Ontology-Based Data Access with Ontop

Benjamin Cogrel
benjamin.cogrel@unibz.it

KRDB Research Centre for Knowledge and Data
Free University of Bozen-Bolzano, Italy

Freie Universität Bozen
Libera Università di Bolzano
Free University of Bozen-Bolzano

Optique

ESSLLI,
Bozen-Bolzano, 26 August 2016

Ontology-Based Data Access (OBDA)

Outline

Outline

unibz

Relational database

uni1-course

c_id	title
1234	Linear algebra

uni1-academic

a_id	first_name	last_name	position
1	Anna	Chambers	1
2	Edward	May	9

uni1-teaching

c_id	a_id
1234	1
1234	2

N-ary facts

uni1-course(1234, 'Linear algebra')
uni1-academic(1, 'Anna', 'Chambers', 1)   uni1-teaching(1234, 1)
uni1-academic(2, 'Edward', 'May', 9)   uni1-teaching(1234, 2)

unibz

Relational database

uni1-course

c_id	title
1234	Linear algebra

uni1-academic

a_id	first_name	last_name	position
1	Anna	Chambers	1
2	Edward	May	9

uni1-teaching

c_id	a_id
1234	1
1234	2

N-ary facts

uni1-course(1234, 'Linear algebra')
uni1-academic(1, 'Anna', 'Chambers', 1)
uni1-academic(2, 'Edward', 'May', 9)
uni1-teaching(1234, 1)
uni1-teaching(1234, 2)

Closed-World Assumption (over DB-specific predicates)

uni1-course(5432, 'Machine learning') does not hold (not in the table)

Querying relational databases
With First-Order Queries

Datalog notation and Structured Query Language (SQL)

Conjunctive Query (CQ)

$\boldsymbol{q}(l) \leftarrow \texttt{uni1-academic}(a, f, l, p), \texttt{uni1-teaching}(c, a)$

Union of CQs (UCQ)

$\boldsymbol{q}(a) \leftarrow \texttt{uni1-academic}(a, f, l, p), p = 1$
$\boldsymbol{q}(a) \leftarrow \texttt{uni1-teaching}(c, a)$

Complex FO queries (SQL without recursion)

```
SELECT teach."c_id", ac."last_name"
FROM "uni1"."teaching" teach
     LEFT JOIN "uni1"."academic" ac
ON ac."a_id" = teach."a_id"
```

Complex algebra (e.g. outer joins), aggregation, etc.

Toy example: University Information System

Relational source

uni1-course

c_id	title
1234	Linear algebra

uni1-academic

a_id	first_name	last_name	position
1	Anna	Chambers	1
2	Edward	May	9
3	Rachel	Ward	8

uni1-teaching

c_id	a_id
1234	1
1234	2

uni1-student

s_id	first_name	last_name
1	Mary	Smith
2	John	Doe

Foreign keys:
from `uni1-teaching.c_id` to `uni1-course.c_id`
from `uni1-teaching.a_id` to `uni1-academic.a_id`

Information need	Datalog and SQL queries

1. First and last names of the students	$\boldsymbol{q}(f, l) \leftarrow \texttt{uni1-student}(s, f, l)$
	
	`SELECT DISTINCT "first_name", "last_name"`
	`FROM "uni1"."student"`

2. First and last names of the persons	$\boldsymbol{q}(f, l) \leftarrow \texttt{uni1-student}(s, f, l)$
	$\boldsymbol{q}(f, l) \leftarrow \texttt{uni1-academic}(a, f, l, p)$
	
	`SELECT DISTINCT "first_name", "last_name"`
	`FROM "uni1"."student"`
	`UNION`
	`SELECT DISTINCT "first_name", "last_name"`
	`FROM "uni1"."academic"`

Information need	Datalog and SQL queries

3. Course titles and teacher names	$\boldsymbol{q}(t, l) \leftarrow \texttt{uni1-course}(c, t), \texttt{uni1-academic}(a, f, l, p), \texttt{uni1-teaching}(c, a)$
	
	`SELECT DISTINCT co."title", ac."last_name"`
	`FROM "uni1"."course" co,`
	`"uni1"."academic" ac,`
	`"uni1"."teaching" teach`
	`WHERE co."c_id" = teach."c_id"`
	`AND ac."a_id" = teach."a_id"`

4. All the teachers	$\boldsymbol{q}(a) \leftarrow \texttt{uni1-teaching}(c, a)$
	$\boldsymbol{q}(a) \leftarrow \texttt{uni1-academic}(a, f, l, p), p \in [1, 8]$
	
	`SELECT DISTINCT "a_id"`
	`FROM "uni1"."teaching" UNION`
	`SELECT DISTINCT "a_id"`
	`FROM "uni1"."academic"`
	`WHERE "position" BETWEEN 1 AND 8`

Integration of a second source

Fusion of two universities

uni2-person

pid	fname	lname	status
1	Zak	Lane	8
2	Mattie	Moses	1
3	Céline	Mendez	2

uni2-course

cid	lecturer	lab_teacher	topic
1	1	3	Information security

Foreign keys:

- from `uni2-course.lecturer` to `uni2-person.pid`
- from `uni2-course.lab_teacher` to `uni2-person.pid`

unibz

Translation of information needs (cont.)

Two datasources

<table>
<tr><th>Information need</th><th>Datalog and SQL queries</th></tr>
<tr><td>1. First and last names of the students</td><td>

$$\boldsymbol{q}(f,l) \leftarrow \texttt{uni1-student}(p,f,l)$$
$$\boldsymbol{q}(f,l) \leftarrow \texttt{uni2-person}(p,f,l,s), s \in [1,2]$$

```
SELECT DISTINCT "first_name", "last_name"
FROM "uni1"."student"
UNION
SELECT DISTINCT "fname" AS "first_name",
                "lname" AS "last_name"
FROM "uni2"."person"
WHERE "status" BETWEEN 1 and 2
```

</td></tr>
</table>

Translation of information needs (cont.)

Two datasources

Information need	Datalog and SQL queries

2. First and last names of the persons	$\boldsymbol{q}(f,l) \leftarrow \texttt{uni1-student}(p,f,l)$
	$\boldsymbol{q}(f,l) \leftarrow \texttt{uni1-academic}(p,f,l,s)$
	$\boldsymbol{q}(f,l) \leftarrow \texttt{uni2-person}(p,f,l,s)$
	
	`SELECT DISTINCT "first_name", "last_name"`
	`FROM "uni1"."student"`
	`UNION`
	`SELECT DISTINCT "first_name", "last_name"`
	`FROM "uni1"."academic"`
	`UNION`
	`SELECT DISTINCT "fname" AS "first_name",`
	`"lname" AS "last_name"`
	`FROM "uni2"."person"`

Translation of information needs (cont.)

Two datasources

Information need	Datalog and SQL queries
3. Course titles and teacher names	$q(t, l) \leftarrow$ uni1-course(c, t), uni1-academic(a, f, l, p), uni1-teaching(c, a) $q(t, l) \leftarrow$ uni2-person(p, f, l, s), uni2-course(c, e, b, t), $(p = e) \vee (p = b)$

$$
\begin{aligned}
\boldsymbol{q}(t,l) \leftarrow\ & \texttt{uni1-course}(c,t), \\
& \texttt{uni1-academic}(a,f,l,p), \\
& \texttt{uni1-teaching}(c,a) \\
\boldsymbol{q}(t,l) \leftarrow\ & \texttt{uni2-person}(p,f,l,s), \\
& \texttt{uni2-course}(c,e,b,t), \\
& (p = e) \vee (p = b)
\end{aligned}
$$

```sql
SELECT DISTINCT co."title", ac."last_name"
FROM "uni1"."course" co,
     "uni1"."academic" ac,
     "uni1"."teaching" teach
WHERE co."c_id" = teach."c_id"
      AND ac."a_id" = teach."a_id"
UNION
SELECT DISTINCT co."topic" AS "title",
                pe."lname" AS "last_name"
FROM "uni2"."person" pe,
     "uni2"."course" co
WHERE pe."pid" = co."lecturer"
      OR pe."pid" = co."lab_teacher"
```

Translation of information needs (cont.)

