Methods and Tools for Developing Ontology-Based Data Management Solutions
Concepts for Ontology-Based Data Management

Domenico Lembo, Valerio Santarelli, Domenico Fabio Savo

SEMANTICS 2018
Vienna, Austria, September 11, 2018
Fragment of a relational table in a Bank Information system:

<table>
<thead>
<tr>
<th>CUC</th>
<th>TS_START</th>
<th>TS_END</th>
<th>ID_GRUP</th>
<th>FLAG_CP</th>
<th>FLAG_CF</th>
<th>FATTURATO</th>
<th>FLAG_FATT</th>
</tr>
</thead>
<tbody>
<tr>
<td>124589</td>
<td>30-lug-2004</td>
<td>1-gen-9999</td>
<td>92736</td>
<td>S</td>
<td>N</td>
<td>195000,00</td>
<td>N</td>
</tr>
<tr>
<td>140904</td>
<td>15-mag-2001</td>
<td>15-giu-2005</td>
<td>35060</td>
<td>N</td>
<td>N</td>
<td>230600,00</td>
<td>N</td>
</tr>
<tr>
<td>124589</td>
<td>5-mag-2001</td>
<td>30-lug-2004</td>
<td>92736</td>
<td>N</td>
<td>S</td>
<td>195000,00</td>
<td>S</td>
</tr>
<tr>
<td>-452901</td>
<td>13-mag-2001</td>
<td>27-lug-2004</td>
<td>92770</td>
<td>S</td>
<td>N</td>
<td>392000,00</td>
<td>N</td>
</tr>
<tr>
<td>129008</td>
<td>10-mag-2001</td>
<td>1-gen-9999</td>
<td>62010</td>
<td>N</td>
<td>S</td>
<td>247000,00</td>
<td>S</td>
</tr>
<tr>
<td>-472900</td>
<td>10-mag-2001</td>
<td>1-gen-9999</td>
<td>62010</td>
<td>S</td>
<td>N</td>
<td>00</td>
<td>N</td>
</tr>
<tr>
<td>130976</td>
<td>7-mag-2001</td>
<td>9-lug-2003</td>
<td>75680</td>
<td>S</td>
<td>N</td>
<td>00</td>
<td>N</td>
</tr>
<tr>
<td>CUC</td>
<td>TS_START</td>
<td>TS_END</td>
<td>ID_GRUP</td>
<td>FLAG_CP</td>
<td>FLAG_CF</td>
<td>FATTURATO</td>
<td>FLAG_FATT</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>-------------</td>
<td>-----------</td>
</tr>
<tr>
<td>124589</td>
<td>30-lug-2004</td>
<td>1-gen-9999</td>
<td>92736</td>
<td>S</td>
<td>N</td>
<td>195000,00</td>
<td>N</td>
</tr>
<tr>
<td>140904</td>
<td>15-mag-2001</td>
<td>15-giu-2005</td>
<td>35060</td>
<td>N</td>
<td>N</td>
<td>230600,00</td>
<td>N</td>
</tr>
<tr>
<td>124589</td>
<td>5-mag-2001</td>
<td>30-lug-2004</td>
<td>92736</td>
<td>N</td>
<td>S</td>
<td>195000,00</td>
<td>S</td>
</tr>
<tr>
<td>-452901</td>
<td>13-mag-2001</td>
<td>27-lug-2004</td>
<td>92770</td>
<td>S</td>
<td>N</td>
<td>392000,00</td>
<td>N</td>
</tr>
<tr>
<td>129008</td>
<td>10-mag-2001</td>
<td>1-gen-9999</td>
<td>62010</td>
<td>N</td>
<td>S</td>
<td>247000,00</td>
<td>S</td>
</tr>
<tr>
<td>-472900</td>
<td>10-mag-2001</td>
<td>1-gen-9999</td>
<td>62010</td>
<td>S</td>
<td>N</td>
<td>0 00</td>
<td>N</td>
</tr>
<tr>
<td>130976</td>
<td>7-mag-2001</td>
<td>9-lug-2003</td>
<td>75680</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Negative value denotes a holding**
