

GEOALCHEMY



Speaker notes

This talk is about GeoAlchemy, which is an extension to SQLAlchemy for working with geospatial databases.

I am actually going to talk about three things:

- Geospatial databases, focusing on PostGIS,
- SQLAlchemy, the Python SQL toolkit,
- and GeoAlchemy, which makes it possible to use SQLAlchemy with geospatial databases.

ÉRIC LEMOINE

Developer @ Oslandia

✉ eric.lemoine@oslandia.com
🐙 [@elemoine](#)
🐦 [@erilem](#)

Speaker notes

My name is Éric Lemoine. I work at Oslandia. I've been using Postgres, PostGIS and SQLAlchemy for about 10 years.



Oslandia provides service on open-source software

- GIS
- 3D
- DATA

Speaker notes

Oslandia is an open-source company working on GIS, 3D and Data Science. QGIS and PostGIS are examples of software components we are working on.

GEOALCHEMY

Use SQLAlchemy with Spatial Databases



Speaker notes

GeoAlchemy is a Python library extending SQLAlchemy to work with geospatial databases.

WHAT'S A SPATIAL DATABASE?

Quoting Wikipedia:

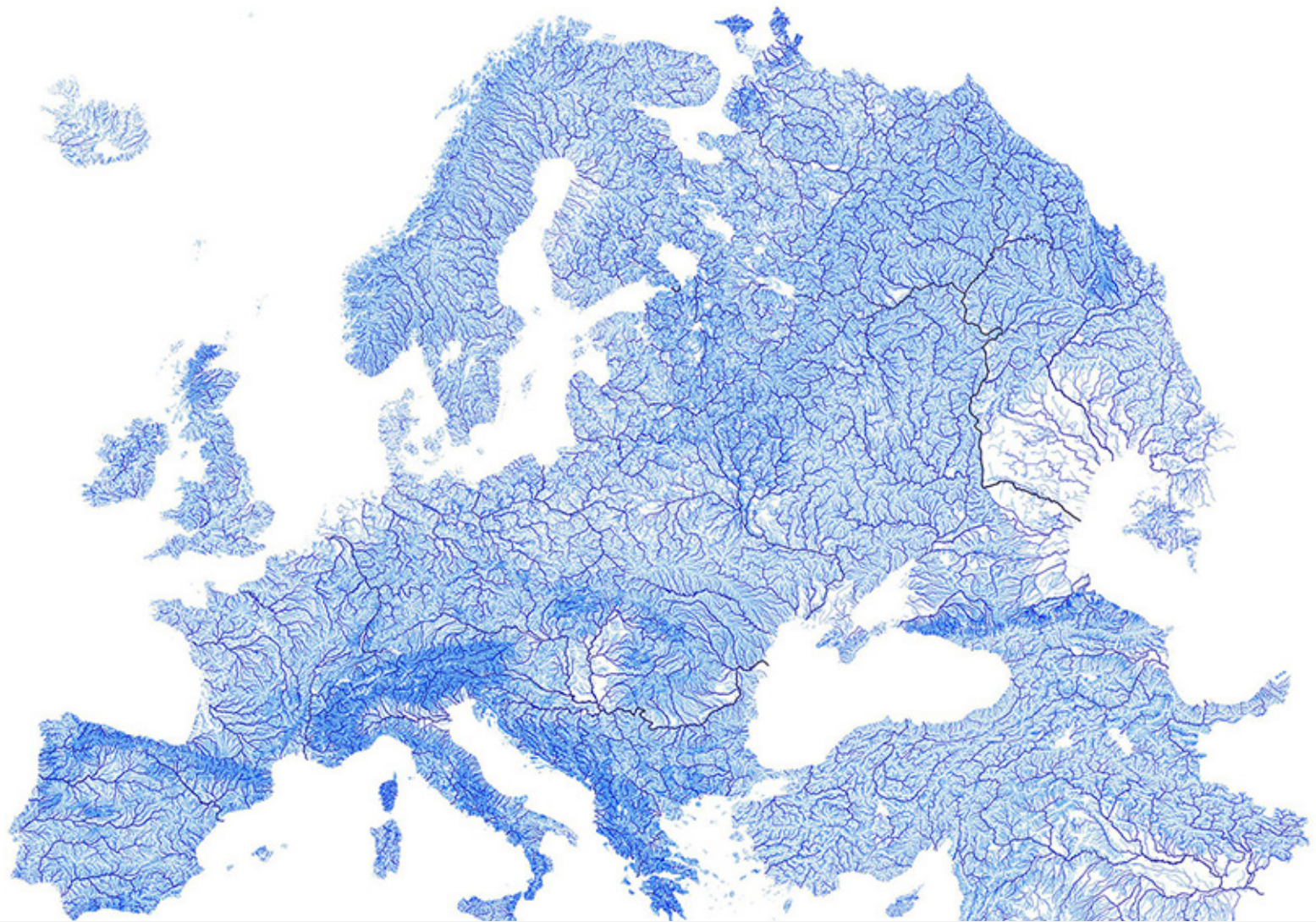
A spatial database, or geodatabase is a database that is optimized to store and query data that represents objects defined in a geometric space.

