

# Parliament und GeoSPARQL

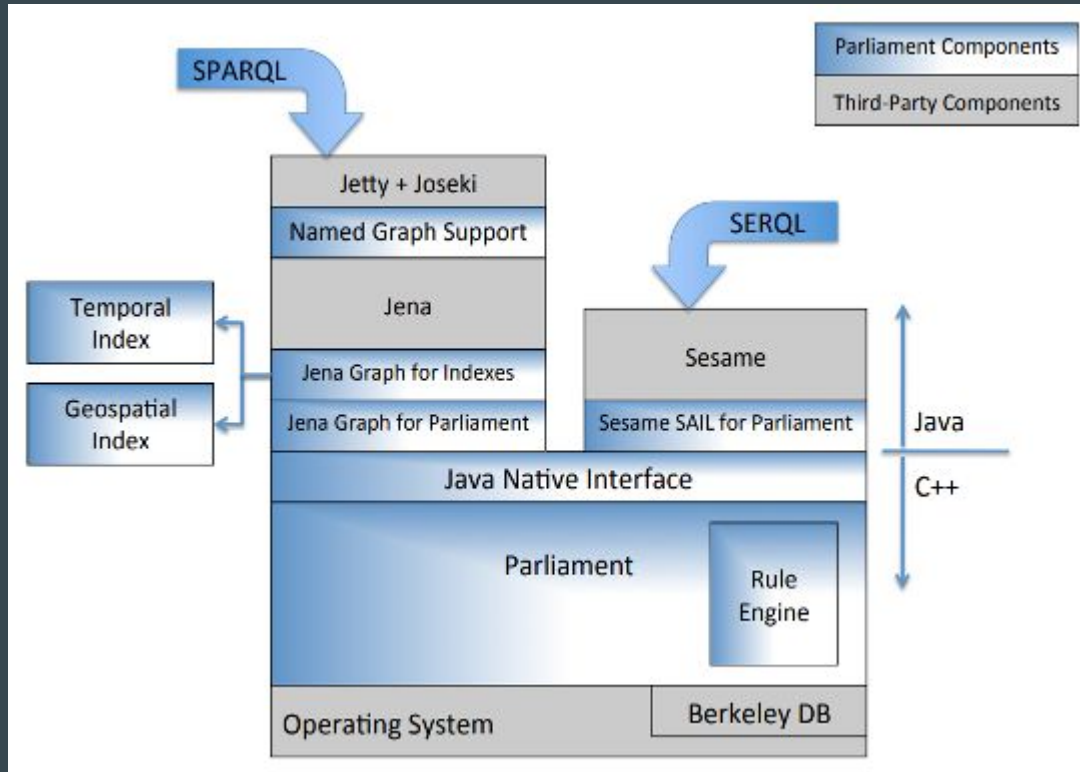
...

Elias Kechter

# Parliament

- Triple Store + Rule Engine von Raytheon BBN Technologies
- als "DAML\_DB" seit 2001 in Entwicklung
- ab 2009 als "Parliament" unter BSD Lizenz
- Triple Store Engine kein komplettes DMS, baut aber auf Drittanbieter
- z.B. Sesame oder Jena, dadurch RDF, OWL und SPARQL Standards
- Rule Engine implementiert alle RDFS Inferenzstandards und einige OWL-Lite

# Parliament



# Parliament

Speicherstruktur besteht aus 3 Komponenten:

- Ressourcen Tabelle
- Statement Tabelle
- Ressourcen Wörterbuch

# Parliament

## Ressourcen Tabelle:

- Datei mit Einträgen die eine fixe Länge haben, sequenziell durchnummeriert
- Jeder Eintrag hat 8 Komponenten:
  - 3 Statement ID Felder; zeigen auf erstes Statement als S, P bzw. O nutzt
  - 3 Zählerfelder; beinhalten die Anzahl der Statements als S, P bzw. O nutzen
  - Offset im Ressourcen Wörterbuch; dadurch bekommt man die String-Darstellung
  - Bit Feld; kodiert etwaige Attribute dieser Ressource

# Parliament

## Statement Tabelle:

- Datei mit Einträgen die eine fixe Länge haben, sequenziell durchnummeriert
- Jeder Eintrag hat 7 Komponenten:
  - 3 Ressourcen ID Felder; zeigen auf Ressource des S, P bzw. O
  - 3 Statement ID Felder; zeigen auf das nächste Statement, welches als S,P bzw. O nutzt
  - Bit Feld; kodiert etwaige Attribute dieses Statements

