COMP318: SPARQL

www.csc.liv.ac.uk/~valli/Comp318

Dr Valentina Tamma

Room: Ashton 2.12

Dept of computer science

University of Liverpool

V.Tamma@liverpool.ac.uk

Based on material by O. Hartig and Cambridge Semantics

Where were we

- SPARQL
  - RDF query language
  - `SELECT` Queries
    - `SELECT` … `FROM` … `WHERE`
  - `Basic` and `OPTIONAL` pattern

GRAPH PATTERNS

- Different types of graph patterns for the query pattern (WHERE clause):
    - Basic graph pattern (BGP)
    - **`Group`** graph pattern
    - **`Optional`** graph pattern
    - **`Union`** graph pattern
    - **`Graph`** graph pattern (Constraints)

Example dataset (in Turtle)

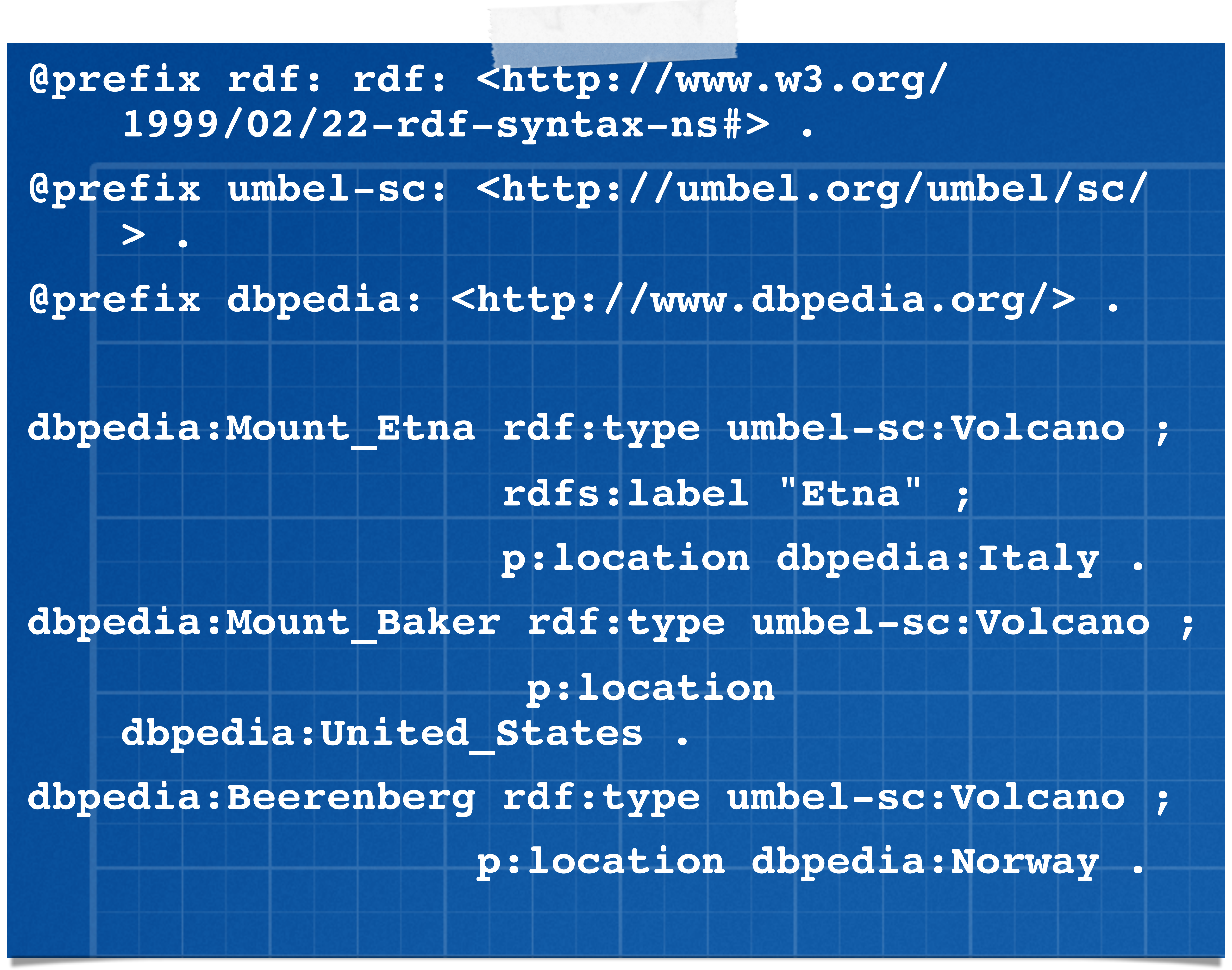
```
@prefix rdf: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
  ns#> .
@prefix umbel-sc: <http://umbel.org/umbel/sc/> .
@prefix dbpedia: <http://www.dbpedia.org/> .

dbpedia:Mount_Etna rdf:type umbel-sc:Volcano ;
                   rdfs:label "Etna" ;
                   p:location dbpedia:Italy .
dbpedia:Mount_Baker rdf:type umbel-sc:Volcano ;
                    p:location dbpedia:United_States .
dbpedia:Beerenberg rdf:type umbel-sc:Volcano ;
                   rdfs:label "Beerenberg"@en ;
                   rdfs:label "Бееренберг"@ru .
                   p:location dbpedia:Norway .
```

