

A2017-GMQ715

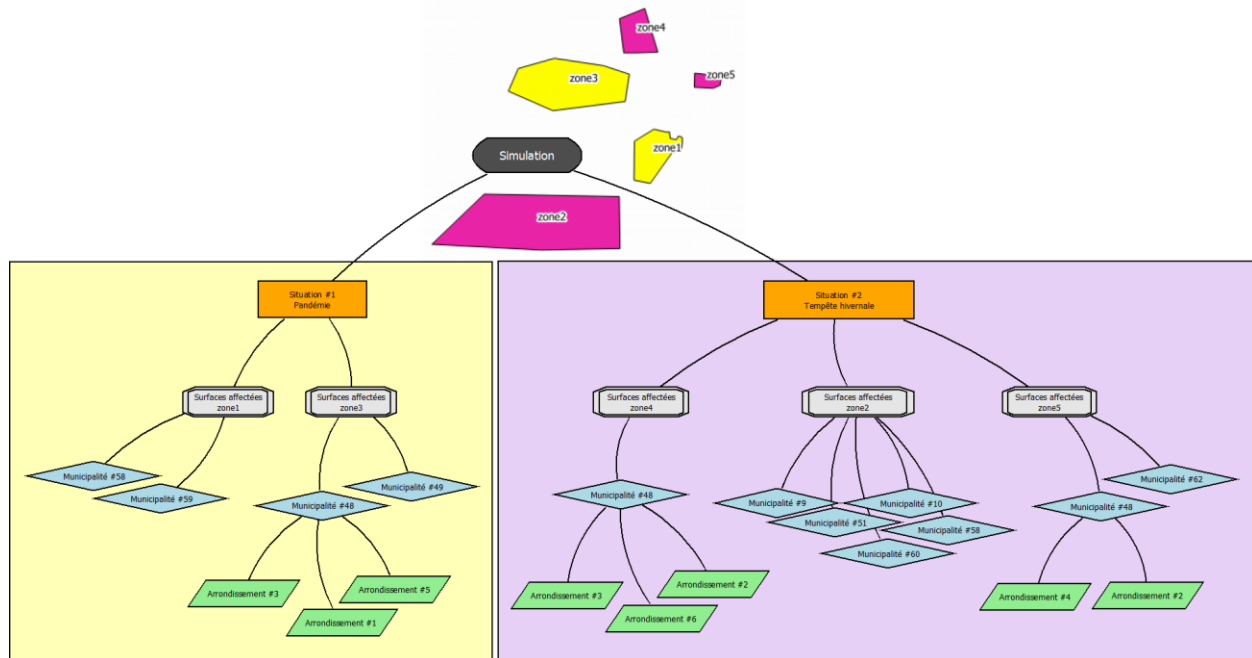
TP3

JONATHAN GARIEPY-ROY
Jonathan.gariepy-roy@usherbrooke.ca
Garj2115

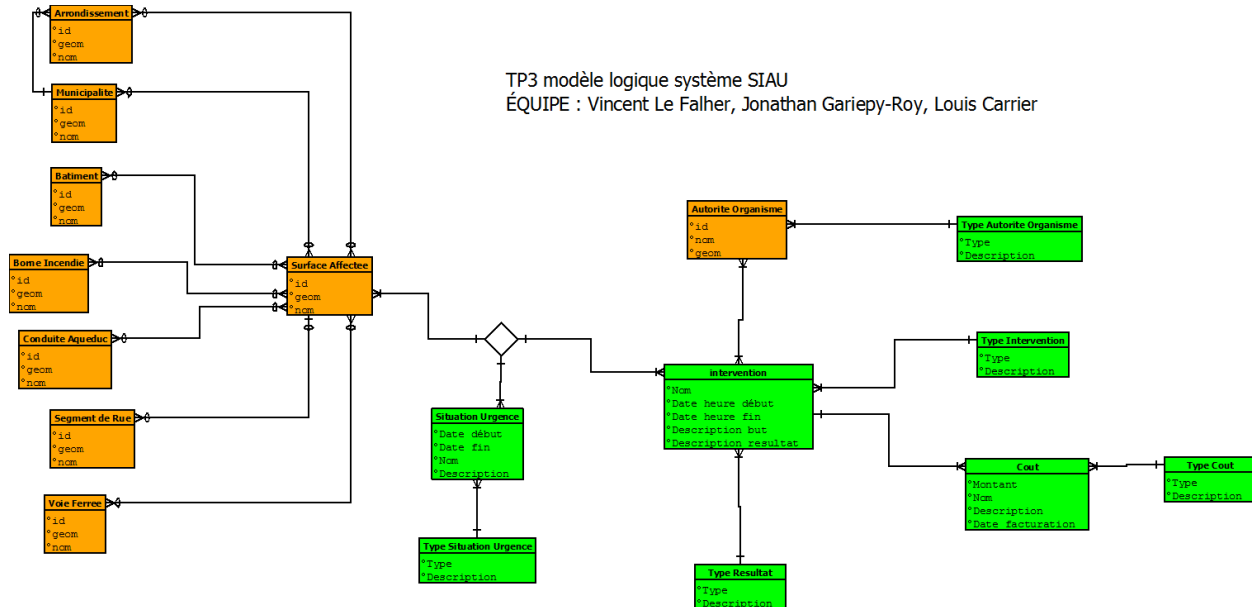
LOUIS CARRIER
louis.carrier@usherbrooke.ca
carl2621