<table>
<thead>
<tr>
<th>CUC</th>
<th>TS_START</th>
<th>TS_END</th>
<th>ID_GRUP</th>
<th>FLAG_CP</th>
<th>FLAG_CF</th>
<th>FATTURATO</th>
<th>FLAG_FATT</th>
</tr>
</thead>
<tbody>
<tr>
<td>124589</td>
<td>30-lug-2004</td>
<td>1-gen-9999</td>
<td>92736</td>
<td>S</td>
<td>N</td>
<td>195000,00</td>
<td>N</td>
</tr>
<tr>
<td>140904</td>
<td>15-mag-2001</td>
<td>15-giu-2005</td>
<td>35060</td>
<td>N</td>
<td>N</td>
<td>230600,00</td>
<td>N</td>
</tr>
<tr>
<td>124589</td>
<td>5-mag-2001</td>
<td>30-lug-2004</td>
<td>92736</td>
<td>N</td>
<td>S</td>
<td>195000,00</td>
<td>S</td>
</tr>
<tr>
<td>-452901</td>
<td>13-mag-2001</td>
<td>27-lug-2004</td>
<td>92770</td>
<td>S</td>
<td>N</td>
<td>392000,00</td>
<td>N</td>
</tr>
<tr>
<td>129008</td>
<td>10-mag-2001</td>
<td>1-gen-9999</td>
<td>62010</td>
<td>N</td>
<td>S</td>
<td>247000,00</td>
<td>S</td>
</tr>
<tr>
<td>-472900</td>
<td>10-mag-2001</td>
<td>1-gen-9999</td>
<td>62010</td>
<td>S</td>
<td>N</td>
<td>0 00</td>
<td>N</td>
</tr>
<tr>
<td>130976</td>
<td>7-mag-2001</td>
<td>9-lug-2003</td>
<td>75680</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUC</td>
<td>TS_START</td>
<td>TS_END</td>
<td>ID_GRUP</td>
<td>FLAG_CP</td>
<td>FLAG_CF</td>
<td>FATTURATO</td>
<td>FLAG_FATT</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>-----------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>124589</td>
<td>30-lug-2004</td>
<td>1-gen-9999</td>
<td>92736</td>
<td>S</td>
<td>N</td>
<td>195000,00</td>
<td>N</td>
</tr>
<tr>
<td>140904</td>
<td>15-mag-2001</td>
<td>15-giu-2005</td>
<td>35060</td>
<td>N</td>
<td>N</td>
<td>230600,00</td>
<td>N</td>
</tr>
<tr>
<td>124589</td>
<td>5-mag-2001</td>
<td>30-lug-2004</td>
<td>92736</td>
<td>N</td>
<td>S</td>
<td>195000,00</td>
<td>S</td>
</tr>
<tr>
<td>-452901</td>
<td>13-mag-2001</td>
<td>27-lug-2004</td>
<td>92770</td>
<td>S</td>
<td>N</td>
<td>392000,00</td>
<td>N</td>
</tr>
<tr>
<td>129008</td>
<td>10-mag-2001</td>
<td>1-gen-9999</td>
<td>62010</td>
<td>N</td>
<td>S</td>
<td>247000,00</td>
<td>S</td>
</tr>
<tr>
<td>-472900</td>
<td>10-mag-2001</td>
<td>1-gen-9999</td>
<td>62010</td>
<td>S</td>
<td>N</td>
<td>00</td>
<td>N</td>
</tr>
<tr>
<td>130976</td>
<td>7-mag-2001</td>
<td>9-lug-2003</td>
<td>75680</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*N means that the FATTURATO field is not valid*
The example shows that the meaning of data in a table is not always clear. Understanding such meaning is crucial if one wants to correctly manage the data in the table and extract information out of it.

