**Rješenja:**

**1.**     **(5 bodova)**

(SELECT terUnit.terUnitId, subservTerUnitId.terUnitId

   FROM terUnit, terUnit subservTerUnitId

  WHERE terUnit.terUnitId = subservTerUnitId.supTerUnitId

   AND terUnit.supTerUnitId IS NULL

UNION

SELECT terUnitResuts.terUnitId, subservTerUnitId.terUnitId

  FROM terUnitResuts, terUnit subservTerUnitId

WHERE terUnitResuts.terUnitResutsId = subservTerUnitId.supTerUnitId

)

  SELECT terUnit.terUnitName, SUM(result.voteFor), SUM(1-result.voteFor)

    FROM  terUnitResuts, poolPlace, result, terUnit

  WHERE terUnitResuts.terUnitResutsID = poolPlace.terUnitId

     AND result.poolPlaceId = poolPlace.poolPlaceId

     AND terUnitResuts.terUnitId = terUnit.terUnitId

     AND valid  = 1

   GROUP BY terUnit.terUnitName, terUnit.terUnitId

**2.**     **(4 boda)**

Shemu ***poolPlace***treba proširiti atributom polPlacePeriod tipa DATERANGE, ažurirati vrijednost tog atributa, uništiti postojeći PRIMARY KEY i kreirati temporalni primarni ključ.

Shemu ***personRef***treba proširiti atributom pollPlacePeriod tipa DATERANGE, ažurirati mu vrijednost i kreirati temporalni strani ključ. PostgreSQl ga ne podržava.

CREATE EXTENSION bTree\_gist; //studenti ne moraju napisati, ali inače treba

ALTER TABLE poolPlace ADD poolPlacePeriod DATERANGE;

ALTER TABLE poolPlace ADD  PRIMARY KEY (poolPlaceId, poolPlacePeriod) ;

ALTER TABLE poolPlace ADD CONSTRAINT temporalPKPoolPlace EXCLUDE USING gist

 (poolPlaceId WITH =, poolPlacePeriod WITH &&)

ALTER TABLE personRef ADD polPlacePeriod DATERANGE;

PostgreSQL ne podržava temporalni referencijski integritet.

SQL standard za temporalni referencijski integritet predviđa da je period referencirajuće n-torke u potpunosti sadržan u periodu jedne referencirane n-torke ili u kombiniranom periodu dvije ili više uzastopnih referenciranih n-torki.

Primijenjeno na gornje dvije relacije to bi značilo sljedeće:

Za isti poolPlaceId (poolPlace.poolPlaceId = personRef.poolPlaceId) personRef.poolPlacePeriod treba biti u potpunosti sadržan u poolPlace.poolPlacePeriod -u jedne n-torke ili u kombiniranom periodu dvije ili više uzastopnih referenciranih n-torki

**3.**     **(4 boda)**

Upiti se sporo izvode jer se za 100 miliona zapisa „on the fly“ izračunava vrijednost funkcijeTO\_tsVector – tj određuje sortirana lista leksema kao prezentacija dokumenta nakon provedenog postupka „normalizacije“ teksta sadržanog u atributu contractContent.

Upit se može ubrzati tako da se:

1.  shema relacije contract proširi atributom npr contractContentTSV tipa TSVECTOR, koji će sadržavati reprezentaciju teksta u normaliziranom obliku:  
ALTER TABLE contract ADD contractContentTSV TSVECTOR;  
i sadržaj tog atributa ažurira i održava ažurnim pomoću okidača:  
CREATE TRIGGER contract\_InsUpd\_Trigg BEFORE INSERT OR UPDATE ON contract

FOR EACH ROW EXECUTE PROCEDURE tsvector\_update\_trigger (contractContentTSV,'english', contractContent);

2.       kreira invertirani indeks nad atributom contractContentTSV:

CREATE INDEX contractContentTSVIdx ON contract   
       USING gin(contractContentTSV); //ili gist

