

# Semantic GIS, GeoSPARQL

Michal Med

January 2, 2020

## 1 GraphDB

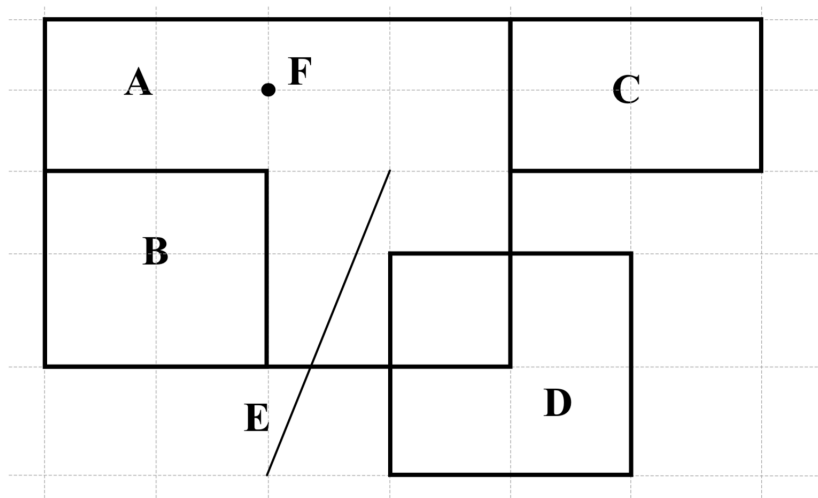
GeoSPARQL has full support in GraphDB.

### 1.1 Get ready

Download and import data into Graph DB – [http://graphdb.ontotext.com/documentation/8.4/free/\\_downloads/geosparql-example.rdf](http://graphdb.ontotext.com/documentation/8.4/free/_downloads/geosparql-example.rdf) and [http://graphdb.ontotext.com/documentation/8.4/free/\\_downloads/geosparql-simple-f.rdf](http://graphdb.ontotext.com/documentation/8.4/free/_downloads/geosparql-simple-f.rdf) and enable GeoSPARQL.

```
PREFIX : <http://www.ontotext.com/plugins/geosparql#>
```

```
INSERT DATA {  
  _:s :enabled "true" .  
}
```



## 1.2 Spatial filters

Select:

- all features that touches each other,
- all features overlapping rectangle D,
- closest feature to the point

```
"<http://www.opengis.net/def/crs/OGC/1.3/CRS84>  
Point(-83.1 34.4)"^^geo:wktLiteral
```

### 1.2.1 Solution

All features that touches each other:

```
PREFIX my: <http://example.org/ApplicationSchema#>  
PREFIX geo: <http://www.opengis.net/ont/geosparql#>  
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>  
  
SELECT ?f  
WHERE {  
  ?a my:hasExactGeometry ?aGeom .  
  ?aGeom geo:asWKT ?aWKT .  
  ?f my:hasExactGeometry ?fGeom .  
  ?fGeom geo:asWKT ?fWKT .  
  FILTER (geof:sfTouches(?aWKT, ?fWKT) && !sameTerm(?aGeom, ?fGeom))  
}
```

All features overlapping rectangle D:

```
PREFIX my: <http://example.org/ApplicationSchema#>  
PREFIX geo: <http://www.opengis.net/ont/geosparql#>  
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>  
  
SELECT ?f  
WHERE {  
  my:D my:hasExactGeometry ?dGeom .  
  ?dGeom geo:asWKT ?dWKT .  
  ?f my:hasExactGeometry ?fGeom .  
  ?fGeom geo:asWKT ?fWKT .  
  FILTER (geof:sfOverlaps(?dWKT, ?fWKT))  
}
```

Closest feature to the given point:

```
PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>
PREFIX my: <http://example.org/ApplicationSchema#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>

SELECT ?f ?dist
WHERE {
  ?f my:hasExactGeometry ?fGeom .
  ?fGeom geo:asWKT ?fWKT .
  BIND((geof:distance(?fWKT, '''
    <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
    Point(-83.1 34.4)''')^geo:wktLiteral, uom:metre)) as ?dist)
} ORDER BY ASC(?dist) LIMIT 1
```

## 1.3 Spatial queries

Try following tasks:

- Create feature AC as a union of features A and C,
- find all features closer than 10 kilometers to line E.

### 1.3.1 Solution

Create feature AC as a union of features A and C (query takes little longer):

```
PREFIX my: <http://example.org/ApplicationSchema#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>

SELECT ?AC
WHERE {
  my:A my:hasExactGeometry ?aGeom .
  ?aGeom geo:asWKT ?aWKT .
  my:C my:hasExactGeometry ?cGeom .
  ?cGeom geo:asWKT ?cWKT .
  BIND((geof:union(?aWKT, ?cWKT)) as ?AC)
}
```

Find all features closer than 10 kilometers to line E:

```
PREFIX uom: <http://www.opengis.net/def/uom/OGC/1.0/>
PREFIX my: <http://example.org/ApplicationSchema#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>

SELECT ?f ?distance
WHERE {
  my:E my:hasExactGeometry ?eGeom .
  ?eGeom geo:asWKT ?eWKT .
  ?f my:hasExactGeometry ?fGeom .
  ?fGeom geo:asWKT ?fWKT .
  FILTER (?fGeom != ?eGeom)
  BIND((geof:distance(?eWKT, ?fWKT, uom:metre)) as ?distance)
  FILTER (?distance < 10000 && (?fGeom != ?eGeom))
} ORDER BY ASC(?distance)
```