Learning Database Queries By Snapping Blocks

Yasin Silva       Jaime Chon
Arizona State University
What is DBSnap?

• Web-based application to build database queries (relational algebra) by snapping blocks
• DBSnap supports the construction of intuitive database query trees
• Highly dynamic
  – Shows query results as the query is being built
  – User can inspect intermediate results of any query node
DBSnap Blocks

- Datasets
  - Table Students
  - Table Course_Student

- Operators
  - Select
  - Natural Join
DBSnap Operators

• Selection
  • List only the senior students
  • \( \sigma_{\text{LEVEL} = "Senior"}(\text{Students}) \)

• Projection
  • List the ID and last name of students
  • \( \pi_{\text{SID, LNAME}}(\text{Students}) \)
DBSnap Operators

• Natural Join
  • List the students and the courses they are taking
  • Students $\bowtie$ Course_Student

• Aggregation
  • Compute the number of students per level
  • $\text{G}_{\text{count}(\text{SID})}(\text{Students})$
DBSnap Query Example 1

- List the professors with last name "Kelly" and the courses (CID) they are teaching
- \( \sigma_{\text{LNAME}="Kelly"}(\text{Professors} \bowtie \text{Course}_\text{Professors}) \)
• Compute the number of students of each level registered in each course considering only Junior and Senior students

\[ \sigma_{\text{Level}=\text{"Junior" OR Level=\text{"Senior"}}} (\text{CID, CName, Level}) \sigma_{\text{Level}=\text{"Junior" OR Level=\text{"Senior"}}} (\text{count(SID)}) \]

\[ \sigma_{\text{Level}=\text{"Junior" OR Level=\text{"Senior"}}} (\text{CID, CName, Level} G_{\text{count(SID)}} \text{Level}) \]

\[ (\text{Students} \bowtie \text{Course_Student}) \bowtie \text{Courses} \]
1. Operator palette  
2. Dataset palette  
3. Query area  
4. Relational algebra panel  
5. Query result panel  
6. Node result panel
DBSnap's Architecture
## Comparison of Educational Tools

<table>
<thead>
<tr>
<th>Feature</th>
<th>DBSnap</th>
<th>WinRDBI</th>
<th>iDFQL</th>
<th>Relational</th>
<th>RALT</th>
<th>Query Visualiser</th>
<th>Bags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation technologies</td>
<td>HTML5,</td>
<td>Java</td>
<td>Borland C++</td>
<td>Python</td>
<td>Java Swing</td>
<td>.NET, Mono</td>
<td>HTML5, JavaScript</td>
</tr>
<tr>
<td>Publicly available</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Open source code</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Block-based query editor</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Shows RA expression</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Web application</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Uses tree-based representation to build queries</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Automatically updates query result</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Intermediate results</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Side-by-side queries</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Can work without DB connection</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Build-in datasets</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Allows importing or connecting to custom data</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Views</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>
DBSnap Demo
Thank you!

Try DBSnap now!
http://www.public.asu.edu/~ynsilva/dbsnap/