
Ontology-Based Data Access

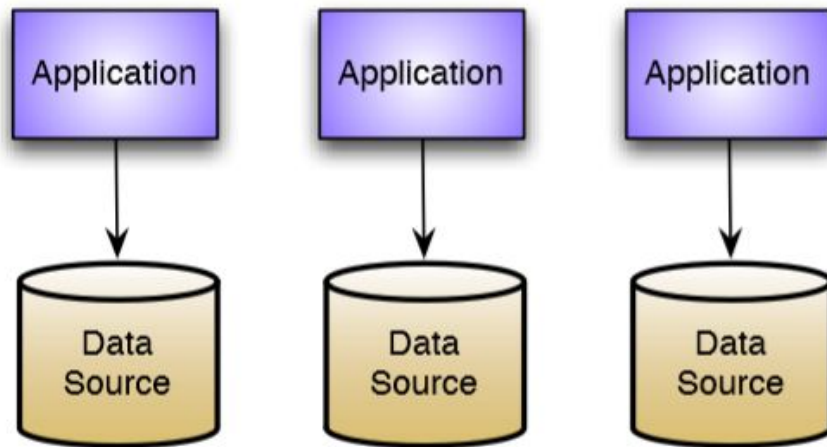
— Diogo Espinhara Oliveira —
Banco de Dados - 2017.1

Sumário

1. Motivação e Objetivo
2. *Ontology Based Data Access* (OBDA)
3. Ontologia e Lógica de Descrição
4. OBDA com Banco de Dados Relacionais
5. R2RML
6. OWL 2 QL
7. Ferramentas
8. Uso
9. Trabalhos Futuros
10. Conclusões
11. Referências

1. Motivação

- Dados distribuídos;
- Dados heterogêneos;



Fonte: [1]

1. Motivação

Caso Simples:



1. Motivação

Caso Complexo:



1. Exemplo 1: Statoil Exploration

Fatos

- 1.000 TB de dados relacionais;
- Diversos esquemas;
- Mais de 2.000 tabelas, em várias bases de dados individuais;

Acesso de Dados para Exploração

- 900 especialistas;
- Até quatro dias para elaborar novas consultas de acesso aos dados;
- Necessitando de assistência de especialistas em TI;
- 30-70% do tempo gasto na coleta de dados.



1. Objetivo



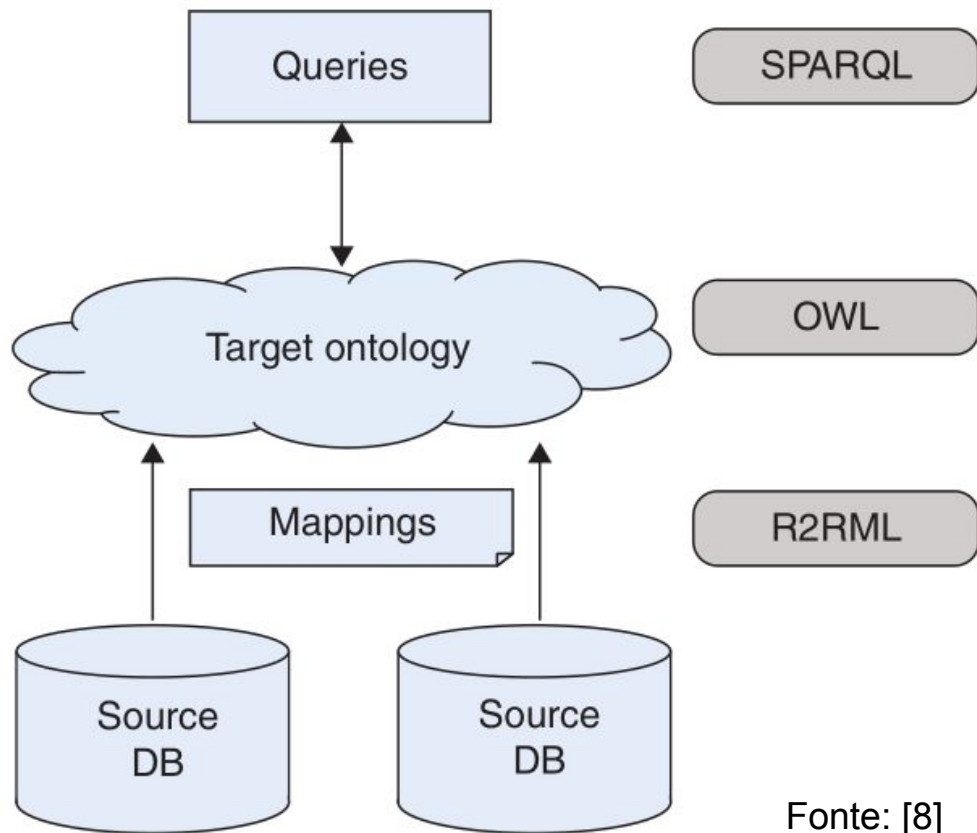
2. Ontology Based Data Access (OBDA)