# Parliament

Ressourcen Wörterbuch:

- Bidirektionales 1-zu-1 Mapping von Ressourcen Literal  $\Leftrightarrow$  Ressourcen ID
- Bsp: “<http://www.example.org/POI#Restaurant>”  $\Leftrightarrow$  1

# Parliament

	S	P	O
1	1	2	3
2	1	4	5
3	1	4	6
4	3	2	7
5	3	4	6

The diagram shows a sequence of statements 1 through 5. Arrows indicate dependencies between columns S, P, and O. Statement 1 has arrows pointing to S=1, P=2, and O=3. Statement 2 has arrows pointing to S=1, P=4, and O=5. Statement 3 has arrows pointing to S=1, P=4, and O=6. Statement 4 has arrows pointing to S=3, P=2, and O=7. Statement 5 has arrows pointing to S=3, P=4, and O=6. There are also curved arrows indicating a sequence: from statement 1 to 2, 2 to 3, 3 to 4, and 4 to 5.

Statement  
List

	FS	FP	FO	SC	PC	OC
1	3	-	-	3	0	0
2	-	4	-	0	2	0
3	5	-	1	2	0	1
4	-	5	-	0	3	0
5	-	-	2	0	0	1
6	-	-	5	0	0	2
7	-	-	4	0	0	1

Resource Table



# Parliament

	S	P	O
1	1	2	3
2	1	4	5
3	1	4	6
4	3	2	7
5	3	4	6
6	8	2	5

Statement  
List

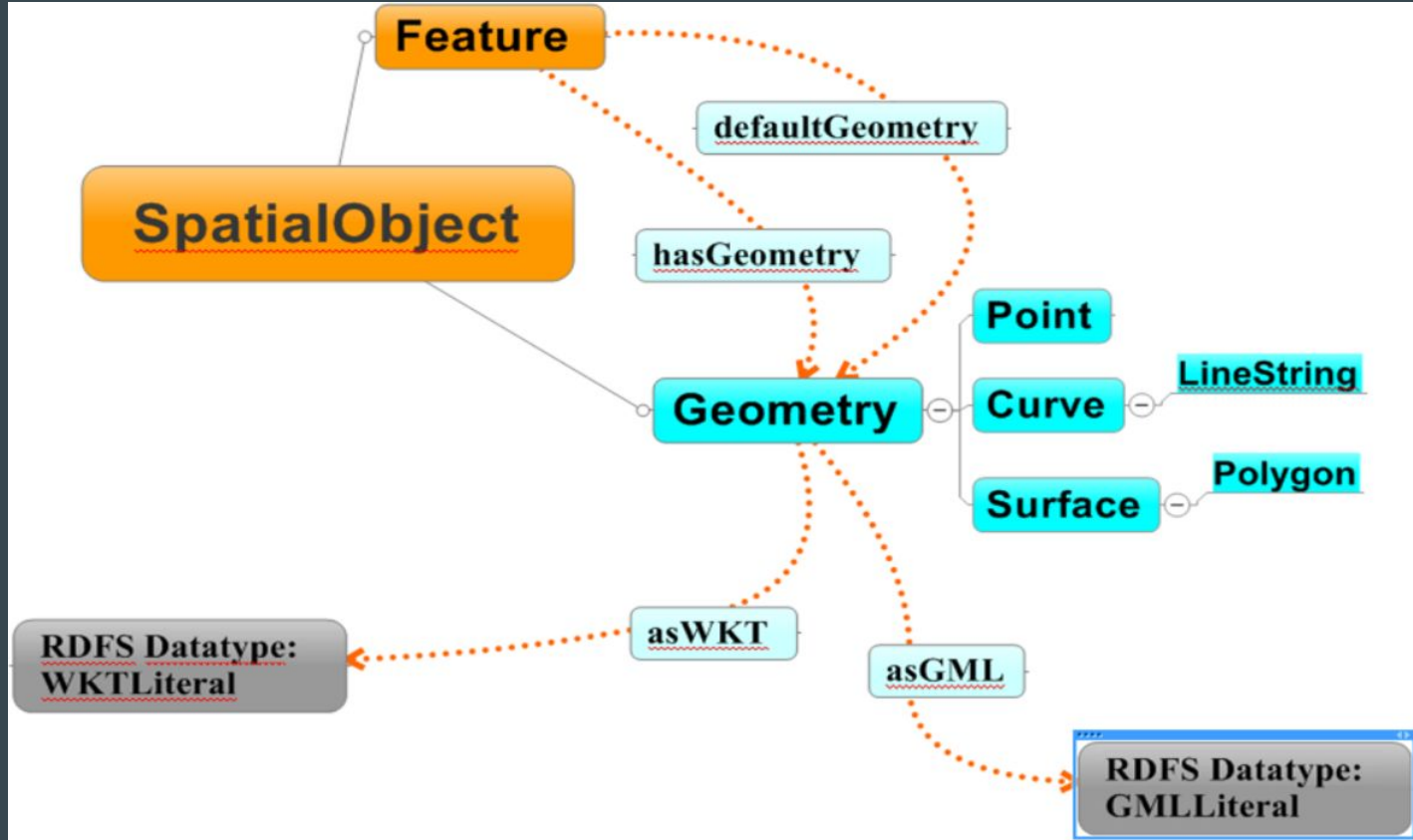
	FS	FP	FO	SC	PC	OC
1	3	-	-	3	0	0
2	-	<del>4</del> 6	-	0	<del>2</del> 3	0
3	5	-	1	2	0	1
4	-	5	-	0	3	0
5	-	-	<del>2</del> 6	0	0	<del>1</del> 2
6	-	-	5	0	0	2
7	-	-	4	0	0	1
8	6	-	-	1	0	0

