SPARQL

Mauricio Espinoza Mejía

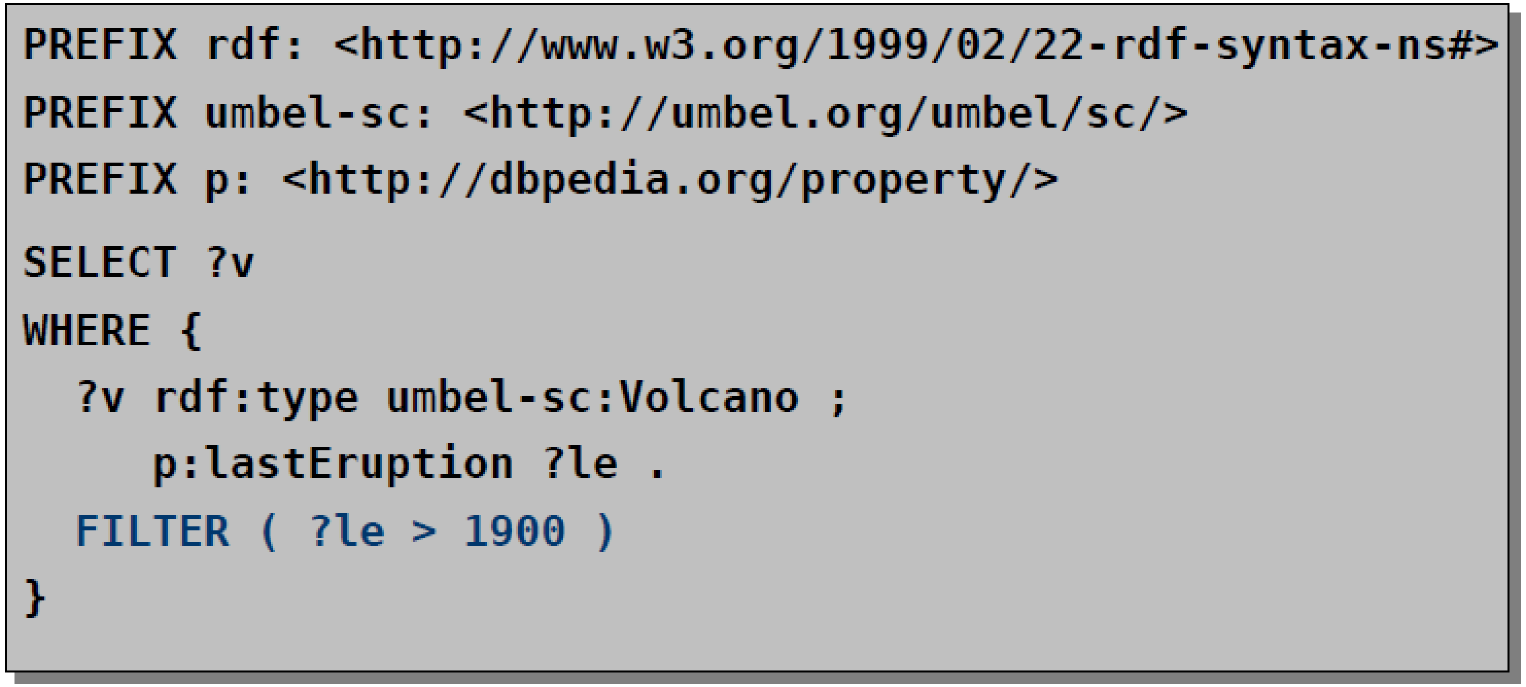
mauricio.espinoza@ucuenca.edu.ec

Content

- Main Idea of SPARQL Queries
- Components of SPARQL Queries
- Graph Patterns
- Constraints on Solutions
- Components of SPARQL Queries: Other feautures
- Solution Modifiers
- SPARQL 1.1

Constraints on Solutions

- **Syntax: Keyword FILTER followed by filter expression**

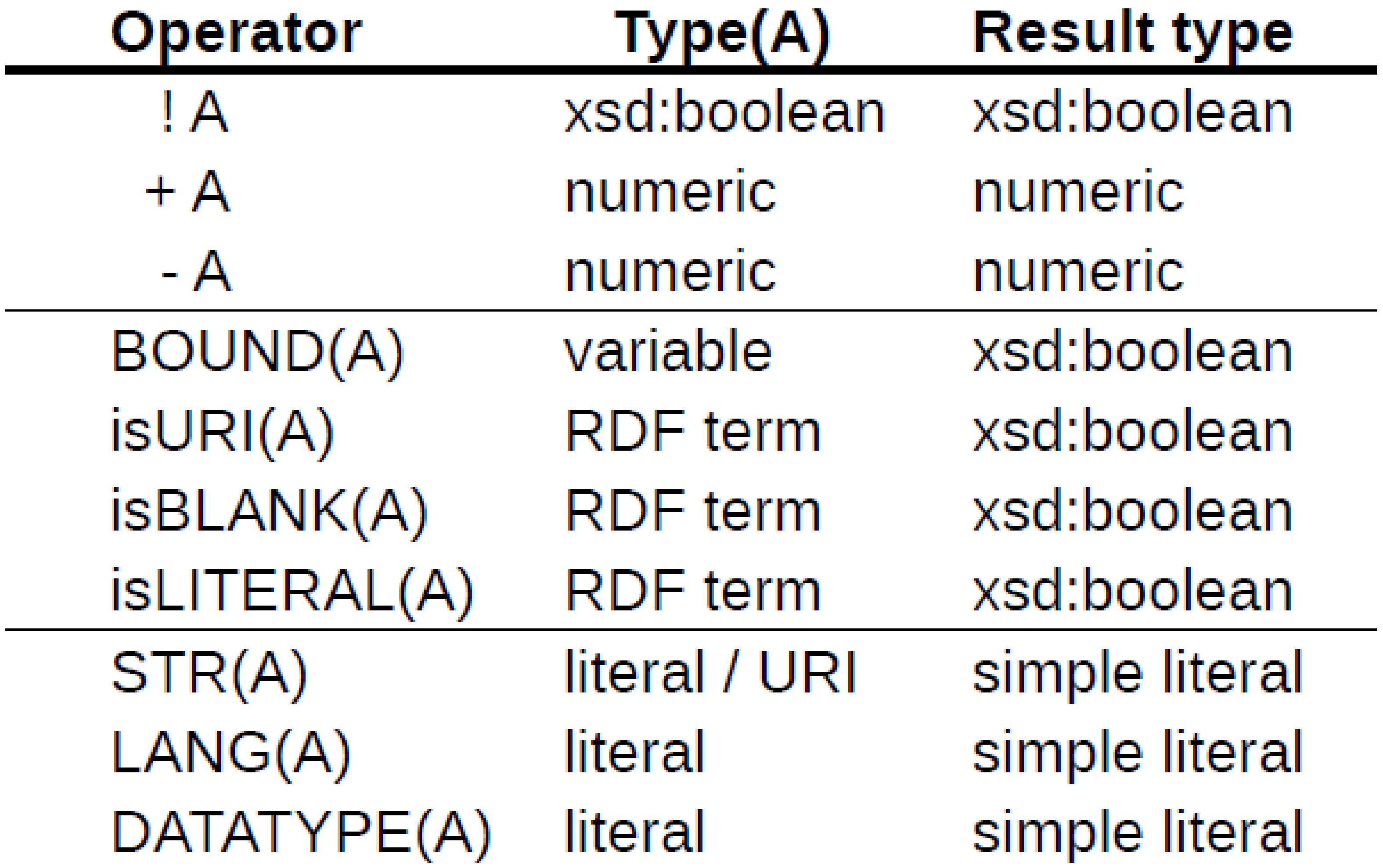


```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX umbel-sc: <http://umbel.org/umbel/sc/>
PREFIX p: <http://dbpedia.org/property/>

SELECT ?v
WHERE {
  ?v rdf:type umbel-sc:Volcano ;
     p:lastEruption ?le .
  FILTER ( ?le > 1900 )
}
```

