

# TDDD43 Advanced Data Models and Databases

RDF, SPARQL and Linked Data

Huanyu Li  
[huanyu.li@liu.se](mailto:huanyu.li@liu.se)

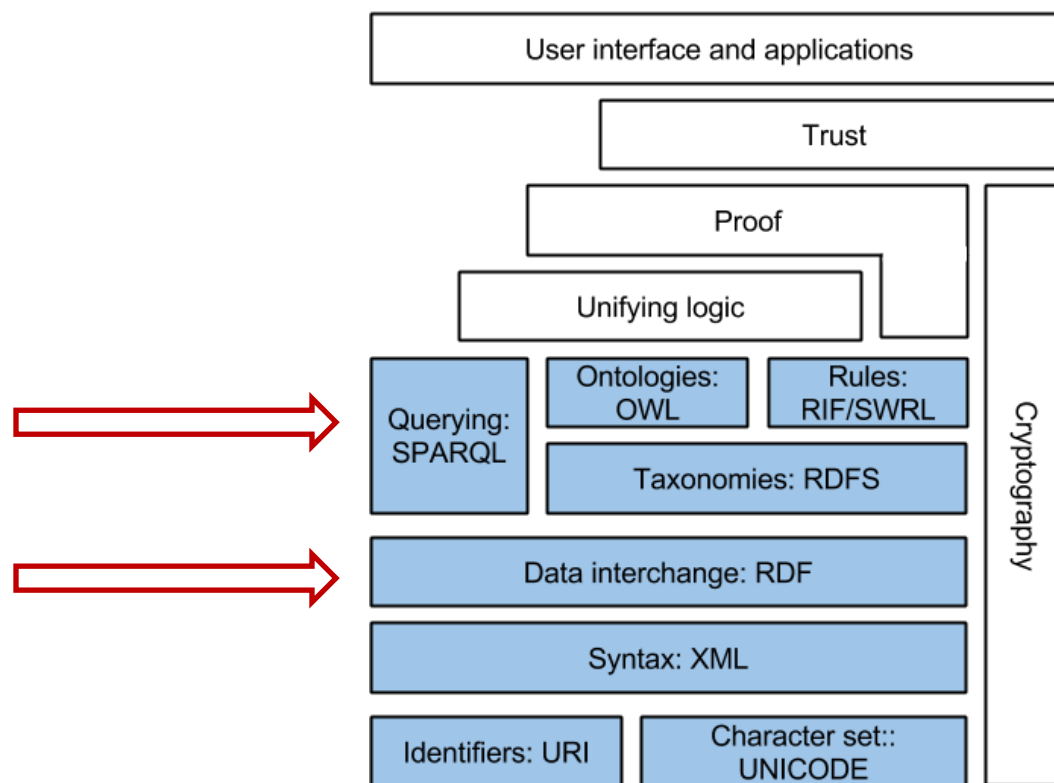
# Recap

## ➤ Semantic Web

... to put machine-understandable data on the Web...

... data can be shared and processed by automated tools as well as by people...

- W3C recommendations
  - XML, DTD, XSD
  - XPath, XQuery
- JSON and GraphQL



# Outline

- RDF (Resource Description Framework)
- SPARQL (SPARQL Protocol And RDF Query Language)
- Linked Data

# RDF (*R*esource *D*escription *F*ramework)

- In Semantic Web, a resource can be basically anything
  - e.g., persons, places, “ the temperature of today”, “all the cars in Linköping”
- To describe a resource
  - Attributes
  - Relationships
- The framework contains
  - A data model
  - Languages and syntaxes
- To represent graph data and support data exchange



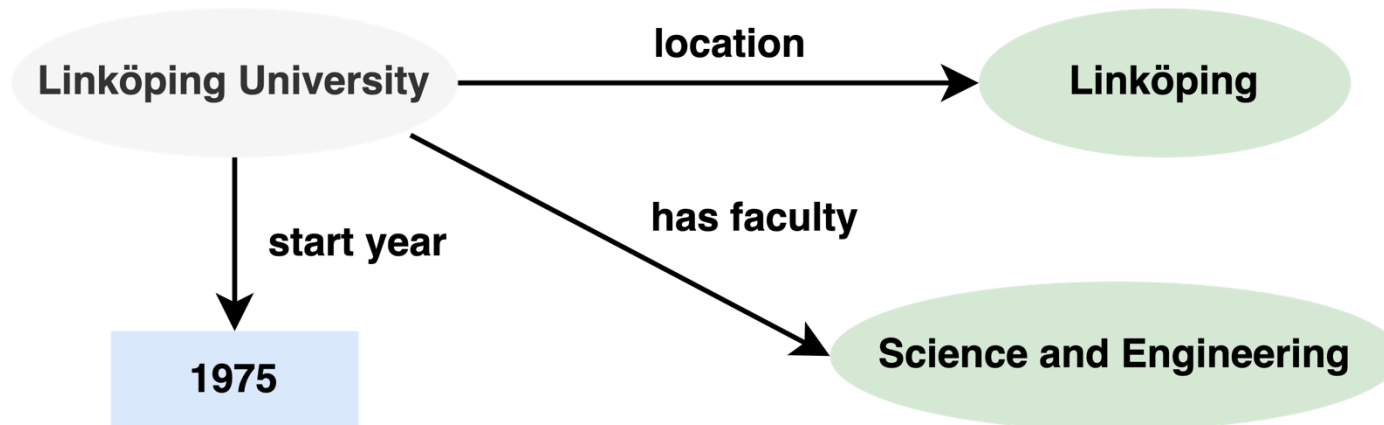
# RDF

- Data is represented as a set of **triples**
  - A triple: (subject, predicate, object)
- Subject: resources
- Predicate: properties
- Object: literals or resources
- Examples:
  - (Linköping University, start year, 1975)
  - (Linköping University, location, Linköping)
  - (Linköping University, location, Norrköping)
  - (Linköping University, has faculty, Science and Engineering)

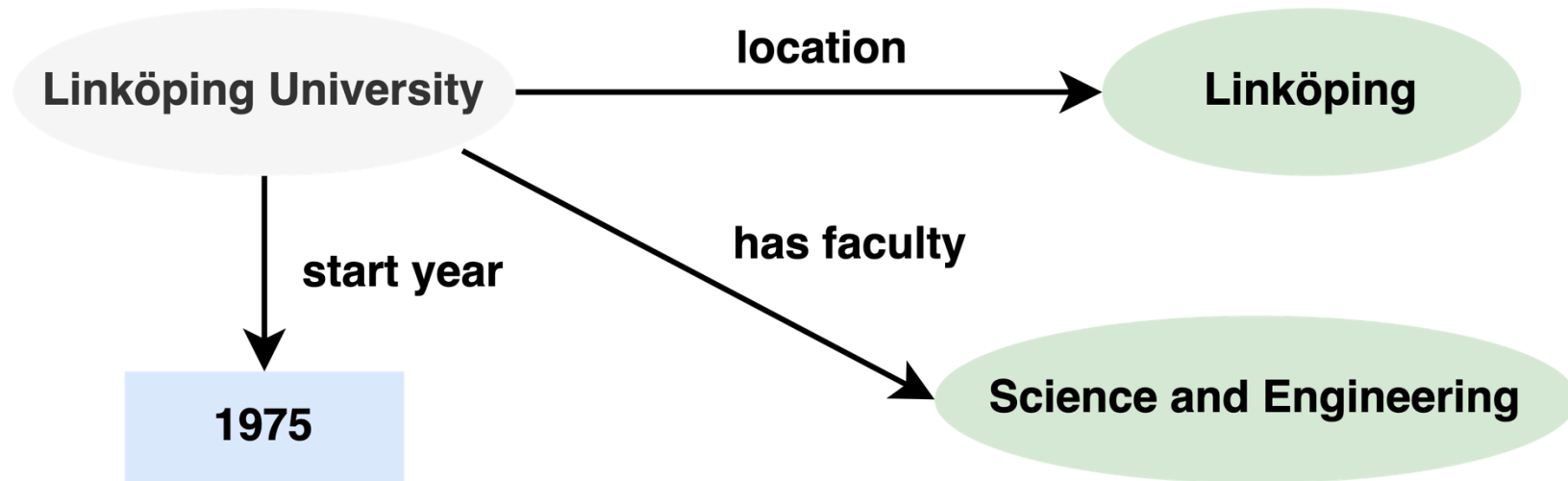
# View Triples as a directed graph

- Subjects and Objects as vertices/nodes
- Each triple as an edge from its subject to its object
- Each edge labeled by a predicate

(Linköping University, start year, 1975)  
(Linköping University, location, Linköping)  
(Linköping University, has faculty, Science and Engineering)



# Uniform Resource Identifier (URI)



Other universities may also have faculties named “Science and Engineering”...  
Therefore, how do we determine whether two nodes are really the same or not?

