

# Hands on: Transformation, Store, Query, visualize



Theofilos Ioannidis

@tioannid1

tioannid@di.uoa.gr

tioannid@yahoo.com

# Store & Query – Demo : Preconditions

---

- **PostgreSQL+PostGIS** installed (see <https://strabon.di.uoa.gr/download.html> "User Guide: Installation Instructions for Linux")
- **Strabon** spatiotemporal RDF store installed (see <https://strabon.di.uoa.gr/download.html> "User Guide: Installation Instructions for Linux")
- **Apache Tomcat** installed (see <https://strabon.di.uoa.gr/download.html> "User Guide: Install Strabon Endpoint")
- **Scalability Workload** (datasets and querysets) from the Geographica2 benchmark (see <https://geographica2.di.uoa.gr/> "The scalability workload")
- ◆ **Demo System:**
  - VirtualBox guest: with Ubuntu 18.04.6 LTS, 24 GB RAM, 4 vCores, 80 GB disk, Java 8, PostgreSQL v9.5 + PostGIS v2.2, Tomcat v9.0.1
  - VirtualBox host: Windows 10, 32 GB RAM, i7-7700HQ CPU 4-core 8-threads

# Store & Query – Demo : Create Spatial databases

- Create 2 spatial PostgreSQL+PostGIS databases
  - *scalability10k, scalability100k*
  - *using postgres user from terminal*

```
tioannid@ubuntu-vma-tioa:~$ sudo -i -u postgres
postgres@ubuntu-vma-tioa:~$ createdb scalability10k -T template_postgis
postgres@ubuntu-vma-tioa:~$ createdb scalability100k -T template_postgis
postgres@ubuntu-vma-tioa:~$ psql
psql (15.0 (Ubuntu 15.0-1.pgdg18.04+1), server 9.5.25)
Type "help" for help.

postgres=# \l
                                List of databases
  Name          | Owner   | Encoding | Collate | Ctype   | ICU Locale | Locale Provider | Access privileges
-----+-----+-----+-----+-----+-----+-----+-----
geographica3   | geographica3 | UTF8     | en_US.UTF-8 | en_US.UTF-8 |              | libc              |
nyc             | postgres    | UTF8     | en_US.UTF-8 | en_US.UTF-8 |              | libc              |
postgres       | postgres    | UTF8     | en_US.UTF-8 | en_US.UTF-8 |              | libc              |
scalability100k | postgres    | UTF8     | en_US.UTF-8 | en_US.UTF-8 |              | libc              |
scalability10k  | postgres    | UTF8     | en_US.UTF-8 | en_US.UTF-8 |              | libc              |
synthetic_64   | postgres    | UTF8     | en_US.UTF-8 | en_US.UTF-8 |              | libc              |
template0      | postgres    | UTF8     | en_US.UTF-8 | en_US.UTF-8 |              | libc              |
                |           |          |           |           |           |                 | =c/postgres      +
                |           |          |           |           |           |                 | postgres=CTc/postgres
template1      | postgres    | UTF8     | en_US.UTF-8 | en_US.UTF-8 |              | libc              |
                |           |          |           |           |           |                 | =c/postgres      +
                |           |          |           |           |           |                 | postgres=CTc/postgres
template_postgis | postgres    | UTF8     | en_US.UTF-8 | en_US.UTF-8 |              | libc              |
(9 rows)

postgres=#
```