UNION Graph patterns

- Union graph patterns allow us to query for possible alternatives

“Which volcanoes are located in Italy or in Norway?”

```
SELECT ?v WHERE
{?v rdf:type umbel-sc:Volcano .
  {?v p:location dbpedia:Italy}
  UNION
  {?v p:location dbpedia:Norway}
}
?v
=================
dbpedia:Mount_Etna
dbpedia:Beerenberg
```



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

dbpedia:Mount_Etna rdf:type umbel-sc:Volcano ;
                   rdfs:label "Etna" ;
                   p:location dbpedia:Italy .
dbpedia:Mount_Baker rdf:type umbel-sc:Volcano ;
                    p:location
    dbpedia:United_States .
dbpedia:Beerenberg rdf:type umbel-sc:Volcano ;
                  p:location dbpedia:Norway .
```

GROUP Graph patterns

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

  {?v p:location dbpedia:Italy}

  UNION

  {?v p:location dbpedia:Norway}

}
```

Semantically equivalent to

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

{ { ?v p:location dbpedia:Italy }

  UNION

  {?v p:location dbpedia:Norway } }

}
```

Constraints: Filters in Query Patterns

- Conditions on literal values with operators and functions

- Different forms
  - Value comparison, e.g., >, !=, >=
  - Numeric functions, e.g., +, *
  - SPARQL test, e.g., `BOUND(?x)`, `isLITERAL(?y)`
  - Negation, e.g., `!BOUND(?x)`

- Syntax: `FILTER` expression

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

SPARQL built-in filter functions

SPARQL		SPARQL 1.1	
Logical	!, &&, ||	Conditionals	IF, COALESCE
Math	+, -, *, /	Constructors	URI, BNODE, STRDT, STRLANG
Comparison	>, <, !=, =, ...	Strings	STRLEN, SUBSTR, UCASE, LCASE, STRSTARTS, STRENDS, CONTAIS, CONCAT, ...

SPARQL built-in filter functions

SPARQL		SPARQL 1.1	
SPARQL Tests	`isURI, isBlank, isLiteral, bound`	More math	`abs, round, ceil, floor, RAND`
SPARQL accessors	`str, lang, datatype`	Sate/Time	`now, year, month, day, hours, minutes, seconds, timezone`
Others	`sameTerm, langMatches, regex`	Hashing	`MD5, SHA1, SHA224, SHA256, SHA384, SHA512`

Solution Modifiers

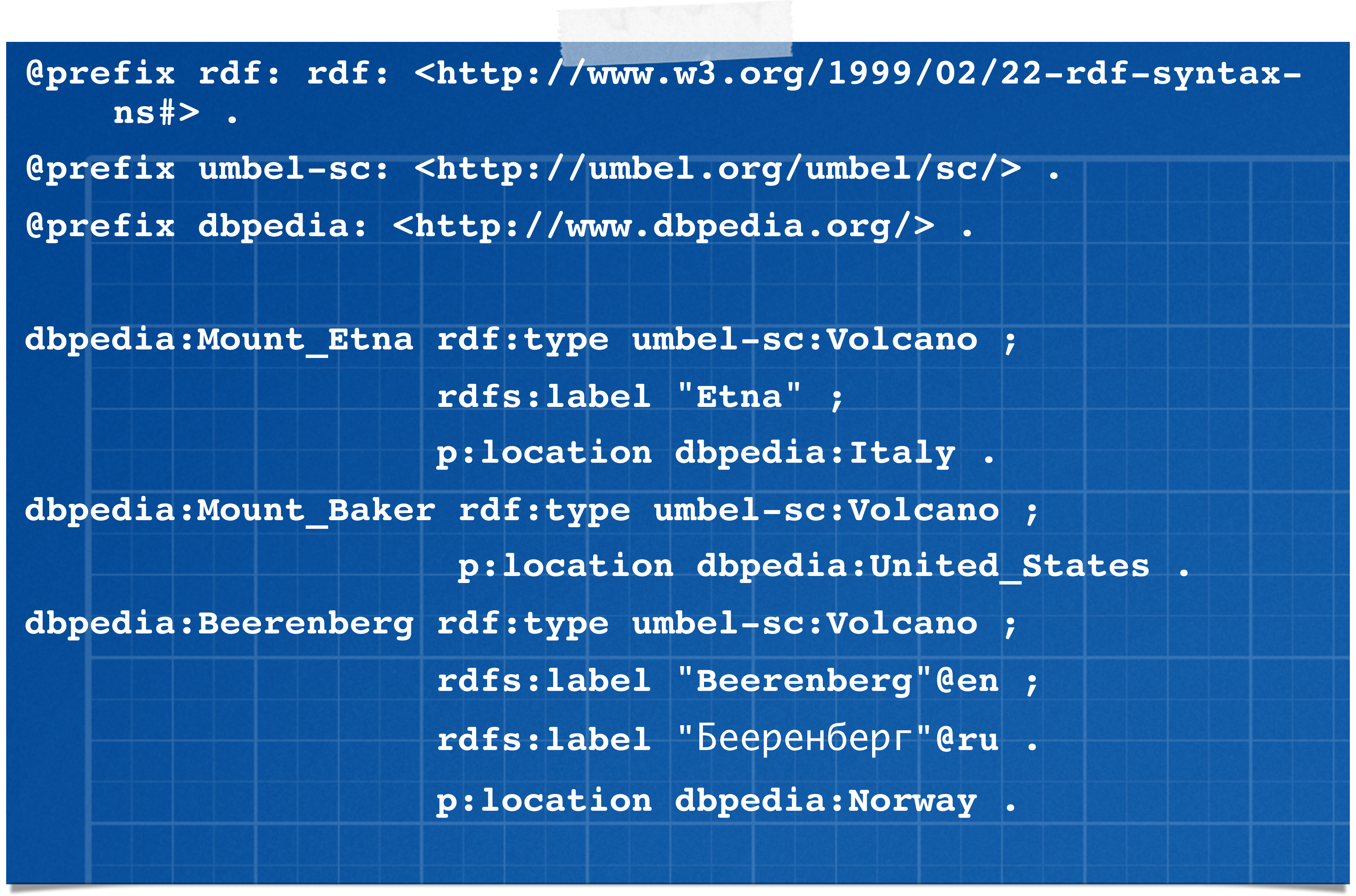
- Modify the result set, but not single results
- Syntax: `ORDER BY`, `LIMIT`, `OFFSET`

NEGATION

“Which volcanoes do not have a name (rdfs:label)?”

