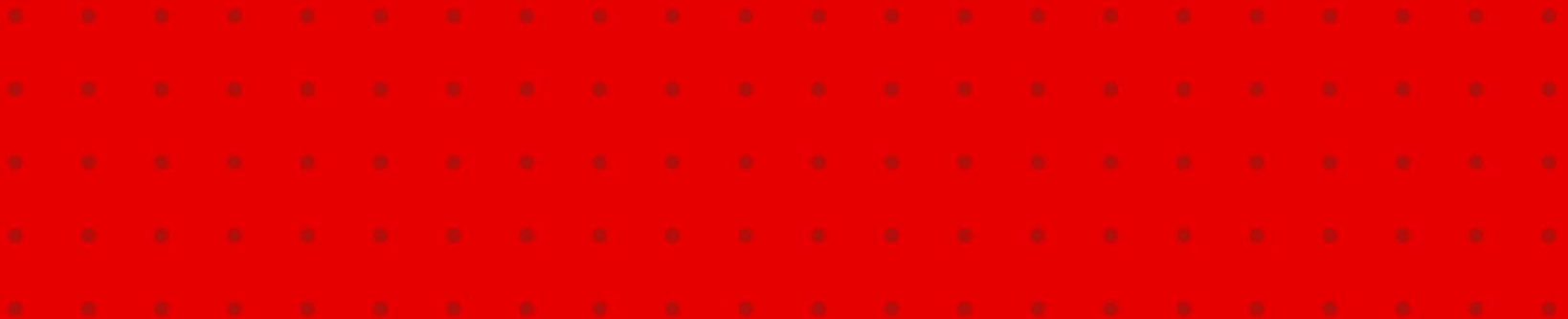


Utilizarea avansată a tabelelor compuse în PostGIS



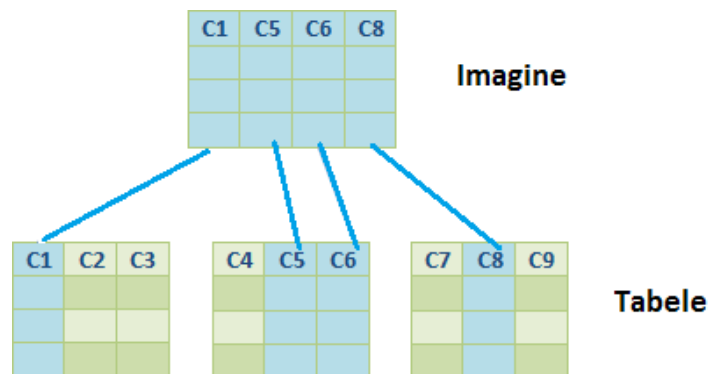
Cuprins

- Ce sunt tabelele compuse?
- De ce este nevoie de ele?
- Editarea geometriei in tabele compuse
- Concluzii

1

Ce sunt tabele compuse?

• Tabele compuse



- Imagini ale bazei de date (VIEW)
- Rezultatul unor operații de compunere pe baza unui coloane cu valori comune (JOIN ... ON ...)