Resource Table

# GeoSPARQL

- Standard zum Repräsentieren und Erfragen von räumlich bezogenen verlinkten Daten im semantischen Web vom OGC herausgegeben
- Der Standard bietet:
  - kleine topologische Ontologie in RDFS/OWL zum Repräsentieren; dazu benutzt es Geography Markup Language (GML) und Well-Known Text (WKT) Literale
  - Simple Features, RCC8 und Egenhofer topologische Beziehungsbeschreibungen für das qualitative Erfragen
  - SPARQL Interface welches einen Satz an topologischen SPARQL Erweiterungsfunktionen benutzt für das quantitative Erfragen

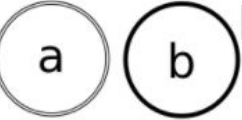
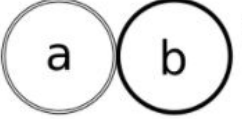
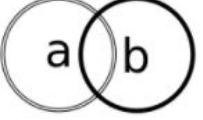
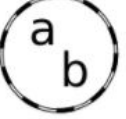

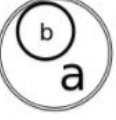


# GeoSPARQL



# GeoSPARQL

Simple Features	Egenhofer	RCC8
equals	equal	EQ
disjoint	disjoint	DC
intersects	$\neg$ disjoint	$\neg$ DC
touches	meet	EC
within	inside+coveredBy	NTPP+TPP
contains	contains+covers	NTPPi+TPPi
overlaps	overlap	PO

# GeoSPARQL

		Region Connection Calculus 8			
		RCC and OGC relationships			
		RCC8 property	RCC8 OGC relation URI	OGC property	OGC property URI
DC(a,b)					
EC(a,b)		DC	geo:rcc8-dc	disjoint	geo:sf-disjoint
		EC	geo:rcc8-ec	touches	geo:sf-touches
		PO	geo:rcc8-po	overlaps	geo:sf-overlaps
PO(a,b)		EQ	geo:rcc8-eq	equals	geo:sf-equals
		TPP	geo:rcc8-tpp	within	geo:sf-within
		TPPi	geo:rcc8-tppi	contains	geo:sf-contains
		nTPP	geo:rcc8-ntpp	within	geo:sf-within
		nTPPi	geo:rcc8-ntppi	contains	geo:sf-contains
EQ(a,b)		*	*	intersects	geo:sf-intersects
* logically represented as $\neg DC$ (the formal way of writing not(DC))					
TPP(a,b)		TPPi(b,a)		DC: Disconnected	
				EC: Externally Connected	
				PO: Partially Overlapping	
				EQ: Equal	
				TPP: Tangential Proper Part	
				TPPi: Tangential Proper Part inverse	
nTPP(a,b)		nTPPi(b,a)		nTPP: non-Tangential Proper Part	
				nTPPi: non-Tangential Proper Part inverse	

# GeoSPARQL

The following example SPARQL query could help model the question "What is within the bounding box defined by [38.913574°N 77.089005°W](#) and [38.886321°N 77.029953°W](#)?"<sup>[6]</sup>

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

SELECT ?what
WHERE {
  ?what geo:hasGeometry ?geometry .