- Tecnologia para fornecer acesso uniforme a dados armazenados em múltiplas e heterogêneas fontes;
- A ideia foi formalizada em 2008;
- Objetivos principais:
 - Fornecer uma visão conceitual de alto nível sobre os dados;
 - Proporcionar um vocabulário conveniente para consultas;
 - Aprimorar dados incompletos com conhecimento da ontologia;

2. OBDA: Características

- Utiliza ontologia para consultar dados, classificar e mais;
- Omite detalhes sobre os esquemas das fontes de dados;
- Mapeia consultas sobre a ontologia para as fontes de dados (por exemplo SQL);
- Não migra os dados para a ontologia, deixa-os nas fontes de dados;
- Usa serviços de inferência no processo de mapeamento das consultas;

2. OBDA: Modelo de Arquitetura



3. Ontologia

- Completamente independente de como os dados estão armazenados;
- Dividida em duas partes;
- TBox descreve os axiomas terminológicos;
 - Armazenados em uma ontologia;
- ABox descreve os axiomas sobre os dados do mundo;
 - Armazenados em um banco de dados relacional;
- Construída através de uma sub-linguagem de Lógica de Descrição (DL);

3. Lógica de Descrição (DL)

- Família de linguagens formais de representação do conhecimento;
- Importante formalismo lógico para ontologias e Web Semântica;
- Existem várias sub-linguagens:
 - OBDA com banco de dados;
 - OBDA com datalog engines (motores de dados);
 - OBDA com DL mais expressiva;
- Foco da apresentação é o tipo OBDA com Banco de Dados;

3. Ontologia: Exemplo de DL

Natural Language	Logic
<i>Ontology (TBox)</i>	
A woman is a person	$Woman \sqsubseteq Person$
A mother is a woman that has a child	$Mother \equiv Woman \sqcap \exists hasChild$
<i>Data (ABox)</i>	
Maria is a woman	$Woman(Maria)$
Maria has the child Jesus	$hasChild(Maria, Jesus)$

3. Ontologias: Exemplo de Inferência

$Mother \equiv Woman \sqcap \exists hasChild$ ← Ontologia **T**

Mãe é equivalente a uma mulher que possui um filho

$Woman(Maria)$

$hasChild(Maria, Jesus)$ ← Dados **A**

Maria é mulher e tem um filho Jesus

$Mother(Maria)$

Inferência: Maria é mãe

4. OBDA com Banco de Dados

- A ontologia T (TBox) é acessada pelo usuário;
- A estrutura dos dados A (ABox) não é conhecida pelo usuário;
- Usuário formula uma consulta q no vocabulário da ontologia T ;
- A consulta q é '**reescrita**' em uma nova consulta q' sobre o vocabulário dos dados A ;
- q é chamada de consulta conjuntiva;
- q' é chamada de consulta de primeira ordem;
 - Representa as consultas expressíveis em SQL;

4. Exemplo

woman	
w_id	w_name
1	Maria
2	Sara

man	
m_id	m_name
3	Jesus
4	Alan Turing

has_child	
w_id	m_id
1	3
2	4

Fonte: [4]