To make matters worse...typically, information systems in the real world use different heterogeneous data sources, both internal and external to the organization.

Organizations spend a significant amount of time and money trying to govern the resources (data, meta-data, services, processes, etc.) in their information systems.
The example shows that the meaning of data in a table is not always clear. Understanding such meaning is crucial if one wants to correctly manage the data in the table and extract information out of it.

To make matters worse...typically, information systems in the real world use different heterogeneous data sources, both internal and external to the organization.

Organizations spend a significant amount of time and money trying to govern the resources (data, meta-data, services, processes, etc.) in their information systems.
Managing data through the lens of an ontology

Manage data by adopting principles and techniques studied in Knowledge Representation.

- Provide a conceptual, high level representation of the domain of interest in terms of an ontology.
- Data may reside where they are (no need to move data).
- Map the ontology to the data sources.
- Specify all information requests to the data in terms of the ontology.
- Use inference services to automatically translate the requests into queries to the data sources.

Ontology-based Data Management (OBDM)
Managing data through the lens of an ontology

Manage data by adopting principles and techniques studied in Knowledge Representation.

- Provide a conceptual, high level representation of the domain of interest in terms of an ontology.
- Data may reside where they are (no need to move data).
- Map the ontology to the data sources.
- Specify all information requests to the data in terms of the ontology.
- Use inference services to automatically translate the requests into queries to the data sources.

Ontology-based Data Management (OBDM)
Ontologies among the key technology of the Next Decade

According to Gartner, AI technologies will be the most disruptive class of technologies in digital business in the next decade, and Enterprise Ontology Management will be a mainstream technology in 5 to 10 years.\(^1\)

\(^1\)http://www.gartner.com/newsroom/id/3784363
What is an ontology?

**Ontologies** are logical specifications describing the things of a domain of knowledge and the relationships between them.

They are typically specified in languages that allow abstraction away from data structures and implementation strategies.

Commonly used languages are:

- Description Logics
- OWL 2 (W3C ontology standard language)
**Description Logics** (DLs) are a family of logic languages used for representing the knowledge of a domain.

The domain of interest is described in terms of **objects** (individuals), **concepts** (sets of objects), and **roles** (binary relations between objects).

A DL ontology is a pair $\langle T, A \rangle$, where $T$ is the Terminological Box ($T$Box), containing assertions that describe the general aspects of the domain, while $A$ is the Assertional Box ($A$Box) providing extensional information on the domain (i.e., data).

**TBox** $T$:

- $\exists \text{playsIn} \sqsubseteq \text{Actor}$
- $\exists \text{playsIn}^- \sqsubseteq \text{Movie}$
- $\text{married} \equiv \text{married}^-$
- $\exists \text{directs} \sqsubseteq \text{Director}$
- $\text{Movie} \sqsubseteq \exists \text{directedBy}$
- $\text{directs} \equiv \text{directedBy}^-$

**ABox** $A$:

- $\text{Oscar}$
- $\text{Tom Cruise}$
- $\text{Nicole Kidman}$
- $\text{Stanley Kubrick}$
- $\text{Steven Spielberg}$
- $\text{Minority Report}$
- $\text{Eyes Wide Shut}$
- $\text{The Others}$
- $\text{Director}$
- $\text{playedIn}$
- $\text{earned}$
- $\text{directedBy}$
- $\text{married}$
Ontology-based data management: architecture

Ontology provides a global vocabulary and is used as a conceptual view

Mappings semantically link sources and ontology

Data Sources external and heterogeneous
Designing a proper OBDM framework

Which is the “right” query language?

Which is the “right” ontology language?

Which is the “right” mapping language?

Trade-off expressive power vs. efficiency of query answering!

We access big data, so efficiency w.r.t. the data is crucial.
Designing a proper OBDM framework

Which is the “right” query language?

Which is the “right” ontology language?

Which is the “right” mapping language?

Trade-off expressive power vs. efficiency of query answering!

We access big data, so efficiency w.r.t. the data is crucial.
Abstracting from the mapping

Here, we abstract away from the mappings, and consider **ontology-based query answering** (as opposed to data access).

One can see the **ABox** as a database whose predicates are the same of the ontology and containing both objects and values.
Ontology-based query answering

Which is the “right” query language?

Which is the “right” ontology language?

We are in a setting of incomplete information.

Certain answers

Answering a query $q(x)$ amounts to finding the certain answers $\text{cert}(q, \mathcal{O})$, i.e., those answers that hold in all models of the ontology $\mathcal{O}$. 
Ontology-based query answering

Which is the “right” query language?

Which is the “right” ontology language?

We are in a setting of incomplete information.

Certain answers

Answering a query $q(x)$ amounts to finding the certain answers $\text{cert}(q, \mathcal{O})$, i.e., those answers that hold in all models of the ontology $\mathcal{O}$. 
Which is the right query language?

Two borderline cases for the language to use for querying ontologies:

- Use the ontology language as query language.
  - Ontology languages are tailored for capturing intensional relationships.
  - They are quite poor as query languages.
- Full SQL (or equivalently, first-order logic).
  - **Problem:** in the presence of incomplete information, query answering becomes **undecidable** (FOL validity).