Two datasources

Information need	SQL query
4. All the teachers	`SELECT DISTINCT 'uni1/' || "a_id" AS "id" FROM "uni1"."teaching" UNION SELECT DISTINCT 'uni1/' || "a_id" AS "id" FROM "uni1"."academic" WHERE "position" BETWEEN 1 AND 8 UNION SELECT DISTINCT 'uni2/' || "lecturer" AS "id" FROM "uni2"."course" UNION SELECT DISTINCT 'uni2/' || "lab_teacher" AS "id" FROM "uni2"."course" UNION SELECT DISTINCT 'uni2/' || "pid" AS "id" FROM "uni2"."person" WHERE "status" BETWEEN 6 AND 9`

```sql
SELECT DISTINCT 'uni1/' || "a_id"  AS "id"
FROM "uni1"."teaching"
UNION
SELECT DISTINCT 'uni1/' || "a_id"  AS "id"
FROM "uni1"."academic"
WHERE "position" BETWEEN 1 AND 8
UNION
SELECT DISTINCT 'uni2/' || "lecturer" AS "id"
FROM "uni2"."course"
UNION
SELECT DISTINCT 'uni2/' || "lab_teacher" AS "id"
FROM "uni2"."course"
UNION
SELECT DISTINCT 'uni2/' || "pid" AS "id"
FROM "uni2"."person"
WHERE "status" BETWEEN 6 AND 9
```

Industrial case: stratigraphic model design

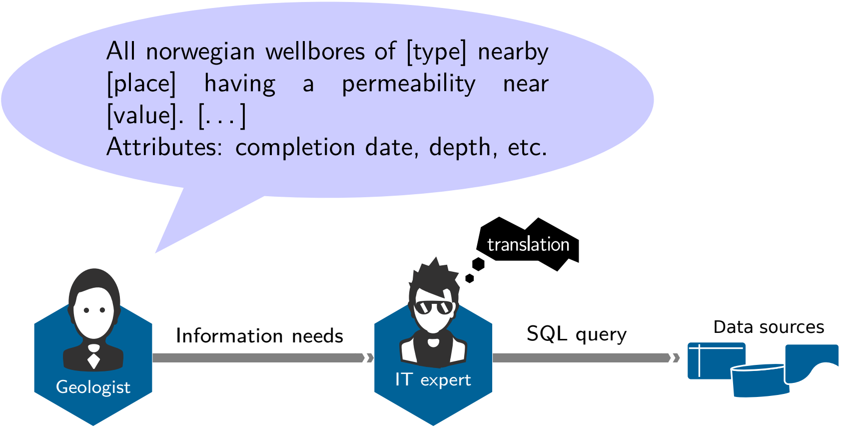
Users: domain experts

- ∼ 900 geologists et geophysicists
- Data collecting: 30-70% of their time

Sources

- *Exploitation and Production Data Store (EPDS)*: ∼ 1500 tables (100s GB)
- *OpenWorks*
- *Norwegian Petroleum Directorate FactPages (NPD FactPages)*

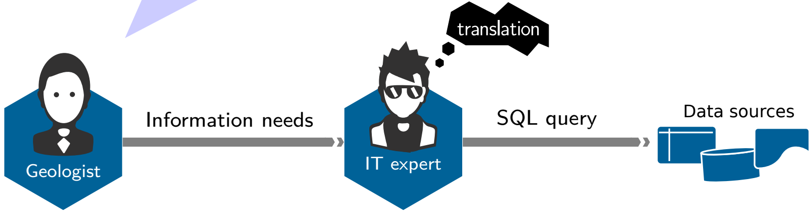
Designing a new (ad-hoc) query



NB: Simplified information needs

Designing a new (ad-hoc) query

All norwegian wellbores of [type] nearby [place] having a permeability near [value]. [. . .]
Attributes: completion date, depth, etc.



Takes 4 days in average (with EPDS only)

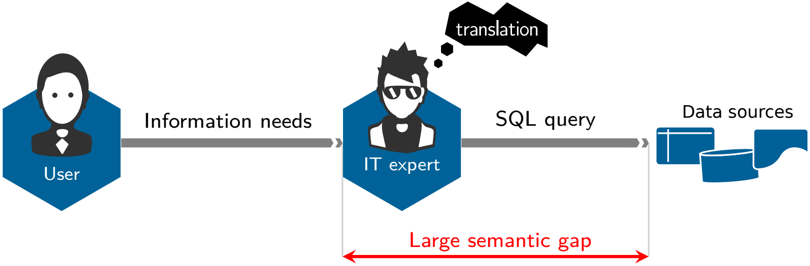
NB: Simplified information needs

unibz

Anonymized extract of a typical query

```
SELECT [...]
FROM
db_name.table1 table1,
db_name.table2 table2a,
db_name.table2 table2b,
db_name.table3 table3a,
db_name.table3 table3b,
db_name.table3 table3c,
db_name.table3 table3d,
db_name.table4 table4a,
db_name.table4 table4b,
db_name.table4 table4c,
db_name.table4 table4d,
db_name.table4 table4e,
db_name.table4 table4f,
db_name.table5 table5a,
db_name.table5 table5b,
db_name.table6 table6a,
db_name.table6 table6b,
db_name.table7 table7a,
db_name.table7 table7b,
db_name.table8 table8,
db_name.table9 table9,
db_name.table10 table10a,
db_name.table10 table10b,
db_name.table10 table10c,
db_name.table11 table11,
db_name.table12 table12,
db_name.table13 table13,
db_name.table14 table14,
db_name.table15 table15,
db_name.table16 table16
WHERE [...]
table2a.attr1=‘keyword’ AND
table3a.attr2=table10c.attr1 AND
table3a.attr6=table6a.attr3 AND
table3a.attr9=‘keyword’ AND
table4a.attr10 IN (‘keyword’) AND
table4a.attr1 IN (‘keyword’) AND
table5a.kinds=table4a.attr13 AND
table5b.kinds=table4c.attr74 AND
table5b.name=‘keyword’ AND
(table6a.attr19=table10c.attr17 OR
(table6a.attr2 IS NULL AND
table10c.attr4 IS NULL)) AND
table6a.attr14=table5b.attr14 AND
table6a.attr2=‘keyword’ AND
(table6b.attr14=table10c.attr8 OR
(table6b.attr4 IS NULL AND
table10c.attr7 IS NULL)) AND
table6b.attr19=table5a.attr55 AND
table6b.attr2=‘keyword’ AND
table7a.attr19=table2b.attr19 AND
table7a.attr17=table15.attr19 AND
table4b.attr11=‘keyword’ AND
table8.attr19=table7a.attr80 AND
table8.attr19=table13.attr20 AND
table8.attr4=‘keyword’ AND
table9.attr10=table16.attr11 AND
table3b.attr19=table10c.attr18 AND
table3b.attr22=table12.attr63 AND
table3b.attr66=‘keyword’ AND
table10a.attr54=table7a.attr8 AND
table10a.attr70=table10c.attr10 AND
table10a.attr16=table4d.attr11 AND
table4c.attr99=‘keyword’ AND
table4c.attr1=‘keyword’ AND
table11.attr10=table5a.attr10 AND
table11.attr40=‘keyword’ AND
table11.attr50=‘keyword’ AND
table2b.attr1=table1.attr8 AND
table2b.attr9 IN (‘keyword’) AND
table2b.attr2 LIKE ‘keyword’% AND
table12.attr9 IN (‘keyword’) AND
table7b.attr1=table2a.attr10 AND
table3c.attr13=table10c.attr1 AND
table3c.attr10=table6b.attr20 AND
table3c.attr13=‘keyword’ AND
table10b.attr16=table10a.attr7 AND
table10b.attr11=table7b.attr8 AND
table10b.attr13=table4b.attr89 AND
table13.attr1=table2b.attr10 AND
table13.attr20=’‘keyword’’ AND
table13.attr15=‘keyword’ AND
table3d.attr49=table12.attr18 AND
table3d.attr18=table10c.attr11 AND
table3d.attr14=‘keyword’ AND
table4d.attr17 IN (‘keyword’) AND
table4d.attr19 IN (‘keyword’) AND
table16.attr28=table11.attr56 AND
table16.attr16=table10b.attr78 AND
table16.attr5=table14.attr56 AND
table4e.attr34 IN (‘keyword’) AND
table4e.attr48 IN (‘keyword’) AND
table4f.attr89=table5b.attr7 AND
table4f.attr45 IN (‘keyword’) AND
table4f.attr1=‘keyword’ AND
table10c.attr2=table4e.attr19 AND
(table10c.attr78=table12.attr56 OR
(table10c.attr55 IS NULL AND
table12.attr17 IS NULL))
```

