Using Ontop-spatial

In this page we provide an example on how one can write geospatial mappings and pose GeoSPARQL queries in ontop-spatial.

As an example database, we assume the database creating from importing the OpenStreatMap (OSM) shapefiles of Bremen,as explained here

Creating the example database

Please note that a geospatial database is needed in order to exploit the geospatial features of ontop-spatial. In this example we assume a PostGIS database. The SQL dump file located in this repository (lgd-bremen.sql) can be imported to your PostGIS database using the following command:

```
(sudo -u postgres) psql -d <_databasename_> -f <path-to>lgd-bremen.sql
```

Creating the Mappings

Ontop supports both .obda and R2RML mappings. We will show how one can write geospatial mappings in both of these notations (See lgd-bremen.obda and lgd-bremen.ttl mapping files in the repository). An extract from lgd-bremen.obda is provided as follows:

```
mappingId       lgd_landuse
target          lgd:{gid}  lgd:landUse lgd:{type} . lgd:{gid}  lgd:hasName
{name} . lgd:{gid} geo:asWKT {geom}^^geo:wktLiteral .
source          select  gid, type, geom, name from landuse
```

In the mapping provided above, we want to map the columns *gid*, *type*, *geom* and *name* from the table *landuse*. The column geom contains geometries in a binary format, the Well Known Binary (WKB) format. These geometries will be mapped to the respective GeoSPARQL-compliant literals as specified by the GeoSPARQL vocabulary, i.e., literals of the datatype geosparql:wktLiteral. WKT and GML are the two standard text serializations of geometries specified by the OGC. GeoSPARQL incorporates these standards by defining the respective *geosparql:wktLiteral* and *geosparql:gmlLiteral* datatypes in the GeoSPARQL vocabulary component. Ontop-spatial currently maps geometries that are stored in WKB format to the respective literals of the geosparql:wktLiteral datatype, as shown in the example mapping.

The R2RML notation of the mapping described above is as follows:

```
[ a                     rr:TriplesMap ;
  rr:logicalTable       [ a            rr:R2RMLView ;
                          rr:sqlQuery  "select  gid, type, geom, name from landuse"
                        ] ;
  rr:predicateObjectMap [ a             rr:PredicateObjectMap ;
                          rr:objectMap  [ a            rr:TermMap , rr:ObjectMap ;
                                          rr:datatype  <http://www.opengis.net/ont/geosparql#wktLiteral> ;
                                          rr:template  "{geom}" ;
                                          rr:termType  rr:Literal
                                        ] ;
                          rr:predicate  <http://www.opengis.net/ont/geosparql#asWKT>
                        ] ;
  rr:predicateObjectMap [ a             rr:PredicateObjectMap ;
                          rr:objectMap  [ a            rr:TermMap , rr:ObjectMap ;
                                          rr:column    "name" ;
                                          rr:termType  rr:Literal
                                        ] ;
                          rr:predicate  <http://linkedgeodata.org/ontology#hasName>
                        ] ;
  rr:predicateObjectMap [ a             rr:PredicateObjectMap ;
                          rr:objectMap  [ a            rr:TermMap , rr:ObjectMap ;
                                          rr:template  "http://linkedgeodata.org/ontology#{type}" ;
                                          rr:termType  rr:IRI
                                        ] ;
                          rr:predicate  <http://linkedgeodata.org/ontology#landUse>
                        ] ;
  rr:subjectMap         [ a            rr:SubjectMap , rr:TermMap ;
                          rr:template  "http://linkedgeodata.org/ontology#{gid}" ;
                          rr:termType  rr:IRI
                        ]
] .
```

Example of geospatial R2RML mappings

ATTENTION: Ontop-spatial assumes that all geometries in the database are expressed in the Universal Coordinate System (with code: 4326). The geometries that are expressed in a different Coordinate Reference System, should be tranformed first (Or another geometry column can be added with the geometries expressed in 4326). Geometries can be transformed using the PostGIS function ST_Transform. For more information, please visit the PostGIS reference page : http://postgis.net/docs/reference.html

Posing GeoSPARQL queries

Now we are ready to query our database using the OGC standard GeoSPARQL, a geospatial extension of the query language SPARQL. For example, let us suppose that we want to retrieve roads that intersect with ports and project also the geometry and the type of these roads. Such a query would be expressed in GeoSPARQL as follows:

```
        select distinct ?roadtype ?geo1
            where {
            ?x lgd:landUse lgd:port .
            ?x geo:asWKT ?geo .
            ?x1 geo:asWKT ?geo1 .
            ?x1 lgd:roadType ?roadtype .

FILTER(<http://www.opengis.net/def/function/geosparql/sfIntersects>(?geo,?geo
1))}
```

The query described above is a GeoSPARQL query that retrieves ports whose geometries intersect with the geometries of roads and projects the geometries and the types of these roads. Please pay attention to the use of the respective triple patterns that retrieve the geometries in order to be used as arguments in the GeoSPARQL filter function that checks if the topology relation "sf-intersects" holds between these geometries. Another way to express this query is the following:

```
select distinct ?roadtype ?geo1
    where {
    ?x lgd:landUse lgd:port .
    ?x1 geo:sfIntersects ?x .
    ?x1 lgd:roadType ?roadtype .
    ?x1 geo:sfIntersects ?x .}
```

The query described above uses the triple pattern ?x1 geo:sfIntersects ?x instead of using the respective filter function that is described above. Internally, this query and the previous one get translated into the same SQL query that gets evaluated in the DBMS, as Ontop-spatial implements the query rewrite extension, transforming qualitative geospatial queries (like this one) into their quantitative counterparts (like the previous one). Please note though, that the mapping file **must not be changed** in either of the two versions of the query.