# Fakultet elektrotehnike i računarstva Zavod za primjenjeno računarstvo

# Napredni modeli i baze podataka

1. Vježba za laboratorij profila

frajer

U prvom zadatku bilo je potrebno spajanjem županija podijeliti Hrvatsku u četiri regije i te regije prikazati na karti.

To je postignuto korištenjem funkcije ST\_Union, tj. ugnježđivanjem iste nekoliko puta s obzirom da prima samo 2 parametra tipa 'geom'. Na sve skupa još je primijenjena funkcija ST\_SetSRID kako bi se postavio odgovarajući SRID na novonastalim 'geom' tipovima i omogućio daljnji rad s njima.

```
NTE OR REPLACE VIEW zupanije AS

CT row_number() over() as id,

Inion(ST_Union(z1.geom, z2.geom), ST_Union(z3.geom, z4.geom)) as prva,

Inion(ST_Union(z5.geom, z6.geom), ST_Union(z7.geom, z8.geom)) as druga,

Inion(ST_Union(ST_Union(ST_Union(z6.geom, z10.geom), ST_Union(z11.geom, z12.geom)), ST_Union(z13.geom, z14.geom)), z15.geom) as treca,

Inion(ST_Union(ST_Union(ST_Union(z16.geom, z17.geom), ST_Union(z18.geom, z19.geom)), ST_Union(z20.geom, z21.geom)) as cetvrta

if hrv_adm1 as z1, hrv_adm1 as z2, hrv_adm1 as z3, hrv_adm1 as z4,

adm1 as z5, hrv_adm1 as z6, hrv_adm1 as z7, hrv_adm1 as z8,

adm1 as z9, hrv_adm1 as z10, hrv_adm1 as z11, hrv_adm1 as z12, hrv_adm1 as z13, hrv_adm1 as z14, hrv_adm1 as z15,

adm1 as z16, hrv_adm1 as z17, hrv_adm1 as z18, hrv_adm1 as z19, hrv_adm1 as z20, hrv_adm1 as z21

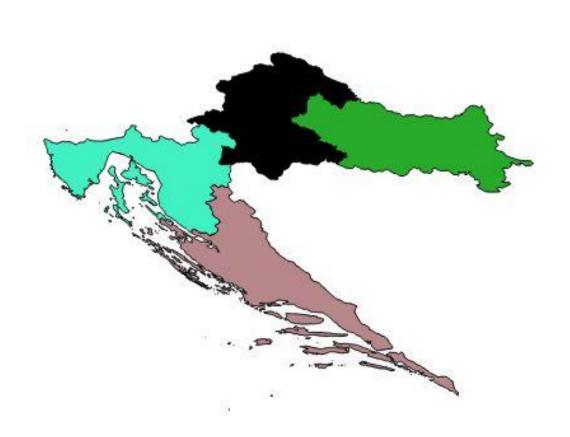
EE z1.name_1='Zadarska' AND z2.name_1='Splitsko-Dalmatinska' AND z3.name_1='Dubrovacko-Neretvanska' AND z4.name_1='Sibensko-Kninska' AND

name_1='Primorsko-Goranska' AND z10.name_1='Istarska' AND z7.name_1='Licko-Senjska' AND z8.name_1='Karlovacka' AND

name_1='Medimurska' AND z10.name_1='Splacko-Moslavacka' AND z11.name_1='Grad Zagreb' AND z12.name_1='Krapinsko-Zagorska' AND z13.name_1='Varaždinska' AND

name_1='Brodsko-Posavska' AND z19.name_1='Požeško-Slavonska' AND z10.name_1='Viroviticko-Podravska' AND z11.name_1='Bjelovarska-Bilogorska'

AND z19.name_1='Požeško-Slavonska' AND z20.name_1='Viroviticko-Podravska' AND z21.name_1='Bjelovarska-Bilogorska'
```



U drugom zadatku trebalo je označiti na karti rijeku Savu i odrediti kolika je duljina rijeke u našim regijama kreiranim u prethodnom zadatku.