Semantic gap



Querying over tables

Requires a lot of knowledge about:

1. Magic numbers
   (e.g. 1 → full professor)
2. Cardinalities and normal forms
3. Closely-related information spread over many tables

Data integration

- Exacerbate these issues
- Variety: **challenge #1** for most Big Data initiatives

unibz

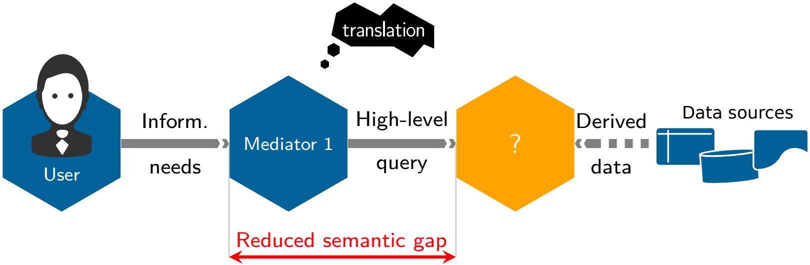
High-level translation

Main bottleneck: translation

- of the information needs
- ... into a **formal query**

Goal

Make such a translation easy
(Ideally: IT expertise not required)



Mediator 1 could be a user, an IT expert or a GUI

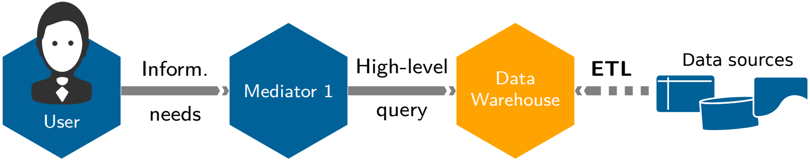
General approach: two steps

1. Translate the information needs into a **high-level query**
2. Answer the high-level query **automatically**

Choice 1: Generating a new representation

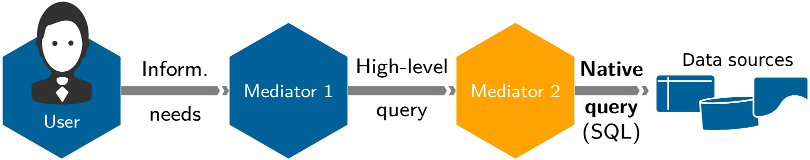
Extract Transform Load (ETL) process

E.g. relational data warehouse, triplestore



Virtual views

E.g. virtual databases (Teiid, Apache Drill, Exareme), **OBDA** (Ontop)



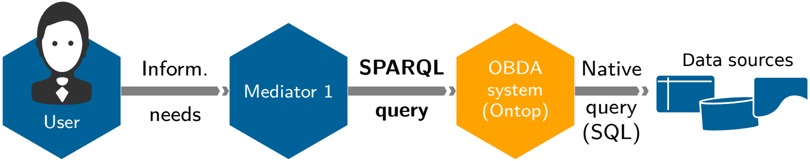
Choice 2: Which data format

New data format	Corresponding query language
Relational schema	SQL
JSON document	Mongo Aggregate, SQL (with e.g. Drill or Teiid)
XML document	XPath, XQuery, SQL (with e.g. Teiid)
RDF graph	SPARQL

unibz

Ontop approach

Ontology-Based Data Access (OBDA)



Choice 1: Generating a new representation

1. Extract Transform Load (ETL) process
2. **Virtual views**

Choice 2: Which data format

1. New relational schema, JSON or XML documents
2. **Resource Description Framework (RDF)**

Outline

1. Querying relational databases
2. RDF and other Semantic Web standards
   - RDF
   - SPARQL
   - OWL 2 QL
   - Mapping assertions
3. Ontology-Based Data Access
4. Optique platform
5. Conclusion

unibz

Resource Description Framework (RDF)

Subject

Predicate/Property

Object/Attribute

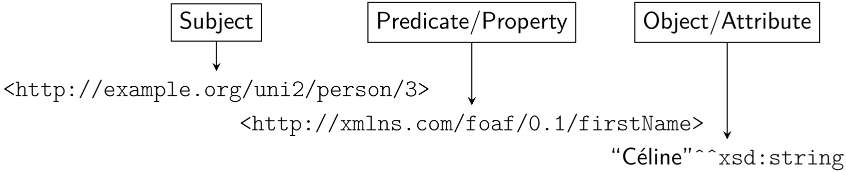
`<http://example.org/uni2/person/3>`

`<http://xmlns.com/foaf/0.1/firstName>`

“Céline”`^^xsd:string`

unibz

Resource Description Framework (RDF)




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

ABox assertion	AssociateProfessor(*uni2/person/1*)
RDF triple	`<uni2/person/1> a AssociateProfessor`

ABox assertion	lastName(*uni2/person/3*, 'Mendez')
RDF triple	`<uni2/person/3> foaf:lastName "Mendez"`

unibz

Resource Description Framework (RDF) (cont.)

ABox assertion	givesLecture(*uni2/person/3*, *uni2/course/1*)
RDF triple	`<uni2/person/3> :givesLecture <uni2/course/1>`

unibz

Resource Description Framework (RDF) (cont.)

ABox assertion	givesLecture(*uni2/person/3*, *uni2/course/1*)
RDF triple	<uni2/person/3> :givesLecture <uni2/course/1>

TBox axiom	givesLecture $\sqsubseteq$ teaches
RDF triple	:givesLecture rdfs:subPropertyOf :teaches

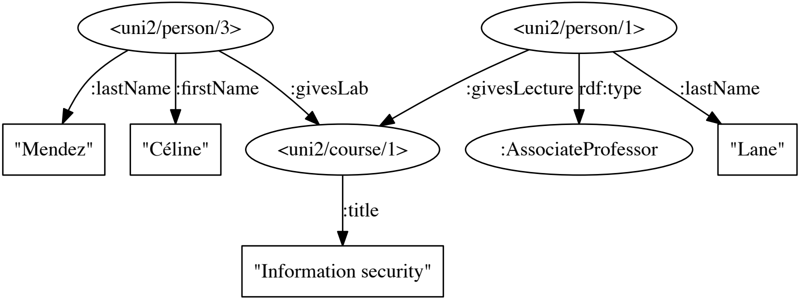
Resource Description Framework (RDF) (cont.)