**Conjunctive queries**

A good trade-off is to use **conjunctive queries** (CQs) or unions of CQs (UCQs), corresponding to SQL/relational algebra (union) **select-project-join queries**.
Ontology-based query answering – Example

**TBox** $\mathcal{T}$:
- $\exists \text{playsIn} \sqsubseteq \text{Actor}$
- $\exists \text{playsIn} \sqsubseteq \text{Movie}$
- $\text{married} \equiv \text{married}^-$
- $\exists \text{directs} \sqsubseteq \text{Director}$
- $\text{Movie} \sqsubseteq \exists \text{directedBy}$
- $\text{directs} \equiv \text{directedBy}^-$

**ABox** $\mathcal{A}$:

Leveraging the TBox axioms, we infer additional ABox facts.

**Query**: Return all pairs of married actors who played in movies whose directors have earned the same award.

$q(x_1, x_2) \leftrightarrow \exists m_1, m_2, d_1, d_2, a. \ \text{Actor}(x_1) \land \text{playsIn}(x_1, m_1) \land \text{Movie}(m_1) \land \text{directedBy}(m_1, d_1) \land \text{Director}(d_1) \land \text{earned}(d_1, a) \land \text{Actor}(x_2) \land \text{playsIn}(x_2, m_2) \land \text{Movie}(m_2) \land \text{directedBy}(m_2, d_2) \land \text{Director}(d_2) \land \text{earned}(d_2, a) \land \text{married}(x_1, x_2) \land \text{Award}(a)$

We have that the pair (‘Tom Cruise’, ‘Nicole Kidman’) is in $\text{cert}(q, (\mathcal{T}, \mathcal{A}))$. 
Ontology-based query answering – Example

TBox $\mathcal{T}$:  
- $\exists \text{playsIn} \sqsubseteq \text{Actor}$  
- $\exists \text{playsIn}^\neg \sqsubseteq \text{Movie}$  
- $\text{married} \equiv \text{married}^\neg$  
- $\exists \text{directs} \sqsubseteq \text{Director}$  
- Movie $\sqsubseteq \exists \text{directedBy}$  
- directedBy $\equiv \text{directedBy}^\neg$

ABox $\mathcal{A}$:

Query: Return all pairs of married actors who played in movies whose directors have earned the same award.

$q(x_1, x_2) \leftarrow \exists m_1, m_2, d_1, d_2, a$.  
- $\text{Actor}(x_1) \land \text{playsIn}(x_1, m_1) \land \text{Movie}(m_1) \land \text{directedBy}(m_1, d_1) \land \text{Director}(d_1) \land \text{earned}(d_1, a) \land \text{Actor}(x_2) \land \text{playsIn}(x_2, m_2) \land \text{Movie}(m_2) \land \text{directedBy}(m_2, d_2) \land \text{Director}(d_2) \land \text{earned}(d_2, a) \land \text{married}(x_1, x_2) \land \text{Award}(a)$

We have that the pair (‘Tom Cruise’, ‘Nicole Kidman’) is in $\text{cert}(q, \langle \mathcal{T}, \mathcal{A} \rangle)$. 

Leveraging the TBox axioms, we infer additional ABox facts.
Ontology-based query answering – Example

TBox $T$: 

\[ \exists \text{playsIn} \sqsubseteq \text{Actor} \]
\[ \exists \text{playsIn}^- \sqsubseteq \text{Movie} \]
\[ \text{married} \equiv \text{married}^- \]
\[ \exists \text{directs} \sqsubseteq \text{Director} \]
\[ \text{Movie} \sqsubseteq \exists \text{directedBy} \]

ABox $A$: 

Leveraging the TBox axioms, we infer additional ABox facts.

Query: Return all pairs of married actors who played in movies whose directors have earned the same award.

$$q(x_1, x_2) \leftarrow \exists m_1, m_2, d_1, d_2, a. \quad \text{Actor}(x_1) \land \text{playsIn}(x_1, m_1) \land \text{Movie}(m_1) \land \text{directedBy}(m_1, d_1) \land \text{Director}(d_1) \land \text{earned}(d_1, a) \land \text{Actor}(x_2) \land \text{playsIn}(x_2, m_2) \land \text{Movie}(m_2) \land \text{directedBy}(m_2, d_2) \land \text{Director}(d_2) \land \text{earned}(d_2, a) \land \text{married}(x_1, x_2) \land \text{Award}(a)$$

We have that the pair ('Tom Cruise', 'Nicole Kidman') is in $\text{cert}(q, (T, A))$. 
Ontology-based query answering – Example

TBox \( T \): \( \exists \text{playsIn} \sqsubseteq \text{Actor} \)
\( \exists \text{playsIn} \sqsubseteq \text{Movie} \)
\( \text{married} \equiv \text{married}^- \)
\( \exists \text{directs} \sqsubseteq \text{Director} \)
Movie \( \sqsubseteq \exists \text{directedBy} \)
directs \( \equiv \text{directedBy}^- \)

ABox \( A \):

Leveraging the TBox axioms, we infer additional ABox facts.