4. Exemplo (cont.)

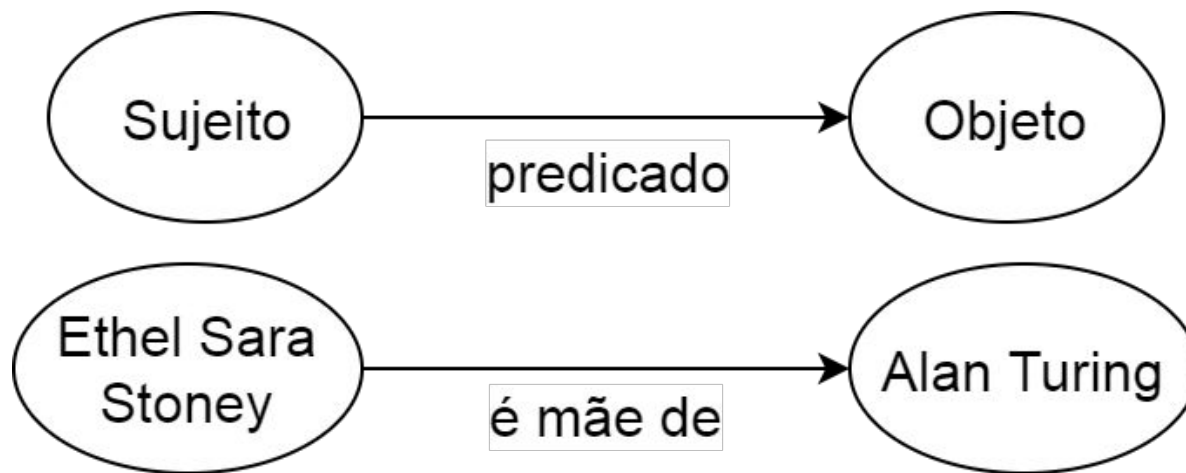
```
SELECT DISTINCT  
wo.w_name, ma.m_name  
FROM woman wo,  
      man ma,  
      has_child hc  
WHERE hc.w_id = wo.w_id  
AND   hc.m_id = ma.m_id
```



w_name	m_name
Maria	Jesus
Sara	Alan Turing

5. R2RML

- Linguagem para mapear bancos relacionais para conjuntos RDFS;
- Permite a visualização de dados relacionais em modelo de dados RDF;
- O conhecimento é representado através de triplas RDF:



Fonte: [7]

