically compiled into data and graphical transformations.
Features of Polaris

- The visual specifications can be rapidly & incrementally developed, giving the users visual feedback as they construct complex queries & visualization.

- Ability to generate a precise set of relational queries from the visual specifications.

- Users can incrementally construct complex queries, receiving visual feedback as they assemble and alter the specifications.
Visualizing Multidimensional Data

Several characteristics to tables make them particularly effective for displaying multi-dimensional data:

- **Multivariate** - multiple dimensions of the data can be explicitly encoded in the structure of the table, enabling the display of high-dimensional data.

- **Comparative** - tables generate small multiple displays of information, which are easily compared, exposing patterns and trends across dimensions of the data.

- **Familiar** - Statisticians are accustomed to using tabular displays of graphs, such as scatterplot matrices and Trellis displays, for analysis. Pivot Tables are a common interface to large data warehouses.
**Polaris Display: UI**

**Database Schema:**
The user drags fields from the database schema to shelves to define the visual specification.

**Import:**
Data from multiple data sources can be imported. Each dataset maps to a different layer.

**Layer Tabs:**
Each layer has its own tab; different transformations and mappings can be specified for each layer.

**Axis Shelves:**
The fields placed here determine the structure of the table and the types of graphs in each table pane.

**Layer Shelf:**
The fields placed here determine how records are partitioned into layers.

**Grouping and Sorting Shelves:**
The fields placed here determine how records are grouped and sorted within the table panes.

**Mark Pulldown:**
Relations in each pane are mapped to marks of the selected type.

**Retinal Property Shelves:**
The fields placed here determine how data is encoded in the retinal properties of the marks.

**Legends:**
Legends enable the user to see and modify the mappings from data to retinal properties.
Design Decision: Use a Formalism

- Why a formalism?
  - **unification**: unify tables and graphs
  - **expressiveness**: build visualizations designers did not think of
  - **interface simplicity**: clearly defined semantics and operations
  - **code simplicity**: composable language versus monolithic objects
  - **declarative**: can state what, not how - allows for optimization, etc.
Polaris Formalism

- Interface interpreted as **visual specification** in formal language that defines:
  - table configuration
  - type of graphic in each pane
  - encoding of data as visual properties of marks

- Specification compiled into data & graphical transformations to generate display
Example specification

```xml
<visual name='table'>
  <tablelayout>
    <tableexpr axis='X'> ProductType * Profit </tableexpr>
    <tableexpr axis='Y'> Market </tableexpr>
    <aggregationflag value='false'/>
  </tablelayout>

  <layer name='profit'>
    <datasource name='starbucks'/>
    
    <pane>
      <mark type='Circle'>
        <defaultsize>ll</defaultsize>
        <colorencoding><dimension name='Decaf'/></colorencoding>
      </mark>
    </pane>
  </layer>
</visual>
```
Formalism Example: Specifying Table Configurations

- **Interface:** define table configuration by dropping fields on shelves

- **Formalism:** shelf content interpreted as expressions in table algebra

- Can express extremely wide range of table configurations
Operands are the database fields
- each operand interpreted as a set \{\ldots\}
- quantitative and ordinal fields interpreted differently

Three main operators:
- concatenation (+), cross (X), nest (/)
- Additionally: dot (.) operator
Table Algebra: Operands

- **Ordinal fields** - interpret domain as a set that partitions table into rows and columns:
  
  \[ \text{QUARTER} = \{\text{Quarter1, Quarter2, Quarter3, Quarter4}\} \]

<table>
<thead>
<tr>
<th>Quarter 1</th>
<th>Quarter 2</th>
<th>Quarter 3</th>
<th>Quarter 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>31,400</td>
<td>35,600</td>
<td>37,120</td>
<td>30,900</td>
</tr>
</tbody>
</table>

- **Quantitative fields** – treat domain as single element set and encode spatially as axes:
  
  \[ \text{PROFIT} = \{P[0 - 65,000]\} \]

  ![Profit Scale](image)
Table Algebra:
Concatenation (+) Operator

- Ordered union of set interpretations:

\[
\text{QUARTER} + \text{PRODUCT\_TYPE} \\
= \{\text{QTR1, QTR2, QTR3, QTR4}\} + \{\text{Coffee, Tea}\} \\
= \{\text{QTR1, QTR2, QTR3, QTR4, Coffee, Tea}\}
\]

