GeoSPARQL
Spatial operations and functions

Michal Med

michal.med@fel.cvut.cz

January 3, 2019

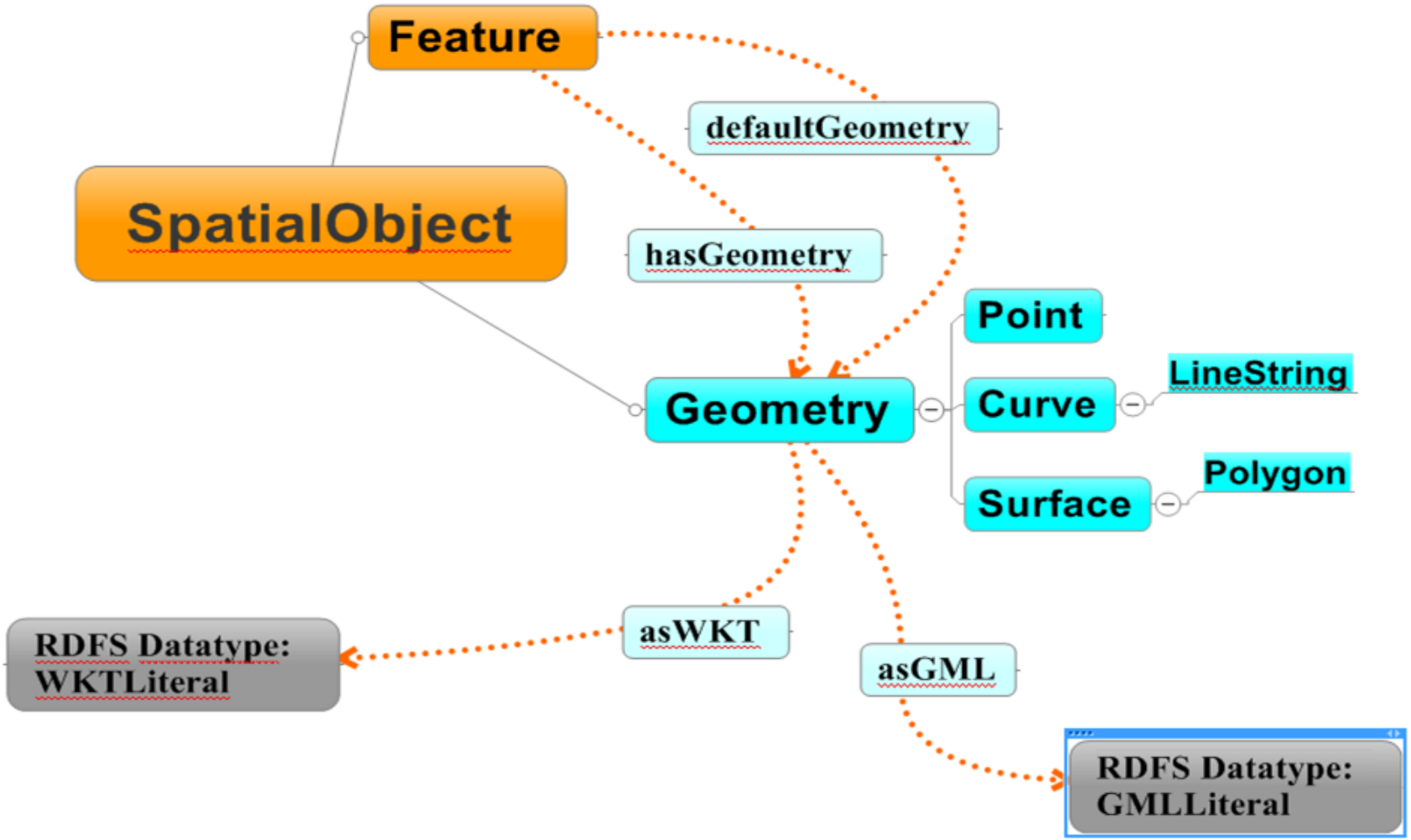
Outline

Linked Geo Data

How is it usually done?

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#">
  <geo:Point>
    <geo:lat>49.701</geo:lat>
    <geo:long>14.552</geo:long>
  </geo:Point>
</rdf:RDF>
```

GeoSPARQL ontology



GeoSPARQL representation

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix geosparql: <http://www.opengis.net/ont/geosparql#> .
@prefix ds-par: <http://onto.fel.cvut.cz/ontologies/town-plan/parcely/> .
@prefix databaseTableParcely: <http://onto.fel.cvut.cz/ontologies/town-plan/databaseTableParcely/>
@prefix par-geometry: <http://onto.fel.cvut.cz/ontologies/town-plan/parcelakn_dokm_p/geometry/>
@prefix townplan: <http://onto.fel.cvut.cz/ontologies/town-plan/>

townplan:parcelakn_dokm_p/1/2018-01-29T14:36:24.178617 a ds-par:Parcely,
        geosparql:Feature ;
    rdfs:label <parcelakn_dokm_p/1/2018-01-29T14:36:24.178617> ;
    databaseTableParcely:dat_vznik "2008-09-25"^^xsd:date ;
    databaseTableParcely:existujedi "A" ;
    databaseTableParcely:id 2087553101.0 ;
    databaseTableParcely:id_poskyt 397 ;
    databaseTableParcely:katuze_kod 727164 ;
    databaseTableParcely:nazev_ku "Vinohrady" ;
    databaseTableParcely:ogc_fid 1 ;
    databaseTableParcely:par_id 2087553101.0 ;
    databaseTableParcely:parcela "1057" ;
    databaseTableParcely:shape_area 260.475900002 ;
    databaseTableParcely:shape_length 65.6304823872 ;
    databaseTableParcely:tid_parcelakn_dokm_p 61534.0 ;
    databaseTableParcely:vymera 260 ;
    geosparql:hasGeometry par-geometry:1/2018-01-29T14:36:24.178617 .

    par-geometry:1/2018-01-29T14:36:24.178617 a geosparql:Geometry ;
    rdfs:label <parcelakn_dokm_p/geometry/1/2018-01-29T14:36:24.178617> ;
    geosparql:asWKT "MULTIPOLYGON(((-742241.02 -1045480.81,-742242.84 -1045482.35,
                    -742257.059 -1045469.76,-742246.0798 -1045456.9,-742237.98
                    -1045465.82,-742241.02 -1045480.81)))" .
```