5. Exemplo



TopManager		
empCode	salary	project
10	2500	102

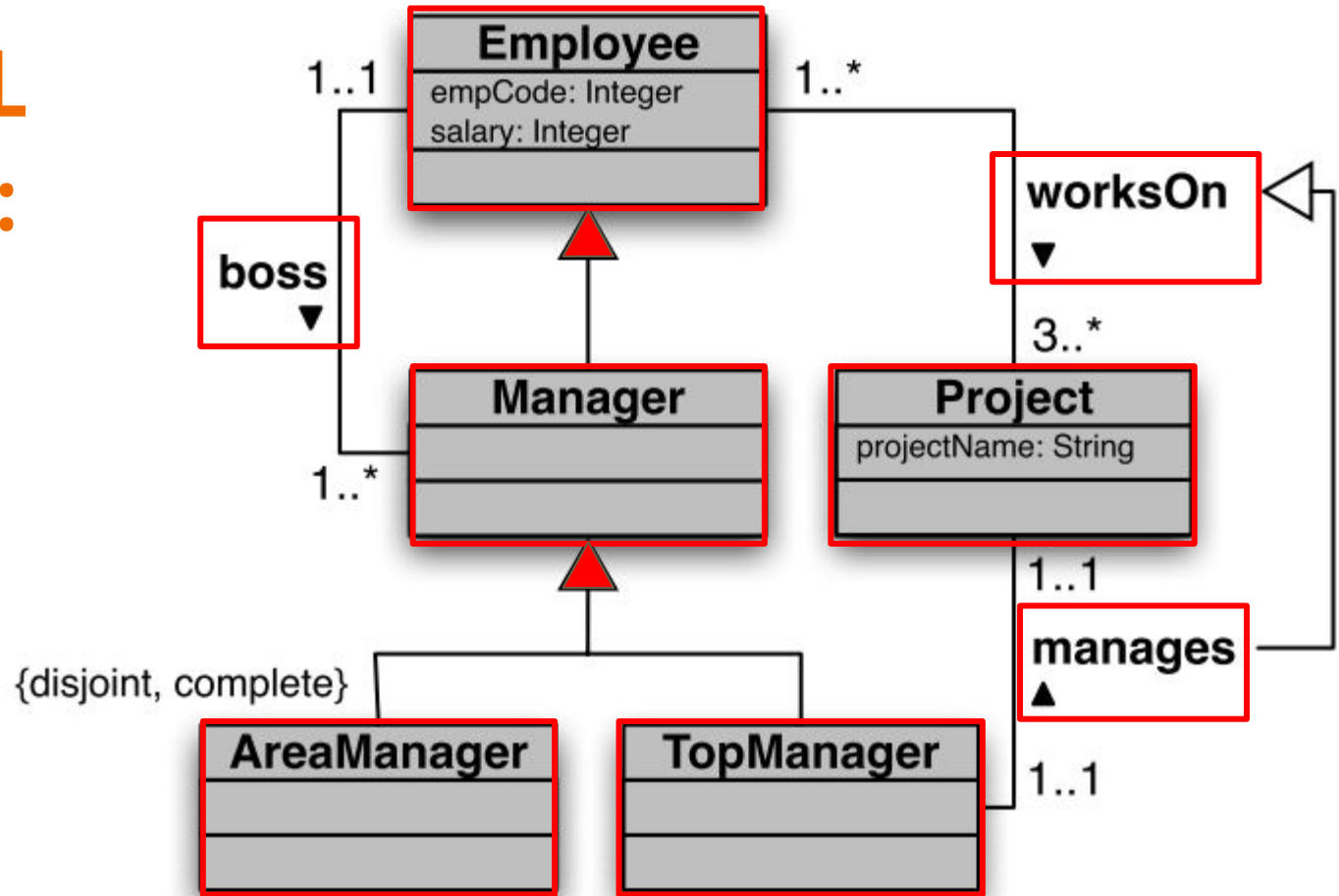
Project	
projCode	projName
102	OBDA

```
<http://ex.com/project/102> rdf:type :Project.  
<http://ex.com/project/102> :projName "OBDA".  
  
<http://ex.com/topmanager/10> rdf:type :TopManager.  
<http://ex.com/topmanager/10> :salary 2500.  
<http://ex.com/topmanager/10>  
  :manages <http://ex.com/project/102>.
```