# Store & Query – Demo : Store RDF dataset with Strabon (console)

- **Strabon** is a geospatial RDF store that has as its main persistence option the PostgreSQL+PostGIS database
- The RDF dataset to load is the **Scalability 100K** dataset from the **Scalability** workload of the **Geographica 2** benchmarking framework. The Scalability reference dataset is OSM based and with the matching utility we can produce the following list of scalability datasets: *10K, 100K, 1M, 10M, 100M, 500M*.
- Load the Scalability 100K N-Triples dataset to the **scalability100k** database
  - *using java command line and Strabon's eu.earthobservatory.runtime.postgis.StoreOp class (see [https://strabon.di.uoa.gr/files/stSPARQL\\_tutorial.pdf](https://strabon.di.uoa.gr/files/stSPARQL_tutorial.pdf) , Section §4.3.2)*

```
tioannid@ubuntu-vm-tioa:~$ cd ~/NetBeansProjects/PhD/Strabon/runtime/target
tioannid@ubuntu-vm-tioa:~/NetBeansProjects/PhD/Strabon/runtime/target$ export STRABON_CLASSPATH=$(for file in `ls -1 *.jar`; do myVar=$myVar./$file:"";done;echo $myVar;)
tioannid@ubuntu-vm-tioa:~/NetBeansProjects/PhD/Strabon/runtime/target$ java -cp $$STRABON_CLASSPATH eu.earthobservatory.runtime.postgis.StoreOp localhost 5432 scalability100k postgres postgres /media/sf_VM_Shared/PHD/Geographica2_Datasets/Scalability/100K/scalability_100K.nt
2022-10-23 09:56:29,739 [main] INFO eu.earthobservatory.runtime.postgis.Strabon - [Strabon] Checking for locks...
2022-10-23 09:56:31,877 [main] INFO eu.earthobservatory.runtime.postgis.Strabon - [Strabon] Initialization completed.
2022-10-23 09:56:31,878 [main] INFO eu.earthobservatory.runtime.generaldb.Strabon - [Strabon] Initialization took 1496 ms.
2022-10-23 09:56:31,883 [main] INFO eu.earthobservatory.runtime.generaldb.Strabon - [Strabon.storeURL] Storing file.
2022-10-23 09:56:31,883 [main] INFO eu.earthobservatory.runtime.generaldb.Strabon - [Strabon.storeURL] URL : file:/media/sf_VM_Shared/PHD/Geographica2_Datasets/Scalability/100K/scalability_100K.nt
2022-10-23 09:56:31,883 [main] INFO eu.earthobservatory.runtime.generaldb.Strabon - [Strabon.storeURL] Context : default
2022-10-23 09:56:31,883 [main] INFO eu.earthobservatory.runtime.generaldb.Strabon - [Strabon.storeURL] Base URI : null
2022-10-23 09:56:31,883 [main] INFO eu.earthobservatory.runtime.generaldb.Strabon - [Strabon.storeURL] Format : N-Triples (mimeType=text/plain; ext=nt)
2022-10-23 09:57:13,151 [main] INFO eu.earthobservatory.runtime.generaldb.Strabon - [Strabon.storeURL] Storing was successful.
2022-10-23 09:57:13,151 [main] INFO eu.earthobservatory.runtime.generaldb.Strabon - [Strabon.close] Closing connection...
2022-10-23 09:57:13,190 [main] INFO eu.earthobservatory.runtime.postgis.Strabon - [Strabon] Cleaning...
2022-10-23 09:57:13,204 [main] INFO eu.earthobservatory.runtime.postgis.Strabon - [Strabon] Clearing Successful.
2022-10-23 09:57:13,204 [main] INFO eu.earthobservatory.runtime.generaldb.Strabon - [Strabon.close] Connection closed.
```

# Store & Query – Demo : Query Strabon database (console)

- The query to execute is SC1 (spatial selection) from the Scalability workload queryset of the Geographica 2 benchmarking framework. The Scalability queryset comprises 1 spatial selection (SC1) and 2 spatial join queries (SC2, SC3). These queries can run against any of the scalability datasets: 10K, 100K, 1M, 10M, 100M, 500M.
- Query the **scalability100k** database using the SC1 query from Scalability Workload dataset, retrieve results in HTML format
  - *using java command line and Strabon's `eu.earthobservatory.runtime.postgis.QueryOp` class (see [https://strabon.di.uoa.gr/files/stSPARQL\\_tutorial.pdf](https://strabon.di.uoa.gr/files/stSPARQL_tutorial.pdf), Section §4.3.3)*

```
tioannid@ubuntu-vm-a-tioa:~$ cd ~/NetBeansProjects/PhD/Strabon/runtime/target
tioannid@ubuntu-vm-a-tioa:~/NetBeansProjects/PhD/Strabon/runtime/target$ export STRABON_CLASSPATH=$(for file in `ls -1 *.jar`; do my
Var=$myVar./$file";done;echo $myVar;)
tioannid@ubuntu-vm-a-tioa:~/NetBeansProjects/PhD/Strabon/runtime/target$ export SC1_QUERY=`cat /media/sf_VM_Shared/PHD/Geographica2_
Datasets/QuerySets/Scalability/SC1/SC1_Geometries_Intersects_GivenPolygon.qry`
tioannid@ubuntu-vm-a-tioa:~/NetBeansProjects/PhD/Strabon/runtime/target$ echo $$SC1_QUERY
PREFIX geof: <http://www.opengis.net/def/function/geosparql/> PREFIX geo: <http://www.opengis.net/ont/geosparql#> PREFIX lgd: <http
://data.linkedeodata.eu/ontology#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> SELECT ?s1 ?o1 WHERE { ?s1 geo:asWKT ?o1 . FILTE
R(geof:sfIntersects(?o1, "POLYGON((23.708496093749996 37.95719224376526,22.906494140625 40.659805938378526,11.524658203125002 48.16
425348854739,-0.1181030273437499 51.49506473014367,-3.2189941406250004 55.92766341247031,-5.940856933593749 54.59116279530599,-3.16
68090820312504 51.47967237816337,23.708496093749996 37.95719224376526))"^^<http://www.opengis.net/ont/geosparql#wktLiteral>)). }
tioannid@ubuntu-vm-a-tioa:~/NetBeansProjects/PhD/Strabon/runtime/target$ java -cp $$STRABON_CLASSPATH eu.earthobservatory.runtime.pos
tgis.QueryOp localhost 5432 scalability100k postgres postgres "$SC1_QUERY" false
```

