OpenLink Virtuoso

Virtuoso

- Cross-platform server for SQL, XML and RDF data management
  - Reduces the cost of bringing together data from different data sources

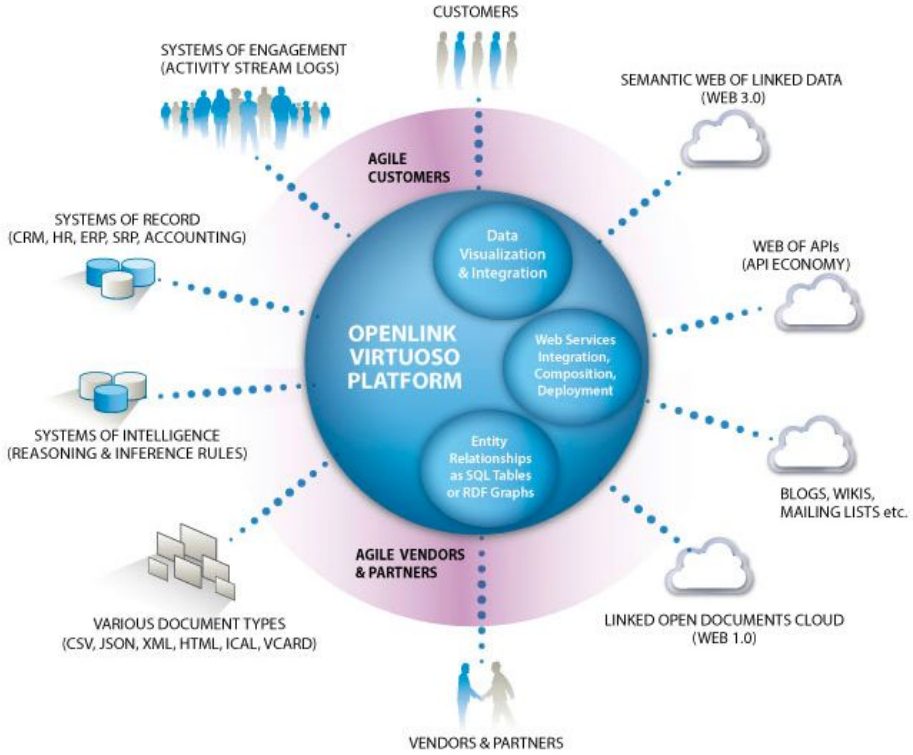
- Provides transparent access to your existing data sources, which are typically databases from different database vendors
  - All your databases are treated as single logical unit.

Virtuoso

Includes:

- Virtual database engine
- Web Services deployment platform
- Web application server
- SPARQL support and an RDF data store tightly integrated with its relational storage engine

Virtuoso


CUSTOMERS
SYSTEMS OF ENGAGEMENT
(ACTIVITY STREAM LOGS)
SEMANTIC WEB OF LINKED DATA
(WEB 3.0)
AGILE
CUSTOMERS
SYSTEMS OF RECORD
(CRM, HR, ERP, SRP, ACCOUNTING)
Data
Visualization
& Integration
WEB OF APIs
(API ECONOMY)
OPENLINK
VIRTUOSO
PLATFORM
Web Services
Integration,
Composition,
Deployment
SYSTEMS OF INTELLIGENCE
(REASONING & INFERENCE RULES)
Entity
Relationships
as SQL Tables
or RDF Graphs
BLOGS, WIKIS,
MAILING LISTS etc.
AGILE VENDORS
& PARTNERS
VARIOUS DOCUMENT TYPES
(CSV, JSON, XML, HTML, ICAL, VCARD)
LINKED OPEN DOCUMENTS CLOUD
(WEB 1.0)
VENDORS & PARTNERS

Mapping Relational Data to RDF with Virtuoso's RDF Views

[1] https://virtuoso.openlinksw.com/whitepapers/relational%20rdf%20views%20mapping.html