SPATIAL QUERIES

« Give me all the POIs within a given area »

SPATIAL QUERIES

« Give me all the POIs within a certain distance to a point »



Speaker notes

And what you can ultimately do with geospatial databases is create beautiful maps. This one is a map of Europe drawn by its rivers and streams.

POSTGIS

*The Spatial Database extender for
PostgreSQL*



<http://postgis.net/>

POSTGIS

PostGIS provides "spatial" types, functions and operators, and indexes.

POSTGIS EXAMPLE #1

Enable PostGIS in a database

```
$ psql -d my-database  
my-database=# create extension postgis;
```

POSTGIS EXAMPLE #2

Create a table with a "geometry" column

```
CREATE TABLE users (  
  id SERIAL,  
  name TEXT,  
  fullname TEXT,  
  geom GEOMETRY(POINT)  
);  
CREATE INDEX users_geom_idx ON users USING GIST (geom);
```

POSTGIS EXAMPLE #3

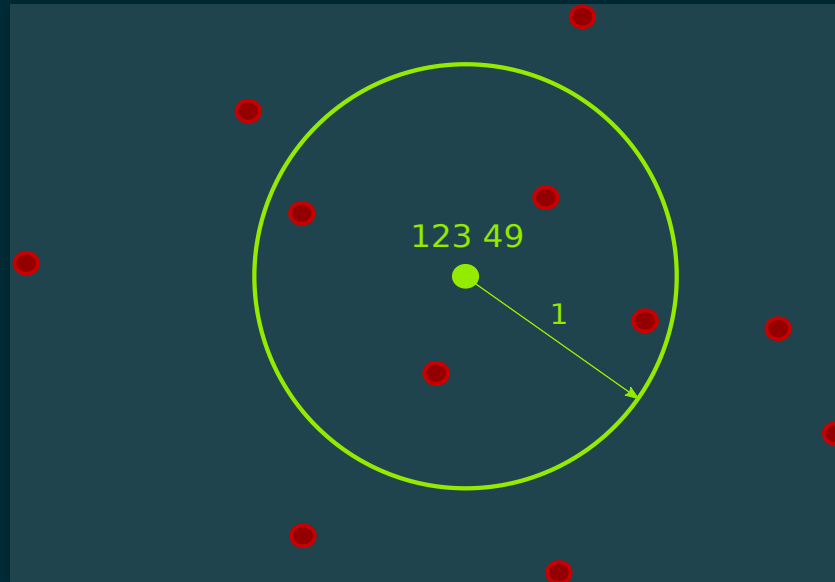
Insert a record with a geometry

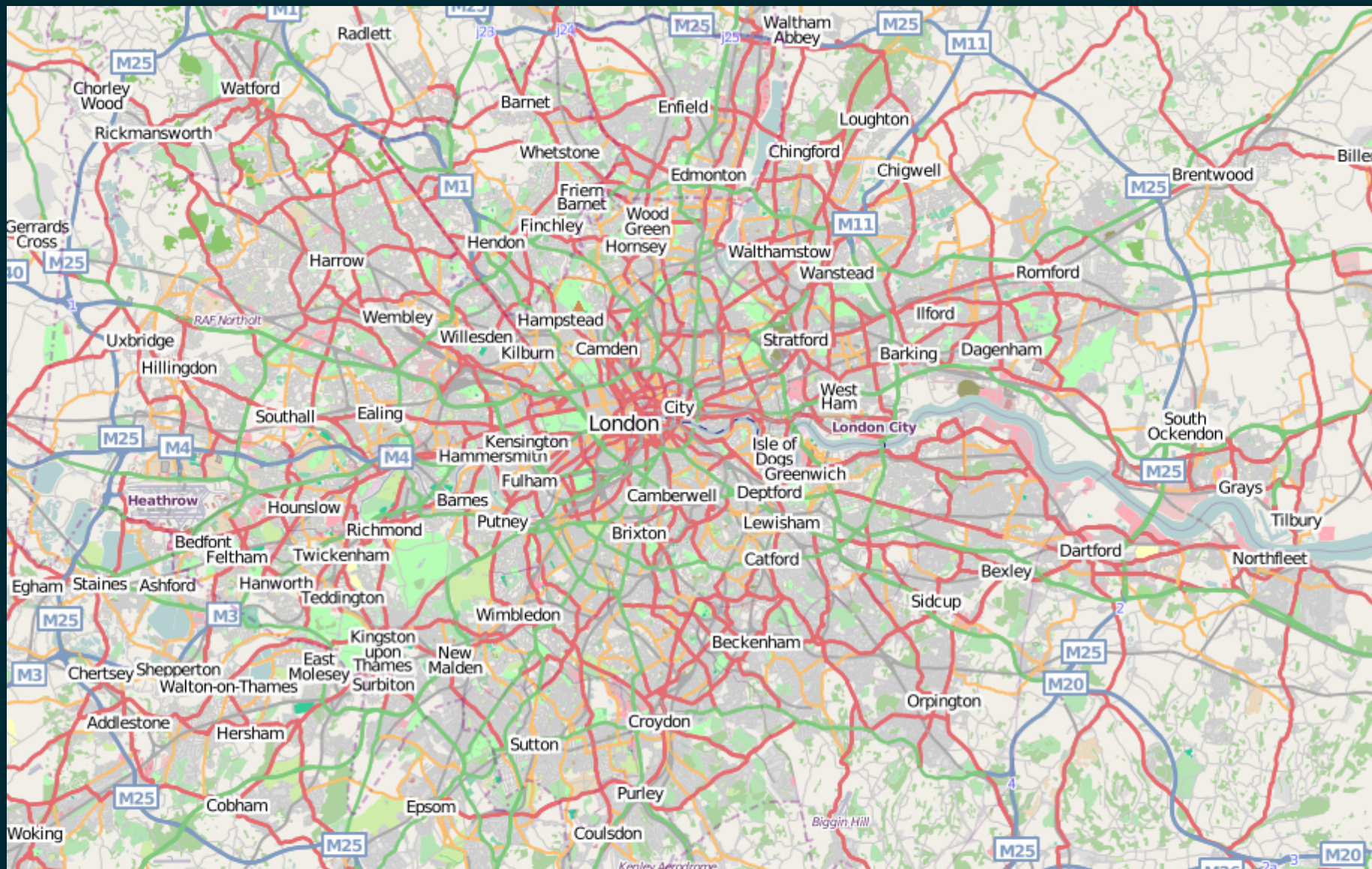
```
INSERT INTO users (name, fullname, geom)
VALUES('pramsey', 'Paul Ramsey',
       ST_GeomFromText('POINT(-123 48)'));
```

POSTGIS EXAMPLE #4

Select users within a distance of a point

```
SELECT name FROM users  
WHERE ST_DWithin(users.geom, 'POINT(-123 49)', 1);
```





Speaker notes

OpenStreetMap uses PostGIS.

SQLALCHEMY

The Database Toolkit for Python



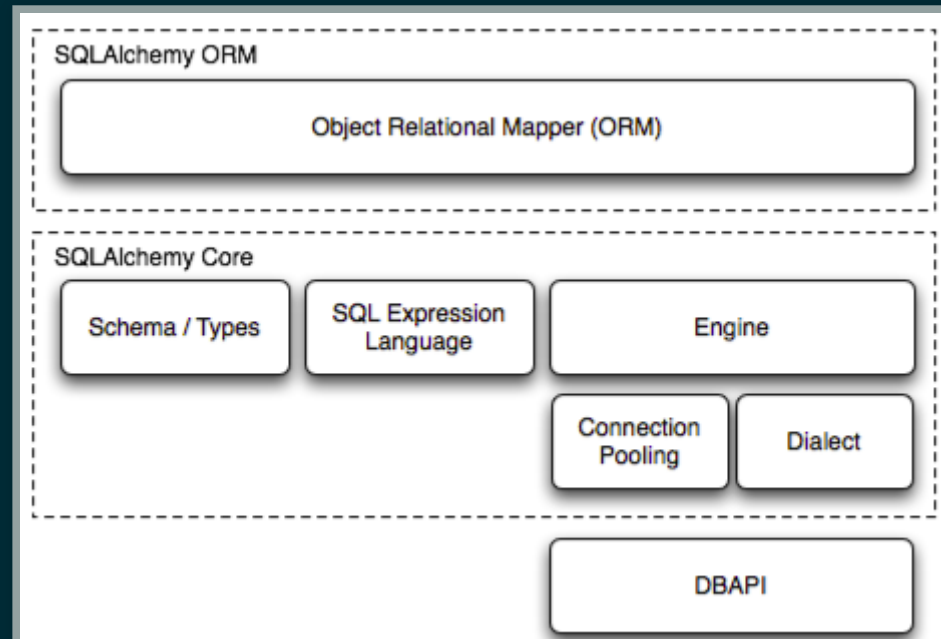
<https://www.sqlalchemy.org/>

SQLALCHEMY PHILOSOPHY

- Not about hiding the DB
- Relational form of data is preserved
- SQLA provides a rich vocabulary to express decisions made by the developer