<table>
<thead>
<tr>
<th>Quarter 1</th>
<th>Quarter 2</th>
<th>Quarter 3</th>
<th>Quarter 4</th>
<th>Coffee</th>
<th>Tea</th>
</tr>
</thead>
<tbody>
<tr>
<td>31,400</td>
<td>35,600</td>
<td>37,120</td>
<td>30,900</td>
<td>37,120</td>
<td>30,900</td>
</tr>
</tbody>
</table>

\[
\text{PROFIT} + \text{SALES} = \{P[0-65,000], S[0-125,000]\}
\]

<table>
<thead>
<tr>
<th>Profit (in thousands)</th>
<th>Sales (in thousands)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 20 30 40 50 60</td>
<td>20 40 60 80 100 120</td>
</tr>
</tbody>
</table>
Table Algebra: Cross (X) Operator

- Cross-product of set interpretations:

\[
\text{QUARTER} \times \text{PRODUCT\_TYPE} = \\
\{(\text{Qtr1, Coffee}), (\text{Qtr1, Tea}), (\text{Qtr2, Coffee}), (\text{Qtr2, Tea}), (\text{Qtr3, Coffee}), (\text{Qtr3, Tea}), (\text{Qtr4, Coffee}), (\text{Qtr4, Tea})\}\\
\]

\[
\text{PRODUCT\_TYPE} \times \text{PROFIT} = \\
\begin{array}{c|c|c|c|c|c}
\text{Coffee} & \text{Tea} \\
\hline
\text{Profit (in thousands)} & 10 & 20 & 30 & 40 & 50 & 60 \\
\hline
\text{Coffee} & 10 & 20 & 30 & 40 & 50 & 60 \\
\text{Tea} & 10 & 20 & 30 & 40 & 50 & 60 \\
\end{array}
\]
Table Algebra: Nest (/) Operator

- QUARTER X MONTH
  - would create entry twelve entries for each quarter
    i.e. (Qtr1, December)

- QUARTER / MONTH
  - would only create three entries per quarter
  - based on tuples in database not semantics
  - can be expensive to compute
Ordinal fields: Quarter, Months, Product
Quantitative fields: Profit, Sales

$O = \text{Quarter} = \{\text{Qtr1}, \text{Qtr2}, \text{Qtr3}, \text{Qtr4}\} = \text{Qtr1} + \text{Qtr2} + \text{Qtr3} + \text{Qtr4}$:

<table>
<thead>
<tr>
<th>Qtr1</th>
<th>Qtr2</th>
<th>Qtr3</th>
<th>Qtr4</th>
</tr>
</thead>
</table>

$O + O = \text{Quarter} + \text{Product} = \{\text{Qtr1}, \text{Qtr2}, \text{Qtr3}, \text{Qtr4}, \text{Coffee}, \text{Espresso}, \text{Herbal Tea}, \text{Tea}\}$:

<table>
<thead>
<tr>
<th>Qtr1</th>
<th>Qtr2</th>
<th>Qtr3</th>
<th>Qtr4</th>
<th>Coffee</th>
<th>Espresso</th>
<th>Herbal Tea</th>
<th>Tea</th>
</tr>
</thead>
</table>

$O \times O = \text{Quarter} \times \text{Product} = \{(\text{Qtr1,Coffee}), (\text{Qtr1,Espresso}), (\text{Qtr1,Herbal Tea}), (\text{Qtr1,Tea}), (\text{Qtr2, Coffee}) \ldots (\text{Qtr4, Tea})\}$:

<table>
<thead>
<tr>
<th>Qtr1</th>
<th>Qtr2</th>
<th>Qtr3</th>
<th>Qtr4</th>
<th>Coffee</th>
<th>Espresso</th>
<th>Herbal Tea</th>
<th>Tea</th>
</tr>
</thead>
</table>

$O / O = \text{Quarter} / \text{Month} = \{(\text{Qtr1,Jan}), (\text{Qtr1,Feb}), (\text{Qtr1,Mar}), (\text{Qtr2, Apr}), (\text{Qtr2, May}) \ldots (\text{Qtr4, Dec})\}$:

<table>
<thead>
<tr>
<th>Qtr1</th>
<th>Qtr2</th>
<th>Qtr3</th>
<th>Qtr4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>Feb</td>
<td>Mar</td>
<td>Apr</td>
</tr>
</tbody>
</table>

$Q = \text{Profit} = \{\text{Profit}\}$:

<table>
<thead>
<tr>
<th>Profit (in thousands)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>50</td>
</tr>
<tr>
<td>60</td>
</tr>
</tbody>
</table>

$Q + Q = \text{Profit} + \text{Sales} = \{\text{Profit}, \text{Sales}\}$:

<table>
<thead>
<tr>
<th>Profit (in thousands)</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>150</td>
</tr>
<tr>
<td>20</td>
<td>170</td>
</tr>
<tr>
<td>30</td>
<td>190</td>
</tr>
<tr>
<td>40</td>
<td>210</td>
</tr>
<tr>
<td>50</td>
<td>230</td>
</tr>
</tbody>
</table>

$Q \times Q = \text{Quarter} \times \text{Profit} = \{(\text{Qtr1,Profit}), (\text{Qtr2, Profit}), (\text{Qtr3, Profit}), (\text{Qtr4, Profit})\}$:
Polaris Display: UI

Database Schema:
The user drags fields from the database schema to shelves to define the visual specification.