Savu smo 'uzeli' iz tablice 'waterways' u kojoj je ona rastavljena u nekoliko manjih dijelova.

```
CREATE OR REPLACE VIEW sava AS
SELECT *
FROM waterways
WHERE name = 'Sava'
```

Nakon toga, da bi dobili rijeku Savu u jednom komadu iskorišteni su podaci iz pogleda 'sava' i napravljen je novi pogled 'sava2'.

```
CREATE OR REPLACE VIEW sava2 AS

SELECT row_number() OVER () AS id,

st_setsrid(st_union(st_union(st_union(st_union(s1.geom, s2.geom), st_union(s3.geom, s4.geom)), st_union(st_union(s5.geom, s6.geom), st_union(s7.geom, s8.geom))), s9.geom), 4326)

AS sava

FROM sava s1, sava s2, sava s3, sava s4, sava s5, sava s6, sava s7, sava s8, sava s9

WHERE s1.gid = 7 AND s2.gid = 466 AND s3.gid = 1210 AND s4.gid = 1211 AND s5.gid = 1234 AND s6.gid = 1235 AND s7.gid = 1518 AND s8.gid = 3085 AND s9.gid = 3773;
```

Budući da je očigledno da kroz dvije od četiri regije Sava uopće ne prolazi, izračunali smo samo za dvije regije kroz koje prolazi. Koristeći funkcije ST\_Intersection i ST\_Length te funkciju ST\_Transform (kako bi rezultat bio u metrima)

```
CREATE OR REPLACE VIEW savatreca AS

SELECT row_number() OVER () AS id,

st_intersection(sava2.sava, zupanije.treca) AS st_intersection,

st_length(st_transform(st_intersection(sava2.sava, zupanije.treca), 3765)) AS duljina

FROM sava2,

zupanije;
```

```
CREATE OR REPLACE VIEW savacetvrta AS

SELECT row_number() OVER () AS id,

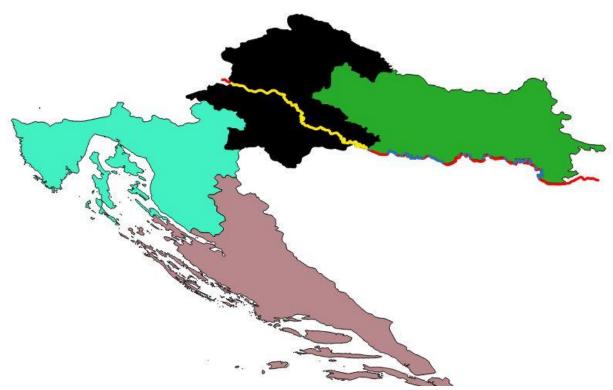
st_intersection(sava2.sava, zupanije.cetvrta) AS st_intersection,

st_length(st_transform(st_intersection(sava2.sava, zupanije.cetvrta), 3765)) AS duljina

FROM sava2,

zupanije;
```

Rezultat toga u QGISu je sljedeći:



Sava u cjelini je označena crvenom bojom, žuto je dio koji se nalazi u trećoj regiji, a plavo dio u četvrtoj. S obzirom da Sava djelomično u četvrtoj regiji zalazi u teritorij Bosne i Hercegovine duljina Sava u toj regiji je zapravo manja nego u trećoj: 83km naspram 219km.

#### 3. Zadatak

U trećem zadatku trebalo je odrediti koliko se najmanje mora izliti rijeka Sava da bi pogodila prvu zgradu i to prikazati na karti. Prvi pokušaj bio je izračun udaljenosti napraviti unutar ST\_Buffer funkcije no taj upit trajao je predugo i otvaranje toga u QGIS-u bilo je zbog toga nemoguće. Zato smo napravili novu tablicu u kojoj je pohranjena ta udaljenost:

```
CREATE TABLE dist AS (

SELECT MIN(ST_Distance(sava,geom))

FROM sava2, buildings
```

Zatim je napravljen pogled 'buffer' koji koristi prethodno izračunatu minimalnu udaljenost do prve zgrade.

```
CREATE OR REPLACE VIEW buffer AS

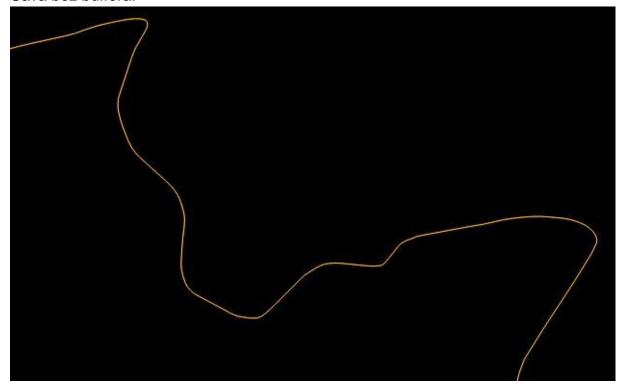
SELECT row_number() OVER () AS id, st_buffer(sava2.sava, dist) AS st_buffer

FROM sava2, dist

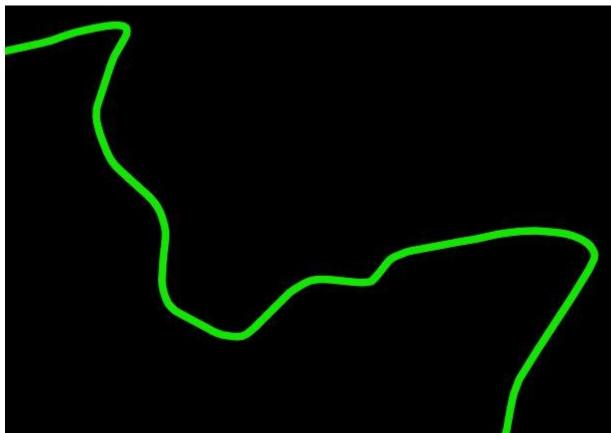
GROUP BY sava2.sava, dist;
```