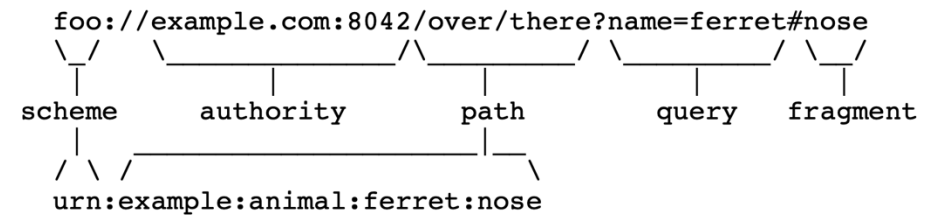
# Uniform Resource Identifier (URI)

- URIs extend the concept of URLs (Uniform Resource Locator)
  - Globally unique identifier for resources
  - URL of a Web document usually used as its URI
  - However, URIs identify more than web documents
    - RDF document about me: <http://huanyuli.se/foaf.ttl>
    - HTML document about me: <http://huanyuli.se/index.html>
    - To identify anything in RDF data
      - Concept: <http://huanyuli.se/teaching.ttl#Course>
      - Instance: [http://huanyuli.se/teaching.ttl#Huanyu\\_Li](http://huanyuli.se/teaching.ttl#Huanyu_Li)
      - Relationship: <http://huanyuli.se/university.ttl#hasFaculty>



# URI Syntax

- Syntax [1]:
  - [Schema]://[authority][Path]?[query]#[fragment]



- Examples [2]:

The following examples illustrate URI that are in common use.

```
ftp://ftp.is.co.za/rfc/rfc1808.txt
```

-- ftp scheme for File Transfer Protocol services

```
gopher://spinaltap.micro.umn.edu/00/Weather/California/Los%20Angeles
```

-- gopher scheme for Gopher and Gopher+ Protocol services

```
http://www.math.uio.no/faq/compression-faq/part1.html
```

-- http scheme for Hypertext Transfer Protocol services

```
mailto:mduerst@ifi.unizh.ch
```

-- mailto scheme for electronic mail addresses

```
news:comp.infosystems.www.servers.unix
```

-- news scheme for USENET news groups and articles

```
telnet://melvyl.ucop.edu/
```

-- telnet scheme for interactive services via the TELNET Protocol

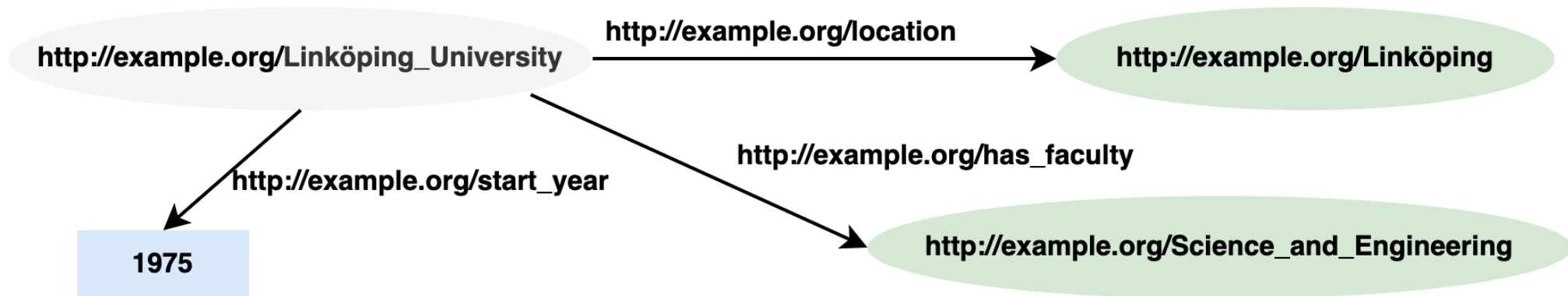
- Concept: <http://huanyuli.se/teaching.ttl#Course>
- Instance: [http://huanyuli.se/teaching.ttl#Huanyu\\_Li](http://huanyuli.se/teaching.ttl#Huanyu_Li)
- Relationship: <http://huanyuli.se/university.ttl#hasFaculty>

# URI

([http://example.org/Linköping\\_University](http://example.org/Linköping_University), [http://example.org/start\\_year](http://example.org/start_year), 1975)

([http://example.org/Linköping\\_University](http://example.org/Linköping_University), <http://example.org/location>, <http://example.org/Linköping>)

([http://example.org/Linköping\\_University](http://example.org/Linköping_University), [http://example.org/has\\_faculty](http://example.org/has_faculty), [http://example.org/Science\\_and\\_Engineering](http://example.org/Science_and_Engineering))



# Compact URI (CURIE)

- A syntax for expressing Compact URIs
- To easily express models based on RDF
- Abbreviated Notation for URIs
  - Prefix name (references a prefix of the URI)
  - Colon character (":")
  - Reference part
- URI as concatenating the **prefix** and the **reference part** (namespace and **identifier**)
- Example:
  - ex for prefix <http://example.org/>

([http://example.org/Linköping\\_University](http://example.org/Linköping_University), [http://example.org/start\\_year](http://example.org/start_year), 1975)

([http://example.org/Linköping\\_University](http://example.org/Linköping_University), <http://example.org/location>, <http://example.org/Linköping>)

([http://example.org/Linköping\\_University](http://example.org/Linköping_University), [http://example.org/has\\_faculty](http://example.org/has_faculty), [http://example.org/Science\\_and\\_Engineering](http://example.org/Science_and_Engineering))

([ex:Linköping\\_University](http://example.org/Linköping_University), [ex:start\\_year](http://example.org/start_year), 1975)

([ex:Linköping\\_University](http://example.org/Linköping_University), [ex:location](http://example.org/location), [ex:Linköping](http://example.org/Linköping))

([ex:Linköping\\_University](http://example.org/Linköping_University), [ex:has\\_faculty](http://example.org/has_faculty), [ex:Science\\_and\\_Engineering](http://example.org/Science_and_Engineering))

# Example with CURIEs

- dbpedia for prefix <http://dbpedia.org/resource/>
- p for prefix <http://example.org/property/>
- ex for prefix <http://example.org/>

([ex:Linköping\\_University](#), [p:start\\_year](#), 1975)

([ex:Linköping\\_University](#), [p:location](#), [dbpedia:Linköping](#))

# Literals

- In the object position of RDF triples
- Represented by strings and interpreted by datatypes
  - Datatype identified by a URI
  - We usually use the XML schema datatypes (e.g., xsd:integer, xsd:string)
  - If no datatype provided, then interpreted as xsd:string
- Untyped literals may have language tags (e.g., @en, @sv)

([ex:Linköping\\_University](#), [p:start\\_year](#), 1975)

([ex:Linköping\\_University](#), [p:start\\_year](#), "1975"^^xsd:integer)

([ex:Linköping\\_University](#), [p:name](#), "Linköping University"@en)

([ex:Linköping\\_University](#), [p:name](#), "Linköpings Universitet"@sv)

# RDF Syntax or Serialization formats

- Different ways to represent RDF data in text
  - N-Triples
  - Turtle
  - RDF/XML
  - JSON-LD
  - RDFa

# N-Triples

- The simplest form
- As raw RDF triples
  - Using fully unabbreviated URIs
  - Each URI written between angle brackets <URI>
  - A triple (three resources) written in order, followed by a period (.)

```
<http://example.org/tddd43/Linköping_University> <http://dbpedia.org/property/has_faculty>  
<http://example.org/tddd43/liu/Science_and_Engineering> .
```

# Turtle

- Simple, and human-readable notation
- @prefix directive binds a prefix to a namespace URI
- A whitespace character between elements in a triple
- A period (.) after a triple

```
@prefix p: <http://dbpedia.org/property/> .
```

```
@prefix ex: <http://example.org/tddd43/> .
```

```
@prefix ex-liu: <http://example.org/tddd43/liu/> .
```

```
@prefix dbpedia: <http://dbpedia.org/resource/> .
```

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
ex:Linköping_University p:has_faculty ex-liu:Science_and_Engineering .
```

```
ex:Linköping_University p:location dbpedia:Linköping .
```

```
ex:Linköping_University p:location dbpedia:Norrköping .
```

```
ex:Linköping_University p:name "Linköping University"@en .
```

```
ex:Linköping_University p:start_year "1975"^^xsd:integer .
```



# Turtle provides some syntactic sugar

- Property lists separated by a semicolon (;) character
- Object lists separated by a comma (,) character

```
@prefix p: <http://dbpedia.org/property/> .  
@prefix ex: <http://example.org/tddd43/> .  
@prefix ex-liu: <http://example.org/tddd43/liu/> .  
@prefix dbpedia: <http://dbpedia.org/resource/> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

```
ex:Linköping_University p:has_faculty ex-liu:Science_and_Engineering ;  
    p:location dbpedia:Linköping, dbpedia:Norrköping ;  
    p:name "Linköping University"@en ;  
    p:start_year "1975"^^xsd:integer .
```