ABox assertion	givesLecture(*uni2/person/3*, *uni2/course/1*)
RDF triple	`<uni2/person/3> :givesLecture <uni2/course/1>`

TBox axiom	givesLecture $\sqsubseteq$ teaches
RDF triple	`:givesLecture rdfs:subPropertyOf :teaches`

Advanced notions: blank nodes, named graph

RDF graph example (ABox)



SPARQL

SPARQL Protocol and RDF Query Language

Basic Graph Pattern (BGP) ≈ CQ

$\boldsymbol{q}(x, y) \leftarrow \mathsf{teaches}(x, y)$

```
PREFIX : <http://example.org/voc#>

SELECT ?x ?y
WHERE {
  ?x :teaches ?y .
}
```

UCQ ≈ BGP + UNION

$\boldsymbol{q}(x) \leftarrow \mathsf{Professor}(x)$
$\boldsymbol{q}(x) \leftarrow \mathsf{Student}(x)$

```
SELECT ?x
WHERE {
  { ?x a :Professor }
  UNION
  { ?x a :Student }
}
```

SPARQL (cont.)

SPARQL Protocol and RDF Query Language

Title of courses taught by a professor and professor names

```
PREFIX : <http://example.org/voc#>
# Other prefixes omitted

SELECT ?title ?fName ?lName {

  ?teacher a :Professor .
  ?teacher :teaches ?course .
  ?teacher foaf:lastName ?lName .

  ?course :title ?title .

  OPTIONAL {
    ?teacher foaf:firstName ?fName .
  }
}
```

Algebra (goes beyond UCQ)

- Basic Graph Pattern
- UNION
- OPTIONAL
- GROUP BY
- MINUS
- FILTER NOT EXISTS

OWL 2 QL profile ($DL\text{-}Lite_{\mathcal{R}}$)

Supported constructs

- Class and property hierarchies (`rdfs:subClassOf` and (`rdfs:subPropertyOf`)
- Property domain and range (`rdfs:domain`, `rdfs:range`)
- Inverse properties (`owl:inverseOf`)
- Class disjointness (`owl:disjointWith`)
- Mandatory participation (`owl:someValuesFrom` in the superclass expression)

Not supported

- Individual identities (`owl:sameAs`)
- Cardinality constraints (functional property, etc.)
- Many other constructs

Characteristics

- Lightweight ontologies
- First-order rewritability (rewritable into a SQL query)

unibz

RDF Schema (RDFS)

Lightweight ontology language

rdfs:subClassOf $(A \sqsubseteq B)$

	:AssociateProfessor rdfs:subClassOf :Professor . <uni1/academic/1> a :AssociateProfessor .
$\Longrightarrow$	<uni1/academic/1> a :Professor .

rdfs:subPropertyOf $(P \sqsubseteq R)$

	:givesLecture rdfs:subPropertyOf :teaches . <uni2/academic/2> :givesLecture <uni2/course/1> .
$\Longrightarrow$	<uni2/academic/2> :teaches <uni2/course/1> .

rdfs:domain $(\exists P \sqsubseteq A)$

	:teaches rdfs:domain :Teacher . <uni2/academic/2> :teaches <uni2/course/1> .
$\Longrightarrow$	<uni2/academic/2> a :Teacher .

rdfs:range $(\exists P^{-} \sqsubseteq A)$

	:teaches rdfs:range :Course . <uni2/academic/2> :teaches <uni2/course/1> .
$\Longrightarrow$	<uni2/course/1> a :Course .

Other constructs of OWL 2 QL (cont.)

owl:inverseOf $(P^- \sqsubseteq R,\ R^- \sqsubseteq P)$

	`:isTaughtBy owl:inverseOf :teaches .` `<uni2/academic/2> :teaches <uni2/course/1> .`
$\Longrightarrow$	`<uni2/course/1> :isTaughtBy <uni2/academic/2> .`

owl:someValuesFrom in the superclass expression $(A \sqsubseteq \exists R.B)$

	`:PhDStudent rdfs:subClassOf`
	`[ a owl:Restriction ;`
	`owl:onProperty :isSupervisedBy ;`
	`owl:someValuesFrom :Professor ] .`
	`<uni2/person/10> a :PhDStudent .`

$\Longrightarrow$	`<uni2/person/10> a`
	`[ a owl:Restriction ;`
	`owl:onProperty :isSupervisedBy ;`
	`owl:someValuesFrom :Professor ] .`

unibz

Other constructs of OWL 2 QL (cont.)

owl:disjointWith $(A \sqsubseteq \neg B,\ B \sqsubseteq \neg A)$

	`:Student owl:disjointWith :Professor .`
	`<uni1/academic/19> a :Professor .`
	`<uni1/academic/19> a :Student .`

$\Longrightarrow$	Inconsistent RDF graph

Mapping assertions RDB-RDF

Global-As-View (GAV) mapping assertion $\varphi \rightsquigarrow \psi$

- φ: FO query
- ψ: atom

unibz

Mapping assertions RDB-RDF

Global-As-View (GAV) mapping assertion $\varphi \rightsquigarrow \psi$

- φ: FO query (over DB predicates)
- ψ: atom (over an RDF predicate)

unibz

Mapping assertions RDB-RDF

Global-As-View (GAV) mapping assertion $\varphi \rightsquigarrow \psi$

- φ: FO query (over DB predicates)
- ψ: atom (over an RDF predicate)

Class instance (`:Student`)

Source	$\boldsymbol{q}(s) \leftarrow \texttt{uni1-student}(s, f, l)$
	`SELECT s_id`
	`FROM uni1.student`

Target	$\text{Student}(\text{URI}_1(s))$ `ex:uni1/student/{s_id} a :Student .`

unibz

Mapping assertions RDB-RDF

Global-As-View (GAV) mapping assertion $\varphi \rightsquigarrow \psi$

- φ: FO query (over DB predicates)
- ψ: atom (over an RDF predicate)
- Open-World Assumption (by default)

Class instance (`:Student`)

Source	$\boldsymbol{q}(s) \leftarrow \texttt{uni1-student}(s, f, l)$
	
	`SELECT s_id`
	`FROM uni1.student`

Target	$\text{Student}(\mathrm{URI}_1(s))$ `ex:uni1/student/{s_id} a :Student .`

Mapping assertions RDB-RDF

Ontop native format (similar to the R2RML standard)

Data property (foaf:firstName)

Source (SQL)	`SELECT s_id, firstName, lastName` `FROM uni1.student`

Target (RDF)	`ex:uni1/student/{s_id}`
	`foaf:firstName "{firstName}"^^xsd:string ;`
	`foaf:lastName "{lastName}"^^xsd:string .`

Mapping assertions RDB-RDF

Ontop native format (similar to the R2RML standard)

Data property (foaf:firstName)

Source (SQL)	`SELECT s_id, firstName, lastName` `FROM uni1.student`

Target (RDF)	`ex:uni1/student/{s_id}`
	`foaf:firstName "{firstName}"^^xsd:string ;`
	`foaf:lastName "{lastName}"^^xsd:string .`