SQLALCHEMY ARCHITECTURE

Two parts: SQLAlchemy Core and SQLAlchemy ORM



Speaker notes

SQLAlchemy Core includes an SQL Expression Language for forming SQL constructs.

SQLALCHEMY CORE EXAMPLE #1

Define and create tables

```
from sqlalchemy import (Table, Column, Integer,
    String, MetaData, ForeignKey)
metadata = MetaData()
users = Table('users', metadata,
    Column('id', Integer, primary_key=True),
    Column('name', String),
    Column('fullname', String),
    )
addresses = Table('addresses', metadata,
    Column('id', Integer, primary_key=True),
    Column('user_id', None, ForeignKey('users.id')),
    Column('email_address', String, nullable=False)
    )
metadata.create_all()
```

SQLALCHEMY CORE EXAMPLE #2

Insert records

```
insert = users.insert().values(name='jack', fullname='Jack Jones')
conn = engine.connect() # get a connection
result = conn.execute(insert)
user_id = result.inserted_primary_key
```

SQLALCHEMY CORE EXAMPLE #3

Select recods

```
from sqlalchemy.sql import select

s = select([users, addresses]).where(users.c.id == addresses.c.user_id)
result = conn.execute(s)
for row in result:
    print(row['name'], row['fullname'])
```

WITH JUST PSYCOPG2 (DBAPI)

```
sql = "SELECT id, diameter, ST_AsGeoJSON(geom) FROM mytable"
if bbox:
    sql_ = "{} WHERE ST_Intersects(geom, ST_MakeEnvelope("
        "%(xmin)s, %(ymin)s, %(xmax)s, %(ymax)s))".format(sql)
    vars_["xmin"] = bbox[0]
    vars_["ymin"] = bbox[1]
    vars_["xmax"] = bbox[2]
    vars_["ymax"] = bbox[3]

if limit:
    sql = "{} LIMIT %(limit)s".format(sql)
    vars_['limit'] = limit

with conn.cursor() as cursor:
    cursor.execute(sql, vars_)
```

WITH SQLALCHEMY

```
q = select([column("id"),
               column("diameter"),
               func.ST_ASGeoJSON(column("geom"))])
q = q.select_from("mytable")

if bbox:
    q = q.where(
        func.ST_Intersects(
            column("geom"),
            func.ST_MakeEnvelope(bbox[0], bbox[1], bbox[2], bbox[3])
        )
    )

if limit:
    q = q.limit(limit)

conn.execute(q)
```

SQLALCHEMY ECOSYSTEM

- Alembic – DB migrations
- Flask-SQLAlchemy – Flask extension for SQLA
- ...

Speaker notes

Flask-User depends on Flask-SQLAlchemy.

GEOALCHEMY

Provides extensions to SQLAlchemy for working with
Spatial databases



<https://geoalchemy-2.readthedocs.io/>

A BIT OF HISTORY

- GeoAlchemy 1 created in 2009 (PostGIS only)
- MySQL, SpatiaLite, Oracle, MS SQL support added in 2010
- GeoAlchemy 2 created in 2012 (PostGIS only)
- SpatiaLite support added in 2018

GEOALCHEMY FEATURES

- Supports Geometry, Geography and Raster types
- Supports many PostGIS functions and operators
- Works with SQLA Core and SQLA ORM
- Integrates with Shapely

Speaker notes

GeoAlchemy is actually a thin layer on top of SQLAlchemy.