```
SELECT ?v WHERE {
  ?v rdf:type umbel-sc:Volcano .
  OPTIONAL { ?v rdfs:label ?name }
  FILTER( ! BOUND(?name) )
}
```

```
?v
=================
dbpedia:Mount_Baker
```



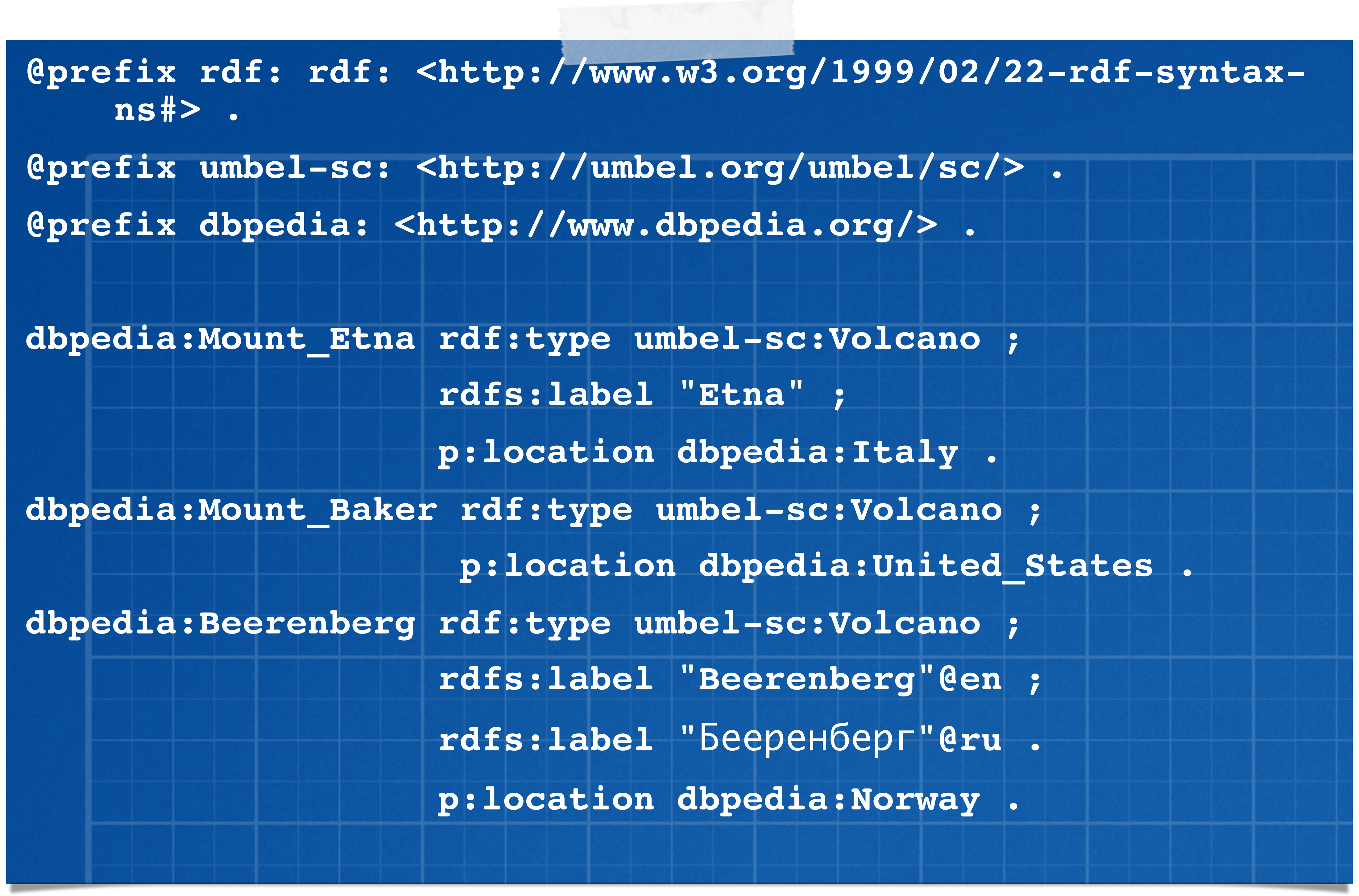
```
@prefix rdf: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
    ns#> .
@prefix umbel-sc: <http://umbel.org/umbel/sc/> .
@prefix dbpedia: <http://www.dbpedia.org/> .