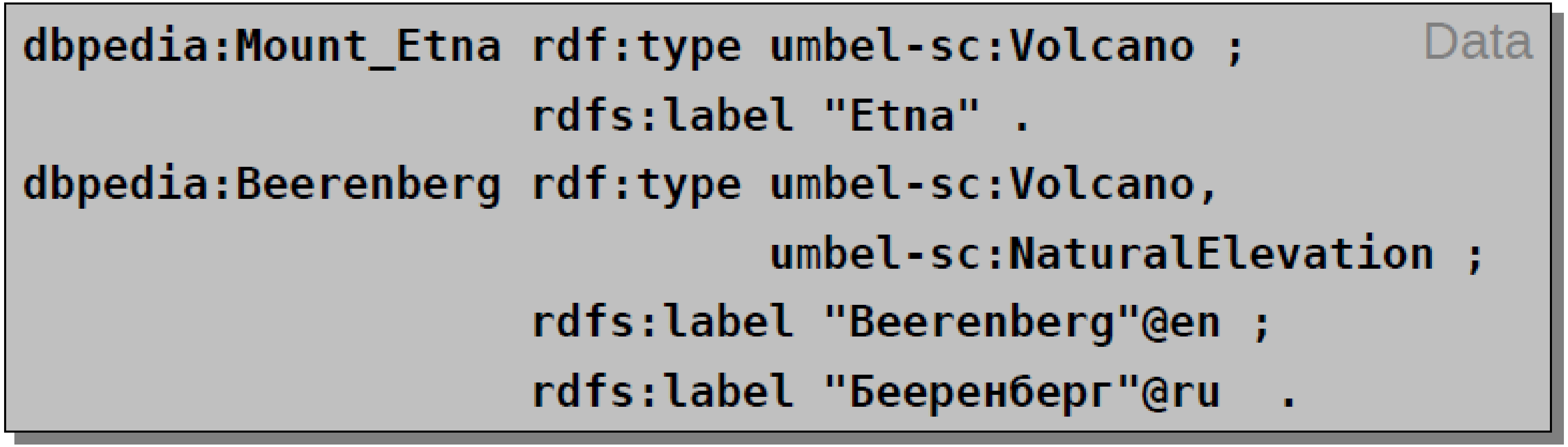
- **Filter expressions contain operators and functions**
- **Operators and functions operate on RDF terms**

Unary Operators in 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	simple literal

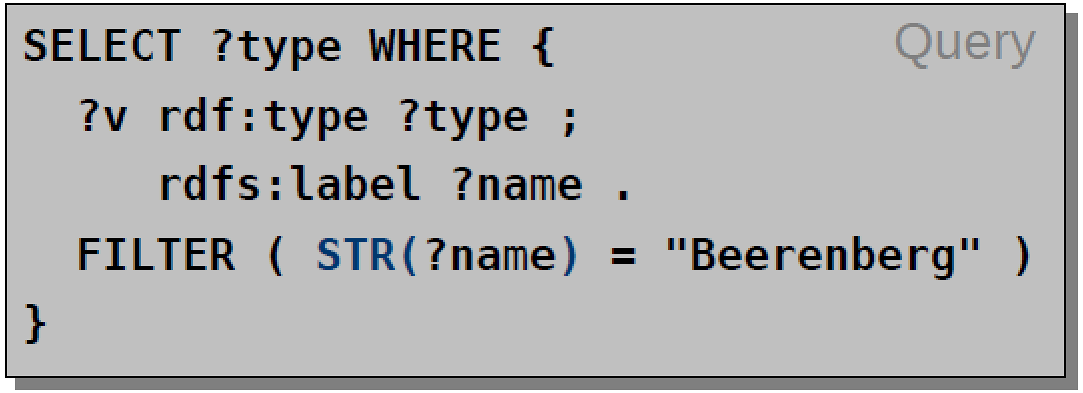
Constraints (Example)



Data

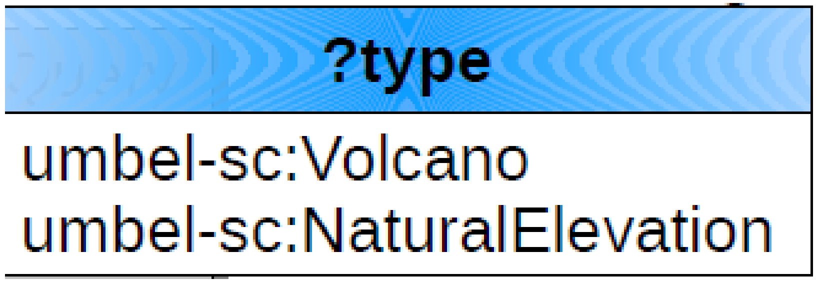
```
dbpedia:Mount_Etna rdf:type umbel-sc:Volcano ;
                   rdfs:label "Etna" .
dbpedia:Beerenberg rdf:type umbel-sc:Volcano,
                            umbel-sc:NaturalElevation ;
                   rdfs:label "Beerenberg"@en ;
                   rdfs:label "Беерендерг"@ru  .
```