# Turtle provides some syntactic sugar

- Shortcuts for number literals

```
ex:Linköping_University p:has_faculty ex-liu:Science_and_Engineering ;  
  p:location dbpedia:Linköping, dbpedia:Norrköping ;  
  p:name "Linköping University"@en ;  
  p:start_year "1975"^^xsd:integer .
```

Equivalent to

```
ex:Linköping_University p:has_faculty ex-liu:Science_and_Engineering ;  
  p:location dbpedia:Linköping, dbpedia:Norrköping ;  
  p:name "Linköping University"@en ;  
  p:start_year 1975.
```

# RDF/XML

- Historically, W3C recommended the XML serialization of RDF, RDF/XML
  - Web infrastructures are accustomed to representing information in HTML, or more generally, XML
- RDF/XML also supports CURIEs

```
<rdf:RDF xmlns:p="http://dbpedia.org/property/"
  xmlns:ex="http://example.org/tddd43/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex-liu="http://example.org/tddd43/liu/"
  xmlns:dbpedia="http://dbpedia.org/resource/">
  <rdf:Description rdf:about="ex:Linköping_University">
    <p:has_faculty rdf:resource="ex-liu:Science_and_Engineering"/>
    <p:location rdf:resource="dbpedia:Linköping"/>
    <p:location rdf:resource="dbpedia:Norrköping"/>
    <p:name xml:lang="en">Linköping University</p:name>
    <p:start_year rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1975</p:start_year>
  </rdf:Description>
</rdf:RDF>
```

# RDF/XML vs Turtle

```
@prefix p: <http://dbpedia.org/property/> .  
@prefix ex: <http://example.org/tddd43/> .  
@prefix ex-liu: <http://example.org/tddd43/liu/> .  
@prefix dbpedia: <http://dbpedia.org/resource/> .
```

Turtle

```
ex:Linköping_University p:has_faculty ex-liu:Science_and_Engineering ;  
    p:location dbpedia:Linköping, dbpedia:Norrköping ;  
    p:name "Linköping University"@en ;  
    p:start_year 1975 .
```

```
<rdf:RDF xmlns:p="http://dbpedia.org/property/"  
    xmlns:ex="http://example.org/tddd43/"  
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
    xmlns:ex-liu="http://example.org/tddd43/liu/"  
    xmlns:dbpedia="http://dbpedia.org/resource/">  
  <rdf:Description rdf:about="ex:Linköping_University">  
    <p:has_faculty rdf:resource="ex-liu:Science_and_Engineering"/>  
    <p:location rdf:resource="dbpedia:Linköping"/>  
    <p:location rdf:resource="dbpedia:Norrköping"/>  
    <p:name xml:lang="en">Linköping University</p:name>  
    <p:start_year rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">1975</p:start_year>  
  </rdf:Description>  
</rdf:RDF>
```

RDF/XML

# JSON-LD: JSON for Linked Data

- Encoding Linked Data (RDF data) using JSON
- Mappings data (keys) in JSON to RDF resources
  - @context
- Valid JSON

```
{
  "@context": {
    "ex": "http://example.org/",
    "p": "http://dbpedia.org/property",
    "dbpedia": "http://dbpedia.org/resource"
  }
  "@id" : "ex:Linköping_University",
  "p:has_faculty" : { "@id" : "http://example.org/tddd43/liu/Science_and_Engineering" },
  "p:location" : [
    { "@id" : "dbpedia:Linköping" },
    { "@id" : "dbpedia:Norrköping" }
  ],
  "p:name" :
  {
    "@language" : "en",
    "@value" : "Linköping University"
  },
  .....
}
```

JSON-LD

# JSON-LD

```
@prefix p: <http://dbpedia.org/property/> .  
@prefix ex: <http://example.org/tddd43/> .  
@prefix ex-liu: <http://example.org/tddd43/liu/> .  
@prefix dbpedia: <http://dbpedia.org/resource/> .
```

Turtle

```
ex:Linköping_University p:has_faculty ex-liu:Science_and_Engineering ;  
    p:location dbpedia:Linköping, dbpedia:Norrköping ;  
    p:name "Linköping University"@en ;  
    p:start_year 1975 .
```

```
{  
  "@context": {  
    "ex": "http://example.org/",  
    "p": "http://dbpedia.org/property",  
    "dbpedia": "http://dbpedia.org/resource"  
  }  
  "@id" : "ex:Linköping_University",  
  "p:has_faculty" : { "@id" : "http://example.org/tddd43/liu/Science_and_Engineering" },  
  "p:location" : [  
    { "@id" : "dbpedia:Linköping" },  
    { "@id" : "dbpedia:Norrköping" }  
  ],  
  "p:name" :  
    {  
      "@language" : "en",  
      "@value" : "Linköping University"  
    },  
  .....  
}
```

JSON-LD

# RDFa: *RDF* inside HTML *a*tttributes

- RDF annotation to (X)HTML at the attribute level
- RDFa 1.0 based on XHTML (W3C recommendation, 2008)
- RDFa 1.1 based on HTML5 (W3C recommendation, 2012)
- Attributes:
  - **about**: a URI or CURIE
  - **property**: specifying a property for a resource
  - **rel**: specifying a relationship with another resource
  - **resource**: specifying a resource
  - **datatype**: specifying datatype for a property

# RDFa: *RDF* inside HTML attributes

```
@prefix p: <http://dbpedia.org/property/> .  
@prefix ex: <http://example.org/tddd43/> .  
@prefix ex-liu: <http://example.org/tddd43/liu/> .  
@prefix dbpedia: <http://dbpedia.org/resource/> .
```

Turtle

```
ex:Linköping_University p:has_faculty ex-liu:Science_and_Engineering ;  
    p:location dbpedia:Linköping, dbpedia:Norrköping ;  
    p:name "Linköping University"@en ;  
    p:start_year 1975 .
```

```
<div xmlns="http://www.w3.org/1999/xhtml"  
    prefix="ex: http://example.org/  
        xsd: http://www.w3.org/2001/XMLSchema#  
        ex-liu: http://example.org/tddd43/liu/  
            dbpedia: http://dbpedia.org/resource/  
            p: http://dbpedia.org/property/">
```

RDFa

```
<div about="http://example.org/Linköping_University">  
    <div property="p:start_year" datatype="xsd:integer">1975</div>  
    <div property="p:name" datatype="xsd:string">Linköping University</div>  
    <div rel="p:location" resource="dbpedia:Linköping"></div>  
    <div rel="p:location" resource="dbpedia:Norrköping"></div>  
    <div rel="p:has_faculty" resource="ex-liu:Science_and_Engineering"></div>  
</div>  
</div>
```