dbpedia:Mount_Etna rdf:type umbel-sc:Volcano ;
                   rdfs:label "Etna" ;
                   p:location dbpedia:Italy .
dbpedia:Mount_Baker rdf:type umbel-sc:Volcano ;
                    p:location dbpedia:United_States .
dbpedia:Beerenberg rdf:type umbel-sc:Volcano ;
                   rdfs:label "Beerenberg"@en ;
                   rdfs:label "Бееренберг"@ru .
                   p:location dbpedia:Norway .
```

NEGATION

“What volcanoes are not called Beerenberg?”

```
SELE      v W      {
 ?v r          umbel-sc:Volcano .
    r       el ?name .
 FI     (      ame) != "Beerenberg")
}
```

```
?v
=================
dbpedia:Mount_Etna
dbpedia:Mount_Baker
dbpedia:Beerenberg
```



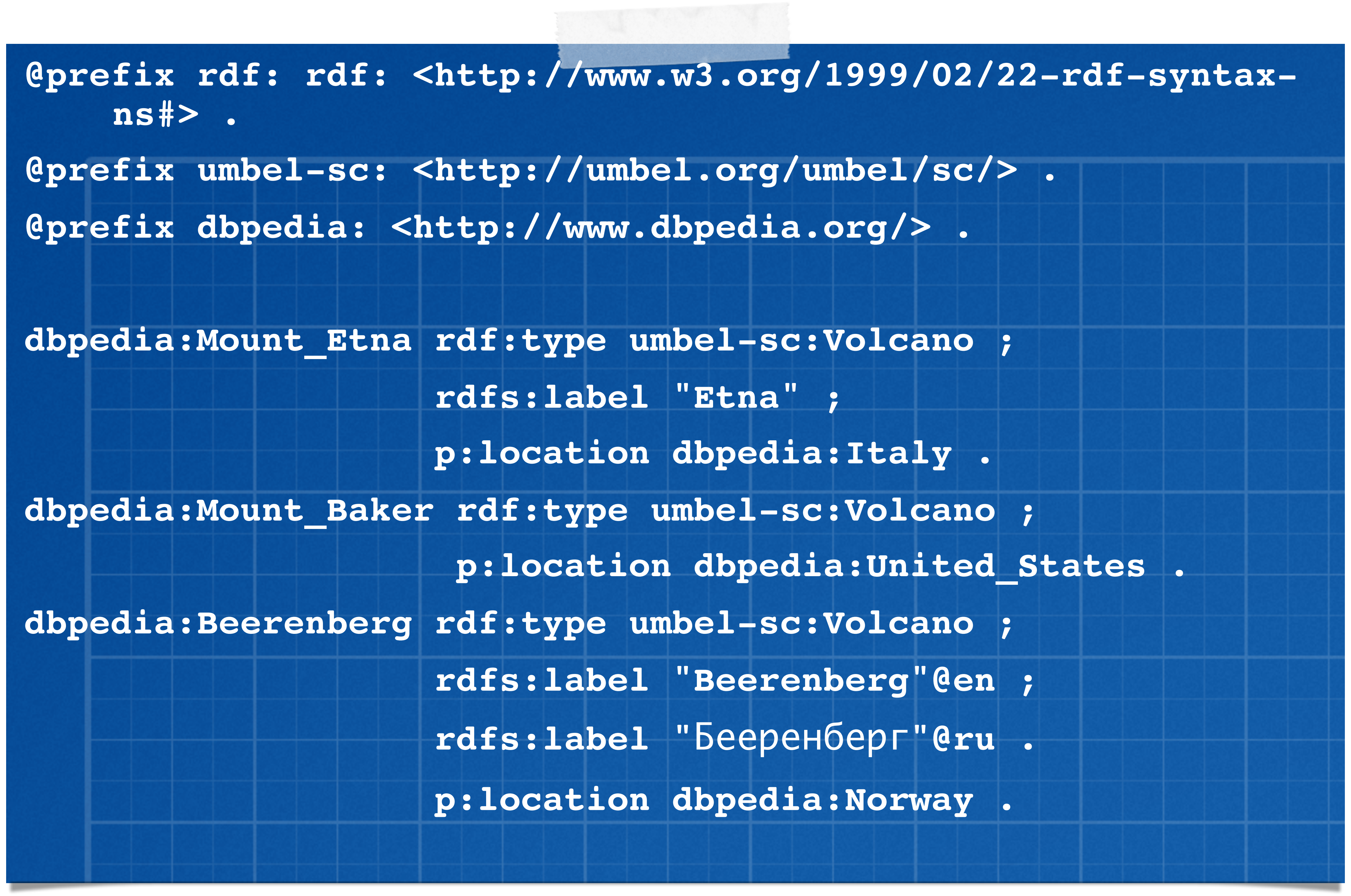
```
@prefix rdf: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
    ns#> .
@prefix umbel-sc: <http://umbel.org/umbel/sc/> .
@prefix dbpedia: <http://www.dbpedia.org/> .