3.       U upitima koristiti contractContentTSV umjesto contractContent:

SELECT \*

  FROM contract

WHERE TO\_tsVector('english', contractBody) @@

      TO\_tsquery('english', 'Europian & independency & I & am');

**GIS**

**(a)**

Napraviti vizualizaciju rezultata izbora po najmanjim teritorijalnim  
jedinicama (najnižima u hijerarhiji), poput ove preuzete s wikipedije:   
Opišite kako biste napravili takvu vizualizaciju.  
Potrebno je opisati potencijalne izmjene na bazi u pogledu dodatnih   
podataka, te SQL naredbe kojima se pripremaju podatci potrebni   
za vizualizaciju.

// priznaje se i ako grupiraju po geom, inače trebalo bi napraviti podupit u select

// listi ili sl.

SELECT terUnitId, geom, 100 \* sum(voteFor) / COUNT(\*) as remain

  FROM terUnit

  JOIN poolPlace ON terUnit.terUnitId = poolPlace.terUnitId

  JOIN result ON poolPlace.poolPlaceId = result.poolPlaceId

WHERE valid = 1

   AND NOT EXISTS (SELECT \* from terUnit child WHERE superUnitId = terUnit.terUnitId)

GROUP BY terUnitId, geom

Ovaj upit je ulaz u alat za vizualizaciju kao QGIS, gdje onda treba jednostavno dodijeli boja na temelju one mjere „remain“.

**(b)**

CREATE VIEW glasaleZa ASSELECT terUnitId, geom  
  FROM terUnit   
  JOIN poolPlace ON terUnit.terUnitId = poolPlace.terUnitId  
  JOIN result ON poolPlace.poolPlaceId = result.poolPlaceId  
WHERE valid = 1  
   AND NOT EXISTS (SELECT \* from terUnit child WHERE superUnitId = terUnit.terUnitId)  
GROUP BY terUnitId, geom  
HAVING SUM(voteFor) >  SUM(1 - voteFor);

MIN = {id: null, relArea: Float.MAX};

**foreach** (unit in glasaleZa) {

  LET cnt= SELECT COUNT(\*)

             FROM glasaleZa

            WHERE st\_touches(unit.geom, glasaleZa.geom)

              AND unit.terUnitId <> glasaleZa.terUnitId

  IF (cnt == 0) {  // Nema susjeda istomišljenika

    currRelArea = SELECT st\_area(unit.geom)/SUM(st\_area(geom))

                    FROM terUnit

                   WHERE st\_touches(unit.geom, terUnit.geom)

                     AND NOT EXISTS (SELECT \* from terUnit child

                             WHERE superUnitId = terUnit.terUnitId);

                   // ne treba gledati glasove, svi susjedi su protiv

    IF (MIN.relArea < currRelArea) {

       MIN.relArea = currRelArea;

       MIN.id = unit.terUnitId;

    }

  }

}

PRINT MIN

**M/R rješenje**:

**map** (k, v) {

   // nije nužno množiti s valid,

   emit(v.poolPlaceId, {

stay:  (v.voteFor == 1) \* v.valid,

leave: (v.voteFor == 0) \* v.valid,

invalid: (1 – v.valid)

});

}

**reduce** (k, vlist) {

   var agg = { stay: 0, leave: 0, invalid: 0};

   foreach (v in vlist) {

      agg.stay    += v.stay;

      agg.leave   += v.leave;

      agg.invalid += v.invalid;

   }

   return agg;

}

// finalize nije potreban

**SEMWEB**

PREFIX foaf:     <http://.../>

PREFIX pets:  [http://.../](https://postar.fer.hr/owa/UrlBlockedError.aspx)

...

SELECT DISTINCT ?name  (COUNT(?friend) AS ?num)

 WHERE {  
   ?user foaf:firstName ?name .  
   ?user foaf:knows ?friend .  
   ?friend pets:hasPet ?pet .

   ?pet rdf:type pets:Dog.

}

GROUPY BY ?name