- **Question: List all types of the volcano called “Beerenberg”**



Query

```
SELECT ?type WHERE {
  ?v rdf:type ?type ;
     rdfs:label ?name .
  FILTER ( STR(?name) = "Beerenberg" )
}
```



?type
umbel-sc:Volcano
umbel-sc:NaturalElevation

Constraints (Further Operators)

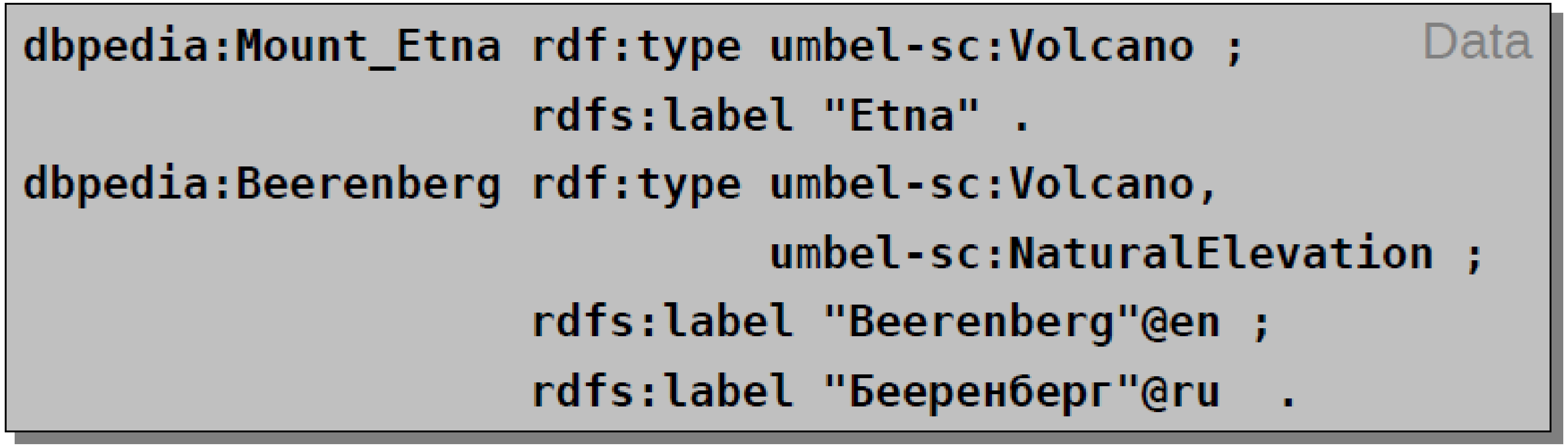
Binary operators:

- Logical connectives && and || for xsd:boolean
- Comparison operators =, !=, <, >, <=, and >= for numeric datatypes, xsd:dateTime, xsd:string, and xsd:boolean
- Comparison operators = and != for other datatypes
- Arithmetic operators +, -, *, and / for numeric datatypes

Furthermore:

- REGEX(String,Pattern) or REGEX(String,Pattern,Flags)
- sameTERM(A,B)
- langMATCHES(A,B)

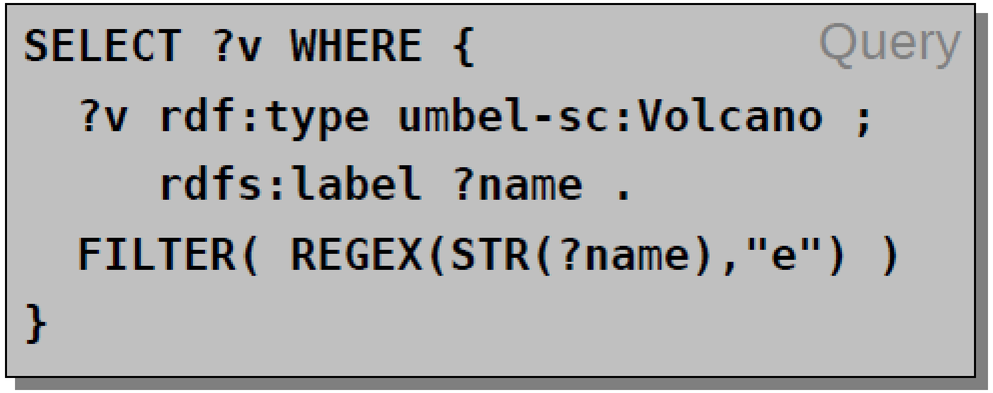
Constraints (Example)



Data

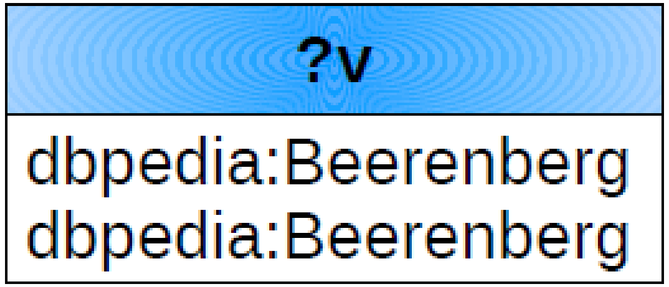
```
dbpedia:Mount_Etna rdf:type umbel-sc:Volcano ;
                   rdfs:label "Etna" .
dbpedia:Beerenberg rdf:type umbel-sc:Volcano,
                            umbel-sc:NaturalElevation ;
                   rdfs:label "Beerenberg"@en ;
                   rdfs:label "Бееренберг"@ru  .
```