dbpedia:Mount_Etna rdf:type umbel-sc:Volcano ;
                   rdfs:label "Etna" ;
                   p:location dbpedia:Italy .
dbpedia:Mount_Baker rdf:type umbel-sc:Volcano ;
                    p:location dbpedia:United_States .
dbpedia:Beerenberg rdf:type umbel-sc:Volcano ;
                   rdfs:label "Beerenberg"@en ;
                   rdfs:label "Беееренберг"@ru .
                   p:location dbpedia:Norway .
```

NEGATION AS FAILURE

“What volcanoes are not called Beerenberg?”

```
SELECT ?v WHERE {
 ?v rdf:type umbel-sc:Volcano .
 OPTIONAL { ?v rdfs:label ?name .
            FILTER (STR(?name) = "Beerenberg") }
FILTER ( ! BOUND(?name) )
}
```

```
?v
==================
dbpedia:Mount_Etna
dbpedia:Mount_Baker
```

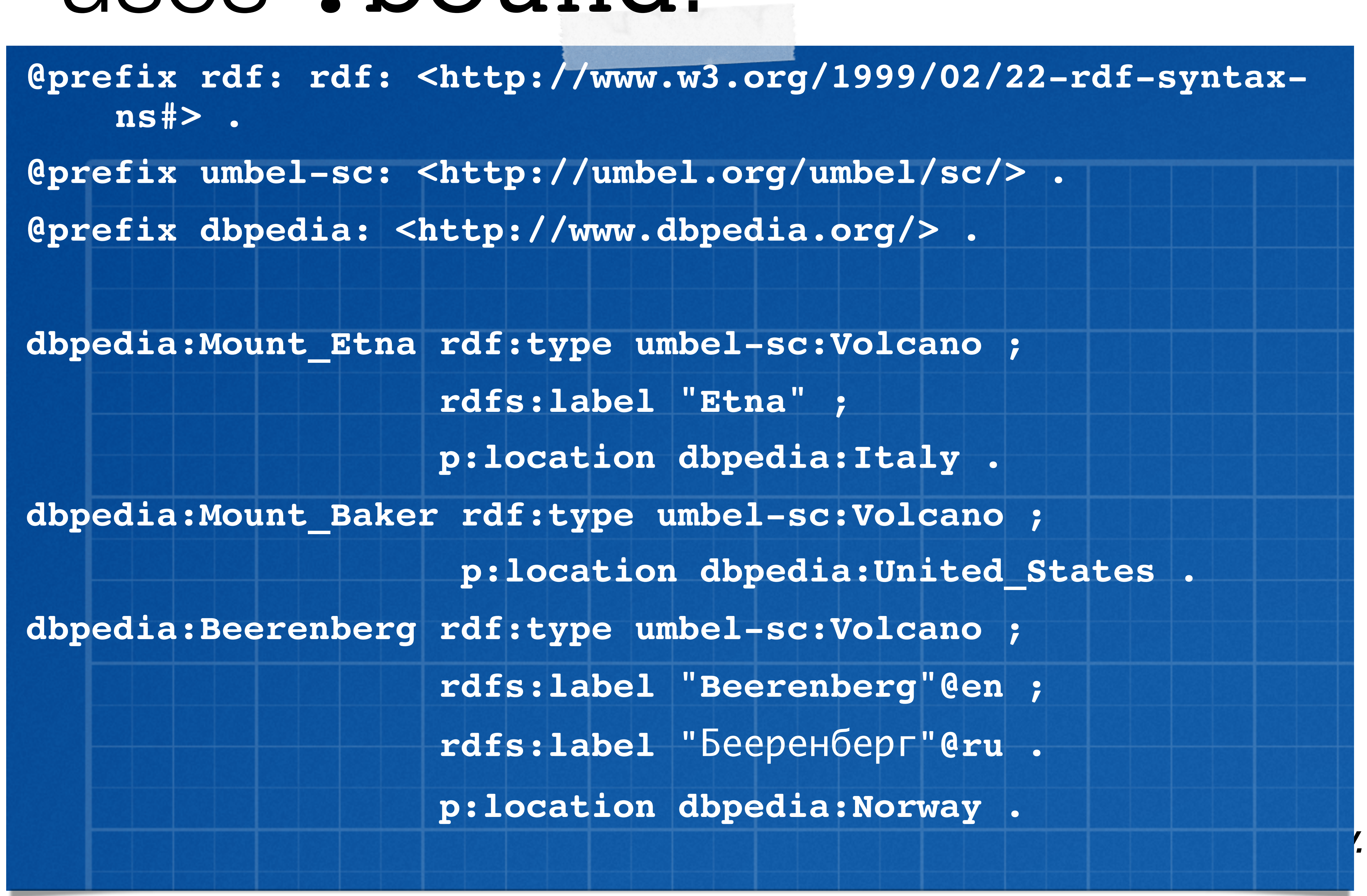