• Tabele compuse - exemplu

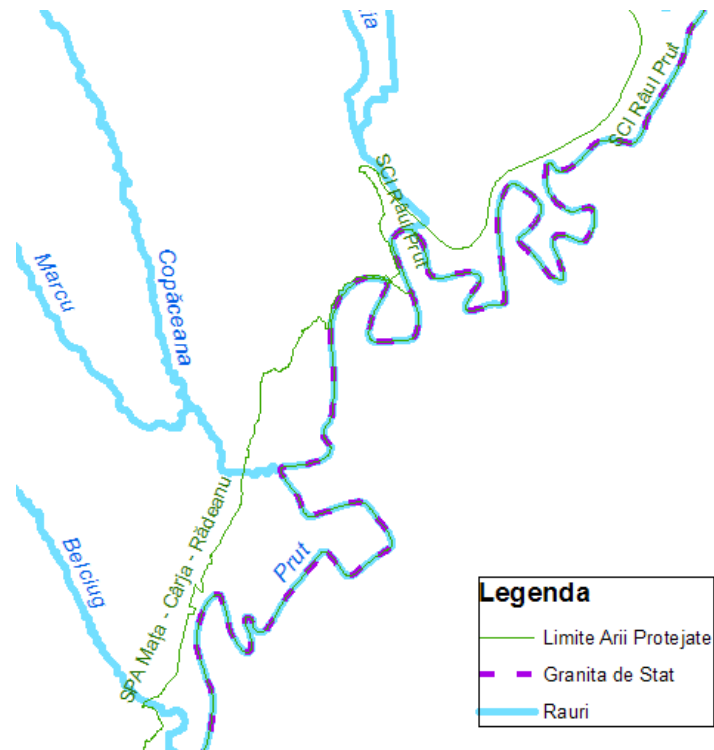
```
CREATE OR REPLACE VIEW public.sv_addressareas AS
SELECT "AddressAreas".id,
       "AddressAreaName"."SIRUTA" AS siruta_loc,
       "AddressAreaName"."VALUE" AS den_loc,
       "AddressAreas".geom,
       "AdministrativUnitName"."SIRUTA" AS siruta_uat,
       "AdministrativUnitName"."VALUE" AS den_uat,
       "AdministrativUnitName"."COUNTY" AS judet
FROM "AddressAreaName",
     "AddressAreas",
     "AdministrativUnitName"
WHERE "AddressAreaName"."IDUAT"::text = "AdministrativUnitName"."ID"::text
AND "AddressAreas"."IDADDRESSAREA"::text = "AddressAreaName"."ID"::text;
```

2

**De ce este nevoie de ele?
(in GIS)**

Utilitatea tabelelor compuse

- Refolosirea atributelor si geometriei
 - reducerea spațiului de stocare
 - asigurarea coincidenței geometriei din straturi din teme diferite
 - utilizarea de domenii



Exemplu - Directiva INSPIRE

Recommendation 2 The objects in the *Hydrography* theme should be positionally consistent with spatial objects from other themes (e.g. with buildings and waterways)

INSPIRE Data Specification on Hydrography



Recommendation 6 The objects in the *Transport Networks* theme should be positionally consistent with spatial objects from other themes (e.g. with buildings and rivers, forestry extents)

INSPIRE Data Specification on Transport Networks

IR Requirement

Annex IV, Section 11.4.1

Theme-specific Requirements – Management Restriction Or Regulation Zones

- (1) Where the geometry of the spatial object is derived from another spatial object, the geometries of the two objects shall be consistent.

INSPIRE Data Specification on Area Management/ Restriction/ Regulation Zones and Reporting Units

3

Editarea geometriei in tabele compuse

Pași

- Identificarea operațiilor de editare (INSERT, UPDATE, DELETE)
- Identificarea coloanei cu geometrie si a tabelului din care provine
- Identificarea altor coloane care asigura consistența imaginii
 - de obicei, cel puțin cheia primară a tabelului cu geometrii
- Definirea operațiilor de efectuat pentru a simula editarea imaginii

Definirea operațiilor de editare

- PostgreSQL
 - [Reguli](#)
 - [Declanșatoare \(TRIGGER\)](#)
- Microsoft SQL Server
 - [Declanșatoare cu rescriere \(INSTEAD OF TRIGGER\)](#)
- Oracle
 - [Declanșatoare cu rescriere \(INSTEAD OF TRIGGER\)](#)

• PostgreSQL 9.5

- Reguli

• PostgreSQL 9.5

- Reguli

```
CREATE [ OR REPLACE ] RULE name AS ON event
  TO table
  DO INSTEAD (command | ( command ; command ... ) )
```

• PostgreSQL 9.5

- Reguli

```
CREATE [ OR REPLACE ] RULE name AS ON event  
  TO table  
  DO INSTEAD (command | ( command ; command ... ) )
```

- comenzi valide: SELECT, INSERT, UPDATE, DELETE

• PostgreSQL 9.5

- Reguli

```
CREATE [ OR REPLACE ] RULE name AS ON event  
  TO table  
  DO INSTEAD (command | ( command ; command ... ) )
```

- comenzi valide: SELECT, INSERT, UPDATE, DELETE

- Declanșatoare cu rescriere

• PostgreSQL 9.5

- Reguli

```
CREATE [ OR REPLACE ] RULE name AS ON event  
  TO table  
  DO INSTEAD ( command | ( command ; command ... ) )
```