# Store & Query – Demo : Deploy the Strabon endpoint WAR to Tomcat

- Deploy the Strabon endpoint WAR to Tomcat and connect to the **scalability10k** database (see <https://strabon.di.uoa.gr/download.html> “User Guide: Install Strabon Endpoint”)
  - Through Tomcat’s Manager app browse to locate the Strabon endpoint WAR (`..Strabon/endpoint/target/strabon-endpoint-3.3.2-SNAPSHOT.war`)
  - Deploy the WAR

<code>/rdf4j-workbench</code>	<i>None specified</i>	RDF4J Workbench	true	<u>0</u>
-------------------------------	-----------------------	-----------------	------	----------

### Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

### WAR file to deploy

Select WAR file to upload  `strabon-endpoint-3.3.2-SNAPSHOT.wa`

<code>/rdf4j-workbench</code>	<i>None specified</i>	RDF4J Workbench	true	<u>0</u>
<code>/strabon-endpoint-3.3.2-SNAPSHOT</code>	<i>None specified</i>	StrabonEndpoint	true	<u>0</u>

### Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

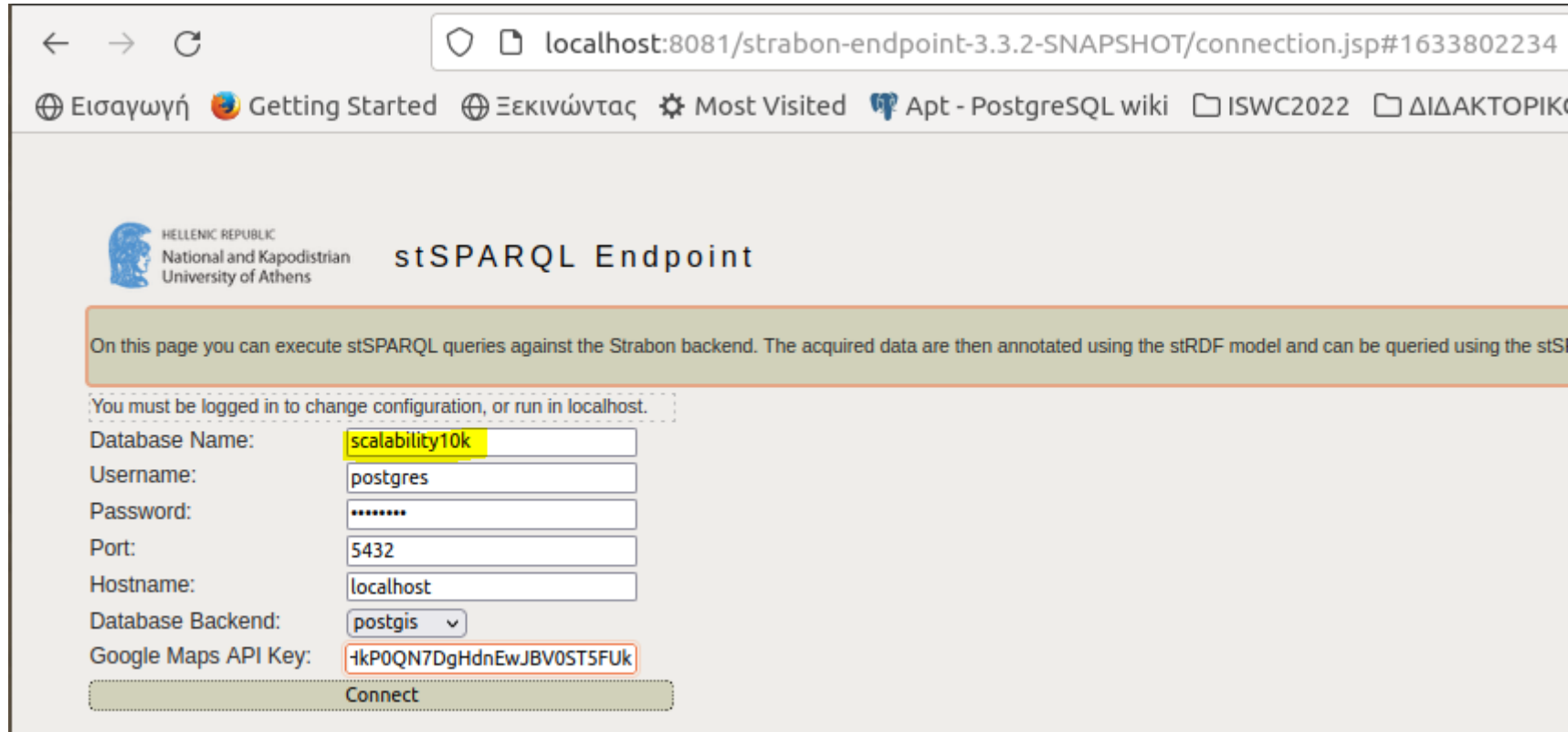
WAR or Directory URL:

### WAR file to deploy

Select WAR file to upload  No file selected.

# Store & Query – Demo : Connect to a DB with Strabon (UI)

- Connect to the **scalability10k** database with the Strabon endpoint UI
  - use the **Explore/Modify operations-->Configuration** form to connect to the empty **scalability10k** database



The screenshot shows a web browser window with the URL `localhost:8081/strabon-endpoint-3.3.2-SNAPSHOT/connection.jsp#1633802234`. The page header includes the logo of the Hellenic Republic National and Kapodistrian University of Athens and the title "stSPARQL Endpoint". A message states: "On this page you can execute stSPARQL queries against the Strabon backend. The acquired data are then annotated using the stRDF model and can be queried using the stSPARQL endpoint." Below this, a note says: "You must be logged in to change configuration, or run in localhost." The configuration form contains the following fields:

Database Name:	<input type="text" value="scalability10k"/>
Username:	<input type="text" value="postgres"/>
Password:	<input type="password" value="*****"/>
Port:	<input type="text" value="5432"/>
Hostname:	<input type="text" value="localhost"/>
Database Backend:	<input type="text" value="postgis"/>
Google Maps API Key:	<input type="text" value="AIzaSyBkP0QN7DgHdnEwJBV0ST5FUK"/>

At the bottom of the form is a "Connect" button.

# Store & Query – Demo : Store RDF dataset with Strabon (UI)

- Load the Scalability 10K dataset to the **scalability10k** database
  - use the **Explore/Modify operations-->Store** form of the Strabon endpoint to load the Scalability 10K dataset from **file:///media/sf\_VM\_Shared/PHD/Geographica2\_Datasets/Scalability/10K/scalability\_10K.nt**

The screenshot shows the 'stSPARQL Endpoint' interface. On the left is a sidebar with a menu under 'Explore/Modify operations' containing 'Store', 'Query/Update', 'Describe', and 'Configuration'. The main area has a 'Direct Input' section with a large empty text area. Below it are two buttons: 'Store Input' and 'Store from URI', with the latter being highlighted in yellow. The 'URI Input' field below contains the file path: 'file:///media/sf\_VM\_Shared/PHD/Geographica2\_Datasets/Scalability/10K/scalability\_10K.nt'. At the top, there is a header with the Hellenic Republic logo and the text 'stSPARQL Endpoint'. A message box at the top states: 'On this page you can execute stSPARQL queries against the Strabon backend. The acquired data are then annotated using the stRDF model and can be queried using the stSPARQL query language. On the left sidebar, some example stSPARQL queries to acquire information on the dataset, are provided.'



