1. Zadatak

```
WITH RECURSIVE TURecursive (terUnitId, terUnitName, supTerUnitId, votesFor, votesAgainst)
AS (SELECT terUnitId,
        supTerUnitId,
        CASE WHEN voteFor = 1 THEN 1 ELSE 0 END AS votesFor,
        CASE WHEN voteFor = 1 THEN 0 ELSE 1 END AS votesAgainst
        FROM result, poolPlace
    WHERE result.poolPlaceId = poolPlace.poolPlaceId
        AND poolPlace.terUnitId = terUnit.terUnitId
        AND valid = 1

    UNION

    SELECT parent.terUnitId, parent.terUnitName childTerUnit.supTerUnitId,
    childResult.votesFor, childResult.votesAgainst
        FROM TURecursive parent, terUnit childTerUnit, poolPlace childPoolPlace,
            result childResult
            WHERE parent.supTerUnitId = childTerUnit.terUnitId AND
             childPoolPlace.poolPlaceId = childResult.poolPlaceId AND
            childPoolPlace.terUnitId = childTerUnit.terUnitId
    )

    SELECT terUnitName,
            SUM(votesFor) AS votesFor,
            SUM(votesAgainst) AS votesAgainst
            FROM TURecursive WHERE terUnitId NOT IN (SELECT DISTINCT
        terUnitId FROM terUnit WHERE supTerUnitId IS NOT NULL)

GROUP BY terUnitId, TURecursive.terUnitName
```

2. Zadatak

```
ALTER TABLE poolPlace ADD terUnitPeriod DATERANGE;
UPDATE poolPlace SET terUnitPeriod = '[dd.mm.yyyy, dd.mm.yyyy]' -- PK ne moze biti NULL
ALTER TABLE poolPlace ADD CONSTRANT pkPoolPlace PRIMARY KEY (poolPlaceId, terUnitPeriod);
```

PRIMARY KEY neće očuvati temporalni primarni ključ u PostgreSQL-u pa se takvo ograničenje implementira:

```
ALTER TABLE poolPlace ADD CONSTRAINT noOverlapPoolPlace EXCLUDE USING gist (poolPlaceID WITH ==, terUnitPeriod WITH &&);
```

ALTER TABLE personRef ADD CONSTRAINT actualPoolPlace FOREIGN KEY (refDate, poolPlaceId) REFERENCES poolPlace (poolPlaceId, terUnitPeriod); -- mozda bi refDate trebalo pretvoriti u RANGE, nemam pojma kako ovaj FK radi.

Temporalni strani ključ u personRef nije moguće dodati u PostgreSQL-u, implementacija bi se mogla napraviti pomoću okidača i procedura.

3. Zadatak

Potrebno je:

- dodati novi atribut contractContent_tsv tipa TSVECTOR koji bi čuvao sadržaj u normaliziranom obliku.
- održavati taj atribut ažurnim za što je potrebno napraviti okidač za INSERT i UPDATE naredbe, a može se dobiti pomoću to_tsvector funkcije.
- napraviti invertirani indeks nad atributom contractContent_tsv:
  CREATE INDEX contractContentIdx ON contract USING gin(contractContent_tsv);

4. Zadatak

A)

```
CREATE VIEW terUnitResults AS
SELECT terUnit.terUnitId, terUnitName,
    1. * SUM(CASE WHEN voteFor = 1 AND valid = 1 THEN 1 ELSE 0 END) AS
    leavePercentage,
    1. * SUM(CASE WHEN voteFor = 0 AND valid = 1THEN 1 ELSE 0 END) AS
    remainPercentage,

    FROM result JOIN poolPlace ON poolPlace.poolPlaceId = result.poolPoolPlaceId
        JOIN poolPlace.terUnitId = terUnit.terUnitId
    WHERE supTerUnitId != NULL --ovo je vjerojatno krivo jer postoji jos nivoa hijerarhije
GROUP BY terUnit.terUnitId, terUnitName
```

Gornji upit spojiti s terUnit, kako bi se dobili podatci: geom, terUnitName, leavePercentage, remainPercantage.

B)

Ne znam bas.

5. Zadatak

```
function map (k, v) {
    if (v.refDate == '23.06.2016') {
        voteAgainst = v.voteFor == 1 ? 0 : 1;
        emit(v.poolPlaceId,
            { voteStay: voteAgainst, voteLeave: foteFor, invalidVote: v.valid });
    }
}

function reduce (k, vlist) {
    let agg = { voteStay: 0, voteAgainst: 0, invalidVote: 0 };
    foreach (v in vlist) {
        agg.voteStay += v.voteStay;
        agg.voteAgainst += v.voteAgainst;
        agg.invalidVote += v.invalidVote;
    }
    return agg;
}

function finalize (k, vlist) {
    return v.list.sort(); // pretpostavka
}
```

6. Zadatak

Za konzistenciju kod pisanja dovoljna je većina: W > N/2
Za konzistenciju kod čitanja dovoljno je: W + R > N (konzistencija kod čitanja ovisi o W)

Možemo imati konzistenciju kod čitanja i kad nemamo kod pisanja - čitati sve čvorove! Što više čvorova je uključeno u operaciju bolja je konzistencija, ali veća latencija i obratno. Primjer: ako želimo brzo čitanje onda žrtvujemo pisanje: N=3, W=3, R=1

7. Zadatak

Ne da mi se crtati pa opisem podatke u N3/Turtle formatu, nekako otprilike.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix pet: <http://..#> .

<http://www.peanuts.com/charlie.brown>
        rdf:type  foaf:Person ;
        foaf:firstName "Charlie" ;
        foaf:knows <http://www.peanuts.com/linus.van.pelt> ;
        foaf:owns [
            rdf:type  pet:Cat ;
            bio:name      “Frodo”
        ]
<http://www.peanuts.com/linus.van.pelt>
        rdf:type  foaf:Person ;
        foaf:firstName      “Linus”
        foaf:knows <http://www.peanuts.com/charlie.brown> ;
        foaf:owns [
            rdf:type  pet:Dog;
            bio:name “Milo”
        ]
<http://www.peanuts.com/chuck.norries>
        rdf:type  foaf:Person ;
        foaf:firstName      “Chuck”
```

Upit:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX pet: <http://..#>

SELECT ?name, COUNT(?friend)
WHERE {
    ?person rdf:type  foaf:Person,
    ?person foaf:firstName      ?name
    ?person foaf:knows    ?friend
    ?friend   foaf:owns     ?pet
}
```