- comenzi valide: SELECT, INSERT, UPDATE, DELETE

- Declanșatoare cu rescriere

```
CREATE TRIGGER name INSTEAD OF ( event [ OR ... ] )  
  ON table  
  FOR EACH ROW  
  EXECUTE PROCEDURE function name ( arguments )
```


Reguli PostgreSQL

```
CREATE OR REPLACE VIEW public.sv_adrese_distante_buc AS
SELECT rom_geom_buc.geom,
       rom_geom_buc.id,
       rom_ldec.den_st,
       rom_ldec.just_no,
       dist2sect.sv_id,
       dist2sect.dist,
FROM positions.rom_geom_buc,
     mapping.dist2sect,
     aaa_rom_ldec
WHERE dist2sect.aep_id = rom_geom_buc.id
AND aaa_rom_ldec.id = rom_geom_buc.id;
```

Reguli PostgreSQL

```
CREATE OR REPLACE VIEW public.sv_adrese_distante_buc AS
SELECT rom_geom_buc.geom,
       rom_geom_buc.id,
       rom_ldec.den_st,
       rom_ldec.just_no,
       dist2sect.sv_id,
       dist2sect.dist,
FROM positions.rom_geom_buc,
     mapping.dist2sect,
     aaa_rom_ldec
WHERE dist2sect.aep_id = rom_geom_buc.id
AND aaa_rom_ldec.id = rom_geom_buc.id;
```

Regulă pentru modificare

```
CREATE OR REPLACE RULE update_geom_sv_ad_dist AS
ON UPDATE TO sv_adrese_distante_buc
DO INSTEAD
UPDATE positions.rom_geom_buc SET geom = new.geom
WHERE rom_geom_buc.id = old.id;
```

Reguli PostgreSQL

```
CREATE OR REPLACE VIEW public.sv_adrese_distanta_buc AS
SELECT rom_geom_buc.geom,
       rom_geom_buc.id,
       rom_ldec.den_st,
       rom_ldec.just_no,
       dist2sect.sv_id,
       dist2sect.dist,
FROM positions.rom_geom_buc,
     mapping.dist2sect,
     aaa_rom_ldec
WHERE dist2sect.aep_id = rom_geom_buc.id
AND aaa_rom_ldec.id = rom_geom_buc.id;
```

Regulă pentru modificare

```
CREATE OR REPLACE RULE update_geom_sv_ad_dist AS
ON UPDATE TO sv_adrese_distanta_buc
DO INSTEAD
UPDATE positions.rom_geom_buc SET geom = new.geom
WHERE rom_geom_buc.id = old.id;
```

Reguli PostgreSQL

```
CREATE OR REPLACE VIEW public.sv_adrese_distante_buc AS
SELECT rom_geom_buc.geom,
       rom_geom_buc.id,
       rom_ldec.den_st,
       rom_ldec.just_no,
       dist2sect.sv_id,
       dist2sect.dist,
FROM   positions.rom_geom_buc,
       mapping.dist2sect,
       aaa_rom_ldec
WHERE  dist2sect.aep_id = rom_geom_buc.id
AND   aaa_rom_ldec.id = rom_geom_buc.id;
```

Regulă pentru ștergere

```
CREATE OR REPLACE RULE delete_geom_sv_ad_dist AS
ON DELETE TO sv_adrese_distante_buc
DO INSTEAD
DELETE FROM positions.rom_geom_buc
WHERE rom_geom_buc.id = old.id;
```

Reguli PostgreSQL

```
CREATE OR REPLACE VIEW public.sv_adrese_distanta_buc AS
SELECT rom_geom_buc.geom,
       rom_geom_buc.id,
       rom_ldec.den_st,
       rom_ldec.just_no,
       dist2sect.sv_id,
       dist2sect.dist,
FROM positions.rom_geom_buc,
     mapping.dist2sect,
     aaa_rom_ldec
WHERE dist2sect.aep_id = rom_geom_buc.id
AND aaa_rom_ldec.id = rom_geom_buc.id;
```

Regulă pentru ștergere

```
CREATE OR REPLACE RULE delete_geom_sv_ad_dist AS
ON DELETE TO sv_adrese_distanta_buc
DO INSTEAD
DELETE FROM positions.rom_geom_buc
WHERE rom_geom_buc.id = old.id;
```