- **Question: What volcanos have an “e” in their name?**



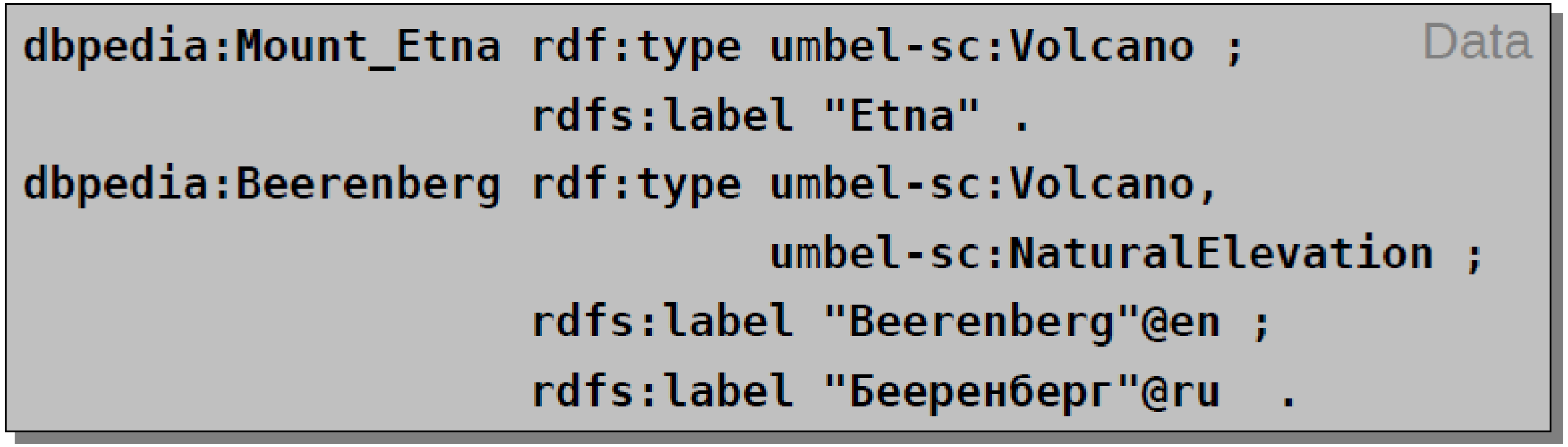
Query

```
SELECT ?v WHERE {
  ?v rdf:type umbel-sc:Volcano ;
     rdfs:label ?name .
  FILTER( REGEX(STR(?name),"e") )
}
```



?v
dbpedia:Beerenberg
dbpedia:Beerenberg

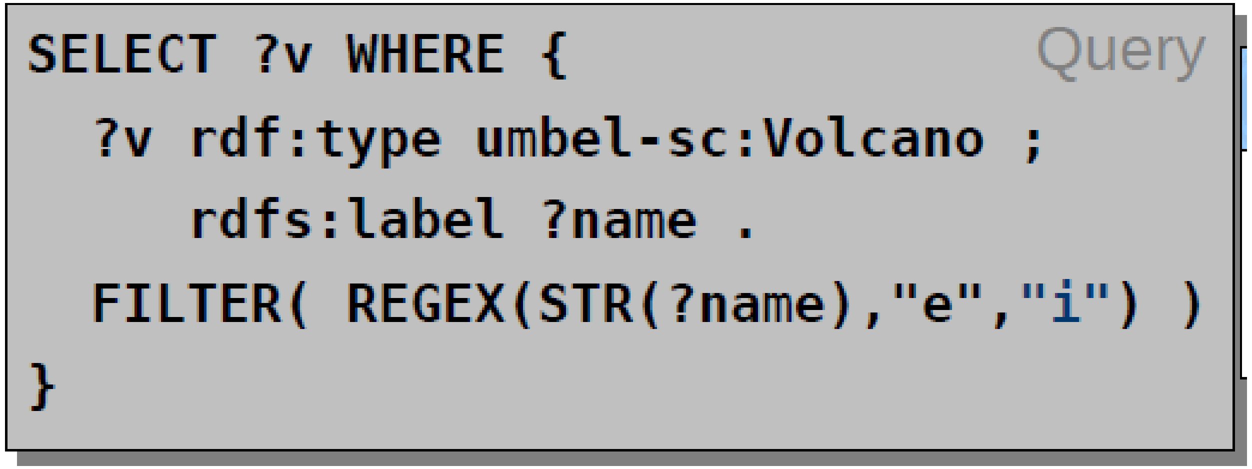
Constraints (Example)



Data

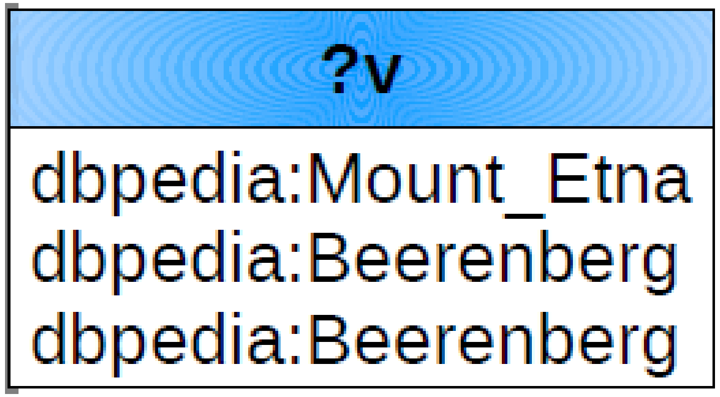
```
dbpedia:Mount_Etna rdf:type umbel-sc:Volcano ;
                   rdfs:label "Etna" .
dbpedia:Beerenberg rdf:type umbel-sc:Volcano,
                            umbel-sc:NaturalElevation ;
                   rdfs:label "Beerenberg"@en ;
                   rdfs:label "Бееренберг"@ru  .
```

- **Question: What volcanos have an “e” in their name?**



Query

```
SELECT ?v WHERE {
  ?v rdf:type umbel-sc:Volcano ;
     rdfs:label ?name .
  FILTER( REGEX(STR(?name),"e","i") )
}
```