```
@prefix rdf: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-
    ns#> .
@prefix umbel-sc: <http://umbel.org/umbel/sc/> .
@prefix dbpedia: <http://www.dbpedia.org/> .

dbpedia:Mount_Etna rdf:type umbel-sc:Volcano ;
                   rdfs:label "Etna" ;
                   p:location dbpedia:Italy .
dbpedia:Mount_Baker rdf:type umbel-sc:Volcano ;
                    p:location dbpedia:United_States .
dbpedia:Beerenberg rdf:type umbel-sc:Volcano ;
                   rdfs:label "Beerenberg"@en ;
                   rdfs:label "Бееренберг"@ru .
                   p:location dbpedia:Norway .
```

NEGATION AS FAILURE

- The `OPTIONAL` pattern in the previous query does not generate bindings in the following two cases:
  - There is no `rdfs:label` property for `?v`
  - There is an `rdfs:label` property for `?v` but its string value is not Bareenberg

- These two cases are then selected for output by the `FILTER` condition that uses `!bound`.



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

dbpedia:Mount_Etna rdf:type umbel-sc:Volcano ;
                   rdfs:label "Etna" ;
                   p:location dbpedia:Italy .
dbpedia:Mount_Baker rdf:type umbel-sc:Volcano ;
                    p:location dbpedia:United_States .
dbpedia:Beerenberg rdf:type umbel-sc:Volcano ;
                   rdfs:label "Beerenberg"@en ;
                   rdfs:label "Беерeнберг"@ru .
                   p:location dbpedia:Norway .
```

GRAPH Graph patterns

- SPARQL queries are executed against RDF datasets
- RDF datasets are composed of the default graph and zero or more **named graphs**
  - identified by a URI
- **Named graphs**
  - specified through the `FROM NAMED` clause
    - which allows us to scope the query being asked (e.g. to the graphs that comprise an application's user-data storage).
  - or hardwired in a particular endpoint
  - the `GRAPH` keyword allows portions of a query to match against the named graphs in the dataset
    - Anything outside the scope of `GRAPH` clause matches only against the default graph
    - Keyword `GRAPH` makes one of the named graphs the active graph used for pattern matching, if there is no named graph specified in the query it consider a merge of all the named graphs