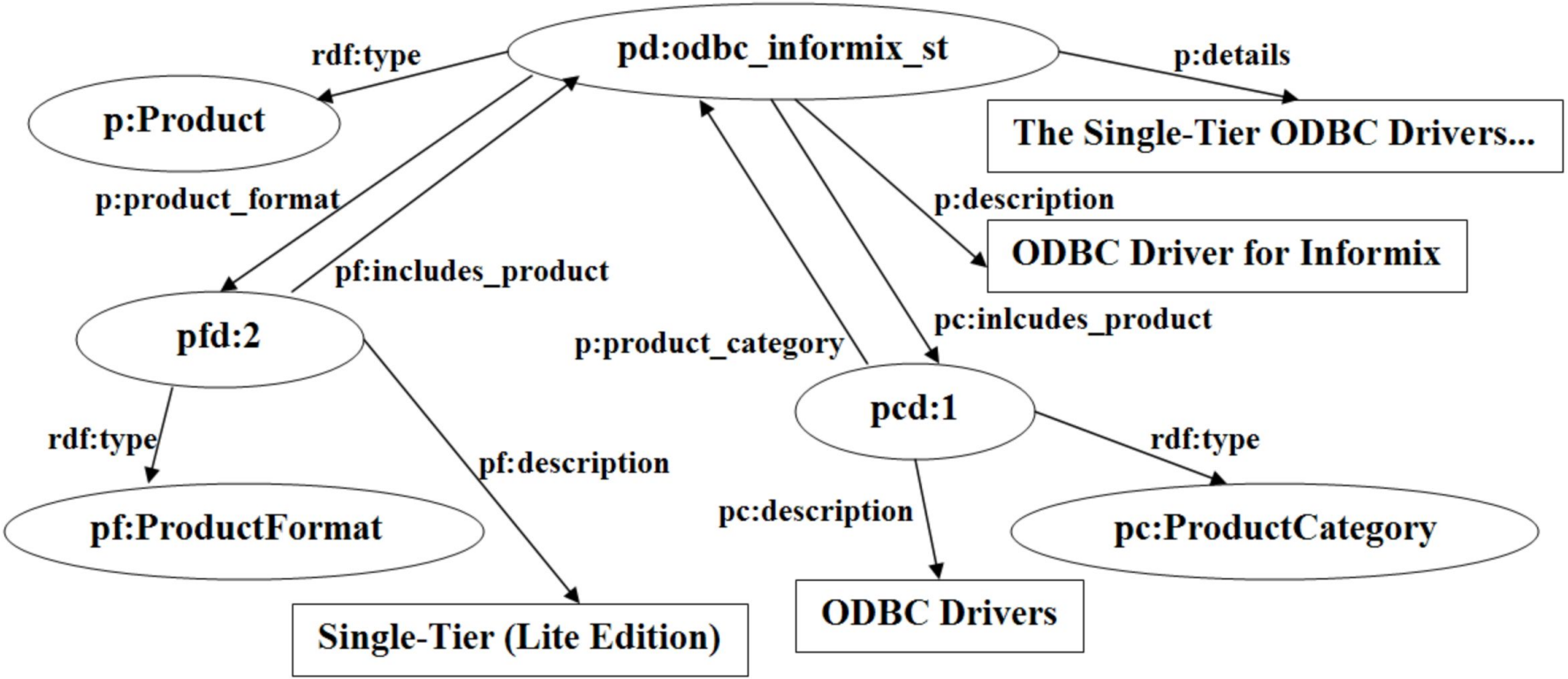

pd:odbc_informix_st
rdf:type
p:Product
p:details
The Single-Tier ODBC Drivers...
p:product_format
p:description
ODBC Driver for Informix
pf:includes_product
pfd:2
p:product_category
pc:inlcudes_product
pcd:1
rdf:type
pf:description
rdf:type
pf:ProductFormat
pc:description
pc:ProductCategory
Single-Tier (Lite Edition)
ODBC Drivers

Table: oplweb2.oplweb.product		
product_id	varchar(25)	primary key
product_description	varchar(125)	
long_description	long varchar	
product_cat_id	integer	(foreign key)
product_format_id	integer	(foreign key)

Table: oplweb2.oplweb.product_category		
product_cat_id	integer	primary key
product_category_description	varchar(50)	

Table: oplweb2.oplweb.product_format		
product_format_id	integer	primary key
product_format_description	varchar(75)	

Virtuoso

- SPARQL support

- RDF data store **integrated** with relational storage engine

Virtuoso - What are RDF Views?