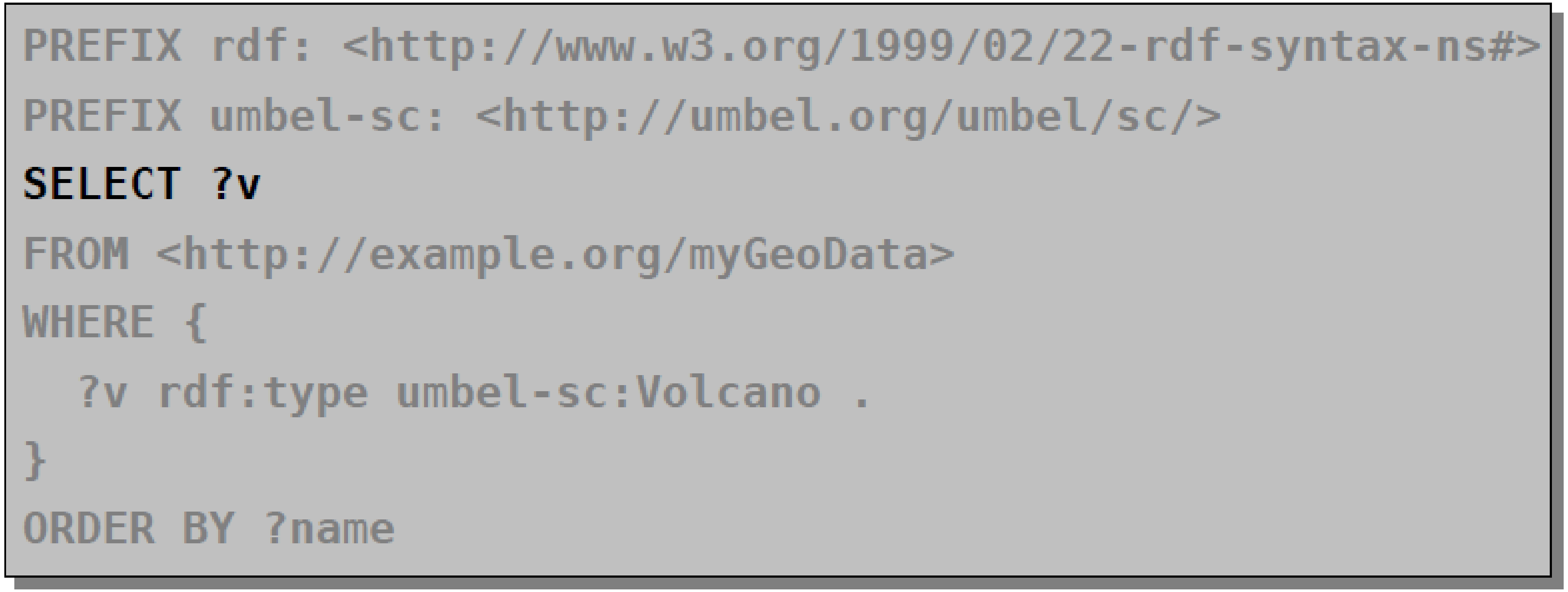


?v
dbpedia:Mount_Etna
dbpedia:Beerenberg
dbpedia:Beerenberg

Content

- Main Idea of SPARQL Queries
- Components of SPARQL Queries
- Graph Patterns
- Constraints on Solutions
- Components of SPARQL Queries: Other feautures
- Solution Modifiers
- SPARQL 1.1

Components of SPARQL Queries

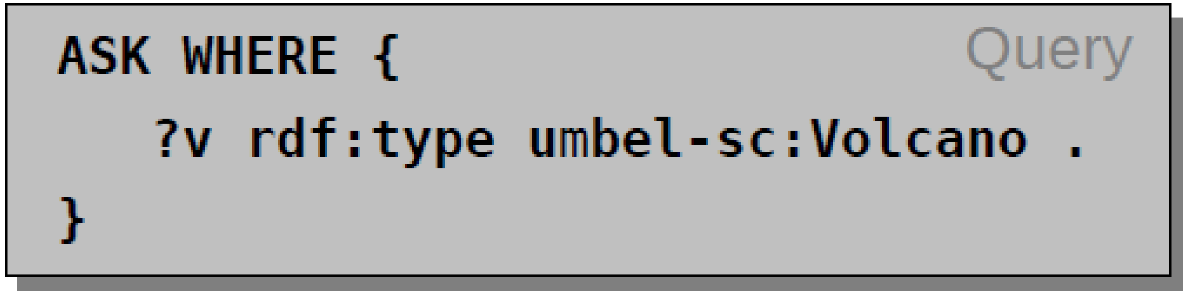


```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX umbel-sc: <http://umbel.org/umbel/sc/>
SELECT ?v
FROM <http://example.org/myGeoData>
WHERE {
  ?v rdf:type umbel-sc:Volcano .
}
ORDER BY ?name
```