Object property (:teaches)

Source	`SELECT *` `FROM "uni1"."teaching"`
Target (RDF)	`ex:uni1/academic/{a_id} :teaches` `ex:uni1/course/{c_id} .`

unibz

Mapping assertions RDB-RDF

Ontop native format (similar to the R2RML standard)

Data property (foaf:firstName)

Source (SQL)	`SELECT s_id, firstName, lastName FROM uni1.student`
Target (RDF)	`ex:uni1/student/{s_id} foaf:firstName "{firstName}"^^xsd:string ; foaf:lastName "{lastName}"^^xsd:string .`

Object property (:teaches)

Source	`SELECT * FROM "uni1"."teaching"`
Target (RDF)	`ex:uni1/academic/{a_id} :teaches ex:uni1/course/{c_id} .`

Magic number

Source	`SELECT * FROM "uni1"."academic" WHERE "position" = 1`
Target (RDF)	`ex:uni1/academic/{a_id} a :FullProfessor .`

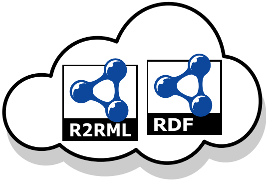
Outline

1. Querying relational databases
2. RDF and other Semantic Web standards
3. Ontology-Based Data Access
   - ABox/RDF graph saturation
   - Query rewriting
   - Ontop
4. Optique platform
5. Conclusion

Querying the OBDA system

OBDA system $\mathcal{K} = \langle \mathcal{T}, \mathcal{M}, \mathcal{D} \rangle$

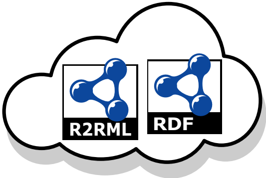
- *DL-Lite*$_{\mathcal{R}}$ TBox $\mathcal{T}$
- RDF graph $\mathcal{G}$ obtained from the mapping $\mathcal{M}$ and the data sources $\mathcal{D}$
- $\mathcal{G}$ can be viewed as the ABox



Querying the OBDA system

OBDA system $\mathcal{K} = \langle \mathcal{T}, \mathcal{M}, \mathcal{D} \rangle$

- *DL-Lite*$_\mathcal{R}$ TBox $\mathcal{T}$
- RDF graph $\mathcal{G}$ obtained from the mapping $\mathcal{M}$ and the data sources $\mathcal{D}$
- $\mathcal{G}$ can be viewed as the ABox



Query answering

- SPARQL query $\boldsymbol{q}$ over $\mathcal{K}$

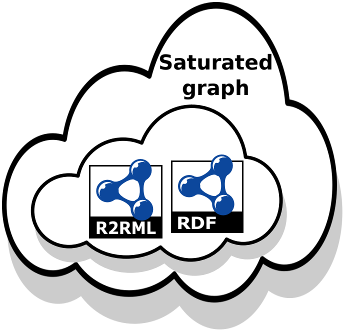
Querying the OBDA system

OBDA system $\mathcal{K} = \langle \mathcal{T}, \mathcal{M}, \mathcal{D} \rangle$

- *DL-Lite*$_\mathcal{R}$ TBox $\mathcal{T}$
- RDF graph $\mathcal{G}$ obtained from the mapping $\mathcal{M}$ and the data sources $\mathcal{D}$
- $\mathcal{G}$ can be viewed as the ABox

Query answering

- SPARQL query $\boldsymbol{q}$ over $\mathcal{K}$



Saturated RDF graph $\mathcal{G}_{\mathsf{sat}}$

- Saturation of $\mathcal{G}$ w.r.t. $\mathcal{T}$
- H-complete ABox

unibz

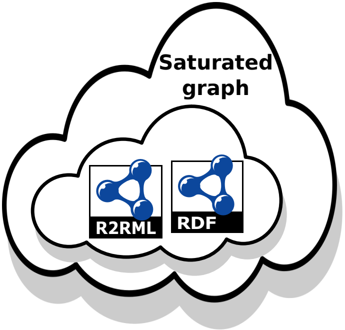
Querying the OBDA system

OBDA system $\mathcal{K} = \langle \mathcal{T}, \mathcal{M}, \mathcal{D} \rangle$

- *DL-Lite*$_{\mathcal{R}}$ TBox $\mathcal{T}$
- RDF graph $\mathcal{G}$ obtained from the mapping $\mathcal{M}$ and the data sources $\mathcal{D}$
- $\mathcal{G}$ can be viewed as the ABox

Query answering

- SPARQL query $\boldsymbol{q}$ over $\mathcal{K}$
- If there is no existential restriction $B \sqsubseteq \exists R.C$ in $\mathcal{T}$, $\boldsymbol{q}$ can be directly evaluated over $\mathcal{G}_{\mathsf{sat}}$



Saturated RDF graph $\mathcal{G}_{\mathsf{sat}}$

- Saturation of $\mathcal{G}$ w.r.t. $\mathcal{T}$
- H-complete ABox

unibz

How to handle the RDF graph $\mathcal{G}_{sat}$ in practice?

By materializing it

- Materialization of $\mathcal{G}$ (ETL) + saturation
- – Large volume
- – Maintenance
- Typical profile: OWL 2 RL

By keeping it virtual

- Query rewriting
- + No materialization required
- Saturated mapping $\mathcal{M}_{sat}$
- Typical profile: OWL 2 QL

unibz

H-complete ABox

[Rodriguez-Muro *et al.*, 2013; Kontchakov and Zakharyaschev, 2014]

ABox saturation

- H-complete ABox: contains all the inferable ABox assertions

H-complete ABox

[Rodriguez-Muro *et al.*, 2013; Kontchakov and Zakharyaschev, 2014]

ABox saturation

- H-complete ABox: contains all the inferable ABox assertions
- Let $\mathcal{K}$ be a *DL-Lite*$_{\mathcal{R}}$ knowledge base, and let $\mathcal{K}'$ be the result of saturating $\mathcal{K}$. For every ABox assertion α, we have:

$$\mathcal{K} \models \alpha \quad \text{iff} \quad \alpha \in \mathcal{K}'$$

H-complete ABox

[Rodriguez-Muro *et al.*, 2013; Kontchakov and Zakharyaschev, 2014]

ABox saturation

- H-complete ABox: contains all the inferable ABox assertions
- Let $\mathcal{K}$ be a *DL-Lite*$_\mathcal{R}$ knowledge base, and let $\mathcal{K}'$ be the result of saturating $\mathcal{K}$. For every ABox assertion α, we have:

$$\mathcal{K} \models \alpha \quad \text{iff} \quad \alpha \in \mathcal{K}'$$

Saturated mapping $\mathcal{M}_{\text{sat}}$ (also called *T-mapping*)

- Composition of the mapping $\mathcal{M}$ and the *DL-Lite*$_\mathcal{R}$ TBox $\mathcal{T}$
- $\mathcal{M}_{\text{sat}} + \mathcal{D} \rightarrow \mathcal{G}_{\text{sat}}$ (H-complete ABox)

H-complete ABox

[Rodriguez-Muro *et al.*, 2013; Kontchakov and Zakharyaschev, 2014]

ABox saturation

- H-complete ABox: contains all the inferable ABox assertions
- Let $\mathcal{K}$ be a *DL-Lite*$_\mathcal{R}$ knowledge base, and let $\mathcal{K}'$ be the result of saturating $\mathcal{K}$. For every ABox assertion α, we have:

$$\mathcal{K} \models \alpha \quad \text{iff} \quad \alpha \in \mathcal{K}'$$

Saturated mapping $\mathcal{M}_{\mathsf{sat}}$ (also called *T-mapping*)