- Virtuoso's RDF Views **map relational data into RDF**
- Allow the RDF representation of the relational data to be **customised**

Virtuoso - What are RDF Views?

- Virtuoso's RDF Views **map relational data into RDF**
- Allow the RDF representation of the relational data to be **customised**
- Declarative Meta Schema Language
  for defining the **mapping of SQL data to RDF ontologies**

Virtuoso - What are RDF Views?

- The mapping is **dynamic**
- Changes to the underlying data are **reflected immediately** in the RDF representation

- **No changes are required** to the underlying relational schema

Virtuoso

- Expose **pre-existing relational data** as virtual RDF graphs
- Available for querying through SPARQL or SPASQL(SPARQL embedded in SQL)
- **No physical regeneration** of relational data

Virtuoso - MSL Building Blocks

- RDF Views =

Virtuoso RDF Meta-Schema + Meta-Schema Language (MSL)

Virtuoso - MSL Building Blocks

- Main building blocks:
  - Quad map patterns
  - IRI classes
  - Literal classes

Virtuoso - MSL Building Blocks

- Organizational enhancements
  - Group map patterns
  - Quad storage
  - Naming

Virtuoso - Organizational Enhancements

- Make it easier to administer large sets of quad map patterns

- **Group map patterns** - group together map patterns sharing a common graph
- **Quad storage** - groups together group map patterns as a named set
- Both allow map patterns to be altered/deleted individually or as a group

Virtuoso - Quad Map Pattern

- Basic unit of **meta schema**
- Defines transformation from one set of **relational columns into triples** that match one SPARQL graph pattern
- Comprises 4 declarations of quad map values - each calculates a triple field value from the SQL data

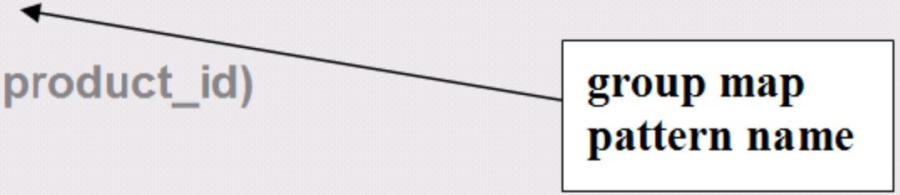
Virtuoso - Named Quad Map Pattern

```
graph
{
        p:product_iri (oplweb2.oplweb.product.product_id) p:description
        oplweb2.oplweb.product.product_description
        as virtrdf:product_product_description .
}
```

Virtuoso - Group Quad Map Pattern

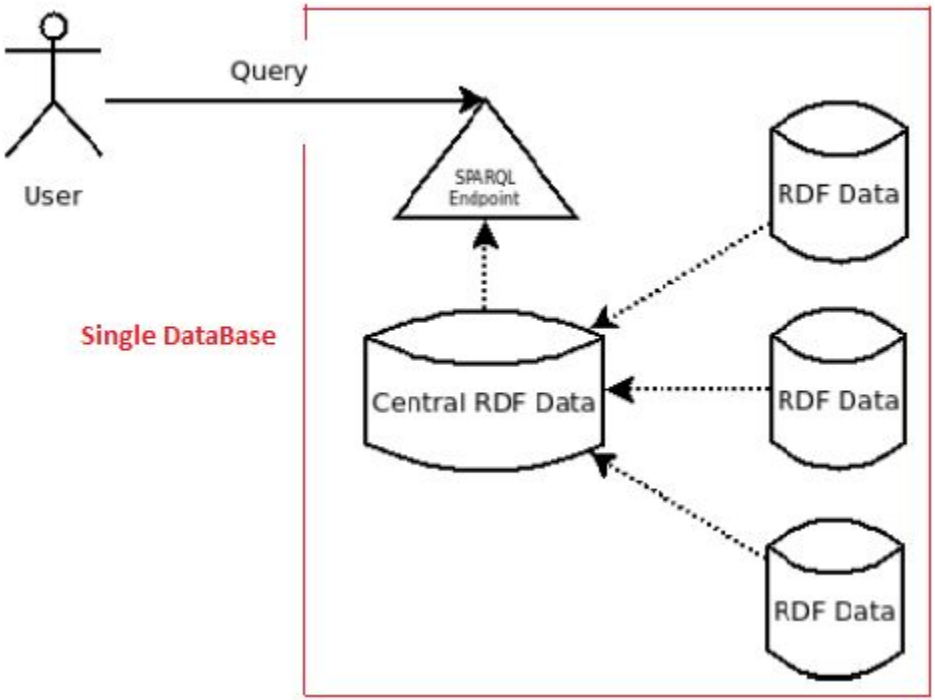
- Patterns for the same graph can be grouped together

```
create virtrdf:product_portfolio as graph
{
  p:product_iri (oplweb2.oplweb.product.product_id)
    a p:Product
    as virtrdf:product_product_id ;
  p:description oplweb2.oplweb.product.product_description
    as virtrdf:product_product_description .
}
```