- **Result form specification:**
- SELECT, DESCRIBE, CONSTRUCT, or ASK

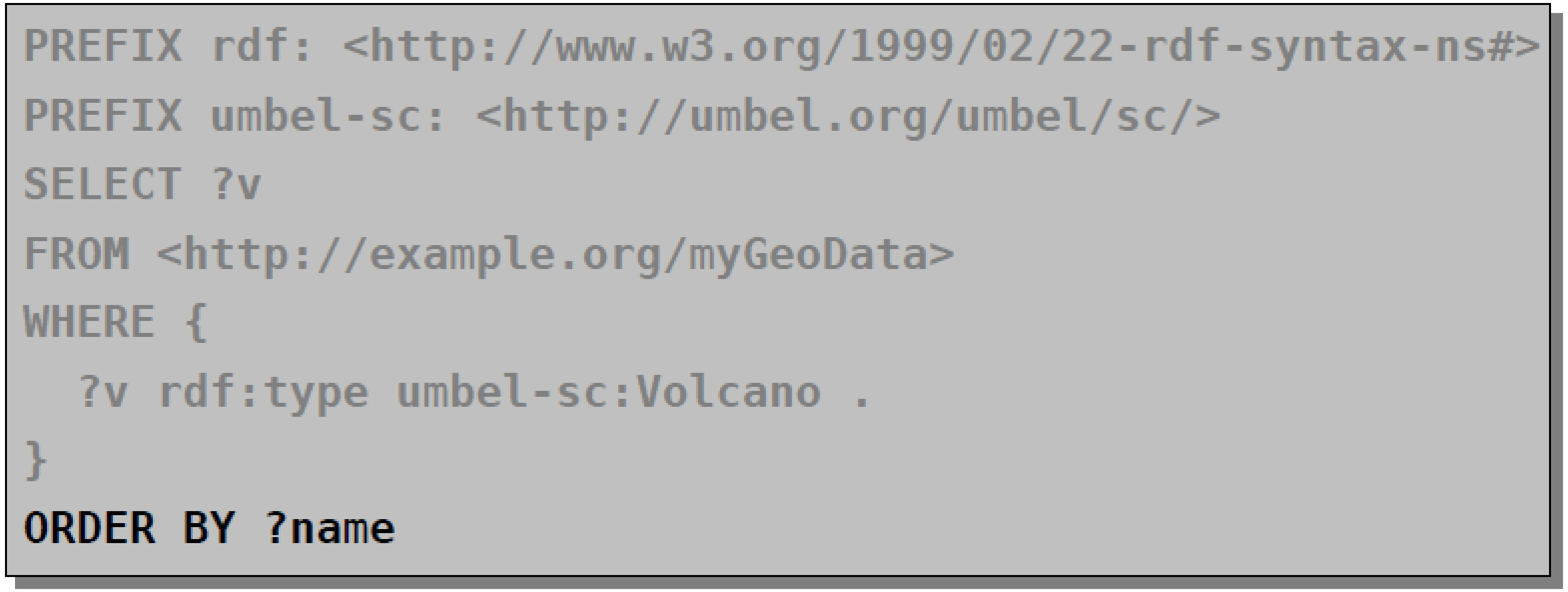
Result Forms

- **SELECT**
  - Result: sequence of solutions (i.e. sets of variable bindings)
  - Selected variables separated by space (not by comma!)
  - Asterisk character (“*”) selects all variables in the pattern
- **ASK**
  - Check whether there is at least one result
  - Result: true or false
  - Example: Do we have data about volcanos?



```
ASK WHERE {
   ?v rdf:type umbel-sc:Volcano .
}
```

Components of SPARQL Queries



```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX umbel-sc: <http://umbel.org/umbel/sc/>
SELECT ?v
FROM <http://example.org/myGeoData>
WHERE {
  ?v rdf:type umbel-sc:Volcano .
}
ORDER BY ?name
```

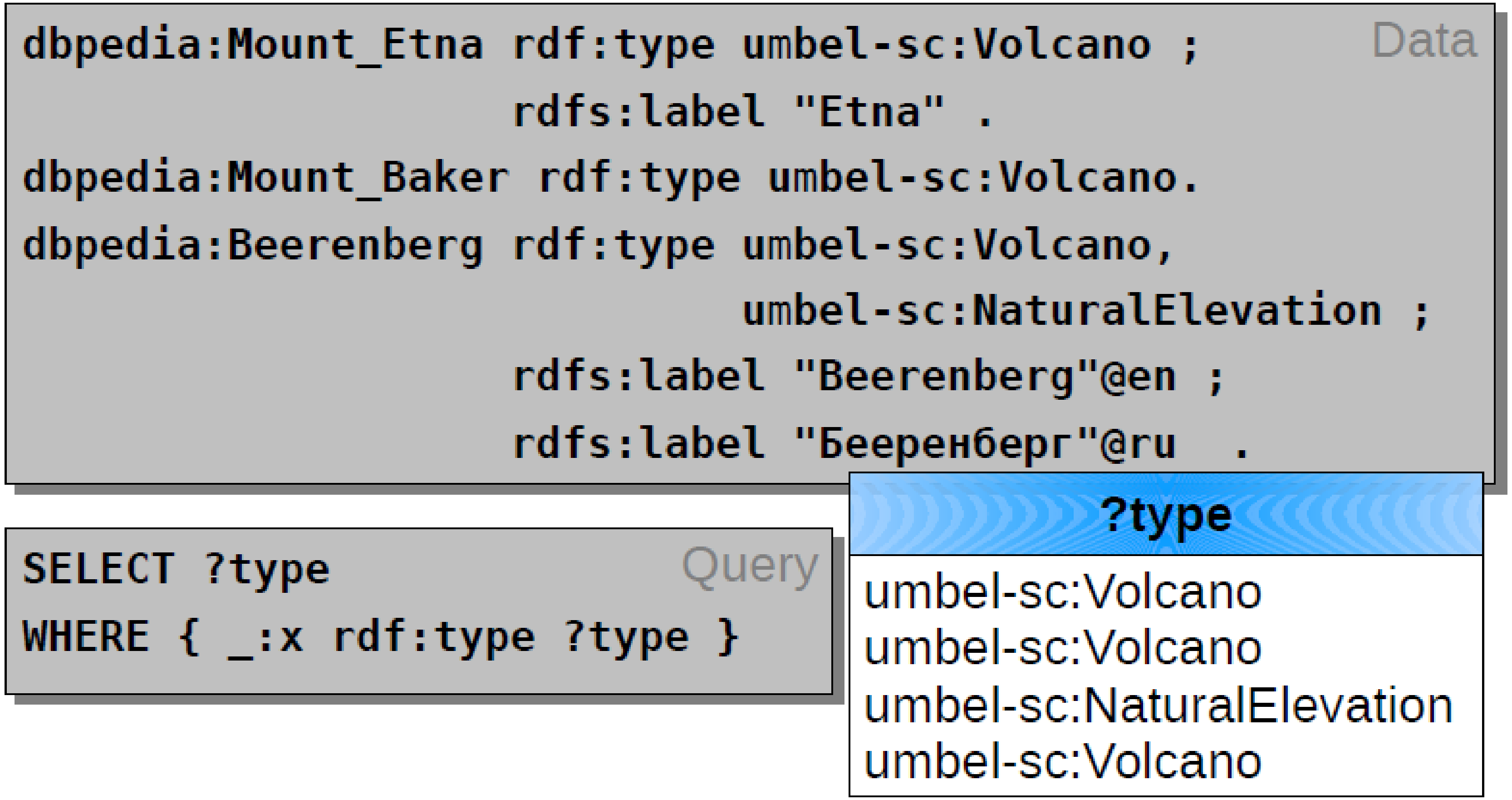
- **Solution modifiers:**
  - Only for SELECT queries
  - Modify the result set as a whole (not single solutions)
  - Keywords: DISTINCT, ORDER BY, LIMIT, and OFFSET

Content

- Main Idea of SPARQL Queries
- Components of SPARQL Queries
- Graph Patterns
- Constraints on Solutions
- Components of SPARQL Queries: Other feautures
- Solution Modifiers
- SPARQL 1.1

Solution Modifiers

- **DISTINCT removes duplicates from the result set**



Data

```
dbpedia:Mount_Etna rdf:type umbel-sc:Volcano ;
                   rdfs:label "Etna" .
dbpedia:Mount_Baker rdf:type umbel-sc:Volcano.
dbpedia:Beerenberg rdf:type umbel-sc:Volcano,
                            umbel-sc:NaturalElevation ;
                   rdfs:label "Beerenberg"@en ;
                   rdfs:label "Беееренберг"@ru  .
```