Import:
Data from multiple data sources can be imported. Each dataset maps to a different layer.

Layer Tabs:
Each layer has its own tab; different transformations and mappings can be specified for each layer.

Axis Shelves:
The fields placed here determine the structure of the table and the types of graphs in each table pane.

Layer Shelf:
The fields placed here determine how records are partitioned into layers.

Grouping and Sorting Shelves:
The fields placed here determine how records are grouped and sorted within the table panes.

Mark Pulldown:
Relations in each pane are mapped to marks of the selected type.

Retinal Property Shelves:
The fields placed here determine how data is encoded in the retinal properties of the marks.

Legends:
Legends enable the user to see and modify the mappings from data to retinal properties.
Polaris Display

- Drag and drop fields from database scheme onto shelves
- May combine multiple data sources, each data source mapping to a separate layer
- Multiple fields may be dragged onto each shelf
- Data may be grouped or sorted, and aggregations may be computed
Polaris Display

- Selecting a single mark in a graphic displays the values for the mark
- Can lasso a set of marks to brush records
- Marks in the graphics use retinal properties
Retinal Properties

- Ordinal/nominal mapping vs. quantitative mapping
- Properties: Shape, size, orientation, and color.
- When encoding a quantitative variables, should only vary one aspect at a time

*Figure 4: The different retinal properties that can be used to encode fields of the data and examples of the default mappings that are generated when a given type of data field is encoded in each of the retinal properties.*
Visual Specification

- Is the configuration of the fields of the tables on shelves
- User does this by dragging and dropping fields onto shelves

- Controls:
  - Mapping of data sources to layers
  - # of rows, columns, and layers, and relative order
  - Selection of tuples from the database
  - Grouping of data within a pane
  - Type of graphic displayed in each pane
  - Mapping of data fields with retinal properties
Graphics

- **Ordinal-Ordinal**: e.g. the table
  - the axis variables are typically independent of each other

- **Ordinal-Quantitative**: e.g. bar chart
  - the quantitative variable is often dependent on the ordinal variable

- **Quantitative-Quantitative**: e.g. maps
  - view distribution of data as a function of one or both variables; discover causal relationships
Ordinal-Ordinal graphics:

Figure shows sales and margin as a function of product type, month and state for items sold by coffee chain.
(c) A detailed view shows that it is costing the company money to sell Caffe Mocha in New York.

Ordinal – Quantitative graphics:

Following slide shows a case where a matrix of bar charts is used to study several functions of the independent variables product and month.
Quantitative – Quantitative graphics:

Following slide shows how flight scheduling varies with the region of the country the flight originated.
Display Types

Gantt charts of events for a parallel graphics application on a 32-processor SGI machine.

Flights between major airports in the USA

Source code colored by cache misses for a parallel graphics application.

Major wars and the births of well known scientists as a timeline.
Data Transformation and Querying

- Derive additional fields by:
  - Simple aggregation of quantitative measures
  - Counting of distinct values in ordinal dimensions
  - Discrete partitioning of quantitative measures
  - Ad hoc grouping within ordinal dimensions
  - Threshold aggregation

- Sorting and Filtering
- Brushing and Tooltips
- Undo and Redo
Querying

- Three steps:
  - Select the records
  - Partition the records into panes
  - Transform the records within the panes

- To create database queries, it is necessary to generate an SQL query per table pane (i.e. must iterate over entire table, executing SQL for each pane).
Transformations and Data Flow

Fig. 5. The transformations and data flow within Polaris. The visual specification generates queries to the database to select subsets of the data for analysis, then to filter, sort, and group the results into panes, and then, finally, to group, sort and aggregate the data within panes.
Generating Database Queries