- Composition of the mapping $\mathcal{M}$ and the *DL-Lite*$_\mathcal{R}$ TBox $\mathcal{T}$
- $\mathcal{M}_{\mathsf{sat}} + \mathcal{D} \rightarrow \mathcal{G}_{\mathsf{sat}}$ (H-complete ABox)
- Independent of the SPARQL query $\boldsymbol{q}$ (can be pre-computed)
- Can be optimized (query containment)

unibz

TBox, user-defined mapping assertions and foreign key

Student ⊔ PostDoc ⊔ AssociateProfessor ⊔ ∃teaches ⊑ Person

$$\mathsf{Student}(\mathrm{URI}_1(p)) \leftarrow \texttt{uni1-student}(p, f, l) \qquad (1)$$

$$\mathsf{PostDoc}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s = 9 \qquad (2)$$

$$\mathsf{AssociateProfessor}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s = 2 \qquad (3)$$

$$\mathsf{Person}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s) \qquad (4)$$

$$\mathsf{teaches}(\mathrm{URI}_2(a), \mathrm{URI}_3(c)) \leftarrow \texttt{uni1-teaching}(c, a) \qquad (5)$$

FK: $\exists y_1.\texttt{uni1-teaching}(y_1, x) \rightarrow \exists y_2 y_3 y_4.\texttt{uni1-academic}(x, y_2, y_3, y_4)$

TBox, user-defined mapping assertions and foreign key

Student ⊔ PostDoc ⊔ AssociateProfessor ⊔ ∃teaches ⊑ Person

$$\text{Student}(\mathrm{URI}_1(p)) \leftarrow \texttt{uni1-student}(p, f, l) \quad (1)$$

$$\text{PostDoc}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s = 9 \quad (2)$$

$$\text{AssociateProfessor}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s = 2 \quad (3)$$

$$\text{Person}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s) \quad (4)$$

$$\text{teaches}(\mathrm{URI}_2(a), \mathrm{URI}_3(c)) \leftarrow \texttt{uni1-teaching}(c, a) \quad (5)$$

FK: $\exists y_1.\texttt{uni1-teaching}(y_1, x) \rightarrow \exists y_2 y_3 y_4.\texttt{uni1-academic}(x, y_2, y_3, y_4)$

Non-optimized saturated mapping assertions for Person

$$\text{Person}(\mathrm{URI}_1(p)) \leftarrow \texttt{uni1-student}(p, f, l) \quad (6)$$

$$\text{Person}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s = 9 \quad (7)$$

$$\text{Person}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s = 2 \quad (8)$$

$$\text{Person}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s) \quad (9)$$

$$\text{Person}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-teaching}(c, a) \quad (10)$$

TBox, user-defined mapping assertions and foreign key

$\mathsf{Student} \sqcup \mathsf{PostDoc} \sqcup \mathsf{AssociateProfessor} \sqcup \exists \mathsf{teaches} \sqsubseteq \mathsf{Person}$

$$\mathsf{Student}(\mathrm{URI}_1(p)) \leftarrow \texttt{uni1-student}(p, f, l) \tag{1}$$
$$\mathsf{PostDoc}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s = 9 \tag{2}$$
$$\mathsf{AssociateProfessor}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s = 2 \tag{3}$$
$$\mathsf{Person}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s) \tag{4}$$
$$\mathsf{teaches}(\mathrm{URI}_2(a), \mathrm{URI}_3(c)) \leftarrow \texttt{uni1-teaching}(c, a) \tag{5}$$

FK: $\exists y_1.\texttt{uni1-teaching}(y_1, x) \rightarrow \exists y_2 y_3 y_4.\texttt{uni1-academic}(x, y_2, y_3, y_4)$

Non-optimized saturated mapping assertions for Person

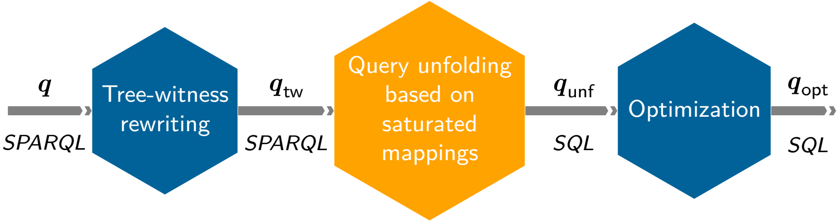
$$\mathsf{Person}(\mathrm{URI}_1(p)) \leftarrow \texttt{uni1-student}(p, f, l) \tag{6}$$
$$\mathsf{Person}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s = 9 \tag{7}$$
$$\mathsf{Person}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s = 2 \tag{8}$$
$$\mathsf{Person}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s) \tag{9}$$
$$\mathsf{Person}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-teaching}(c, a) \tag{10}$$

Mapping assertions for Person after optimization (query containment)

$$\mathsf{Person}(\mathrm{URI}_1(p)) \leftarrow \texttt{uni1-student}(p, f, l) \tag{11}$$
$$\mathsf{Person}(\mathrm{URI}_2(p)) \leftarrow \texttt{uni1-academic}(p, f, l, s) \tag{12}$$

Query rewriting

Implemented by Ontop



Step	Input	Output
1. Tree-witness rewriting	q (SPARQL) and $\mathcal{T}$	q_{tw} (SPARQL)
2. Query unfolding	q_{tw} and $\mathcal{M}_{sat}$	q_{unf} (SQL)
3. Query optimization	q_{unf}, primary and foreign keys	q_{opt} (SQL)

unibz

Saturated mappings

$$\mathsf{firstName}(\mathrm{URI}_1(p), f) \leftarrow \texttt{uni1-student}(p, f, l)$$
$$\mathsf{firstName}(\mathrm{URI}_2(a), f) \leftarrow \texttt{uni1-academic}(a, f, l, s)$$
$$\mathsf{lastName}(\mathrm{URI}_1(p), l) \leftarrow \texttt{uni1-student}(p, f, l)$$
$$\mathsf{lastName}(\mathrm{URI}_2(a), l) \leftarrow \texttt{uni1-academic}(a, f, l, s)$$
$$\mathsf{Teacher}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s \in [1, 8]$$
$$\mathsf{Teacher}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-teaching}(c, a)$$

Query

$$\boldsymbol{q}(x, y, z) \leftarrow \mathsf{Teacher}(x), \mathsf{firstName}(x, y), \mathsf{lastName}(x, z)$$
$$\boldsymbol{q}_{\mathsf{tw}} = \boldsymbol{q}$$

Saturated mappings

$$\mathsf{firstName}(\mathrm{URI}_1(p), f) \leftarrow \texttt{uni1-student}(p, f, l)$$
$$\mathsf{firstName}(\mathrm{URI}_2(a), f) \leftarrow \texttt{uni1-academic}(a, f, l, s)$$
$$\mathsf{lastName}(\mathrm{URI}_1(p), l) \leftarrow \texttt{uni1-student}(p, f, l)$$
$$\mathsf{lastName}(\mathrm{URI}_2(a), l) \leftarrow \texttt{uni1-academic}(a, f, l, s)$$
$$\mathsf{Teacher}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s \in [1, 8]$$
$$\mathsf{Teacher}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-teaching}(c, a)$$

Query

$$\boldsymbol{q}(x, y, z) \leftarrow \mathsf{Teacher}(x), \mathsf{firstName}(x, y), \mathsf{lastName}(x, z)$$
$$\boldsymbol{q}_{\mathsf{tw}} = \boldsymbol{q}$$