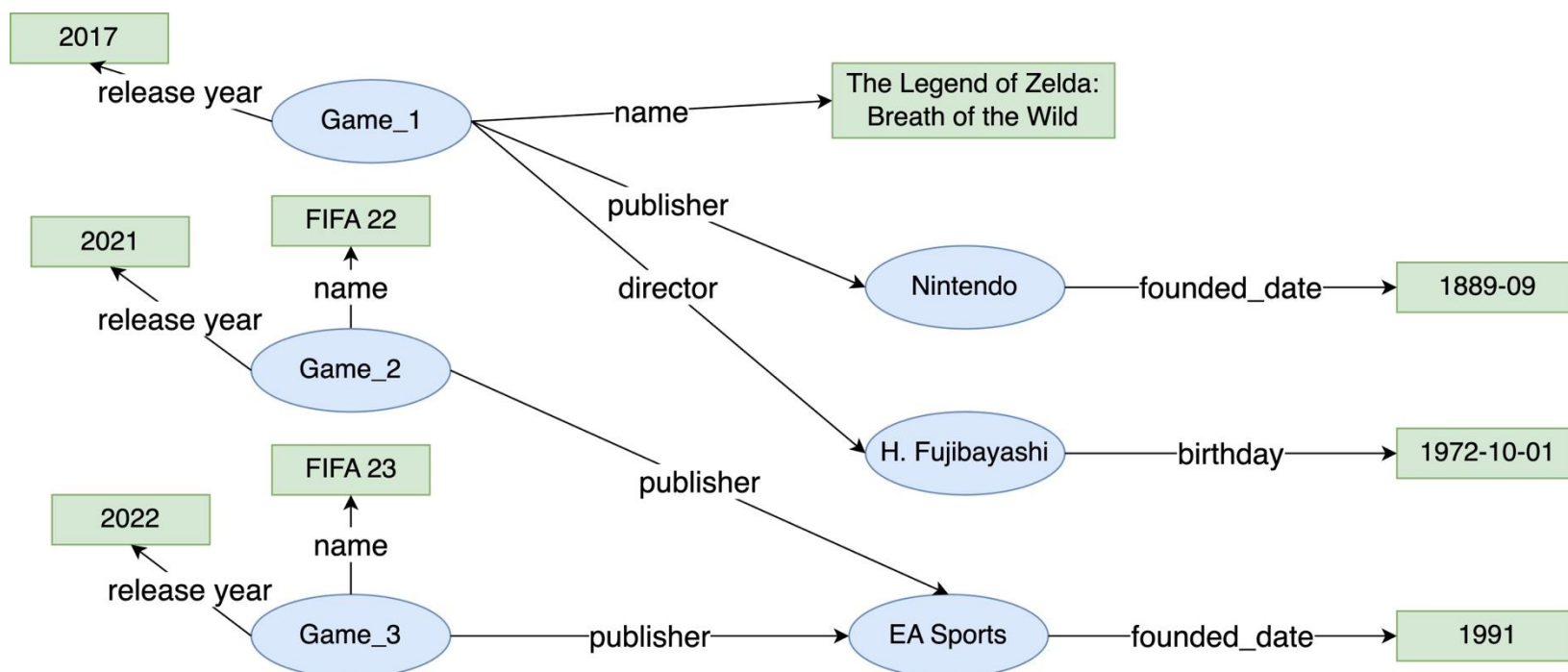


# RDF Syntax or Serialization formats

- Different ways to represent RDF data in text
  - N-Triples, *easy to parse*
  - Turtle, *easy to read and manually edit RDF*
  - RDF/XML, *for XML-based applications*
  - JSON-LD, *make existing JSON-based API support semantic*
  - RDFa, *extend existing HTML-based applications*

# Exercise

- Represent the following graph data in RDF (e.g., Turtle serialization)
  - What are the resources about?
  - What properties are needed to model the resources?



# Exercise

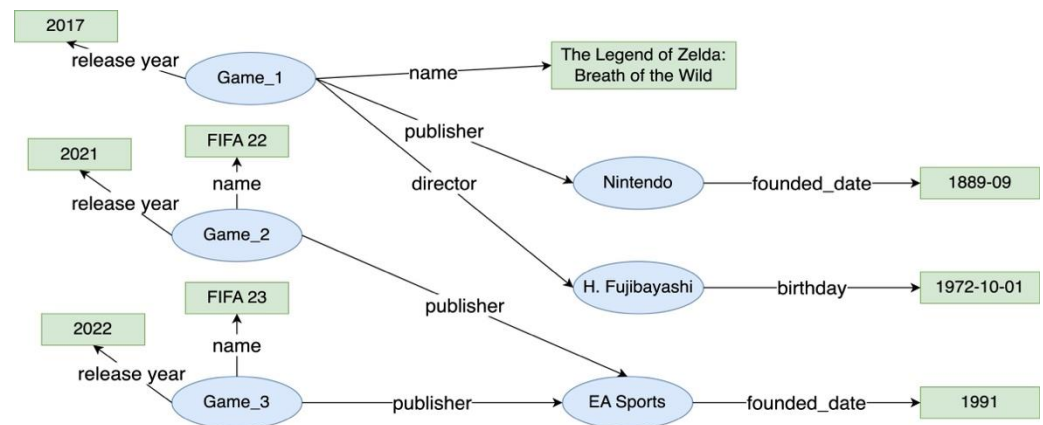
- Represent the following graph data in RDF (e.g., Turtle serialization)
  - What are the resources about?
  - What properties are needed to model the resources?

```
@prefix ex: <http://example.org/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

```
ex:EA_Sports rdf:type ex:Company ;  
            ex:founded_date "1889-09" .  
  
ex:Nintendo rdf:type ex:Company ;  
           ex:founded_date "1991" .  
  
ex:H._Fujibayashi rdf:type ex:Person ;  
                 ex:birthday "1972-10-01" .  
  
ex:Game_1 rdf:type ex:Game ;  
          ex:director ex:H._Fujibayashi ;  
          ex:publisher ex:Nintendo ;  
          ex:name "The Legend of Zelda: Breath of the Wild" ;  
          ex:release_year 2017 .
```

```
ex:Game_2 rdf:type ex:Game ;  
          ex:publisher ex:EA_Sports ;  
          ex:name "FIFA 22" ;  
          ex:release_year "2021" .
```