1. Selecting the Records

\[
A \text{ in } \text{filter}(A)
\]

\[
(P \geq \min(P) \text{ and } P \leq \max(P))
\]

\[
\text{SELECT * WHERE } \{\text{filters}\}
\]
Generating Database Queries

- 2. Partitioning the records into panes
- Putting retrieved records in their corresponding pane

\[
\begin{align*}
\text{Row(1)} &= (A = a_1 \text{ and } B = b_1) \\
\text{Row(2)} &= (A = a_1 \text{ and } B = b_2) \\
\text{Row(3)} &= (A = a_2 \text{ and } B = b_1) \\
\text{Row(4)} &= (A = a_2 \text{ and } B = b_2)
\end{align*}
\]

```sql
SELECT *
WHERE \{(\text{Row}(i) \text{ and Column}(j) \text{ and Layer}(k))\}
```
3. Transforming records within the panes
If aggregation, it is done here

```sql
aggregates = SUM(Profit), AVG(Sales), SUM(Payroll)

G : the field names in the grouping shelf,
S : the field names in the sorting shelf, and
dim : the dimensions in the database.

SELECT {dim},{aggregates}
GROUP BY {G}
HAVING {filters}
ORDER BY {S}

SELECT *
ORDER BY {S}
```
Example application

- Cut expenses for a national coffee store
- Create table of scatterplots showing relationship between marketing costs and profit
- Notice trend; certain products have high marketing costs with no or little profit

(a) Some products are yielding negative profit despite marketing.
POLARIS

Demo: Tableau Software
Related Work

- Single relation visualization
  - APT
  - Sage/SageBrush
  - DEVise

- Multiple relation visualization
  - Visage
  - DataSplash/Tioga-2
  - Rivet/Polaris
  - Sieve
Related Work

- **Formalisms for Graphics**
  - Wilkinson’s Grammar of Graphics
  - Bertin’s Semiology of Graphics
  - Mackinlay’s APT

- **Visual Queries**
  - Trellis display, DeVise, Visage

- **Table-based Visualizations**
  - Table lens, Spreadsheet for Visualization
Interesting, upcoming projects:

- **IBM Many Eyes**: Site allows users to upload data and then produce graphic representations for others to view and comment upon... for free!

- **Processing**: Open source programming language and environment for people who want to create images, animations, and interactions.

- **Prefuse**: Interactive information visualization toolkit

Commercial visualization software:

- **Tableau, Qlikview, Tibco Spotfire, Microsoft BI platform** (PowerPivot, Excel 2010, SQL Server with VertiPaq, SSAS, SSRS and SSIS)
Conclusions

- Novel interface for rapidly constructing table-based graphical displays from multi-dimensional relational databases

- A formalism for specifying complex graphics and tables

- Interpretation of visual specifications as relational (SQL) queries and drawing operations.
Discussion

- Allows overlap between the relations that are divided into each pane of the Polaris display, unlike the basic Pivot Table model.

- Allows more versatile computation of aggregates (e.g., medians and averages, in addition to sums).

- Intuitive drag-and-drop interface, like that seen in Pivot Tables
Remarks

- **Merits:**
  - A cohesive architecture for coordinating visualization components
  - Flexible and easy user interface, no programming needed
  - Supports interactive visual queries
  - Good integration between query and visualization schema

- **Shortcomings:**
  - Not an extensible architecture for the data analysis system
  - Limited support for coordinated data navigation (pan, zoom)
  - Lack of support for hierarchical data (fix: dot operator)
  - Unclear mix of traditional DW (dimensions, measures, etc.) and non-traditional BI (no explicit ETL, filtering) features.
Possible Improvements

- Generate database tables from a selected set of marks. Use selected mark in one display as the data input to another.

- Integrate a table lens, instead of having to click a mark to view its details.

- Exploring interaction techniques for navigating hierarchical structures of multi-dim databases.

- Provide an adapter to “link” external data sources without explicitly storing data in the analysis system.
Polaris: Extended Formalism

- Additional formalism defined in papers*:
  - specification of different graph types
  - encoding of data as retinal properties of marks in graphs
  - data transformations
  - translation of visual specification into SQL queries

* Relevant papers:
Query, Analysis, and Visualization of Hierarchically Structured Data using Polaris
Chris Stolte, Diane Tang and Pat Hanrahan
Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 2002.

Polaris: A System for Query, Analysis and Visualization of Multi-dimensional Relational Databases (extended paper)
Chris Stolte, Diane Tang and Pat Hanrahan
Formalism: Extensions

- Can mix graph types in single visualization:
Dot (.) operator: Hierarchies

- Many data warehouses have hierarchical dimensions:
  - **Time:** Year, Month, Day
  - **Location:** Country, State, Region
- Dot (.) works like Nest (/) except it exploits the defined hierarchies
  - based on semantics not tuples in database
- Demo
QUESTIONS?

http://graphics.stanford.edu/projects/polaris/