VINCENT LE FALHER
vincent.lefalher@usherbrooke.ca
lefv2603

Diagramme de la simulation



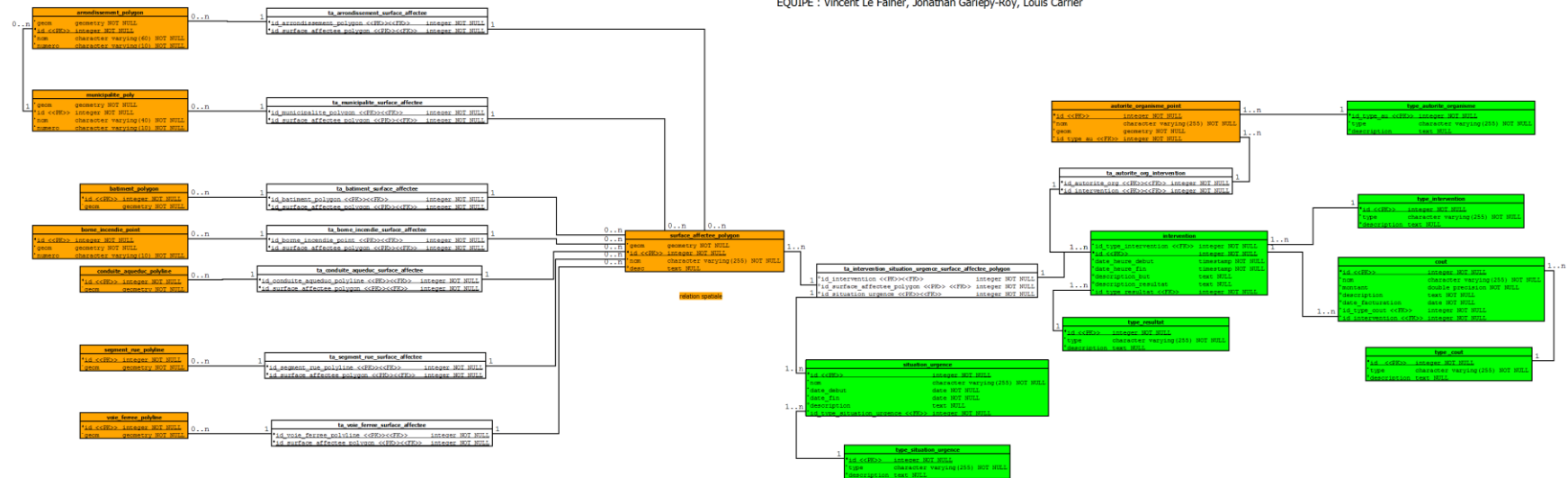
Modèle Logique



TP3 modèle logique système SIAU
 ÉQUIPE : Vincent Le Falher, Jonathan Gariépy-Roy, Louis Carrier

Modèle physique

TP3 modèle physique système SIAU
ÉQUIPE : Vincent Le Falher, Jonathan Gariépy-Roy, Louis Carrier