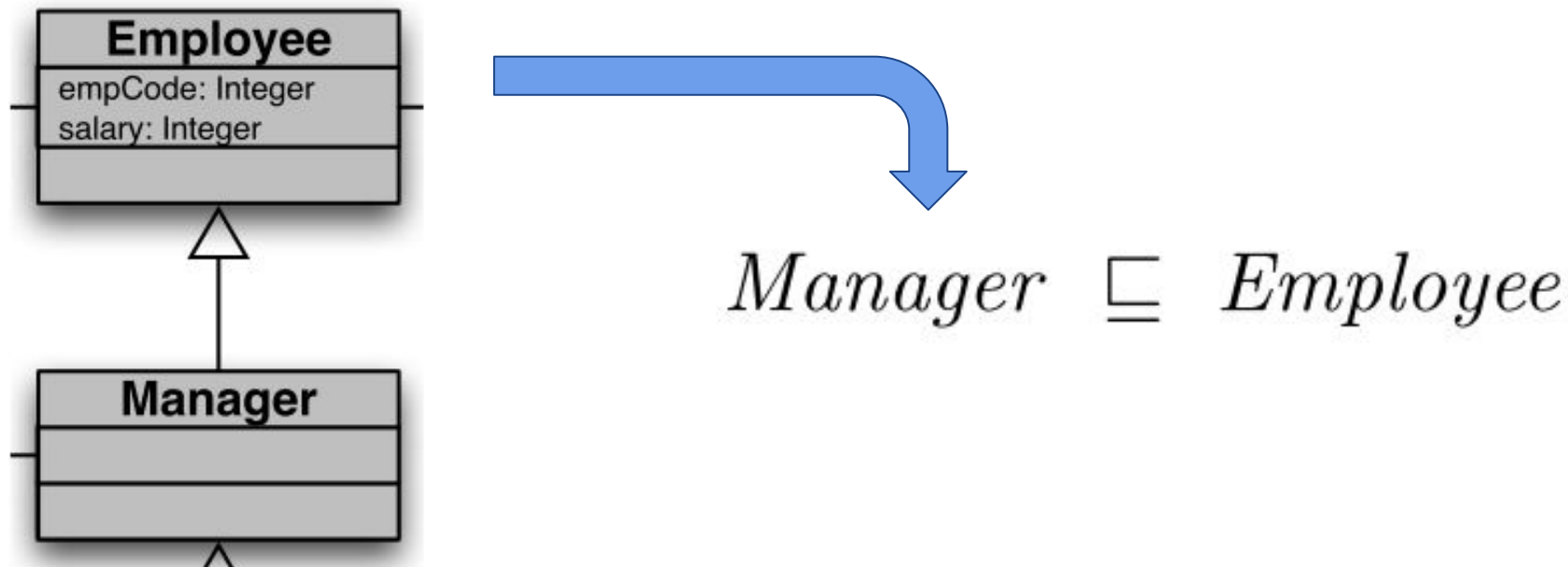
6. OWL 2 QL

- Perfil da linguagem padrão OWL 2 projetada para OBDA com banco de dados relacionais;
- Baseado em uma junção entre OWL 2 DL e RDFS;
- Linguagem possui:
 - Classes;
 - Indivíduos;
 - Axiomas terminológicos;
 - Herança;
 - Disjunção;
 - entre outras coisas....

6. OWL 2 QL Exemplo:



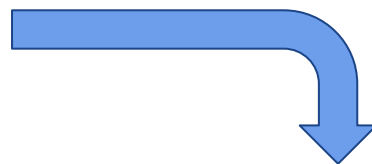
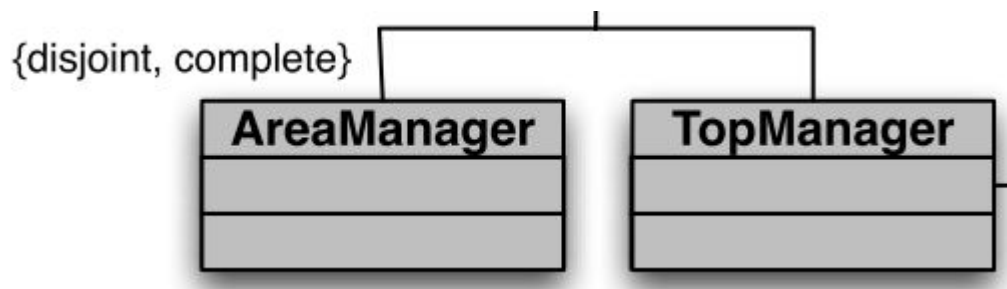
6. OWL 2 QL: Exemplo (cont.)



* *TopManager* e *AreaManager* seguem da mesma forma

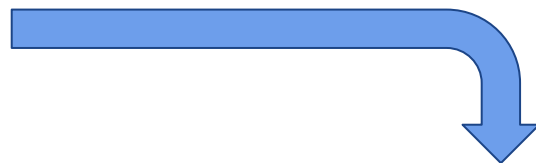
Fonte: [3]