  FILTER(geof:sfWithin(?geometry,
    "POLYGON((
-77.089005 38.913574,
-77.029953 38.913574,
-77.029953 38.886321,
-77.089005 38.886321,
-77.089005 38.913574
))"^^geo:wktLiteral))
}
```

# Parliament + GeoSPARQL

- Parliament unterstützt GeoSPARQL
- Quickstart:
  - .zip runterladen von [http://semwebcentral.org/frs/?group\\_id=159](http://semwebcentral.org/frs/?group_id=159)
  - in einen Ordner mit dem Namen “ParliamentKB” entpacken
  - “/ParliamentKB/startParliament.(bat / sh)” starten
  - über Webbrowser auf “localhost:8080/parliament” zugreifen

# Parliament + GeoSPARQL

The screenshot shows a web browser window with the title 'Parliament Query Server'. The address bar shows 'localhost:8080/parliament/'. The page has a light blue header with the title 'Parliament Query Server'. Below the header, there are three main sections: 'Operations', 'Java Memory Usage (KB)', and 'Graphs'. The 'Operations' section contains a list of links: Query, Explore, SPARQL/Update, Insert Data, Export, Indexes, and Admin. The 'Java Memory Usage (KB)' section contains a table with columns 'Type', 'Heap', and 'Non-Heap'. The 'Graphs' section contains a link 'Default Graph' and a URL 'http://parliament.semwebcentral.org/parliament#MasterGraph'. At the bottom, there is a section titled 'SPARQL is defined by 3 documents:' with a list of links: SPARQL query language, SPARQL protocol, and SPARQL XML results format.