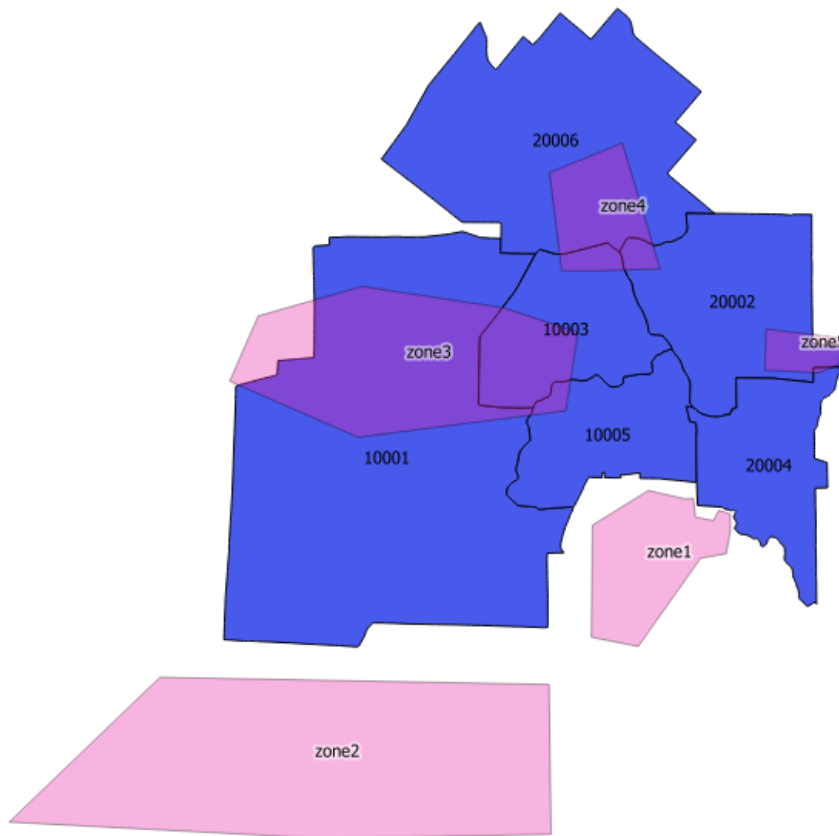
Requêtes

1. Quels sont les arrondissements affectés par chaque situation d'urgence ?

```
1 DROP VIEW question1_arrondissements_affectes;
2 CREATE VIEW question1_arrondissements_affectes AS
3 SELECT DISTINCT
4     (su.id * 10000 + arr.id) id,
5     arr.nom,
6     arr.geom,
7     su.id urg_id,
8     arr.id arr_id
9 FROM
10     arrondissement_polygon arr,
11     surface_affectee_polygon sa,
12     situation_urgence su,
13     ta_intervention_situation_urgence_surface_affectee_polygon ta
14 WHERE
15     ST_INTERSECTS(sa.geom, arr.geom)
16     AND su.id = ta.id_situation_urgence
17     AND sa.id = ta.id_surface_affectee_polygon;
```

Note : le id est créé avec le id des arrondissements * 1000 pour éviter d'avoir un doublon et permettre l'affichage sans erreur dans QGIS.

	id integer	nom character varying (60)	geom geometry	urg_id integer	arr_id integer
1	10001	Arrondissement de Rock Forest--Saint-Élie...	0106000...	1	1
2	10003	Arrondissement de Jacques-Cartier	0106000...	1	3
3	10005	Arrondissement du Mont-Bellevue	0106000...	1	5
4	20002	Arrondissement de Fleurimont	0106000...	2	2
5	20003	Arrondissement de Jacques-Cartier	0106000...	2	3
6	20004	Arrondissement de Lennoxville	0106000...	2	4
7	20006	Arrondissement de Brompton	0106000...	2	6