6. OWL 2 QL: Exemplo (cont.)



$$AreaManager \sqcap TopManager \sqsubseteq \perp$$

6. OWL 2 QL: Exemplo (cont.)

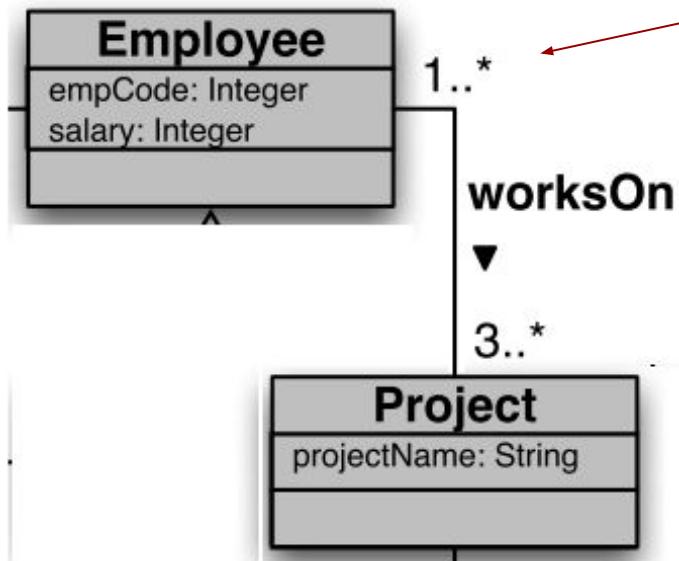


$$\begin{aligned} Employee &\sqsubseteq \exists salary \\ \exists salary^- &\sqsubseteq Integer. \end{aligned}$$

* *empCode* segue da mesma forma

6. OWL 2 QL: Exemplo (cont.)

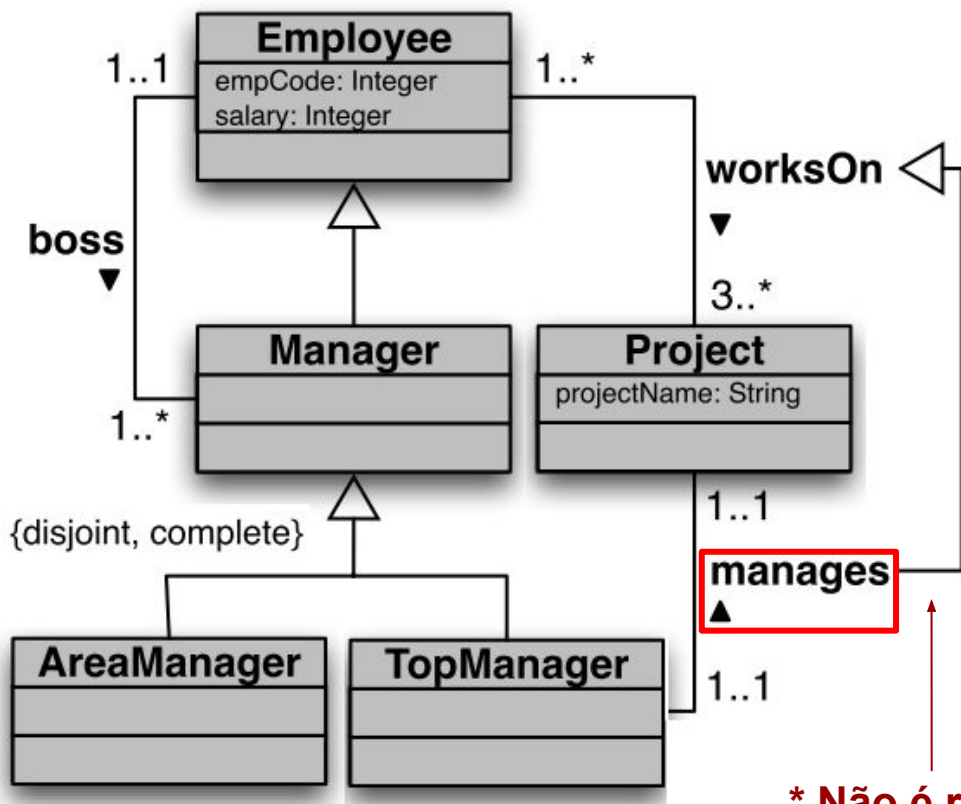
Cardinalidade não é possível
nessa linguagem



$\exists worksOn \sqsubseteq Employee$
 $\exists worksOn^- \sqsubseteq Project$

* a relação '*boss*' segue da mesma forma

6. OWL 2 QL: Exemplo (cont.)



$\exists \text{manages} \sqsubseteq \text{TopManager}$

$\exists \text{manages}^- \sqsubseteq \text{Project}$

$\text{TopManager} \sqsubseteq \exists \text{manages}$

$\text{Project} \sqsubseteq \exists \text{manages}^-$

$\text{manages} \sqsubseteq \text{worksOn}$

* Não é relacionamento sobre relacionamento

6. OWL 2 QL: Implementação - Ontologia

```
<Declaration>
```

```
  <Class IRI="#Employee"/>
```

```
</Declaration>
```

```
<Declaration>
```

```
  <DataProperty IRI="#salary"/>
```

```
</Declaration>
```

```
<Declaration>
```

```
  <ObjectProperty IRI="#manages"/>
```

```
</Declaration>
```

