Ultrawrap: Using SQL Views for RDB2RDF
Juan F. Sequeda1, Conor Cunningham2, Rudy Depena1, Daniel P. Miranker1
1Department of Computer Sciences, The University of Texas at Austin
2Microsoft Corporation

Goals
• Fully automatic publication of legacy relational databases to the Semantic Web
• Assuring real-time consistency between the relational and RDF presentation of the data
• Making maximal use of existing SQL Infrastructure

Do existing commercial SQL query engines already subsume all the algorithms needed to support effective SPARQL execution on relational data?

Current Results

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</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>Jena SDB</td>
<td>4.41</td>
<td>4.33</td>
<td>6.27</td>
<td>7.12</td>
<td>12.36</td>
<td>1.45</td>
<td>11.94</td>
<td>6.69</td>
<td>8.38</td>
<td>5.39</td>
<td>2.76</td>
<td>4.34</td>
</tr>
<tr>
<td>Sesame</td>
<td>2.49</td>
<td>0.86</td>
<td>3.52</td>
<td>3.78</td>
<td>7.31</td>
<td>1.76</td>
<td>15.51</td>
<td>3.02</td>
<td>1.17</td>
<td>3.63</td>
<td>1.49</td>
<td>0.65</td>
</tr>
<tr>
<td>Virtuoso RDF Views</td>
<td>8.29</td>
<td>2.77</td>
<td>9.79</td>
<td>16.13</td>
<td>1.89</td>
<td>0.09</td>
<td>16.59</td>
<td>6.63</td>
<td>2.14</td>
<td>6.96</td>
<td>9.50</td>
<td>3.44</td>
</tr>
<tr>
<td>D2R Server</td>
<td>5.03</td>
<td>5.28</td>
<td>7.93</td>
<td>7.63</td>
<td>222.73</td>
<td>0.94</td>
<td>10.96</td>
<td>12.46</td>
<td>13.37</td>
<td>7.16</td>
<td>30.61</td>
<td>2.55</td>
</tr>
<tr>
<td>Ultrawrap</td>
<td>1</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Native SQL</td>
<td>0.94</td>
<td>0.67</td>
<td>0.90</td>
<td>0.80</td>
<td>1.09</td>
<td>0.62</td>
<td>0.67</td>
<td>0.72</td>
<td>1.03</td>
<td>1.02</td>
<td>0.94</td>
<td>0.30</td>
</tr>
</tbody>
</table>

Current Results
• Current experiment on 1M triples of the Berlin SPARQL Benchmark
• Using Microsoft SQL Server, we currently determine that Ultrawrap is faster than other approaches
• SQL Server does not compile out self joins

Future Work
• Run with other RDBMS
• Scale to 1 Billion triples

Diagram:

Step 0: Start
Step 1: Create PO
Step 2: Virtual Triple Store
Step 3: Naïve SPARQL2SQL
Step 4: SQL Optimizer is the Rewriter

Putative Ontology

Data

Schema

Create View

TripleView

Data

Schema

Translator

Create

View

Current Results and Future Work