

Exercises for geospatial databases

Practical part of the course Advanced databases, on the topic of Geospatial databases will be done using PostgreSQL database management system (with PostGIS add-on). Test data will be available on official course webpages.

To be able to execute spatial queries on your computer you will have to perform following steps:

1. Install PostgreSQL database management system.
 - available for download on <http://www.postgresql.org/>
 - during installation it is recommended to use default settings
2. Install PostGIS add-on for PostgreSQL
 - Use Application Stack Builder included in PostgreSQL installation
3. Load test data to your database
 - You can load your test data using *shp2pgsql* and *psql* command line tools. They are located in /bin folder of your PostgreSQL installation. I.e. folder could be D:\PostgreSQL\8.3\bin.
 - You can use *shp2pgsql* tool to get a list of SQL INSERT instructions you can use to insert data into your database, while you can use *psql* tool to execute those instructions if you previously stored them in a file. Detailed documentation on those two tools and PostGIS in general can be found on <http://postgis.refractory.net/documentation/>.
 - Test data can also be loaded using graphical interface *PostGIS Shapefile Import/Export Manager* (version of PostGIS 2.0 and higher).
 - Test data contains four tables: parcela (parcel), brojparcele (parcel number), stambenazgrada (residential building) and poslovnazgrada (business building).
4. After loading test data, you can view them using QGIS.
 - Quantum GIS is available at: <http://www.qgis.org/>, it is recommended to use latest version
 - To connect to a PostGIS database use option: **Layer -> Add a PostGIS Layer**
5. Now you can pose queries over test data
 - Use option **Tools -> Query Tool** in pgAdmin III included in PostgreSQL installation.

Data model

Test data contains four tables: parcela (parcel), brojparcele (parcel number), stambenazgrada (residential building) and poslovnazgrada (business building). Tables are not connected using foreign key constraints which means that, for example, connection between a building and a parcel can only be determined using spatial properties

Table attributes can be viewed using pgAdmin III included in PostgreSQL installation. Each table contains attribute **gid** which is a key, and attribute **geom** representing spatial attribute. Parcels and buildings are polygons while parcel numbers are points.

Problems

PostGIS supports more than a hundred different functions, and these exercises use only a small number of those. Details on specific functions can be found in PostGIS documentation.

1. Display all data on parcel with gid 303.

```
SELECT * FROM parcela
WHERE gid = 303;
```

2. Display area of parcel with gid 303.

```
SELECT ST_Area(geom)
FROM parcela
WHERE gid = 303
```

3. Display all data on parcels that touch parcel with gid 4366.

```
SELECT parcl.* FROM parcela parcl, parcela parc2
WHERE parc2.gid = 4366
AND ST_Touches(parcl.geom, parc2.geom)
```

4. Display all data on business buildings that are situated on parcel with gid 2996.

```
SELECT * FROM poslovnazgrada pz, parcela parc
WHERE parc.gid = 2996
AND ST_Within(pz.geom, parc.geom)
```

5. Display gid and area of all residential buildings on parcel with gid 2996.

```
SELECT sz.gid, ST_Area(sz.geom)
FROM stambenazgrada sz, parcela parc
WHERE parc.gid = 2996
AND ST_Within(sz.geom, parc.geom)
```

6. Display number and area of residential buildings on parcel with gid 2284.

```
SELECT COUNT(*), SUM(ST_Area(sz.geom))
FROM stambenazgrada sz, parcela parc
WHERE parc.gid = 2284
AND ST_Within(sz.geom, parc.geom)
```

7. Display gid of all parcels with gid less than 1000 that have more than one residential building.

```
SELECT parc.gid
  FROM parcela parc, stambenazgrada sz
 WHERE ST_Within(sz.geom, parc.geom)
       AND parc.gid < 1000
 GROUP BY parc.gid
 HAVING COUNT(*) > 1
```

8. Display gid of all parcels that are completely covered by a residential building. (geometry of a parcel is equal to geometry of a building).

```
SELECT parc.gid FROM parcela parc
 WHERE EXISTS (SELECT * FROM stambenazgrada sz
              WHERE ST_Equals(parc.geom, sz.geom))
```

9. Display all data on parcels that contain residential building and do not contain business building.

```
SELECT * FROM parcela parc
 WHERE NOT EXISTS (SELECT * FROM stambenazgrada sz
                 WHERE ST_within(sz.geom, parc.geom))
 AND EXISTS (SELECT * FROM poslovnazgrada pz
            WHERE ST_Within(pz.geom, parc.geom))
```

10. Display gid and area of the smallest parcel (parcel with the smallest area). Assume there is only one.

```
SELECT gid, ST_Area(parc.geom) AS area
  FROM parcela parc
 ORDER BY area ASC
 LIMIT 1
```

11. Display type of geometry attribute for parcel number with gid 815.

```
SELECT ST_GeometryType(geom) FROM brojparcele
 WHERE gid = 815
```

12. Display distance between two parcel numbers with gid 801 and 816.

```
SELECT ST_Distance(bp1.geom, bp2.geom)
  FROM brojparcele bp1, brojparcele bp2
 WHERE bp1.gid = 801
       AND bp2.gid = 816
```

13. Display all data on business buildings whose distance from parcel number with gid 4152 is less than 50 meters.

```
SELECT pz.*
  FROM poslovnazgrada pz, brojparcele bp
 WHERE bp.gid = 4152
       AND ST_Distance(bp.geom, pz.geom) < 50
```