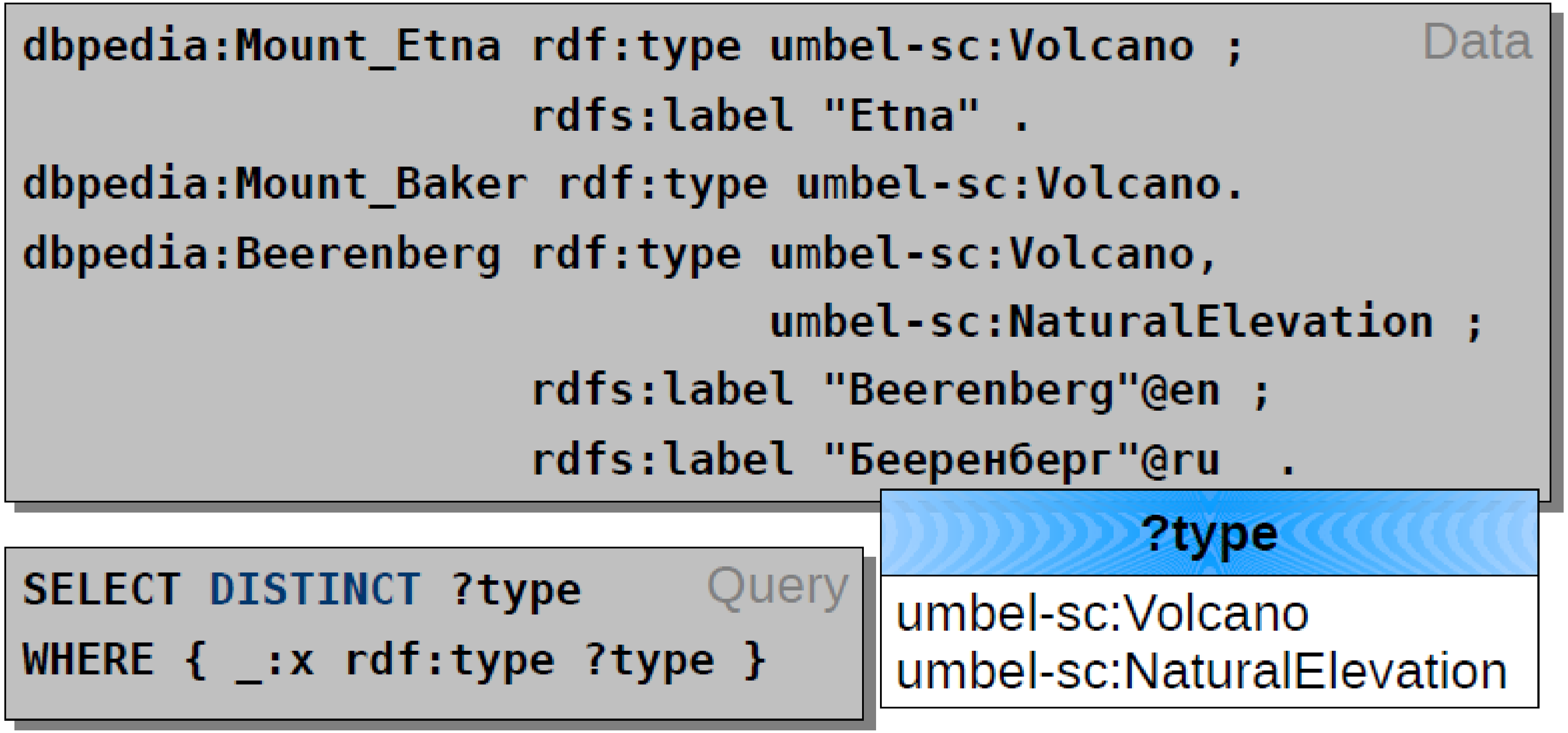
Query

```
SELECT ?type
WHERE { _:x rdf:type ?type }
```

?type
umbel-sc:Volcano
umbel-sc:Volcano
umbel-sc:NaturalElevation
umbel-sc:Volcano

Solution Modifiers

- **DISTINCT removes duplicates from the result set**



Data

```
dbpedia:Mount_Etna rdf:type umbel-sc:Volcano ;
                   rdfs:label "Etna" .
dbpedia:Mount_Baker rdf:type umbel-sc:Volcano.
dbpedia:Beerenberg rdf:type umbel-sc:Volcano,
                            umbel-sc:NaturalElevation ;
                   rdfs:label "Beerenberg"@en ;
                   rdfs:label "Беepeнбepг"@ru  .
```

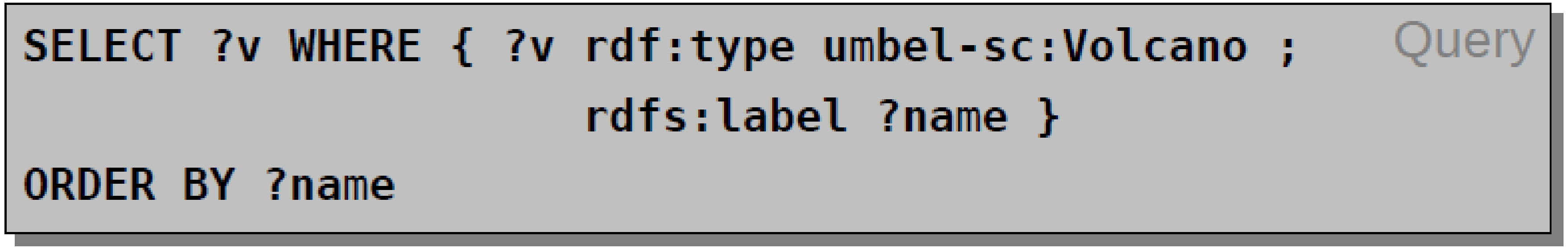
Query

```
SELECT DISTINCT ?type
WHERE { _:x rdf:type ?type }
```