S obzirom da je to samo 54 metra, buffer na karti nije baš vidljiv.

## Sava bez buffera:



## Sava s bufferom:



U četvrtom zadatku bilo je potrebno prikazati hrvatske županije različitim nijansama zelene boje ovisno o površini šuma u svakoj od njih. Prvo je napravljena nova tablica u kojoj su pohranjeni podaci o ukupnom zbroju površine šuma po županijama.

```
CREATE TABLE sume AS (
SELECT name_1, SUM(ST_Area("natural".geom)) as povrsina
FROM "natural", hrv_adm1
WHERE ST_Contains(ST_SetSRID(hrv_adm1.geom, 4326), "natural".geom) = TRUE AND type='forest'
GROUP by name_1
ORDER By povrsina DESC
```

Zatim je napravljen pogled kojim se uzimaju ti podaci te spajaju s 'geom' podacima o županijama kako bi se pravilno prikazali na karti.

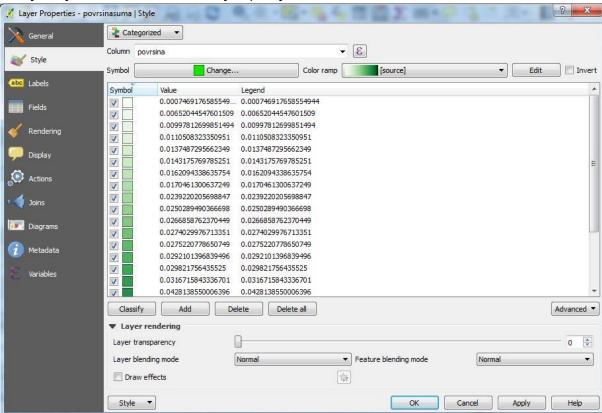
```
CREATE OR REPLACE VIEW povrsinasuma AS

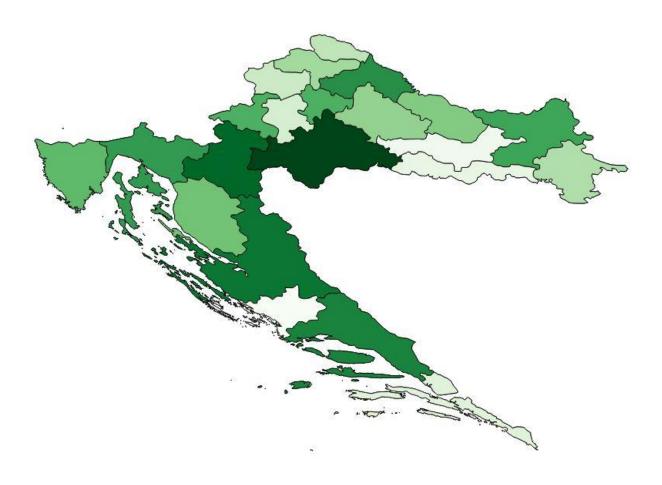
SELECT row_number() OVER () AS id, sume.name_1, sume.povrsina, hrv_adm1.geom

FROM sume, hrv_adm1

WHERE sume.name_1 = hrv_adm1.name_1;
```

Nakon što je taj pogled dodan u qgis, i u postavkama namjestimo Style -> Categorized -> Column: povrsina i podesimo zeleni spektar boja nakon čega na karti dobivamo željeni prikaz. Vidimo da najviše šuma ima u Sisačko-Moslavačkoj, a najmanje u Šibensko-Kninskoj županiji.





Trebalo je prikazati regije koje gravitiraju gradovima s više od 30000 stanovnika. Prvo je napravljen pogled koji je izbacio te gradove.

```
CREATE OR REPLACE VIEW gradovi AS
select row_number() over() as id, name, population, geom
from places
where population > 30000 AND type!='country'
```

Zatim smo u gqisu napravili sljedeće: Vector -> Geometry Tools -> Voronoi polygons te kreirali novi shape file i dodali ga na kartu.