```
ex:Game_3 rdf:type ex:Game ;  
          ex:publisher ex:EA_Sports ;  
          ex:name "FIFA 23" ;  
          ex:release_year "2022" .
```



# RDF

- RDF provides a (simple) way to represent data
  - It is not yet useful, without ways to access such data or manipulate the data

# Outline

- ✓ RDF (Resource Description Framework)
- SPARQL (SPARQL Protocol And RDF Query Language)
- Linked Data

# SPARQL

- Recursive acronym for the SPARQL Protocol And RDF Query Language
- W3C recommendations
  - Query, Update, Protocol, Query result serialization formats
  - As a query language for RDF data
    - **Our focus today!**
  - As an update language for RDF data
  - As a protocol layer for using SPARQL via http
    - Communication between SPARQL processing services and clients

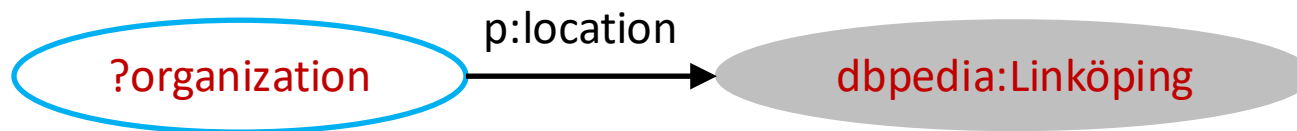
# The basic idea of SPARQL queries

- SPARQL, as a query language, shares features with others, e.g., XQUERY and SQL
- Like general Tell-and-Ask system
  - Get answers based on what we asked or told the system
- SPARQL variables (question words) bound to RDF terms (resource terms)
  - e.g., ?university, ?year
- A lot of overlap between SQL and SPARQL
  - e.g., SELECT, WHERE keywords

?university	?year
ex:Linköping_University	1975
ex:Stockholm_University	1878

# The basic idea of SPARQL queries (Example)

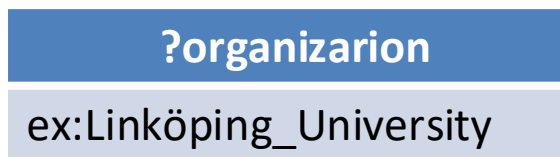
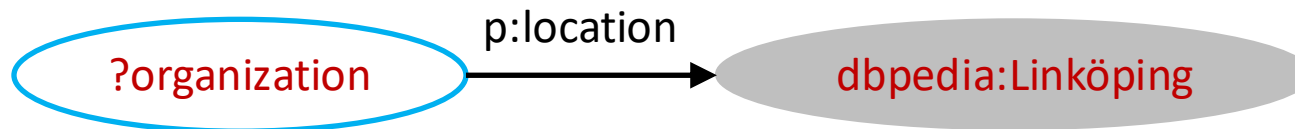
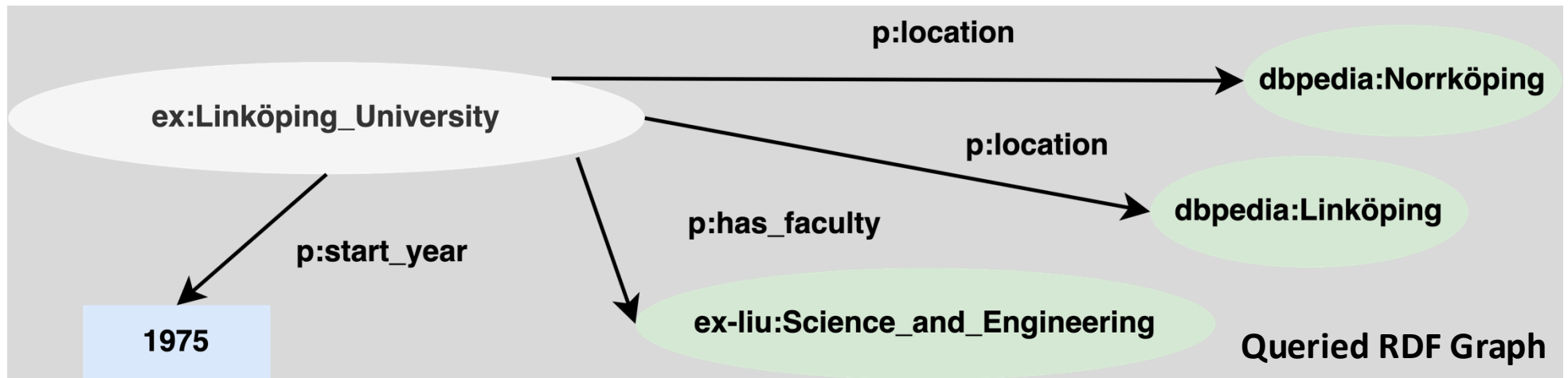
- The syntax of SPARQL is based on Turtle serialization
- Basic Graph Pattern Matching
  - Describe sub-graphs of the queried RDF graph
  - Sub-graphs that match the description yield a result
  - A graph pattern is an RDF triple in Turtle syntax by replacing subject/predicate/object with variables that you want to ask
  - A Basic Graph Pattern (BGP) is a set of Graph Patterns (Triple Patterns)
    - BGP contains one or more Graph Patterns



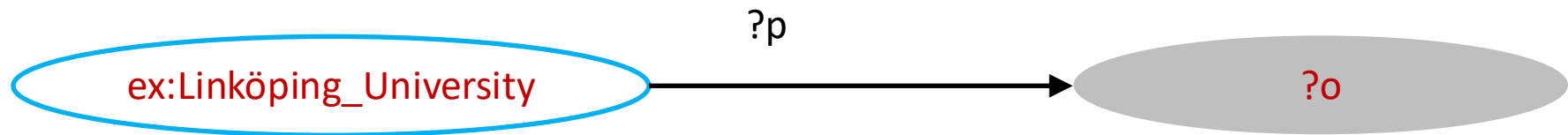
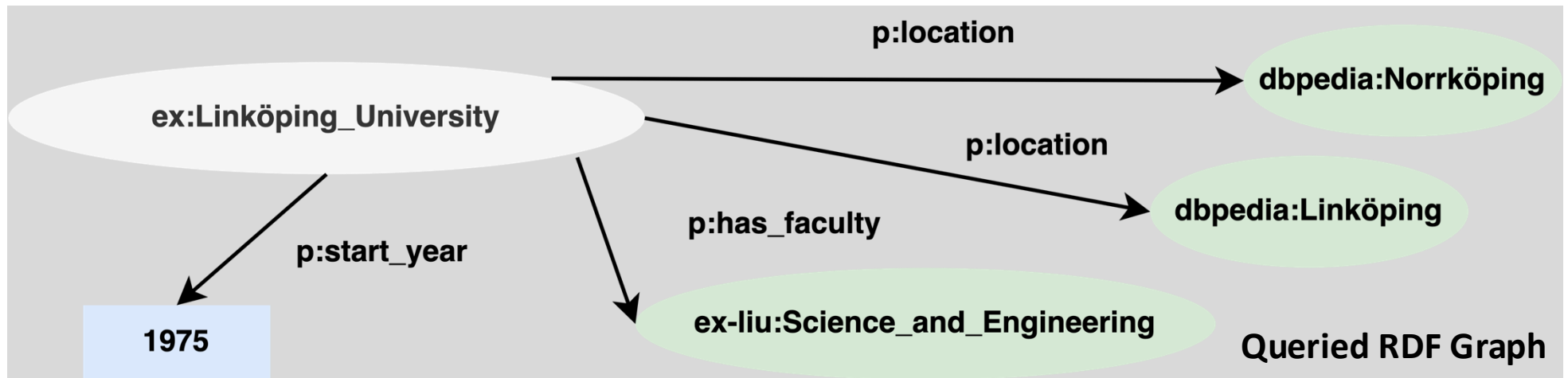
Graph Pattern Example



# The basic idea of SPARQL queries (Example)

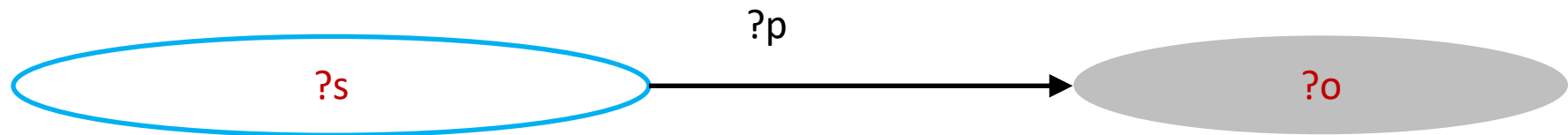
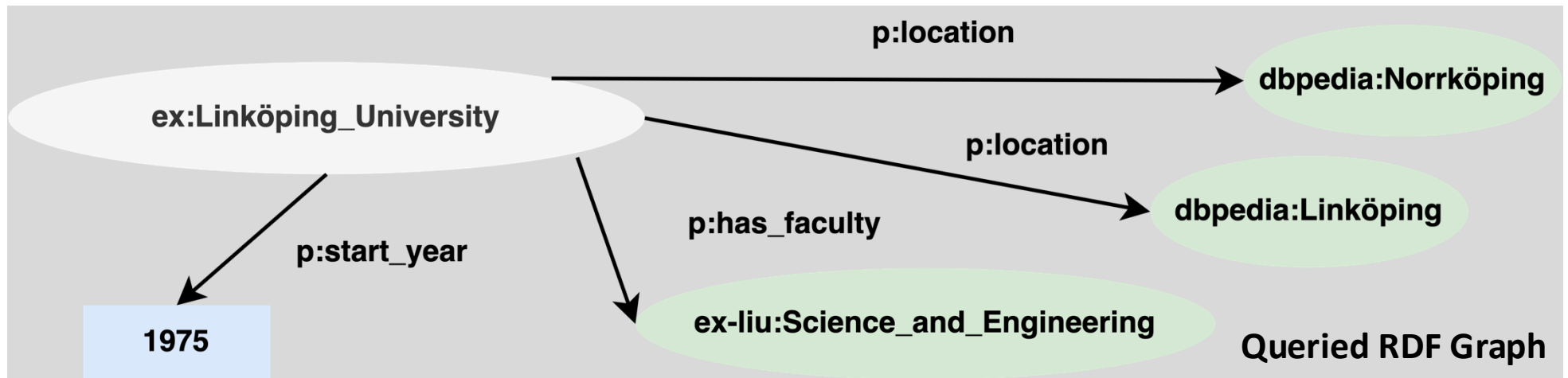


# The basic idea of SPARQL queries (Example)



?p	?o
p:location	dbpedia:Linköping
p:location	dbpedia:Norrköping
p:has_faculty	ex-liu:Science_and_Engineering
p:start_year	1975

# The basic idea of SPARQL queries (Example)



?s	?p	?o
ex:Linköping_University	p:location	dbpedia:Linköping
ex:Linköping_University	p:location	dbpedia:Norrköping
ex:Linköping_University	p:has_faculty	ex-liu:Science_and_Engineering
ex:Linköping_University	p:start_year	1975

# Components of a SPARQL query

**PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>**

**PREFIX ex: <http://example.org/tddd43#>**

**PREFIX p: <http://dbpedia.org/property/>**

**SELECT ?u ?y**

**FROM <http://example.org/mydata>**

**WHERE {**

**?u rdf:type ex:University .**

**?u p:start\_year ?y .**

**}**

**ORDER BY ?y**

# Components of a SPARQL query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX ex: <http://example.org/tddd43#>
```

```
PREFIX p: <http://dbpedia.org/property/>
```

```
SELECT ?u ?y
```

```
FROM <http://example.org/mydata>
```

```
WHERE {
```

```
  ?u rdf:type ex:University .
```

```
  ?u p:start_year ?y .
```

```
}
```

```
ORDER BY ?y
```

- Prologue
  - Prefix definitions by using compact URIs (CURIEs)

# Components of a SPARQL query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX ex: <http://example.org/tddd43#>
```

```
PREFIX p: <http://dbpedia.org/property/>
```

```
SELECT ?u ?y
```

```
FROM <http://example.org/mydata>
```

```
WHERE {
```

```
  ?u rdf:type ex:University .
```

```
  ?u p:start_year ?y .
```

```
}
```

```
ORDER BY ?y
```

- Result form specification:
  - SELECT for projection (similar to projection in relational algebra)
  - Other forms: DESCRIBE, CONSTRUCT, and ASK

# Components of a SPARQL query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX ex: <http://example.org/tddd43#>
```

```
PREFIX p: <http://dbpedia.org/property/>
```

```
SELECT ?u ?y
```

```
FROM <http://example.org/mydata>
```

```
WHERE {
```

```
  ?u rdf:type ex:University .
```

```
  ?u p:start_year ?y .
```

```
}
```

```
ORDER BY ?y
```

- Dataset specification:
  - RDF dataset to be accessed (use URIs to identify particular RDF data)
  - Not mandatory

# Components of a SPARQL query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX ex: <http://example.org/tddd43#>
```

```
PREFIX p: <http://dbpedia.org/property/>
```

```
SELECT ?u ?y
```

```
FROM <http://example.org/mydata>
```

```
WHERE {
```

```
  ?u rdf:type ex:University .
```

```
  ?u p:start_year ?y .
```

```
}
```

```
ORDER BY ?y
```

- Query Pattern
  - WHERE clause specifies the pattern to be matched



# Basic Graph Pattern

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX ex: <http://example.org/tddd43#>
```

```
PREFIX p: <http://dbpedia.org/property/>
```

```
SELECT ?u ?y
```

```
FROM <http://example.org/mydata>
```

```
WHERE {
```

```
  ?u rdf:type ex:University ;
```

```
  p:start_year ?y .
```

```
}
```

```
ORDER BY ?y
```