Federated Query Extension in SPARQL

Virtuoso - Overview Federated DataBase System



Virtuoso - FAQ over federated query

1.4.19. Does Virtuoso support federated triple stores? If there are multiple SPARQL end points, can Virtuoso be used to do queries joining between these?

This is a planned extension. The logic for optimizing message flow between multiple end-points on a wide-area network is similar to the logic for message-optimization on a cluster. This will allow submitting a query with a list of end-points. The query will then consider triples from each of the end points, as if the content of all the end points were in a single store. [2]

Virtuoso - Message-optimization

- Cluster [3]
  - local neighborhoods for specific tasks
  - increase efficiency using closer processors
  - Problem: network topology unknown

- Federated Database
  - Queries in “nearby” SPARQL Endpoints
  - Queries sent to suitable SPARQL Endpoints

Virtuoso - Proposed extension in SPARQL 2.0

- Proposes extensions to answer complex structured queries, including analytics [4]
  - Endpoint Self-Description
  - SPARQL Federation

Virtuoso - Proposed extension in SPARQL 2.0

- Endpoint Self-Description
  - Endpoint SPARQL advertise about your own version
  - Query confirms the existence of the triples

- Information
  - Version Endpoit SPARQL
  - Terminations maximum timeout
  - SPARQL Capabilities supported
  - Xquery functions supported
  - Void Descriptor

Virtuoso - Proposed extension in SPARQL 2.0

- SPARQL Federation
  - Federated query transparents
  - Data matched at several Endpoints SPARQL
  - Solution: Option

Virtuoso - Proposed extension in SPARQL 2.0

- SPARQL Federation

```
select ?contact1
where
 {
  {
    select ?contact1
    where
     {
       ?me foaf:nick "Orri" .
       ?me foaf:knows ?f .
       ?f foaf:name ?contact1
     }
  }
  option (federated <http://www.semanticweb.com/sparql>)
   {
   select ?contact2
   where
     {
       ?me foaf:nick "Orri" .
       ?me foaf:knows ?f .
       ?f foaf:name ?contact2
     }
   }
  option (federated <http://www.myopenlink.com/sparql>) .
  filter (?contact1 = ?contact2)
}
```

Virtuoso - SPARQL 1.1 Federated Query Extension

- Keyword **SERVICE** [5]
  - Invoke a portion of a SPARQL query against a remote SPARQL endpoint

- 2 Examples
  - Simple query to a remote SPARQL endpoint
  - SPARQL query with OPTIONAL to two remote SPARQL endpoints

Virtuoso - SPARQL 1.1 Federated Query Extension

- Simple query to a remote SPARQL endpoint

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
SELECT ?name
FROM <http://example.org/myfoaf.rdf>
WHERE
{
  <http://example.org/myfoaf/I> foaf:knows ?person .
  SERVICE <http://people.example.org/sparql> {
    ?person foaf:name ?name . }
}
```

```
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix : <http://example.org/> .