Reguli PostgreSQL

```
CREATE OR REPLACE VIEW public.sv_adrese_distante_buc AS
SELECT rom_geom_buc.geom,
       rom_geom_buc.id,
       rom_ldec.den_st,
       rom_ldec.just_no,
       dist2sect.sv_id,
       dist2sect.dist,
FROM   positions.rom_geom_buc,
       mapping.dist2sect,
       aaa_rom_ldec
WHERE  dist2sect.aep_id = rom_geom_buc.id
AND    aaa_rom_ldec.id = rom_geom_buc.id;
```

Regulă pentru adăugare

```
CREATE OR REPLACE RULE insert_geom_sv_ad_dist AS
ON DELETE TO sv_adrese_distante_buc
DO INSTEAD
( INSERT INTO positions.rom_geom_buc (id, geom)
  values (new.id, new.geom);
  INSERT INTO mapping.dist2sect (aep_id, sv_id, dist)
  values (new.id, new.sv_id, new.dist);
  INSERT INTO aaa_rom_ldec (id, den_st, just_no)
  values (new.id, new.den_st, new.just_no);)
```

Reguli PostgreSQL

```
CREATE OR REPLACE VIEW public.sv_adrese_distante_buc AS
SELECT rom_geom_buc.geom,
       rom_geom_buc.id,
       rom_ldec.den_st,
       rom_ldec.just_no,
       dist2sect.sv_id,
       dist2sect.dist,
FROM positions.rom_geom_buc,
     mapping.dist2sect,
     aaa_rom_ldec
WHERE dist2sect.aep_id = rom_geom_buc.id
AND aaa_rom_ldec.id = rom_geom_buc.id;
```

Regulă pentru adăugare

```
CREATE OR REPLACE RULE insert_geom_sv_ad_dist AS
ON DELETE TO sv_adrese_distante_buc
DO INSTEAD
(INSERT INTO positions.rom_geom_buc (id, geom)
  values (new.id, new.geom);
INSERT INTO mapping.dist2sect (aep_id, sv_id, dist)
  values (new.id, new.sv_id, new.dist);
INSERT INTO aaa_rom_ldec (id, den_st, just_no)
  values (new.id, new.den_st, new.just_no);)
```

Declanșator PostgreSQL

```
CREATE OR REPLACE VIEW public.sv_adrese_distanta_buc AS
SELECT rom_geom_buc.geom,
       rom_geom_buc.id,
       rom_ldec.den_st,
       rom_ldec.just_no,
       dist2sect.sv_id,
       dist2sect.dist,
FROM   positions.rom_geom_buc,
       mapping.dist2sect,
       aaa_rom_ldec
WHERE  dist2sect.aep_id = rom_geom_buc.id
AND    aaa_rom_ldec.id = rom_geom_buc.id;
```

Declanșator pentru adăugare

```
CREATE TRIGGER add_geom_sv_ad_dist_trig INSTEAD OF INSERT
ON   sv_adrese_distanta_buc
FOR EACH ROW
EXECUTE PROCEDURE add_geom_sv_ad_dist();
```


Declanșator PostgreSQL

Procedură pentru adăugare

```
CREATE OR REPLACE FUNCTION add_geom_sv_ad_dist()
RETURNS TRIGGER
AS $BODY$
DECLARE
new_geom_id bigint;
BEGIN
    INSERT INTO positions.rom_geom_buc(geom) VALUES (NEW.geom)
    RETURNING id INTO new_geom_id;

    INSERT INTO mapping.dist2sect (aep_id, sv_id, dist)
        VALUES (new_geom_id, NEW.sv_id, NEW.dist);

    INSERT INTO aaa_rom_ldec(id, den_st, just_no)
        VALUES (new_geom_id, NEW.den_st, NEW.just_no);

    NEW.id = new_geom_id;
    RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql;
```

4

Concluzii

• Concluzii

- Tabele compuse permit creșterea productivității și pot ajuta la eliminarea erorilor de topologie între teme
- Editarea geometriei tabelelor compuse în PostGis se face prin
 - Reguli
 - pentru compunere simplă
 - în special pentru operații de modificare și ștergere
 - Declanșatoare
 - pentru compunere complexă, coloane autogenerate (ex: secvențe)
 - necesită cel puțin cunoștințe de bază de programare



Întrebări?

Multumesc!