- Set of triple patterns (i.e., RDF triples with variables)
- Variable name prefixed with “?” (or “\$”)
- Turtle syntax

# Basic Graph Pattern (Example)

```
SELECT ?u ?y ?f
WHERE {
  ?u rdf:type ex:University .
  ?u p:start_year ?y .
  ?u p:has_faculty ?f .
}
```

```
ex:Linköping_University rdf:type    ex:University .
ex:Linköping_University p:has_faculty ex-liu:Science_and_Engineering .
ex:Linköping_University p:location  dbpedia:Linköping .
ex:Linköping_University p:location  dbpedia:Norrköping .
ex:Linköping_University p:name      "Linköping University"@en .
ex:Linköping_University p:start_year "1975"^^xsd:integer .
```

# Basic Graph Pattern (Example)

```
SELECT ?u ?y ?f
WHERE {
  ?u rdf:type ex:University .
  ?u p:start_year ?y .
  ?u p:has_faculty ?f .
}
```

```
ex:Linköping_University rdf:type    ex:University .
ex:Linköping_University p:has_faculty ex-liu:Science_and_Engineering .
ex:Linköping_University p:location  dbpedia:Linköping .
ex:Linköping_University p:location  dbpedia:Norrköping .
ex:Linköping_University p:name      "Linköping University"@en .
ex:Linköping_University p:start_year "1975"^^xsd:integer .
```

# Basic Graph Pattern (Example)

```
SELECT ?u ?y ?f
WHERE {
  ?u rdf:type ex:University .
  ?u p:start_year ?y .
  ?u p:has_faculty ?f .
}
```

```
ex:Linköping_University rdf:type    ex:University .
ex:Linköping_University p:has_faculty ex-liu:Science_and_Engineering .
ex:Linköping_University p:location  dbpedia:Linköping .
ex:Linköping_University p:location  dbpedia:Norrköping .
ex:Linköping_University p:name      "Linköping University"@en .
ex:Linköping_University p:start_year "1975"^^xsd:integer .
```

# Basic Graph Pattern (Example)

```
SELECT ?u ?y ?f
WHERE {
  ?u rdf:type ex:University .
  ?u p:start_year ?y .
  ?u p:has_faculty ?f .
}
```

```
ex:Linköping_University rdf:type    ex:University .
ex:Linköping_University p:has_faculty ex-liu:Science_and_Engineering .
ex:Linköping_University p:location  dbpedia:Linköping .
ex:Linköping_University p:location  dbpedia:Norrköping .
ex:Linköping_University p:name "Linköping University"@en .
ex:Linköping_University p:start_year "1975"^^xsd:integer .
```

?u	?y	?f
ex:Linköping_University	1975	ex-liu:Science_and_Engineering

# Basic Graph Pattern (Example)

```
SELECT ?u ?y ?f
WHERE {
  ?u rdf:type ex:University .
  ?u p:start_year ?y .
  ?u p:has_faculty ?f .
}
```

- What if a university is not organized based on faculties (e.g., based on schools, departments instead)?

```
ex:Linköping_University rdf:type ex:University .
ex:Linköping_University p:has_faculty ex-liu:Science_and_Engineering .
ex:Linköping_University p:location dbpedia:Linköping .
ex:Linköping_University p:location dbpedia:Norrköping .
ex:Linköping_University p:name "Linköping University"@en .
ex:Linköping_University p:start_year "1975"^^xsd:integer .
ex:Karolinska_Institute rdf:type ex:University .
ex:Karolinska_Institute p:has_department ex-ki:Neuroscience .
```

?u	?y	?f
ex:Linköping_University	1975	ex-liu:Science_and_Engineering

# Optional Graph Pattern

```
SELECT ?u ?f
WHERE {
  ?u rdf:type ex:University .
  OPTIONAL { ?u p:has_faculty ?f .}
}
```

- What if a university is not organized based on faculties (e.g., schools, departments)?
- Optional Keyword indicates optional patterns
  - May result in unbound variables

?u	?f
ex:Linköping_University	ex-liu:Science_and_Engineering
ex:Karolinska_Institute	

# Optional Graph Pattern

```
SELECT ?u ?f
WHERE {
  ?u rdf:type ex:University .
  OPTIONAL { ?u p:has_faculty ?f .}
  OPTIONAL { ?u p:has_department ?f .}
}
```

?u	?y	?f
ex:Linköping_University	1975	ex-liu:Science_and_Engineering
ex:Karolinska_Institute	1810	ex-ki:NeuroScience



# Constraints on Solutions

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX ex: <http://example.org/tddd43#>
```

```
PREFIX p: <http://dbpedia.org/property/>
```

```
SELECT ?u ?y
```

```
FROM <http://example.org/mydata>
```

```
WHERE {
```

```
  ?u rdf:type ex:University .
```

```
  ?u p:start_year ?y .
```

```
  FILTER ( ?y > 1970 )
```

```
}
```

```
ORDER BY ?y
```

- Syntax: Keyword FILTER followed by filter expression
- Filter expressions contain operators and functions
- Operators and functions operate on RDF terms

# Constraints on Solutions

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX ex: <http://example.org/tddd43#>
```

```
PREFIX p: <http://dbpedia.org/property/>
```

```
SELECT ?u ?y
```

```
FROM <http://example.org/mydata>
```

```
WHERE {
```

```
  ?u rdf:type ex:University .
```

```
  ?u p:start_year ?y .
```

```
  FILTER ( ?y > 1970 )
```

```
}
```

```
ORDER BY ?y
```

- Solution sequence modifiers
- Only for SELECT queries
- Keywords: DISTINCT, ORDER BY, LIMIT, and OFFSET

# Constraints on Solutions

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX ex: <http://example.org/tddd43#>
```

```
PREFIX p: <http://dbpedia.org/property/>
```

```
SELECT ?u ?y
```

```
FROM <http://example.org/mydata>
```

```
WHERE {
```

```
  ?u rdf:type ex:University .
```

```
  ?u p:start_year ?y .
```

```
  FILTER ( ?y > 1970 )
```

```
}
```

```
ORDER BY ?y
```

```
LIMIT 10
```

# Unary Operators in Filter Constraints

Operator	Type (A)	Result Type
!A	xsd:boolean	xsd:boolean
+A	numeric	numeric
-A	numeric	numeric
BOUND(A)	variable	xsd:boolean
isURI(A)	RDF term	xsd:boolean
isBLANK(A)	RDF term	xsd:boolean
isLITERAL(A)	RDF term	xsd:boolean
STR(A)	literal/URI	simple literal
LANG(A)	literal	simple literal
DATATYPE(A)	literal	URI/xsd:String/rdf:lang String

# Constraints on Solutions

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX ex: <http://example.org/tddd43#>

PREFIX p: <http://dbpedia.org/property/>

SELECT ?u ?y

FROM <http://example.org/mydata>

WHERE {

  ?u rdf:type ex:University .

  ?u p:start\_year ?y .

**FILTER ( !(?y > 1970) )**

}

# Binary and other Operators for Filter Constraints

- Logical connectives && and ||
- Comparison operations =, !=, <, >, <=, and >=
  - For numeric datatypes, xsd:dateTime, xsd:string, xsd:boolean
- Comparison operators = and !=
  - For other datatypes
- Furthermore:
  - REGEX (String, Pattern) or REGEX(String, Pattern, Flags)
  - ...

# Constraints on Solutions

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX ex: <http://example.org/tddd43#>
```

```
PREFIX p: <http://dbpedia.org/property/>
```

```
SELECT ?u ?y
```

```
FROM <http://example.org/mydata>
```

```
WHERE {
```

```
  ?u rdf:type ex:University .
```

```
  ?u p:start_year ?y .
```

```
  FILTER ( ?y > 1970 && ?y < 2000 )
```

```
}
```

# Constrains (Example)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX ex: <http://example.org/tddd43#>
```

```
PREFIX p: <http://dbpedia.org/property/>
```

```
SELECT ?u ?y
```

```
FROM <http://example.org/mydata>
```

```
WHERE {
```

```
  ?u rdf:type ex:University ;
```

```
    p:start_year ?y ;
```

```
    p:name ?name .
```

```
  FILTER( REGEX(STR(?name), "technology", "i") )
```

```
}
```



# SPARQL 1.1

- New features of SPARQL 1.1 Query
  - Aggregate functions (e.g., COUNT, SUM, AVG, GROUP BY)
  - Sub-queries
  - Negation (EXISTS, NOT EXISTS)
  - <https://www.w3.org/TR/sparql11-query/>
- SPARQL 1.1 Update
  - Graph update (INSERT DATA, DELETE DATA, etc.)
  - Graph Management (CREATE, DROP, etc.)
- If you want to try more on SPARQL
  - DBpedia SPARQL endpoint: <http://dbpedia.org/sparql>
  - DBpedia structure: [http://dbpedia.org/resource/Linköping\\_University](http://dbpedia.org/resource/Linköping_University)

# Constrains (Example)

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX ex: <http://example.org/tddd43#>
```

```
PREFIX p: <http://dbpedia.org/property/>
```

```
SELECT COUNT(?u)
```

```
FROM <http://example.org/mydata>
```

```
WHERE {
```

```
  ?u rdf:type ex:University ;
```

```
    p:start_year ?y ;
```

```
    p:name ?name .
```

```
  FILTER( REGEX(STR(?name), "technology", "i") )
```

```
}
```