Query: Return all pairs of married actors who played in movies whose directors have earned the same award.

\( q(x_1, x_2) \leftarrow \exists m_1, m_2, d_1, d_2, a. \) Actor\( (x_1) \land \text{playsIn}(x_1, m_1) \land \text{Movie}(m_1) \land \text{directedBy}(m_1, d_1) \land \text{Director}(d_1) \land \text{earned}(d_1, a) \land \text{Actor}(x_2) \land \text{playsIn}(x_2, m_2) \land \text{Movie}(m_2) \land \text{directedBy}(m_2, d_2) \land \text{Director}(d_2) \land \text{earned}(d_2, a) \land \text{married}(x_1, x_2) \land \text{Award}(a) \)

We have that the pair (‘Tom Cruise’, 'Nicole Kidman’) is in \( \text{cert}(q, (T, A)) \).
Ontology-based query answering – Example

TBox $\mathcal{T}$:

\[ \exists \text{playsIn} \sqsubseteq \text{Actor} \]
\[ \exists \text{playsIn} \sqsubseteq \text{Movie} \]
\[ \text{married} \equiv \text{married}^- \]
\[ \exists \text{directs} \sqsubseteq \text{Director} \]
\[ \text{Movie} \sqsubseteq \exists \text{directedBy} \]

ABox $\mathcal{A}$:

Leveraging the TBox axioms, we infer additional ABox facts.

Query: Return all pairs of married actors who played in movies whose directors have earned the same award.

\[ q(x_1, x_2) \leftarrow \exists m_1, m_2, d_1, d_2, a. \quad \text{Actor}(x_1) \land \text{playsIn}(x_1, m_1) \land \text{Movie}(m_1) \land \text{directedBy}(m_1, d_1) \land \text{Director}(d_1) \land \text{earned}(d_1, a) \land \text{Actor}(x_2) \land \text{playsIn}(x_2, m_2) \land \text{Movie}(m_2) \land \text{directedBy}(m_2, d_2) \land \text{Director}(d_2) \land \text{earned}(d_2, a) \land \text{married}(x_1, x_2) \land \text{Award}(a) \]

We have that the pair ('Tom Cruise', 'Nicole Kidman') is in $\text{cert}(q, \langle \mathcal{T}, \mathcal{A} \rangle)$.
Ontology-based query answering – Example

### TBox $\mathcal{T}$:
- $\exists \text{playsIn} \sqsubseteq \text{Actor}$
- $\exists \text{playsIn} \sqsubseteq \text{Movie}$
- $\text{married} \equiv \text{married}^-$
- $\exists \text{directs} \sqsubseteq \text{Director}$
- $\text{Movie} \sqsubseteq \exists \text{directedBy}$