6. OWL 2 QL: Ontologia (cont.)

$\exists salary^- \sqsubseteq Integer.$



```
<DataPropertyRange>  
  <DataProperty IRI="#salary"/>  
  <Datatype abbreviatedIRI="xsd:integer"/>  
</DataPropertyRange>
```

6. OWL 2 QL: Ontologia (cont.)

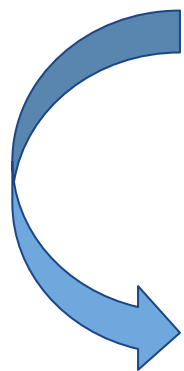
$Employee \sqsubseteq \exists salary$



```
<SubClassOf>
  <Class IRI="#Employee"/>
  <DataSomeValuesFrom>
    <DataProperty IRI="#salary"/>
    <Datatype abbreviatedIRI="xsd:integer"/>
  </DataSomeValuesFrom>
</SubClassOf>
```

* Implementação feita pelo Protégé [5]

6. OWL 2 QL: Ontologia (cont.)



$Manager \sqsubseteq Employee$

```
<SubClassOf>  
  <Class IRI="#Manager"/>  
  <Class IRI="#Employee"/>  
</SubClassOf>
```

6. OWL 2 QL: Ontologia (cont.)

$AreaManager \sqcap TopManager \sqsubseteq \perp$



```
<DisjointClasses>  
  <Class IRI="#AreaManager"/>  
  <Class IRI="#TopManager"/>  
</DisjointClasses>
```

* Implementação feita pelo Protégé [5]

6. OWL 2 QL: Ontologia (cont.)

$\exists worksOn \sqsubseteq Employee$



```
<ObjectPropertyDomain>  
  <ObjectProperty IRI="#worksOn"/>  
  <Class IRI="#Employee"/>  
</ObjectPropertyDomain>
```

* Implementação feita pelo Protégé [5]

6. OWL 2 QL: Ontologia (cont.)



$\exists worksOn^- \sqsubseteq Project.$

```
<ObjectPropertyRange>  
  <ObjectProperty IRI="#worksOn"/>  
  <Class IRI="#Project"/>  
</ObjectPropertyRange>
```

6. OWL 2 QL: Ontologia (cont.)

manages $\sqsubseteq$ *worksOn*.



```
<SubObjectPropertyOf>  
  <ObjectProperty IRI="#manages"/>  
  <ObjectProperty IRI="#worksOn"/>  
</SubObjectPropertyOf>
```

6. OWL 2 QL: Ontologia (cont.)

$TopManager \sqsubseteq \exists manages$



```
<SubClassOf>
  <Class IRI="#TopManager"/>
  <ObjectSomeValuesFrom>
    <ObjectProperty IRI="#manages"/>
    <Class abbreviatedIRI="owl:Thing"/>
  </ObjectSomeValuesFrom>
</SubClassOf>
```

* Implementação feita pelo Protégé [5]

7. Ferramentas

1. O PTIQUE: Ontology-Based Data Access Platform
2. MASTRO
3. **Ontop**

7. Ferramenta de OBDA: Ontop

- Sistema OBDA de código-aberto;
- Permite consultar fontes de dados relacionais através de uma ontologia;
- Mapeia os dados relacionais para a visão da ontologia;
- Pode ser integrado com as principais bases de dados relacionais;
 - Oracle, MySQL, SQL Server, etc
- Segue as recomendações pertinentes da W3C;
 - OWL 2 QL, R2RML e SPARQL;

8. Uso: Ontop

```
[PrefixDeclaration]  
:      http://ex.com/#
```

```
[SourceDeclaration]  
sourceUri      EmployeeDB  
connectionUrl   jdbc:h2:tcp://localhost/database  
username        sa  
password        *****  
driverClass     org.h2.Driver
```