# Exercise

- Q1: Retrieve all games that are published by EA Sports
- Q2: Retrieve games that have specific directors
- Q3: What's the result of the following query?

```
SELECT ?g ?d ?p WHERE {
  ?g rdf:type ex:Game .
  ?g ex:release_date ?d .
  OPTIONAL {
    ?g ex:director ?p .
  }
}
```

```
@prefix ex: <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

ex:EA_Sports rdf:type ex:Company ;
             ex:founded_date "1889-09" .

ex:Nintendo rdf:type ex:Company ;
            ex:founded_date "1991" .

ex:H._Fujibayashi rdf:type ex:Person ;
                  ex:birthday "1972-10-01" .

ex:Game_1 rdf:type ex:Game ;
           ex:director ex:H._Fujibayashi ;
           ex:publisher ex:Nintendo ;
           ex:name "The Legend of Zelda: Breath of the Wild" ;
           ex:release_year 2017 .

ex:Game_2 rdf:type ex:Game ;
           ex:publisher ex:EA_Sports ;
           ex:name "FIFA 22" ;
           ex:release_year "2021" .

ex:Game_3 rdf:type ex:Game ;
           ex:publisher ex:EA_Sports ;
           ex:name "FIFA 23" ;
           ex:release_year "2022" .
```

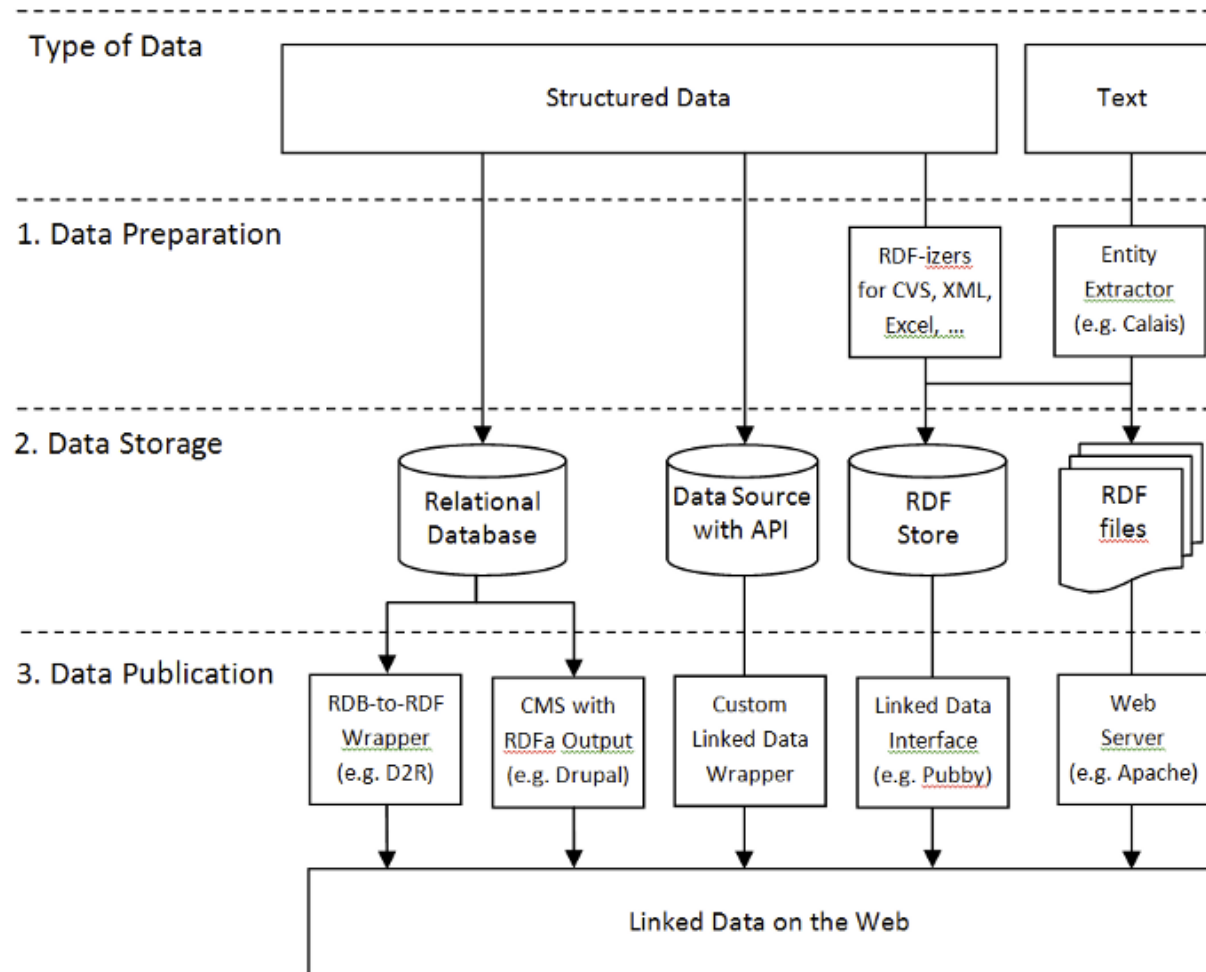
# Outline

- ✓ RDF (Resource Description Framework)
- ✓ SPARQL (SPARQL Protocol And RDF Query Language)
- Linked Data

# Data on the Web

- The goal of linked data is to publish structure data on the WWW in a standardized, machine-readable manner
- From Syntactic Web to Semantic Web (Linked Data),
  - Use HTTP URIs to name everything
  - Provide useful information to describe data based on standards (RDF, SPARQL)
  - Have links to other URIs in the descriptive data
- Linked Open Data
  - [Publicly](#) available (RDF) data on the Web

# Data on the Web



Tom Heath and Christian Bizer (2011) *Linked Data: Evolving the Web into a Global Data Space* (1st edition).

# Data on the Web



## About: [Linköping University](#)

An Entity of Type: [Public university](#), from Named Graph: <http://dbpedia.org>, within Data Space: [dbpedia.org](#)

Linköping University (Swedish: Linköpings universitet, LiU) is a public research university based in Linköping, Sweden. Originally established in 1969, it was granted full university status in 1975 and is one of Sweden's largest academic institutions. The university has four campuses across three cities: Campus Valla and Campus US in Linköping, Campus Norrköping in Norrköping and Campus Lidingö in Stockholm. It is organized into four faculties: Arts and Sciences, Medicine and Health Sciences, Educational Sciences, and the Institute of Technology. In order to facilitate interdisciplinary work, there are 12 large departments combining knowledge from several disciplines and often belonging under more than one faculty. Linköping University emphasises dialogue with the surrounding business sphere

Property	Value
<a href="#">dbo:abstract</a>	<ul style="list-style-type: none"><li>Linköping University (Swedish: Linköpings universitet, LiU) is a public research university based in Linköping, Sweden. Originally established in 1969, it was granted full university status in 1975 and is one of Sweden's largest academic institutions. The university has four campuses across three cities: Campus Valla and Campus US in Linköping, Campus Norrköping in Norrköping and Campus Lidingö in Stockholm. It is organized into four faculties: Arts and Sciences, Medicine and Health Sciences, Educational Sciences, and the Institute of Technology. In order to facilitate interdisciplinary work, there are 12 large departments combining knowledge from several disciplines and often belonging under more than one faculty. Linköping University emphasises dialogue with the surrounding business sphere and the community at large, both in terms of research and education. In 2021 the university was home to 35,900 students and 4,300 employees. It is a member of the European Consortium of Innovative Universities, as well as a founding member of the Conceive Design Implement Operate Initiative. <a href="#">(en)</a></li></ul>
<a href="#">dbo:affiliation</a>	<ul style="list-style-type: none"><li><a href="#">dbr:European_University_Association</a></li><li><a href="#">dbr:European_Consortium_of_Innovative_Universities</a></li><li><a href="#">dbr:European_Society_for_Engineering_Education</a></li><li><a href="#">dbr:Conceive_Design_Implement_Operate</a></li></ul>
<a href="#">dbo:city</a>	<ul style="list-style-type: none"><li><a href="#">dbr:Lidingö</a></li><li><a href="#">dbr:Linköping</a></li><li><a href="#">dbr:Norrköping</a></li></ul>
<a href="#">dbo:country</a>	<ul style="list-style-type: none"><li><a href="#">dbr:Sweden</a></li></ul>

## About: [European University Association](#)

An Entity of Type: [agent](#), from Named Graph: <http://dbpedia.org>, within Data Space: [dbpedia.org](#)

The European University Association (EUA) represents more than 800 institutions of higher education in 48 countries, providing them with a forum for cooperation and exchange of information on higher education and research policies. Members of the Association are European universities involved in teaching and research, national associations of rectors and other organisations active in higher education and research.



