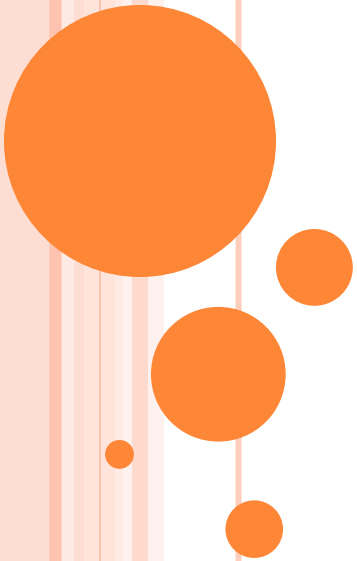

DB Lab
PostgreSQL & spatial Data

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 shapefile (data format of ESRI systems) 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): to be used for 3D geometries
- Functions for inserting a value in a geometric attribute
  - ST_GeomFromText(WKT),
  - ST_GeomFromEWKT(EWKT)

  Starting from a WKT or EWKT string they generate the binary representation of the geometric value as an instance of the type `geometry`.

Exercise

FIRST STEP

- Download from the the elearning platform the files: fiu.sql e lag.sql.
- From pgAdmin 4, open a query tool window and load each file into the window. Now execute the sql code. (If you have problems with the invocation of SQL commands try to specify the path of the scripts: the path for the PC of the lab is: /usr/bin)
- Refresh the database content.
- In the schema **psycho_db** you should now have two additional tables: fiumi_shp and laghi_shp.

EXERCISE

SECOND STEP

- From pgAdmin 4, open a query tool window.
- Write an SQL command to create the table **Fiume_geo** with schema:
  - gid INTEGER PRIMARY KEY
  - idro_id INTEGER NOT NULL,

  in the schema **psycho_db.**

  Add the geometry column **geom** of type **MultiLineString** and **SRID = -1** by using the function:

  AddGeometryColumn(schema, table, attributeGeo, srid, geoType, dimension)
- INSERT into the table just created the content of the table fiumi_shp as **(SELECT gid, idro_id, the_geom FROM psycho_db.fiumi_shp)**

Exercise

THIRD STEP

- From pgAdmin 4, open a query tool window.
- Write an SQL command to create the table **Lago_geo** with schema:
  - gid INTEGER PRIMARY KEY
  - Nome VARCHAR(200),

  in the schema **psycho_db.**

  Add the geometry column **geom** of type **MultiPolygon** and **SRID = -1** by using the function:

  AddGeometryColumn(schema, table, attributeGeo, srid, geoType, dimension)
- INSERT into the table just created the content of the table laghi_shp as **(SELECT gid, nome, the_geom FROM psycho_db.laghi_geo)**

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
- How can we see the content of a column containing geometries?
  - Download openJUMP 1.13
  - Or use QGis if available in the lab computers.