Ultrawrap: Using SQL Views for RDB2RDF

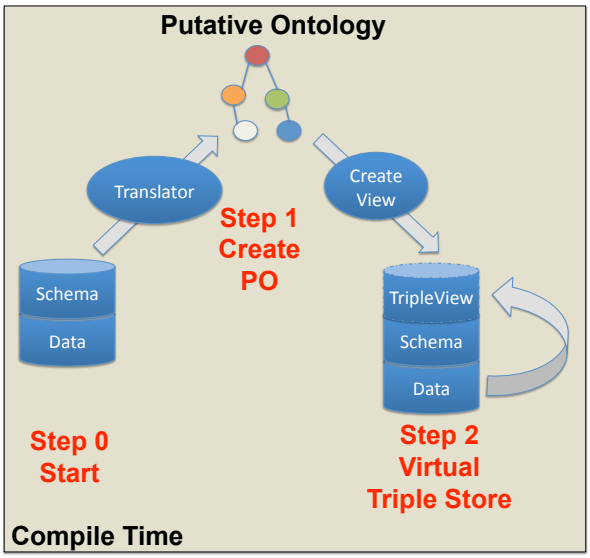
Juan F. Sequeda[1], Conor Cunningham[2], Rudy Depena[1], Daniel P. Miranker[1]
[1]Department of Computer Sciences, The University of Texas at Austin
[2]Microsoft Corporation

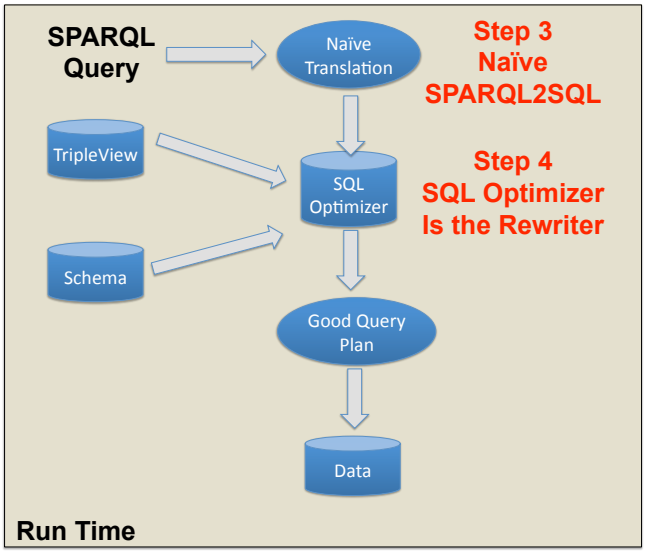
Goals

- 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?

Ultrawrap





Step 1: Create PO

Create a Putative Ontology, which is a syntactic transformation from a relational database schema to an ontology

Step 2: Virtual Triple Store

```
CREATE VIEW TripleView(s,p,o) AS
SELECT "Product" +Product.id as
s, "rdf:type" as p "Product" as
o FROM Product
UNION
SELECT "Product" +Product.id as
s, "label" as p "ABC" as o FROM
Product
UNION
SELECT …
UNION …
```

Step 3: Naïve SPARQL2SQL

```
SELECT ?product ?label
WHERE {
  ?product label ?label.
  ?product propNum1 1.
  ?product propNum2 2.}


SELECT t1.s as product,
t1.o as label,
FROM tripleview t1, t2, t3
WHERE t1.s = t2.s AND t1.p
= 'label'
AND t2.p = 'propNum1' AND
t2.o = 1
AND t1.s = t3.s AND t3.p =
'propNum2'
AND t3.o = 2
```

Step 4: SQL Optimizer is the Rewriter

```
TripleView(1, label, ABC) :- Product(1,ABC, _, _)
TripleView(1, propNum1, 1) :- Product(1,_, 1, _)
TripleView(1, propNum1, 2) :- Product(1,_, _, 2)
```

SQL Query on the TripleView

```
Query(X, Y):-TripleView(X, label, Y),
TripleView(X, propNum1, 1),
TripleView(X, propNum2, 2)
```

SQL Query on the Relational Data

```
SELECT id, label FROM product
WHERE propNum1 = 1 and propNum2 = 2
Query(X, Y) :- Product(X, Y, 1, 2)
```

Evaluate SQL Query on the TripleView

```
Query(X,Y):-Product(X,Y,1,_),Product(X,Y,_, 2)
Query(X,Y):- Product(X, Y, 1, 2)
```

Current Results and Future Work

Query	1	2	3	4	5	6	7	8	9	10	11	12
Jena SDB	4.41	4.33	6.27	7.12	12.36	1.45	11.94	6.69	8.38	5.39	2.76	4.34
Sesame	2.49	**0.86**	3.52	3.78	7.31	1.76	15.51	3.02	1.17	3.63	1.49	**0.65**
Virtuoso RDF Views	8.29	2.77	9.79	16.13	1.89	**0.09**	16.59	6.83	2.14	6.96	9.50	3.44
D2R Server	5.03	5.28	7.93	7.63	222.73	0.94	10.96	12.46	13.37	7.16	30.61	2.55
Ultrawrap	**1**	1.00	**1.00**	**1.00**	**1.00**	1.00	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	1.00
Native SQL	0.94	0.67	0.90	0.80	1.09	0.62	0.67	0.72	1.03	1.02	0.94	0.30

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