:people15  foaf:name     "Alice" .
:people16  foaf:name     "Bob" .
:people17  foaf:name     "Charles" .
:people18  foaf:name     "Daisy" .
```

```
<http://example.org/myfoaf/I> <http://xmlns.com/foaf/0.1/knows>  <http://example.org/people15> .
```

Virtuoso - SPARQL 1.1 Federated Query Extension

- SPARQL query with OPTIONAL to two remote SPARQL endpoints

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
SELECT ?person ?interest ?known
WHERE
{
  SERVICE <http://people.example.org/sparql> {
    ?person foaf:name ?name .
    OPTIONAL {
      ?person foaf:interest ?interest .
      SERVICE <http://people2.example.org/sparql> {
        ?person foaf:knows ?known . } }
  }
}
```

```
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix : <http://example.org/> .

:people15  foaf:knows    :people18 .
:people18  foaf:name     "Mike" .
:people17  foaf:knows    :people19 .
:people19  foaf:name     "Daisy" .
```

```
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
@prefix : <http://example.org/> .

:people15  foaf:name     "Alice" .
:people16  foaf:name     "Bob" .
:people17  foaf:name     "Charles" .
:people17  foaf:interest     <http://www.w3.org/2001/sw/rdb2rdf/> .
```

Virtuoso - SPARQL 1.1 Federated with Virtuoso

- SPARQL 1.1 and Openlink Virtuoso: First steps with federated queries

- Evaluation of Federated Database for Distributed Applications in e-Government

- Querying over Federated SPARQL Endpoints — A State of the Art Survey

Virtuoso - SPARQL 1.1 Federated Query with Virtuoso

- SPARQL 1.1 and Openlink Virtuoso: First steps with federated queries [6]
  - Informal
  - Linux
  - Previous information

```
SELECT ?p ?o
WHERE
{
 SERVICE <http://DBpedia.org/sparql >
 { SELECT ?p ?o
 WHERE { <http://dbpedia.org/resource/Saquarema > ?p ?o . }
 }
}
```

Virtuoso - SPARQL 1.1 Federated with Virtuoso

- Evaluation of Federated Database for Distributed Applications in e-Government [7]
  - Public administration with e-government applications services
  - Distributed municipal administrative units (AUs)
  - real-time information systems

- About Virtuoso
  - RDF graph store with built-in SPARQL for distributed query
  - Several RDF triples

Virtuoso - SPARQL 1.1 Federated with Virtuoso

- Querying over Federated SPARQL Endpoints — A State of the Art Survey [8]
  - SPARQL Federated Extensions
  - Extensions VALUES and BINDING

Framework	Platform	SERVICE	BINDINGS	VALUES
ARQ	Jena	✓	✗	✓
SPARQL-FED	Virtuoso	✓	✗	✓
Sesame	Sesame	✓	✓	✓
SPARQL-DQP	OGSA-DAI and OGSA-DQP	✓	✓	✗

Virtuoso - References

[1] https://virtuoso.openlinksw.com/whitepapers/relational%20rdf%20views%20mapping.html

[2] http://docs.openlinksw.com/virtuoso/virtuosofaq19/

[3] https://en.wikipedia.org/wiki/Message_passing_in_computer_clusters

[4]https://virtuoso.openlinksw.com/tutorials/sparql/OpenLink_Data_Spaces_SIOC_and_SPARQL_Guide/OpenLink_Data_Spaces_SIOC_and_SPARQL_Guide_Part_II/OpenLink_Data_Spaces_SIOC_and_SPARQL_Guide_Part_II.html

[5] https://www.w3.org/TR/sparql11-federated-query/

[6]https://thoughtsasaservice.wordpress.com/2012/05/17/sparql-1-1-and-openlink-virtuoso-first-steps-with-federated-queries

[7] Evaluation of Federated Database for Distributed Applications in e-Government (artigo)

[8] Querying over Federated SPARQL Endpoints — A State of the Art Survey (artigo)

[9] http://docs.openlinksw.com/virtuoso/rdfviewnorthwindexample1/

[10] http://data.legilux.public.lu/sparql?help=views

[11] https://medium.com/virtuoso-blog/rdf-views-generate-b0538101a724