GEOALCHEMY EXAMPLE #1

Specify a geometry column

```
from sqlalchemy import (Table, Column, Integer,
                        String, MetaData, ForeignKey)
from geoalchemy2 import Geometry

metadata = MetaData()
users = Table('users', metadata,
              Column('id', Integer, primary_key=True),
              Column('name', String),
              Column('fullname', String),
              Column('geom', Geometry('POINT'))
              )
```

GEOALCHEMY EXAMPLE #2

Insert a "spatial" record

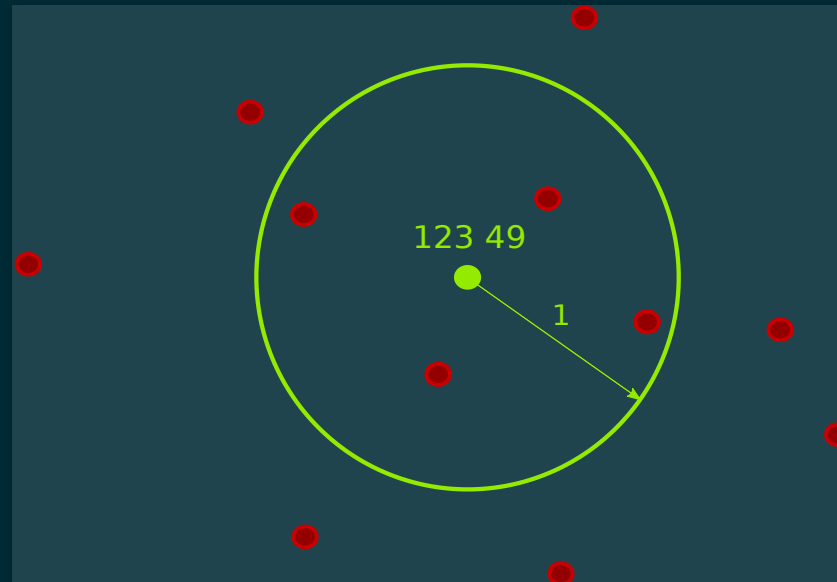
```
insert = users.insert().values(  
    name='jack', fullname='Jack Jones', geom='POINT(90 43)')  
conn = engine.connect()  
result = conn.execute(insert)  
user_id = result.inserted_primary_key
```

GEOALCHEMY EXAMPLE #3

Get the objects that are within a distance to a point

```
from sqlalchemy.sql import select

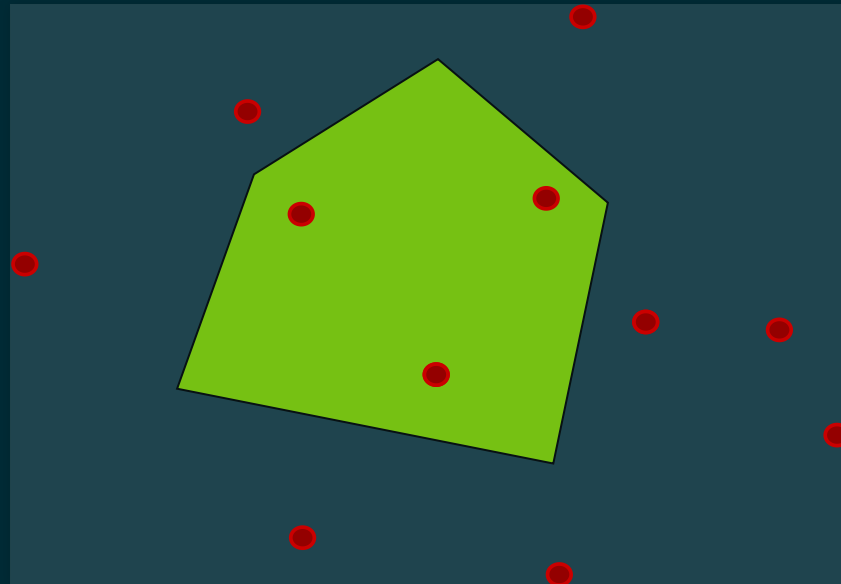
s = select([users]).where(func.ST_DWithin(users.c.geom, 'POINT(90 43)', 1))
result = conn.execute(s)
for row in result:
    print(row['name'], row['fullname'])
```



GEOALCHEMY EXAMPLE #4

Get the objects that are within a polygon

```
s = select([users]).where(
    func.ST_Contains('POLYGON((80 40,100 40,100 50,80 50))', users.c.geom))
result = conn.execute(s)
for row in result:
    print(row['name'], row['fullname'])
```



GEOALCHEMY ECOSYSTEM

GeoAlchemy integrates well with

- Shapely
- geojson
- pyproj

CONCLUSION

- PostGIS is great. Use it!
- SQLA is great when working with DBs in Python
- GeoA is useful when using PostGIS (or SpatiaLite)

THANK YOU!

ÉRIC LEMOINE

Developer @ Oslandia

✉ eric.lemoine@oslandia.com

👤 [@elemoine](#)

🐦 [@erilem](#)