GRAPH Graph Pattern

```
dbpedia:Mount_Etna rdfs:seeAlso <http://example.org/d1>.
  dbpedia:Mount_Baker rdfs:seeAlso <http://example.org/d2>.
```

```
dbpedia:Mount_Etna rdf:type umbel-sc:Volcano ;
                   rdfs:label "Etna" ;
                   p:location dbpedia:Italy .
```

http://example.org/d1

```
dbpedia:Mount_Baker rdf:type umbel-sc:Volcano ;
                    p:location dbpedia:United_States .
```

http://example.org/d2

```
dbpedia:Beerenberg rdf:type umbel-sc:Volcano ;
                   rdfs:label "Beerenberg"@en ;
                   rdfs:label "Беевенберг"@ru .
                   p:location dbpedia:Norway .
```

http://example.org/d3

GRAPH graph pattern

“Find all the volcanoes and the dataset they are described in”

```
SELECT ?g ?v
 WHERE
 GRAPH ?g {
   ?v rdf:type umbel-sc:Volcano . }
```

result:

```
          ?v                name
============================================
dbpedia:Mount_Etna    http://example.org/d1
dbpedia:Mount_Baker   http://example.org/d2
dbpedia:Beerenberg    http://example.org/d3
```

GRAPH graph pattern

“Find all the volcanoes and the dataset they are described in”

```
SELECT ?g ?v
 FROM NAMED http://example.org/d1
 FROM NAMED http://example.org/d2
 WHERE
 GRAPH ?g {
   ?v rdf:type umbel-sc:Volcano . }

          ?v               name
=============================================
dbpedia:Mount_Etna    http://example.org/d1
dbpedia:Mount_Baker   http://example.org/d2
```

SPARQL

- SPARQL is the query language for querying RDF. It allows users to:
    - Pull values from ***structured*** and ***semi-structured*** data
    - Explore data by querying ***unknown relationships***
    - Perform ***complex joins*** of ***disparate databases*** in a single, simple query
    - ***Transform RDF data*** from one vocabulary to another

Result formats

- The results of SPARQL queries can be returned and/or rendered in a variety of formats:
  - ***XML***. SPARQL specifies an XML vocabulary for returning tables of results.
  - ***JSON***. A JSON "port" of the XML vocabulary, particularly useful for Web applications.
  - ***RDF***. Certain SPARQL result clauses trigger RDF responses, which in turn can be serialized in a number of ways (RDF/XML, N-Triples, Turtle, etc.)
  - ***HTML***. When using an interactive form to work with SPARQL queries.
    - Often implemented by applying an XSL transform to XML results.

Query Result Forms

- `SELECT`: Projection of query result
- `CONSTRUCT`: Returning RDF Graph
- `DESCRIBE`: Returning descriptions of RDF resource
  - not treated here
- `ASK`: “yes/no” query

Reconstructing an RDF Graph: CONSTRUCT

- `CONSTRUCT { basic triple pattern* }`
- Query result is an RDF graph
- Form of RDF Graph described using graph template
  - Construct graph for each pattern solution
  - Triples with unbound variables discarded
  - Illegal RDF triples discarded

CONSTRUCT Query Answers: example

- Graph

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@example.org> .
```

- Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
CONSTRUCT { <http://example.org/person#Alice> vcard:FN ?name }
WHERE { ?x foaf:name ?name }
```

- Result

```
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .
<http://example.org/person#Alice> vcard:FN "Alice" .
```

Boolean Queries: Ask

- `ASK { graph pattern }`
- “Does the query have an answer?”
  - `ASK` replaces `WHERE`
  - Queries without variables are meaningful

ASK Query Answers: example

Graph

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name "Alice" .

_:a foaf:homepage <http://work.example.org/alice/> .

_:b foaf:name "Bob" .

_:b foaf:mbox <mailto:bob@work.example> .
```

Query

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

ASK { ?x foaf:name "Alice" }
```

Result

```
yes
```

Recap

- SPARQL syntax