Property	Value
<a href="#">dbo:abstract</a>	<ul style="list-style-type: none"><li>The European University Association (EUA) represents more than 800 institutions of higher education in 48 countries, providing them with a forum for cooperation and exchange of information on higher education and research policies. Members of the Association are European universities involved in teaching and research, national associations of rectors and other organisations active in higher education and research. EUA is the result of a merger between the Association of European Universities and the Confederation of European Union Rectors' Conferences. The merger took place in Salamanca on 31 March 2001. <a href="#">(en)</a></li></ul>

## About: [Linköping](#)

An Entity of Type: [city](#), from Named Graph: <http://dbpedia.org>, within Data Space: [dbpedia.org](#)

Linköping (Swedish: [ˈlɪ̃ŋːçœːpɪŋ]) is a city in southern Sweden, with around 105,000 inhabitants as of 2021. It is the seat of Linköping Municipality and the capital of Östergötland County. Linköping is also the episcopal see of the Diocese of Linköping (Church of Sweden) and is well known for its cathedral. Linköping is the center of an old cultural region and celebrated its 700th anniversary in 1987. Dominating the city's skyline from afar is the steeple of the cathedral, Domkyrka.



Property	Value
<a href="#">dbo:PopulatedPlace/AreaTotal</a>	<ul style="list-style-type: none"><li>42.16</li></ul>
<a href="#">dbo:abstract</a>	<ul style="list-style-type: none"><li>Linköping (Swedish: <span>[ˈlɪ̃ŋːçœːpɪŋ]</span>) is a city in southern Sweden, with around 105,000 inhabitants as of 2021. It is the seat of Linköping Municipality and the capital of Östergötland County. Linköping is also the episcopal see of the Diocese of Linköping (Church of Sweden) and is well known for its cathedral. Linköping is the center of an old cultural region and celebrated its 700th anniversary in 1987. Dominating the city's skyline from afar is the steeple of the cathedral, Domkyrka. Nowadays, Linköping is known for its university and its high-technology industry. Linköping wants to create a sustainable development of the city and therefore plans to become a carbon-neutral community by 2025. Located on the Östergötland Plain, Linköping is closely linked to Norrköping, roughly 40 kilometres (25 mi) to the east, near the sea. <a href="#">(en)</a></li></ul>
<a href="#">dbo:areaCode</a>	<ul style="list-style-type: none"><li><span>(+46) 13</span></li></ul>

## About: [Sweden](#)

An Entity of Type: [place](#), from Named Graph: <http://dbpedia.org>, within Data Space: [dbpedia.org](#)

Sweden, formally the Kingdom of Sweden, is a Nordic country in Scandinavia. It borders Norway to the west and north, Finland to the east, and is connected to Denmark in the southwest by a bridge-tunnel across the Öresund. At 450,295 square kilometres (173,860 sq mi), Sweden is the largest Nordic country, the third-largest country in the European Union, and the fifth-largest country in Europe. The capital and largest city is Stockholm. Sweden has a total population of 10.5 million, and a low population density of 25.5 inhabitants per square kilometre (66/sq mi), with around 87% of Swedes residing in urban areas in the central and southern half of the country.



Property	Value
<a href="#">dbo:PopulatedPlace/area</a>	<ul style="list-style-type: none"><li>450295.0</li><li>455568.5486556611</li></ul>
<a href="#">dbo:PopulatedPlace/populationDensity</a>	<ul style="list-style-type: none"><li>25.0</li><li>25.096640305258983</li></ul>
<a href="#">dbo:abstract</a>	<ul style="list-style-type: none"><li>Sweden, formally the Kingdom of Sweden, is a Nordic country in Scandinavia. It borders Norway to the west and north, Finland to the east, and is connected to Denmark in the southwest by a bridge-tunnel across the Öresund. At 450,295 square kilometres (173,860 sq mi), Sweden is the largest Nordic country, the third-largest country in the European Union, and the fifth-largest country in Europe.</li></ul>

[https://dbpedia.org/page/Linköping\\_University](https://dbpedia.org/page/Linköping_University)



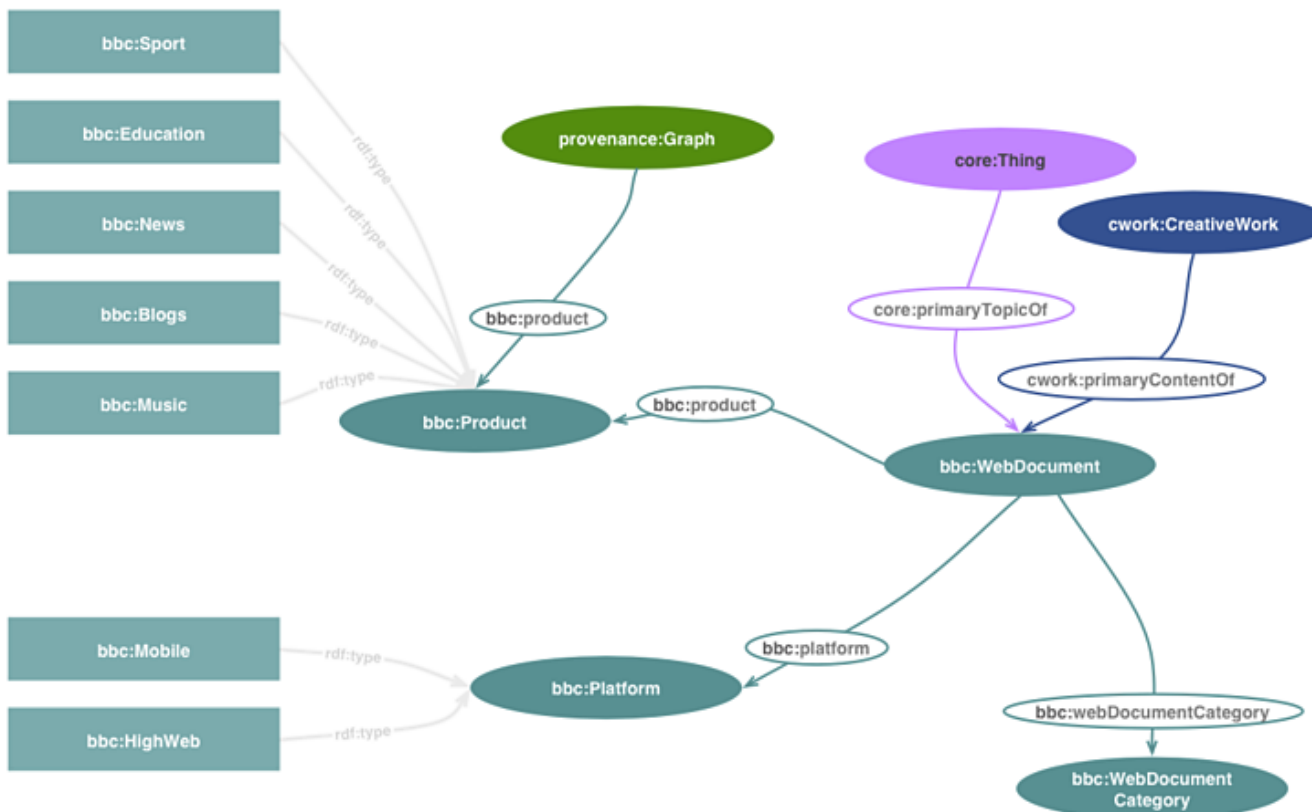
# Linked Data - Adoption

- Started as a grassroots community effort in 2007
  - Publish existing, open license datasets as Linked Data
- Prominent publishers joined the effort
  - BBC, NY Times, UK Government, etc.
- DBpedia, WikiData



# Linked Data - Adoption

- Example: BBC ontology
  - <https://www.bbc.co.uk/ontologies/bbc-ontology/>



# DBpedia endpoint

- <https://dbpedia.org/sparql>

The screenshot shows the SPARQL Query Editor interface. At the top, there is a navigation bar with "SPARQL Query Editor" and "About Tables" on the left, and "Conductor", "Facet Browser", and "Permalink" on the right. Below this, there are "Extensions:" buttons for "cxml", "save to dav", "sponge", and "User: SPARQL". The "Default Data Set Name (Graph IRI)" field contains "http://dbpedia.org". The "Query Text" field contains the following SPARQL query:

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?s ?city
WHERE {
  ?s dbo:type dbr:Public_university.
  ?s dbo:country dbr:Sweden.
  ?s dbo:city ?city.
}
```

The "Results Format" dropdown menu is set to "HTML".

[www.liu.se](http://www.liu.se)