



DB LAB

POSTGRESQL & SPATIAL DATA

Alberto Belussi

2019/2020

POSTRESQL & SPATIAL DATA

PostgreSQL implements the standard OGC: ***Simple Features Specification For SQL*** (currently OGC Simple Features Access – part 2: SQL option)

In particular, by activating the extension PostGIS, it allows one to:

- Define attributes containing geometries
 - By using the `geometry` type
 - or by invoking the function
AddGeometryColumn(schema, table, attributeGeo, srid, geoType, dimension)
- Loading vector data contained in a shape file in a temporary table
 - By using the `shp2pgsql` command.



PostGIS

A rich set of functions is available with the PostGIS extension, among them we will use the following ones:

- Functions for accessing the content of a geometric attribute
 - ST_AsText(geom), ST_AsEWKT(geom)
- Functions for inserting a value in a geometric attribute
 - ST_GeomFromText(WKT),
ST_GeomFromEWKT(EWKT)



PostGIS

- Functions for testing topological relations
 - `boolean ST_Disjoint(geometry A, geometry B);`
 - `boolean ST_Touches(geometry A, geometry B);`
 - `boolean ST_Within(geometry A, geometry B);`
 - `boolean ST_Contains(geometry A, geometry B);`
 - `boolean ST_Equals(geometry A, geometry B);`
 - `boolean ST_Overlaps(geometry A, geometry B);`
 - `boolean ST_Crosses(geometry g1, geometry g2);`
- Functions regarding the distance between geometries:
 - `float ST_Distance(geometry g1, geometry g2);`
 - `geometry ST_Buffer(geometry g1, float radius_of_buffer);`



EXERCISE

FIRST STEP

- Download from the the elearning platform the files: fiu.zip e lag.zip.
- Unzip the files.
- From pgAdmin 4, perform a restore on your database of both files (choose the option “restore...” in the pop up menu on the database name). If you have problems with the restore invocation, check in the preference options if the binary path is correctly set. The path for the PC of the lab is: /usr/bin
- Refresh the database content.
- In the schema public you should now have two additional tables: fiumi_geo and laghi_geo.



EXERCISE

SECOND STEP

- From pgAdmin 4, open a query tool window.
- Write an SQL command to create the table Fiume_geo in the schema psycho_db as (SELECT gid, idro_id, the_geom FROM fiumi_geo)
- Write an SQL command to create the table Lago_geo in the schema psycho_db as (SELECT gid, nome, the_geom FROM laghi_geo)



EXERCISE

THIRD STEP

- Write the following queries on the tables: fiume_geo, lago_geo, comune_geo:
 - Find the WKT of the geometries representing rivers that intersects (not disjoint) municipalities of the province of Verona (cod_pro =23) reporting in the result the WKT and the gid of the river.
 - Find the municipalities that are intersected by a river (not disjoint) returning the name of the municipalities and the idro_id of the river.
 - Find the municipalities that are adjacent to the municipality of Vicenza reporting their names and the number of patients who live or lived there.
 - Find the name of the municipalities that are adjacent to municipalities that intersect the Lake of Garda.



CONNECTION TO POSTGRESQL

- How can you connect to your database using psql?
 - **export PGUSER=loginGIA**
 - **psql -h <server> -d <database>**
psql -h dbserver.scienze.univr.it -d XXXX