8. Uso: Ontop

```
mappingId TopManager
target :db1/topmanager/{empCode} a :TopManager
      ; :salary {salary} .
source SELECT empCode, salary FROM "TopManager"
```

```
mappingId manages
target :db1/topmanager/{empCode}
      :manages :db1/project/{project} .
source SELECT empCode, project FROM "TopManager"
```

8. Uso: Ontop

```
mappingId    Manager
target       :db1/manager/{empCode} a :Manager .
source       SELECT empCode FROM "TopManager"
              UNION ALL
              SELECT empCode FROM "AreaManager"
```

```
mappingId    boss
target       :db1/employee/{bossCode}
              :boss :db1/manager/{empCode} .
source       SELECT empCode, bossCode FROM "TopManager"
              UNION ALL
              SELECT empCode, bossCode FROM "AreaManager"
```

8.

ontop query editor:

Query Editor

```
PREFIX : <http://ex.com/#>

SELECT ?topManager ?salary ?projName
WHERE {
    ?topManager a :TopManager
               ; :salary ?salary
               ; :project ?project .
    ?project :projName ?projName .
}
```

Query

Show: 100 ☐ All ☐ Short IRI  Attach Prefixes  Execute  Save Changes

Results

topManager	salary	projName
<http://ex.com/topmanager/10>	2500	OBDA

Hint: --

Resposta

 Export to CSV...

☐ Synchronising

9. Trabalhos Futuros - Ontop

1. Melhorar o desempenho investigando otimizações;
2. Suportar fragmentos maiores do SPARQL
 - Por exemplo, consultas de agregação e negação;
3. Melhorar a GUI e ampliar os utilitários para tornar o *Ontop* ainda mais fácil de usar;
4. Ir além dos bancos de dados relacionais e apoiar outros tipos de fontes de dados;
 - Por exemplo, grafos e documentos;

10. Conclusões sobre OBDA

- OBDA oferece vantagens no âmbito de dados distribuídos e heterogêneos;
- Adequado para enriquecer dados incompletos;
- Faz uso de benefícios da Inteligência Artificial Simbólica:
 - Ontologias
 - Serviços de Inferência;
- Desvantagens:
 - Necessário conhecimento básico sobre DL;
 - Necessário entendimento sobre DL ou SPARQL para consultar;

11. Referências

- [1] Strandhaug, Marius. (2014). An R2RML Mapping Management API in Java: Making an API Independent of its Dependencies. **Institutt for informatikk**.
- [2] Calvanese, Diego. (2012). Ontology-Based Data Access: From Theory to Practice. **KRDB Research Centre for Knowledge and Data**.
- [3] Kontchakov, R., Rodríguez-Muro, M., & Zakharyashev, M. (2013). Ontology-based data access with databases: A short course. In **Lecture Notes in Computer Science** (Vol. 8067 LNAI, pp. 194–229);
- [4] Cogrel, Benjamin. (2016). **Ontology-Based Data Access with Ontop**.
<[link para download](#)>

11. Referências

- [5] **Protégé**. (2011). The Protégé Ontology Editor. Financial Executive, 19(4), 39–39.
- [6] Calvanese, D., Cogrel, B., & Komla-Ebri, S. (2017). Ontop: Answering SPARQL queries over relational databases. **Semantic**. <[link para download](#)>
- [7] Consortium, W. W. W. (2012). **R2RML**: RDB to RDF mapping language. <[link para download](#)>
- [8] Capsenta, J. F. S., & Miranker, D. P. (2017). **A Pay-As-You-Go Methodology for Ontology-Based Data Access**.

11. Referências

- [9] Calvanese, D., Mosca, A., Remesal, J., Rezk, M., & Rull, G. (2016). A “historical case” of Ontology-Based Data Access. 2015 **Digital Heritage International Congress**, Digital Heritage 2015, 291–298.
- [10] Jim, E. K. E. (n.d.). **O PTIQUE** : Ontology-Based Data Access Platform, 7–10.
- [11] Motik, B., Grau, B., Horrocks, I., & Wu, Z. (2009). Owl 2 web ontology language: Profiles. **W3C**, (April), 1–53;
- [12] Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., ... Savo, D. F. (2011). The MASTRO system for ontology-based data access. **Semantic Web**, 2, 43–53.