?type
umbel-sc:Volcano
umbel-sc:NaturalElevation

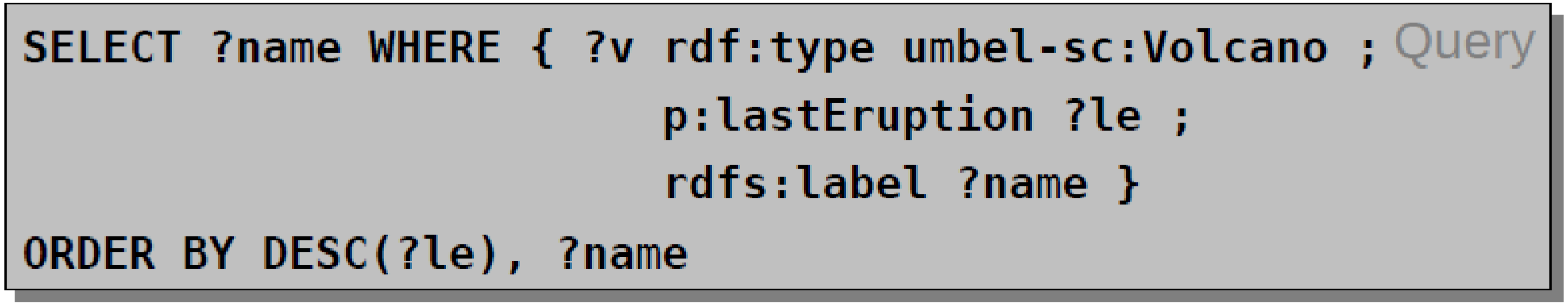
Solution Modifiers

- **ORDER BY orders the results**



```
SELECT ?v WHERE { ?v rdf:type umbel-sc:Volcano ;
                     rdfs:label ?name }
ORDER BY ?name
```

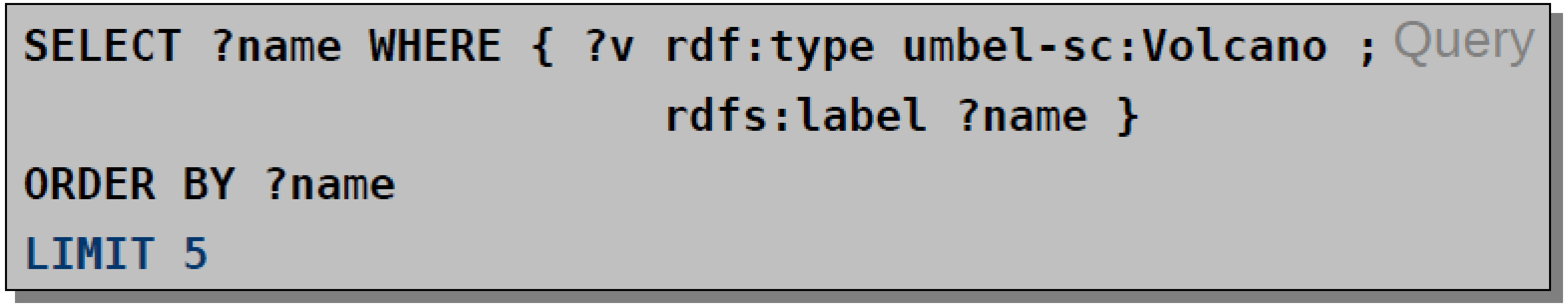
- How do we order different kinds of elements?
  - unbound variable < blank node < URI < literal
  - ASC for ascending (default) and DESC for descending



```
SELECT ?name WHERE { ?v rdf:type umbel-sc:Volcano ;
                        p:lastEruption ?le ;
                        rdfs:label ?name }
ORDER BY DESC(?le), ?name
```

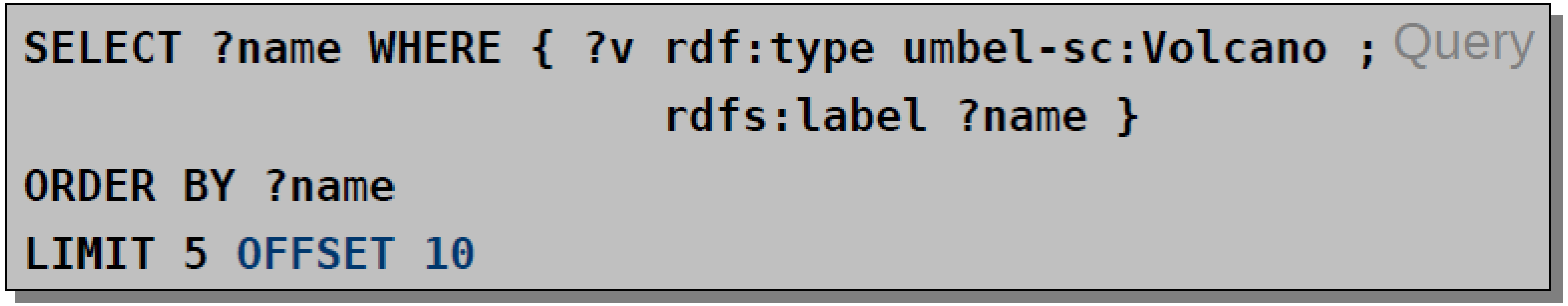
Solution Modifiers

- **LIMIT – limits the number of results**



```
SELECT ?name WHERE { ?v rdf:type umbel-sc:Volcano ;
                        rdfs:label ?name }
ORDER BY ?name
LIMIT 5
```

- **OFFSET – position/index of the first reported results**



```
SELECT ?name WHERE { ?v rdf:type umbel-sc:Volcano ;
                        rdfs:label ?name }
ORDER BY ?name
LIMIT 5 OFFSET 10
```

- **Order of result should be predictable (i.e. combine with ORDER BY)**

Content

- Main Idea of SPARQL Queries
- Components of SPARQL Queries
- Graph Patterns
- Constraints on Solutions
- Components of SPARQL Queries: Other feautures
- Solution Modifiers
- SPARQL 1.1

SPARQL 1.1

- **New features of SPARQL 1.1 Query:**
- Aggregate functions (e.g. COUNT, SUM, AVG)
- Subqueries
- Negation (EXISTS, NOT EXISTS, MINUS)
- Basic query federation (SERVICE, BINDINGS)
- ….

SPARQL 1.1 Update:

- Graph update (INSERT DATA, DELETE DATA, INSERT, DELETE, DELETE WHERE, LOAD, CLEAR)
- Graph management (CREATE, DROP, COPY, MOVE, ADD)

Práctica

Questions?