# Store & Query – Demo : Query RDF dataset with Strabon (UI)

- Query the **scalability10k** database using the SC1 query from Scalability Workload dataset, retrieve results in HTML format
  - use the *Explore/Modify operations-->Query/Update form to run both queries*

the stSPARQL query language. On the left sidebar, some example stSPARQL queries to acquire information on the dataset, are provided.

Discovery Queries

- Find all triples in the dataset.
- Select all distinct subjects that appear in the dataset.
- Select all distinct predicates that appear in the dataset.
- Select all distinct objects that appear in the dataset.
- Find all distinct classes of the dataset.
- Find the number of triples that appear in the dataset.
- Present the first ten triples of the dataset.

Explore/Modify operations

You must be logged in to perform update queries, or run in localhost.

stSPARQL Query:

```
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX lgd: <http://data.linkedeodata.eu/ontology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?s1 ?o1 WHERE {
  ?s1 geo:asWKT ?o1 .
  FILTER(geof:sfIntersects(?o1, "POLYGON((23.708496093749996 37.95719224376526,22.906494140625
40.659805938378526,11.524658203125002 48.16425348854739,-0.1181030273437499 51.49506473014367,-
3.2189941406250004 55.92766341247031,-5.940856933593749 54.59116279530599,-3.1668090820312504
51.47967237816337,23.708496093749996 37.95719224376526))"^^<http://www.opengis.net
/ont/geosparql#wktLiteral>)).
}
```

Output Format:

View Result:

Map Bounds:

s1	o1
<a href="http://data.linkedeodata.eu/osm/wales/transport/Geometry/111686299">http://data.linkedeodata.eu/osm/wales/transport/Geometry/111686299</a>	"MULTIPOLYGON (((-4.0803447 53.0841743, -4.0803 53.0841838, -4.080264 53.084123, -4.0803086 53.0841... <a href="#">more</a>
<a href="http://data.linkedeodata.eu/osm/wales/transport/Geometry/111686304">http://data.linkedeodata.eu/osm/wales/transport/Geometry/111686304</a>	"MULTIPOLYGON (((-4.1180155 53.1046596, -4.1179132 53.1046853, -4.1177627 53.1044688, -4.117865 53... <a href="#">more</a>
<a href="http://data.linkedeodata.eu/osm/wales/transport/Geometry/111686306">http://data.linkedeodata.eu/osm/wales/transport/Geometry/111686306</a>	"MULTIPOLYGON (((-4.0960447 53.0956141, -4.0960086 53.0956258, -4.0959763 53.0955897, -4.0960124 53... <a href="#">more</a>
<a href="http://data.linkedeodata.eu/osm/wales/transport/Geometry/117042377">http://data.linkedeodata.eu/osm/wales/transport/Geometry/117042377</a>	"MULTIPOLYGON (((-2.7197583 51.8100268, -2.7197573 51.8100704, -2.7197355 51.8101557, -2.7197044 51... <a href="#">more</a>

# Store & Query – Demo : Query&Map RDF dataset with Strabon (UI)

- Query the **scalability10k** database using the SC1 query from Scalability Workload dataset, retrieve results in KML and display it as map
  - use the **Explore/Modify operations-->Query/Update** form to run both queries

The screenshot displays the 'stSPARQL Endpoint' interface. At the top, it identifies the 'HELLENIC REPUBLIC National and Kapodistrian University of Athens'. Below this, a message states: 'On this page you can execute stSPARQL queries against the Strabon backend. The acquired data are then annotated using the stRDF model and can be queried using the stSPARQL query language. On th...'. The interface is divided into several sections:

- Discovery Queries:** A list of query types such as 'Find all triples in the dataset', 'Select all distinct subjects that appear in the dataset', etc.
- stSPARQL Query:** A text area containing a query: 

```
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX lgd: <http://data.linkedeodata.eu/ontology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?s1 ?o1 WHERE {
  ?s1 geo:asWKT ?o1 .
  FILTER(geof:sfIntersects(?o1, "POLYGON((23.708496093749996 37.95719224376526,22.906494140625
40.659805938378526,11.524658203125002 48.16425348854739,-0.1181030273437499 51.49506473014367,-
3.2189941406250004 55.92766341247031,-5.940856933593749 54.59116279530599,-3.1668090820312504
51.47967237816337,23.708496093749996 37.95719224376526))"^^<http://www.opengis.net
/ont/geosparql#wktLiteral>)).
}
```
- Output Format:** A dropdown menu set to 'KML'.
- View Result:** A dropdown menu set to 'On a map (localhost)'.
- Map Bounds:** A text field containing a bounding box: `"POLYGON((17.81 49.65, -6.90 49.65, -6.90 56.51, 17.81 56.51, 17.81 49.65))"^^<http://www.opengis.net`
- Buttons:** 'Query' and 'Update' buttons.

Below the interface is a map visualization showing a geographical area with several red location markers. The map includes labels for 'Glasgow', 'Edinburgh', and 'Newcastle upon Tyne' in both Greek and English. The bottom of the map area contains the text 'For development purposes only' repeated several times.

# Store & Query – Demo : Query RDF dataset with Strabon (UI)

- Query the **scalability10k** database using the SC2 query from Scalability Workload dataset, retrieve results in HTML
  - use the **Explore/Modify operations-->Query/Update form to run both queries**

The screenshot shows the stSPARQL Endpoint interface. At the top, it identifies the Hellenic Republic National and Kapodistrian University of Athens. A text box explains that users can execute stSPARQL queries against the Strabon backend. On the left, there are sections for 'Discovery Queries' (listing various search criteria) and 'Explore/Modify operations'. The main area contains a text editor with a stSPARQL query. Below the editor are dropdown menus for 'Output Format' (set to HTML) and 'View Result' (set to Plain). At the bottom, there are 'Query' and 'Update' buttons. The results are displayed in a table with columns labeled s1, o1, s2, and o2.

stSPARQL Endpoint

On this page you can execute stSPARQL queries against the Strabon backend. The acquired data are then annotated using the stRDF model and can be queried using the stSPARQL query language. On the left sidebar, some example stSPARQL queries to acquire information on the dataset, are provided.

You must be logged in to perform update queries, or run in localhost.

Discovery Queries

- Find all triples in the dataset.
- Select all distinct subjects that appear in the dataset.
- Select all distinct predicates that appear in the dataset.
- Select all distinct objects that appear in the dataset.
- Find all distinct classes of the dataset.
- Find the number of triples that appear in the dataset.
- Present the first ten triples of the dataset.

Explore/Modify operations

stSPARQL Query:

```
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX lgd: <http://data.linkededata.eu/ontology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?s1 ?o1 ?s2 ?o2
WHERE {
  ?s1 geo:hasGeometry [ geo:asWKT ?o1 ] ;
  lgd:has_code "1001"^^xsd:integer .
  ?s2 geo:hasGeometry [ geo:asWKT ?o2 ] ;
  lgd:has_code ?code2 .
  FILTER(?code2>5000 && ?code2<6000 && ?code2 != 5260) .
  FILTER(geof:sfIntersects(?o1, ?o2)).
} |
```

Output Format: HTML

View Result: Plain

Map Bounds:

Query Update

s1	o1	s2	o2
<a href="http://data.linkededata.eu/osm/wales/places/id/335184">http://data.linkededata.eu/osm/wales/places/id/335184</a>	"MULTIPOLYGON (((-3.1243762 51.588509, -3.1242394 51.5885257, -3.1242149 51.5885341, -3.1241616 51.... more	<a href="http://data.linkededata.eu/osm/wales/transport/id/256429040">http://data.linkededata.eu/osm/wales/transport/id/256429040</a>	"MULTIPOLYGON (((-2.9955691 51.5893216, -2.9954907 51.5893148, -2.9954612 51.5892948, -2.9954907 51.... more
<a href="http://data.linkededata.eu/osm/wales/places/id/335184">http://data.linkededata.eu/osm/wales/places/id/335184</a>	"MULTIPOLYGON (((-3.1243762 51.588509, -3.1242394 51.5885257, -3.1242149 51.5885341, -3.1241616 51.... more	<a href="http://data.linkededata.eu/osm/wales/transport/id/47224355">http://data.linkededata.eu/osm/wales/transport/id/47224355</a>	"MULTIPOLYGON (((-2.9997833 51.5888969, -2.998691 51.5894817, -2.9985602 51.5894045, -2.9987581 51.... more