### ABox $\mathcal{A}$:

```
ABox $\mathcal{A}$:

Query: Return all pairs of married actors who played in movies whose directors have earned the same award.

$q(x_1, x_2) \leftarrow \exists m_1, m_2, d_1, d_2, a$. $\text{Actor}(x_1) \land \text{playsIn}(x_1, m_1) \land \text{Movie}(m_1) \land \text{directedBy}(m_1, d_1) \land \text{Director}(d_1) \land \text{earned}(d_1, a) \land \text{Actor}(x_2) \land \text{playsIn}(x_2, m_2) \land \text{Movie}(m_2) \land \text{directedBy}(m_2, d_2) \land \text{Director}(d_2) \land \text{earned}(d_2, a) \land \text{married}(x_1, x_2) \land \text{Award}(a)$

We have that the pair (‘Tom Cruise’, ’Nicole Kidman’) is in $\text{cert}(q, \langle \mathcal{T}, \mathcal{A} \rangle)$. ```
Complexity of conjunctive query answering in DLs

Studied extensively for (unions of) CQs and various ontology languages:

<table>
<thead>
<tr>
<th></th>
<th>Combined complexity</th>
<th>Data complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain databases</td>
<td>NP-complete</td>
<td>$AC^0$</td>
</tr>
<tr>
<td>OWL 2</td>
<td>$\geq 2\text{ExpTime}$</td>
<td>coNP-hard</td>
</tr>
</tbody>
</table>

Challenges

- Can we find interesting DLs for which the query answering problem can be solved efficiently (in $AC^0$ wrt data complexity)?
- If yes, can we delegate query answering over ontologies to a relational engine?
Complexity of conjunctive query answering in DLs

Studied extensively for (unions of) CQs and various ontology languages:

<table>
<thead>
<tr>
<th></th>
<th>Combined complexity</th>
<th>Data complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain databases</td>
<td>NP-complete</td>
<td>$AC^0$</td>
</tr>
<tr>
<td>OWL 2</td>
<td>$\geq 2\text{ExpTime}$</td>
<td>coNP-hard</td>
</tr>
</tbody>
</table>

Challenges

- Can we find interesting DLs for which the query answering problem can be solved efficiently (in $AC^0$ wrt data complexity)?
- If yes, can we delegate query answering over ontologies to a relational engine?
To be able to deal with data efficiently, we need to separate the contribution of the data \( A \) from the contribution of \( q \) and \( T \).

\[ \leadsto \text{Query answering by } \text{query rewriting}. \]
Query answering by query rewriting

Query answering is done in two phases:

1. **Perfect rewriting**: produce from \( q \) and \( T \) a new query \( r_{q,T} \) (called perfect rewriting of \( q \) w.r.t. \( T \)).

2. **Query evaluation**: evaluate \( r_{q,T} \) over the \( A \) seen as a complete database (and without considering \( T \)).

\[
\sim \quad \text{cert}(q, (\mathcal{T}, A))
\]

The expressiveness of the ontology language affects the query language into which we are able to express the rewriting \( r_{q,T} \).
FOL-rewritability

We are especially interested in ontology languages for which the rewriting of CQs w.r.t. to $\mathcal{T}$ can be expressed in first-order logic (FOL-rewritability).

Because:

- the rewriting can be expressed in SQL, and so...
- query evaluation can be delegated to a relational DBMS.
We are especially interested in ontology languages for which the rewriting of CQs w.r.t. to $\mathcal{T}$ can be expressed in first-order logic (FOL-rewritability).

Because:

- the rewriting can be expressed in SQL, and so...
- query evaluation can be delegated to a relational DBMS.

Question

- Can we find interesting DLs for which query answering is FOL-rewritable?
The *DL-Lite* family

*DL-Lite* is a family of DLs optimized according to the *tradeoff* between *expressive power* and *complexity* of query answering, with emphasis on *data*.

- The same complexity as relational databases.
- In fact, query answering is FOL-rewritable and hence can be delegated to a relational DB engine.

Nevertheless they have the right expressive power: capture the essential features of conceptual modeling formalisms (except completeness in hierarchies).

*DL-Lite* provides robust foundations for **Ontology-Based Data Management**.

The *DL-Lite* family is at the basis of the *OWL 2 QL* profile of the W3C standard Web Ontology Language OWL 2.
Capturing basic ontology constructs in *DL-Lite*

<table>
<thead>
<tr>
<th>Modeling construct</th>
<th><em>DL-Lite</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>ISA between classes</td>
<td>Student ⊑ Person</td>
</tr>
<tr>
<td>... and or relations</td>
<td>isMatherOf ⊑ isParentOf</td>
</tr>
<tr>
<td>Disjointness between classes</td>
<td>Student ⊑ ¬Professor</td>
</tr>
<tr>
<td>... and or relations</td>
<td>isMatherOf ⊑ ¬isFatherOf</td>
</tr>
<tr>
<td>Domain of properties</td>
<td>∃livesIn ⊑ Person</td>
</tr>
<tr>
<td>Range of properties</td>
<td>∃livesIn ⊑ City</td>
</tr>
<tr>
<td>Mandatory participation <em>(min card = 1)</em></td>
<td>Person ⊑ ∃livesIn</td>
</tr>
<tr>
<td></td>
<td>City ⊑ ∃livesIn⁻</td>
</tr>
<tr>
<td>Functionality of relations <em>(max card = 1)</em></td>
<td>(funct livesIn)</td>
</tr>
<tr>
<td></td>
<td>(funct isFatherOf⁻)</td>
</tr>
</tbody>
</table>

**Note:** *DL-Lite* distinguishes between abstract objects and data values as well *(we can represent concept attributes)* (ignored here).
End Part I
(*) thanks to Freepik.com for some of the icons used in the presentation. (icons created by Sapann-Design - Freepik.com).