Parliament Query Server

## Operations

- [Query](#)
- [Explore](#)
- [SPARQL/Update](#)
- [Insert Data](#)
- [Export](#)
- [Indexes](#)
- [Admin](#)

## Java Memory Usage (KB)

Type	Heap	Non-Heap
Used	114535	57933
Max	466432	0

## Graphs

[Default Graph](#)

<http://parliament.semwebcentral.org/parliament#MasterGraph>

SPARQL is defined by 3 documents:

- [SPARQL query language](#)
- [SPARQL protocol](#)
- [SPARQL XML results format](#)



# Parliament + GeoSPARQL

**Text Insert**

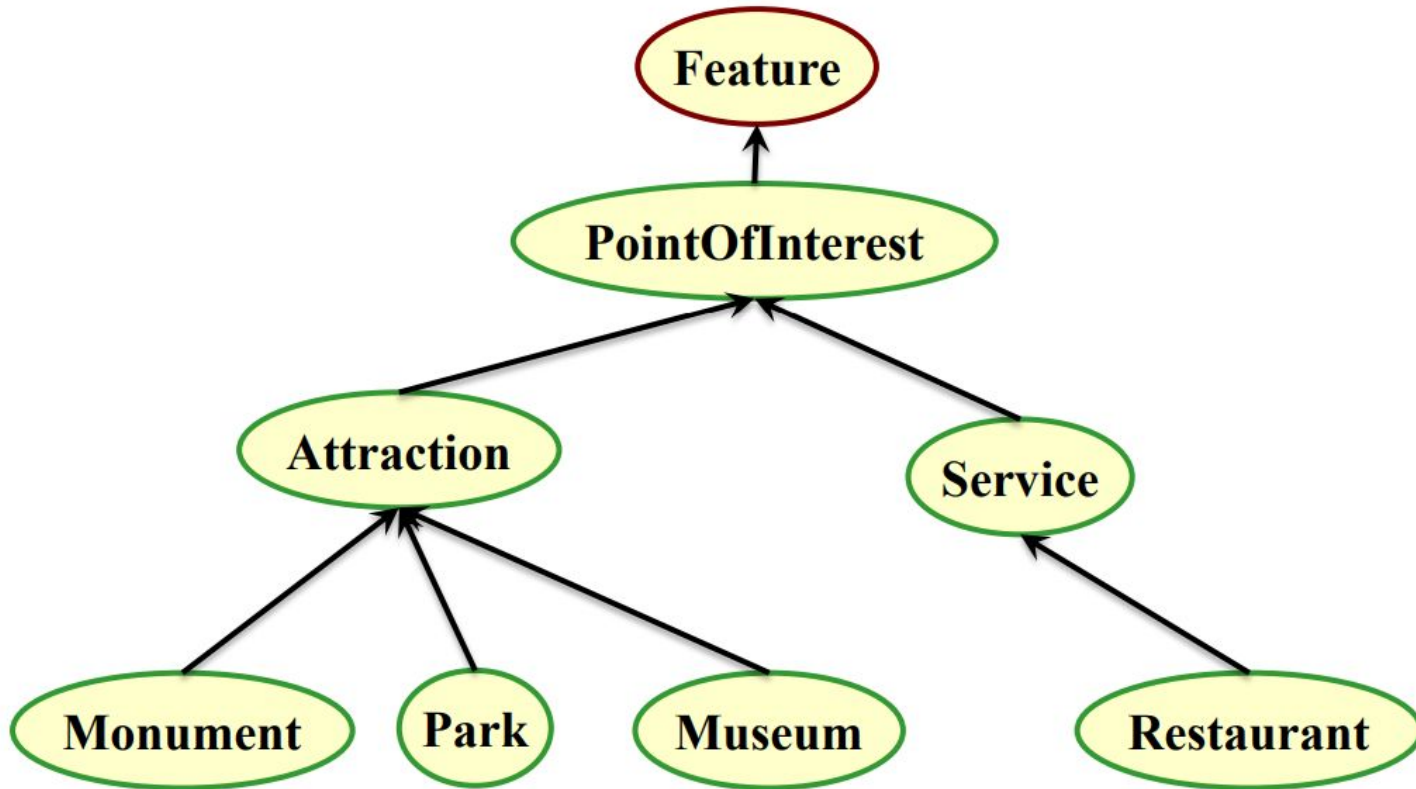
Data to Insert ( )

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix geo: <http://www.opengis.net/ont/geosparql#> .
@prefix ex: <http://www.example.org/POI#> .

ex:Restaurant a owl:Class;
  rdfs:subClassOf ex:Service .
ex:Park a owl:Class;
  rdfs:subClassOf ex:Attraction .
ex:Museum a owl:Class;
  rdfs:subClassOf ex:Attraction .
ex:Monument a owl:Class;
  rdfs:subClassOf ex:Attraction .
ex:Service a owl:Class;
  rdfs:subClassOf ex:PointOfInterest .
ex:Attraction a owl:Class;
  rdfs:subClassOf ex:PointOfInterest .
ex:PointOfInterest a owl:Class;
  rdfs:subClassOf geo:Feature .
```