Query unfolding

$$\boldsymbol{q}_{\mathsf{unf}}(x, y, z) \leftarrow \boldsymbol{q}_{\mathsf{unf1}}(x), \boldsymbol{q}_{\mathsf{unf2}}(x, y), \boldsymbol{q}_{\mathsf{unf3}}(x, z)$$
$$\boldsymbol{q}_{\mathsf{unf1}}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s \in [1, 8]$$
$$\boldsymbol{q}_{\mathsf{unf1}}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-teaching}(c, a)$$
$$\boldsymbol{q}_{\mathsf{unf2}}(\mathrm{URI}_1(p), f) \leftarrow \texttt{uni1-student}(p, f, l)$$
$$\boldsymbol{q}_{\mathsf{unf2}}(\mathrm{URI}_2(a), f) \leftarrow \texttt{uni1-academic}(a, f, l, s)$$
$$\boldsymbol{q}_{\mathsf{unf3}}(\mathrm{URI}_1(p), l) \leftarrow \texttt{uni1-student}(p, f, l)$$
$$\boldsymbol{q}_{\mathsf{unf3}}(\mathrm{URI}_2(a), l) \leftarrow \texttt{uni1-academic}(a, f, l, s)$$

Query unfolding

$$
\begin{aligned}
\boldsymbol{q}_{\mathsf{unf}}(x,y,z) &\leftarrow \boldsymbol{q}_{\mathsf{unf1}}(x), \boldsymbol{q}_{\mathsf{unf2}}(x,y), \boldsymbol{q}_{\mathsf{unf3}}(x,z)\\
\boldsymbol{q}_{\mathsf{unf1}}(\mathrm{URI}_2(a)) &\leftarrow \texttt{uni1-academic}(a,f,l,s), s \in [1,8]\\
\boldsymbol{q}_{\mathsf{unf1}}(\mathrm{URI}_2(a)) &\leftarrow \texttt{uni1-teaching}(c,a)\\
\boldsymbol{q}_{\mathsf{unf2}}(\mathrm{URI}_1(p),f) &\leftarrow \texttt{uni1-student}(p,f,l)\\
\boldsymbol{q}_{\mathsf{unf2}}(\mathrm{URI}_2(a),f) &\leftarrow \texttt{uni1-academic}(a,f,l,s)\\
\boldsymbol{q}_{\mathsf{unf3}}(\mathrm{URI}_1(p),l) &\leftarrow \texttt{uni1-student}(p,f,l)\\
\boldsymbol{q}_{\mathsf{unf3}}(\mathrm{URI}_2(a),l) &\leftarrow \texttt{uni1-academic}(a,f,l,s)
\end{aligned}
$$

Query unfolding

$$\boldsymbol{q}_{\mathsf{unf}}(x, y, z) \leftarrow \boldsymbol{q}_{\mathsf{unf1}}(x), \boldsymbol{q}_{\mathsf{unf2}}(x, y), \boldsymbol{q}_{\mathsf{unf3}}(x, z)$$

$$\boldsymbol{q}_{\mathsf{unf1}}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a, f, l, s), s \in [1, 8]$$

$$\boldsymbol{q}_{\mathsf{unf1}}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-teaching}(c, a)$$

$$\boldsymbol{q}_{\mathsf{unf2}}(\mathrm{URI}_1(p), f) \leftarrow \texttt{uni1-student}(p, f, l)$$

$$\boldsymbol{q}_{\mathsf{unf2}}(\mathrm{URI}_2(a), f) \leftarrow \texttt{uni1-academic}(a, f, l, s)$$

$$\boldsymbol{q}_{\mathsf{unf3}}(\mathrm{URI}_1(p), l) \leftarrow \texttt{uni1-student}(p, f, l)$$

$$\boldsymbol{q}_{\mathsf{unf3}}(\mathrm{URI}_2(a), l) \leftarrow \texttt{uni1-academic}(a, f, l, s)$$

From join-of-unions to union-of-joins (structural optimization)

$$\begin{aligned}\boldsymbol{q}_{\mathsf{lift}}(\mathrm{URI}_2(a), y, z) \leftarrow\; & \texttt{uni1-academic}(a, f_1, l_1, s_1),\\ & \texttt{uni1-academic}(a, y, l_2, s_2),\\ & \texttt{uni1-academic}(a, f_3, z, s_3), s_1 \in [1, 8]\end{aligned}$$

$$\begin{aligned}\boldsymbol{q}_{\mathsf{lift}}(\mathrm{URI}_2(a), y, z) \leftarrow\; & \texttt{uni1-teaching}(c, a),\\ & \texttt{uni1-academic}(a, y, l_2, s_2),\\ & \texttt{uni1-academic}(a, f_3, z, s_3)\end{aligned}$$

Query unfolding

$$\boldsymbol{q}_{\mathsf{unf}}(x,y,z) \leftarrow \boldsymbol{q}_{\mathsf{unf1}}(x), \boldsymbol{q}_{\mathsf{unf2}}(x,y), \boldsymbol{q}_{\mathsf{unf3}}(x,z)$$
$$\boldsymbol{q}_{\mathsf{unf1}}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-academic}(a,f,l,s), s \in [1,8]$$
$$\boldsymbol{q}_{\mathsf{unf1}}(\mathrm{URI}_2(a)) \leftarrow \texttt{uni1-teaching}(c,a)$$
$$\boldsymbol{q}_{\mathsf{unf2}}(\mathrm{URI}_1(p),f) \leftarrow \texttt{uni1-student}(p,f,l)$$
$$\boldsymbol{q}_{\mathsf{unf2}}(\mathrm{URI}_2(a),f) \leftarrow \texttt{uni1-academic}(a,f,l,s)$$
$$\boldsymbol{q}_{\mathsf{unf3}}(\mathrm{URI}_1(p),l) \leftarrow \texttt{uni1-student}(p,f,l)$$
$$\boldsymbol{q}_{\mathsf{unf3}}(\mathrm{URI}_2(a),l) \leftarrow \texttt{uni1-academic}(a,f,l,s)$$

From join-of-unions to union-of-joins (structural optimization)

$$\boldsymbol{q}_{\mathsf{lift}}(\mathrm{URI}_2(a),y,z) \leftarrow \texttt{uni1-academic}(a,f_1,l_1,s_1), \texttt{uni1-academic}(a,y,l_2,s_2), \texttt{uni1-academic}(a,f_3,z,s_3), s_1 \in [1,8]$$
$$\boldsymbol{q}_{\mathsf{lift}}(\mathrm{URI}_2(a),y,z) \leftarrow \texttt{uni1-teaching}(c,a), \texttt{uni1-academic}(a,y,l_2,s_2), \texttt{uni1-academic}(a,f_3,z,s_3)$$

Self-join elimination (semantic optimization)

PK: $\texttt{uni1-academic}(a,b,c,d) \land \texttt{uni1-academic}(a,b',c',d') \rightarrow (b=b') \land (c=c') \land (d=d')$

$$\boldsymbol{q}_{\mathsf{opt}}(\mathrm{URI}_2(a),y,z) \leftarrow \texttt{uni1-academic}(a,y,z,s_1), s_1 \in [1,8]$$
$$\boldsymbol{q}_{\mathsf{opt}}(\mathrm{URI}_2(a),y,z) \leftarrow \texttt{uni1-teaching}(c,a), \texttt{uni1-academic}(a,y,z,s_2)$$