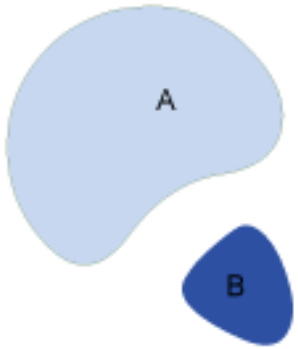
GeoSPARQL querying

```
geo:<http://www.opengis.net/ont/geosparql#>
geof:<http://www.opengis.net/def/function/geosparql/>
```

GeoSPARQL querying

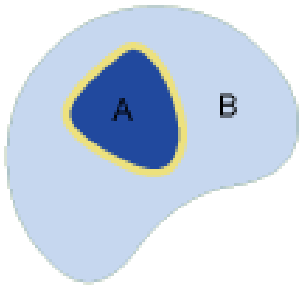
```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX vocab-vyuziti: <http://onto.fel.cvut.cz/ontologies/town-plan/
                       resource/vocab/urk_ss_vyuzitizakl_p/>
PREFIX vocab-fvu: <http://onto.fel.cvut.cz/ontologies/town-plan/
                   resource/vocab/pvp_fvu_p/>

SELECT ?var1 ?var2
WHERE  {
        ?var1 vocab-fvu:wkb_geometry ?geometry1.
        ?var2 vocab-vyuziti:wkb_geometry ?geometry2.
        ?geometry1 geof:intersects ?geometry2
}
```

Topological relations

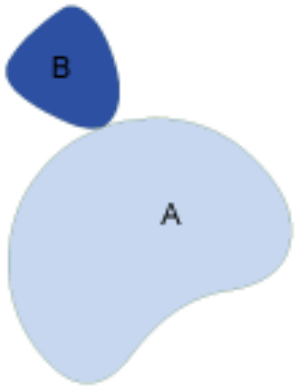


disjoint

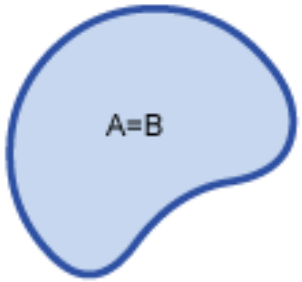
intersects (¬disjoint)



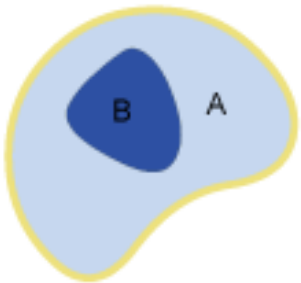
within



touches



equals



contains

GeoSPARQL filter functions

ogc:relate

Returns true if two objects are spatial related.

```
ogcf:relate
(geom1: ogc:GeomLiteral, geom2: ogc:GeomLiteral,
relation: xsd:anyURI): xsd:boolean
```

ogc:distance

Returns distance in given units between two objects.

```
ogcf:distance
(geom1: ogc:GeomLiteral, geom2: ogc:GeomLiteral,
units: xsd:anyURI): xsd:double
```

GeoSPARQL filter functions

ogc:buffer

Returns geometric object representing all points whose distance from geom1 is within radius in units.

```
ogcf:buffer
(geom: ogc:GeomLiteral, radius: xsd:real,
units: xsd:anyURI): ogc:GeomLiteral
```

ogcf:convexHull

Returns geometric object representing all points in convex hull of geom1.

```
ogcf:convexHull
(geom1: ogc:GeomLiteral): ogc:GeomLiteral
```

GeoSPARQL filter functions

ogcf:intersection

Returns a geometric object that represents all Points in the intersection of geom1 with geom2.

```
ogcf:intersection
(geom1: ogc:GeomLiteral, geom2: ogc:GeomLiteral,
): ogc:GeomLiteral
```

ogcf:union

Returns a geometric object that represents all Points in the union of geom1 with geom2.

```
ogcf:union
(geom1: ogc:GeomLiteral, geom2: ogc:GeomLiteral,
): ogc:GeomLiteral
```

GeoSPARQL filter functions

ogcf:difference,

ogcf:symDifference,

ogcf:envelope,

ogcf:boundary.

GeoSPARQL documentation

GeoSPARQL filter example

```
SELECT ?f
WHERE {
     my:A my:hasExactGeometry ?aGeom .
     ?aGeom ogc:asWKT ?aWKT .
     ?f my:hasExactGeometry ?fGeom .
     ?fGeom ogc:asWKT ?fWKT .
     FILTER (ogcf:relate(?aWKT, ?fWKT, ogc:contains))
}
```

Tutorial

Following page contains full useful tutorial in GraphDB including data.
http://graphdb.ontotext.com/documentation/standard/geosparql-support.html