Named graph for insertion:

# Parliament + GeoSPARQL



# Parliament + GeoSPARQL

## Text Insert

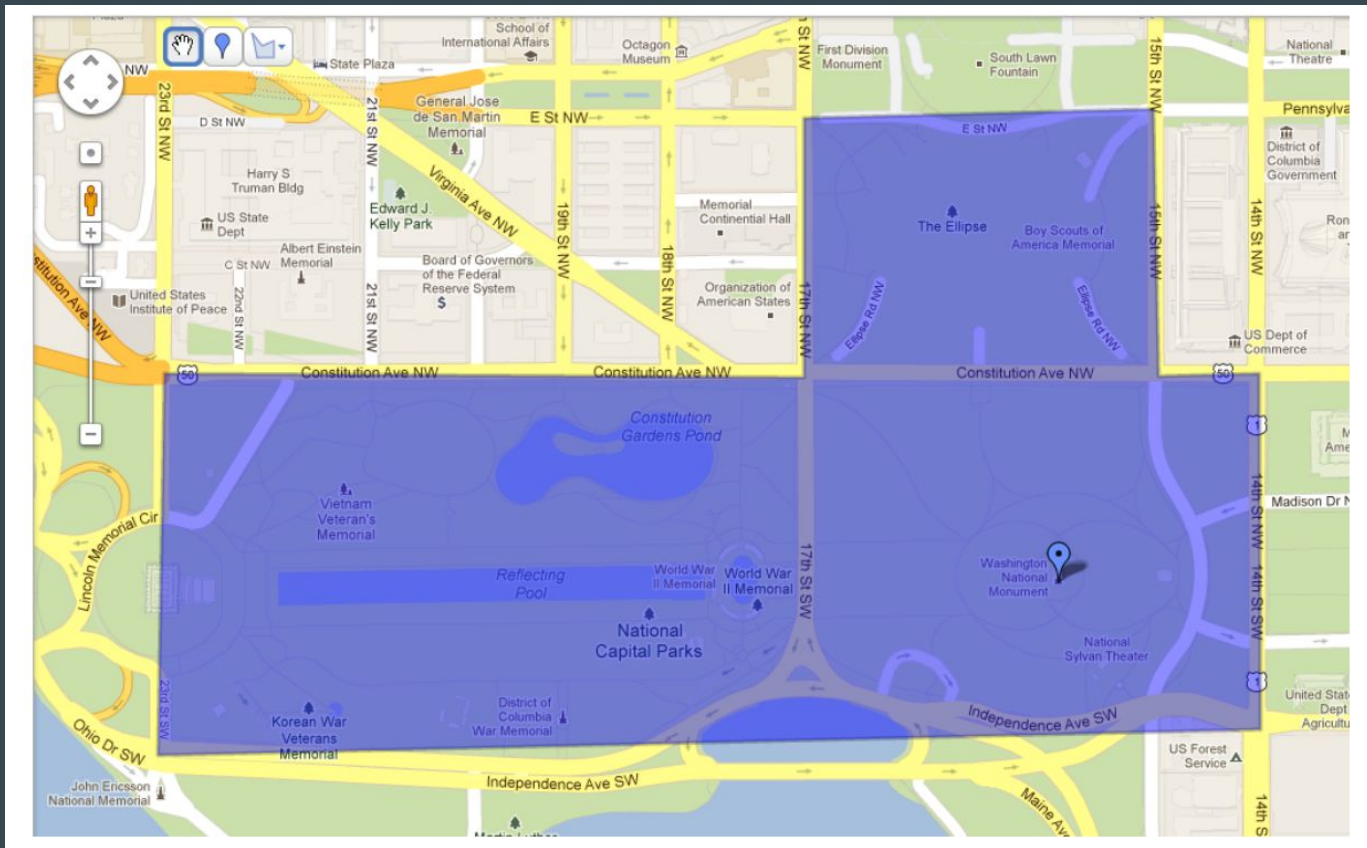
Data to Insert (  Format )

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix geo: <http://www.opengis.net/ont/geosparql#> .
@prefix ex: <http://www.example.org/POI#> .
@prefix sf: <http://www.opengis.net/ont/sf#> .

ex:WashingtonMonument a ex:Monument;
  rdfs:label "Washington Monument";
  geo:hasGeometry ex:WMPoint .
ex:WMPoint a sf:Point;
  geo:asWKT "POINT(-77.03524 38.889468)"^^geo:wktLiteral.
ex:NationalMall a ex:Park;
  rdfs:label "National Mall";
  geo:hasGeometry ex:NMPoly .
ex:NMPoly a sf:Polygon;
  geo:asWKT "POLYGON((-77.050125 38.892086, -77.039482 38.892036, -77.039482 38.895393,
-77.033669 38.895508, -77.033585 38.892052, -77.031906 38.892086, -77.031883 38.887474,
-77.050232 38.887142, -77.050125 38.892086 ))"^^geo:wktLiteral.
```

Named graph for insertion:

# Parliament + GeoSPARQL



# Parliament + GeoSPARQL

## SELECT or CONSTRUCT query

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX ex: <http://www.example.org/POI#>

SELECT ?m ?p
WHERE {
    ?m a ex:Monument ;
        geo:hasGeometry ?mgeo .
    ?p a ex:Park ;
        geo:hasGeometry ?pgeo .
    ?mgeo geo:sfWithin ?pgeo .
}
```

If SELECT query, display as:

☒ HTML table

☐ Count only

☐ CSV

☐ SPARQL result set

☐ Custom XSLT

☐ JSON

# Parliament + GeoSPARQL

The screenshot shows a web browser window with three tabs: "SPARQLer Query Results", "Explore Repository: http://", and "OK 200 Insert operation su". The address bar shows "localhost:8080/parliament/sparql". The page title is "SPARQLer Query Results". Below the title is a navigation bar with links: "Home", "Operations:", "Query", "Explore", "SPARQL/Update", "Insert Data", "Export", "Indexes", and "Admin". The main content area shows "Count: 1" and a table with two columns, "m" and "p". The table contains one row with the following values:

m	p
<a href="http://www.example.org/POI#WashingtonMonument">http://www.example.org/POI#WashingtonMonument</a>	<a href="http://www.example.org/POI#NationalMall">http://www.example.org/POI#NationalMall</a>

**Ende**