Ontop

http://ontop.inf.unibz.it

Ontop framework

- Started in 2010
- Open-source *(Apache 2)*
- W3C standard compliant *(SPARQL, OWL 2 QL, R2RML)*
- Supports all major relational DBs *(Oracle, DB2, Postgres, MySQL, etc.)* and some virtual DBs *(Teiid, Exareme)*

Components

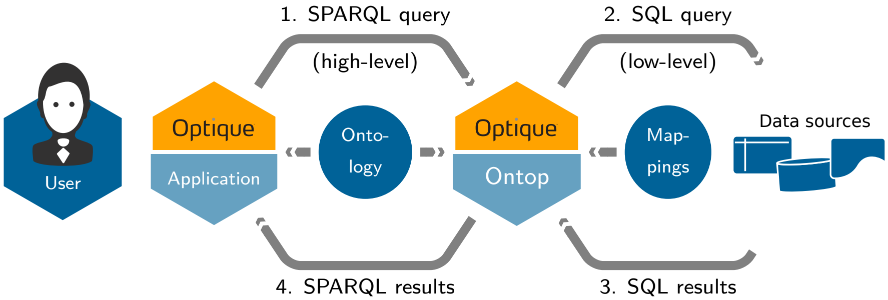
- Java APIs
- Protégé extension (GUI)
- Sesame SPARQL endpoint

Integration

- Optique platform
- Stardog 4.0 (virtual graphs)

Outline

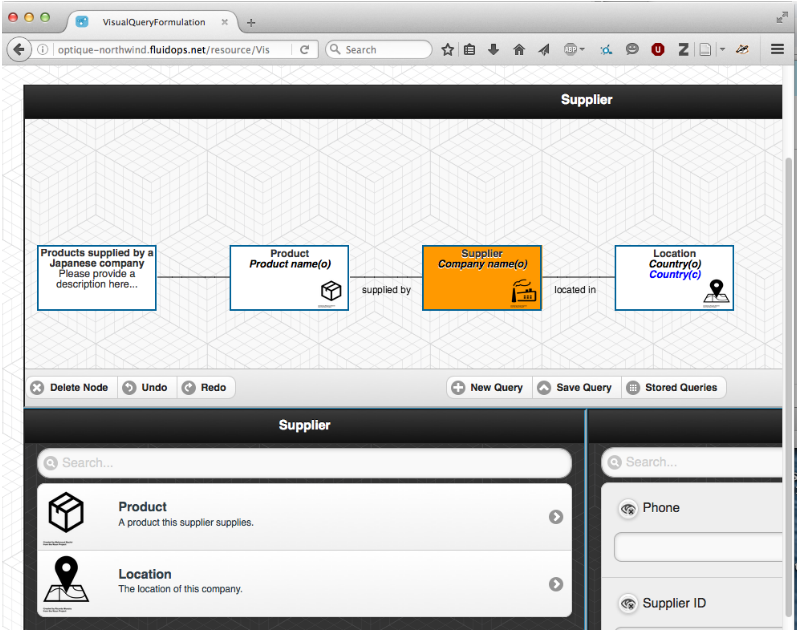
Optique platform



unibz

Visual query formulation (Optique VQS)

`http://optique-northwind.fluidops.net` **demo/demo**



Outline

1. Querying relational databases
2. RDF and other Semantic Web standards
3. Ontology-Based Data Access
4. Optique platform
5. Conclusion

Conclusion

Main message: we need high-level access to data

1. SQL queries over tables can be difficult to write manually (low-level)
2. OBDA is a powerful solution for high-level data access
3. Ontop is an open-source OBDA framework

Recent work and WIP

- Beyond OWL 2 QL
- MongoDB (non-relational)
- `owl:sameAs` partial support
- SPARQL aggregation
- Better SPARQL `OPTIONAL`
- SPARQL `MINUS`

Links

- Github : ontop/ontop
- ontop4obda@googlegroups.com
- Twitter : @ontop4obda
- `http://ontop.inf.unibz.it`
- `http://optique-project.eu`

unibz

Ontop team

- Diego Calvanese
- Guohui Xiao
- Roman Kontchakov (Birkbeck, London)
- Elena Botoeva
- Julien Corman
- Sarah Komla-Ebri
- Elem Güzel Kalayci
- Ugur Dönmez
- Davide Lanti
- Dag Hovland (Oslo)
- Mariano Rodriguez-Muro (now at IBM Research, NY)
- Martin Rezk (now at Rakuten, Tokyo)
- Me

Introductory resources

Journal paper [Calvanese *et al.*, 2016a]

Ontop: Answering SPARQL Queries over Relational Databases.
Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao.
Semantic Web Journal. 2016 http://www.semantic-web-journal.net/content/ontop-answering-sparql-queries-over-relational-databases-1

Tutorial

https://github.com/ontop/ontop-examples/tree/master/swj-2015

University example

https://github.com/ontop/ontop-examples/tree/master/university

EPNET SPARQL endpoint [Calvanese *et al.*, 2016b]

http://136.243.8.213/epnet-pleiades-edh/

References (cont.)

[Botoeva *et al.*, 2016a] Elena Botoeva, Diego Calvanese, Benjamin Cogrel, Martin Rezk, and Guohui Xiao.
OBDA beyond relational DBs: A study for MongoDB.
In *International workshop on Description Logic*, 2016.

[Botoeva *et al.*, 2016b] Elena Botoeva, Diego Calvanese, Valerio Santarelli, Domenico F. Savo, Alessandro Solimando, and Guohui Xiao.
Beyond OWL 2 QL in OBDA: Rewritings and approximations.
In *Proc. of the 30th AAAI Conf. on Artificial Intelligence (AAAI)*, 2016.

[Calvanese *et al.*, 2015] Diego Calvanese, Martin Giese, Dag Hovland, and Martin Rezk.
Ontology-based integration of cross-linked datasets.
In *Proc. of ISWC*, volume 9366 of *LNCS*, pages 199–216. Springer, 2015.

[Calvanese *et al.*, 2016a] Diego Calvanese, Benjamin Cogrel, Sarah Komla-Ebri, Roman Kontchakov, Davide Lanti, Martin Rezk, Mariano Rodriguez-Muro, and Guohui Xiao.
Ontop: Answering SPARQL queries over relational databases.
Semantic Web J., 2016.
DOI: 10.3233/SW-160217.

References (cont.)

[Calvanese *et al.*, 2016b] Diego Calvanese, Pietro Liuzzo, Alessandro Mosca, Jose Remesal, Martin Rezk, and Guillem Rull.
Ontology-based data integration in EPNet: Production and distribution of food during the Roman Empire.
Engineering Applications of Artificial Intelligence, 2016.

[Kontchakov and Zakharyaschev, 2014] Roman Kontchakov and Michael Zakharyaschev.
An introduction to description logics and query rewriting.
In *Reasoning Web. Reasoning on the Web in the Big Data Era - 10th International Summer School 2014, Athens, Greece, September 8-13, 2014. Proceedings*, volume 8714 of *LNCS*, pages 195–244. Springer, 2014.

[Rodriguez-Muro *et al.*, 2013] Mariano Rodriguez-Muro, Roman Kontchakov, and Michael Zakharyaschev.
Ontology-based data access: Ontop of databases.
In *Proc. of ISWC*, volume 8218 of *LNCS*, pages 558–573, 2013.

[Xiao *et al.*, 2014] Guohui Xiao, Martin Rezk, Mariano Rodriguez-Muro, and Diego Calvanese.
Rules and ontology based data access.
In *Proc. of RR*, volume 8741 of *LNCS*, pages 157–172. Springer, 2014.