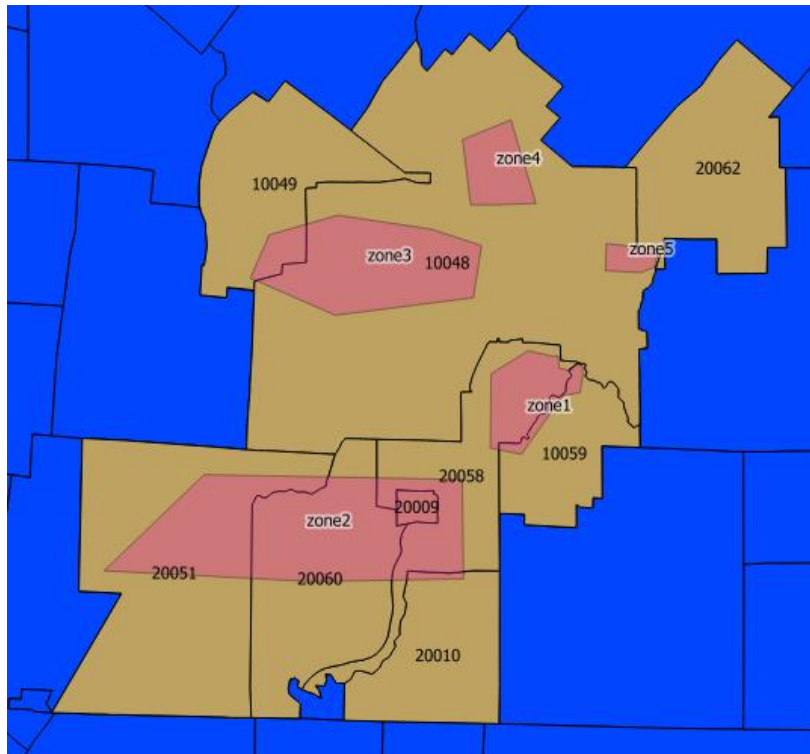


2. Quels sont les municipalités affectées par chaque situation d'urgence ? Créer une vue pour une situation d'urgence spécifique

```
1 DROP VIEW question2_municipalites_affectees;
2 CREATE VIEW question2_municipalites_affectees AS
3 SELECT DISTINCT
4     (su.id * 10000 + mun.id) id,
5     mun.nom,
6     mun.geom,
7     su.id su_id,
8     mun.id mun_id
9 FROM
10    municipalite_polygon mun,
11    surface_affectee_polygon sa,
12    situation_urgence su,
13    ta_intervention_situation_urgence_surface_affectee_polygon ta
14 WHERE
15     ST_INTERSECTS(sa.geom, mun.geom)
16     AND su.id = ta.id_situation_urgence
17     AND sa.id = ta.id_surface_affectee_polygon;
```

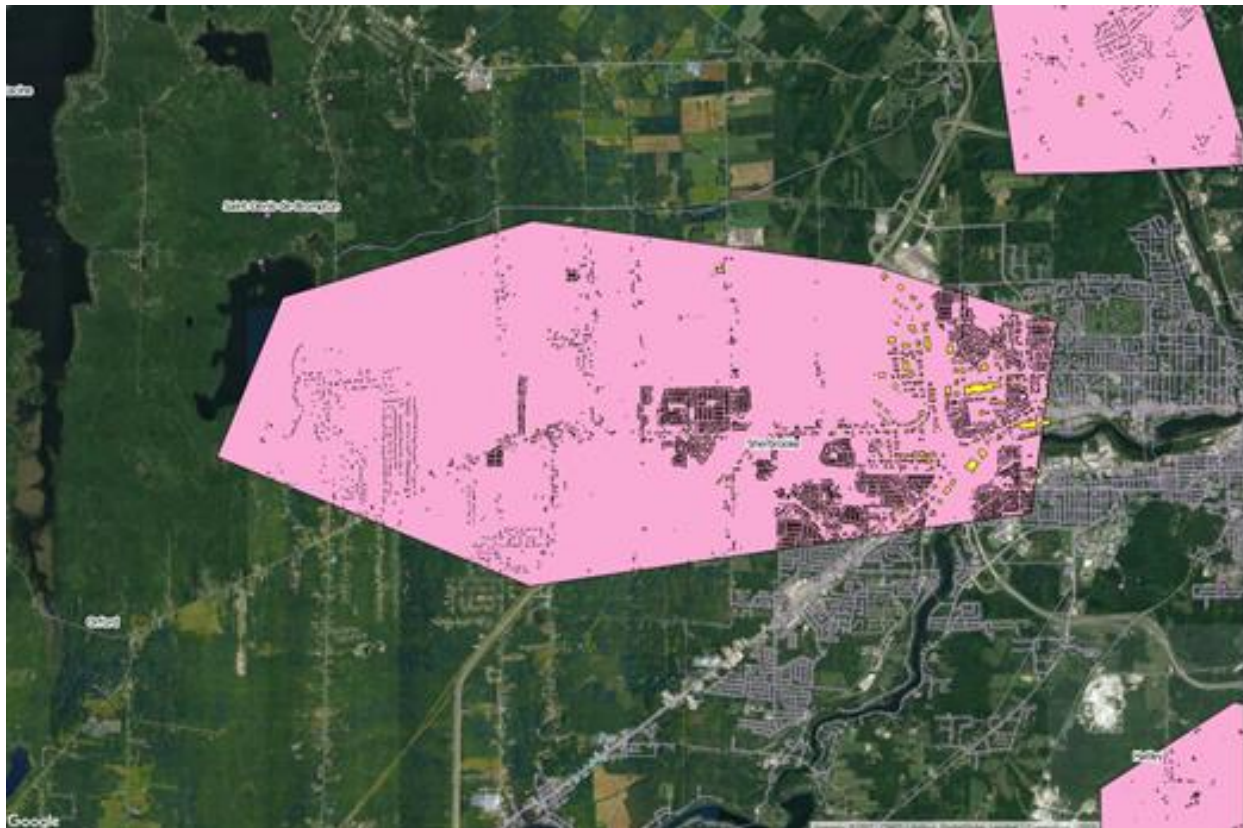
Note : le id est créé avec le id des municipalités * 1000 pour éviter d'avoir un doublon et permettre l'affichage sans erreur dans QGIS.

	id	nom	geom	su_id	mun_id
▲	integer	character varying (40)	geometry	integer	integer
1	10048	Sherbrooke	0106000...	1	48
2	10049	Saint-Denis-de-Bromp...	0106000...	1	49
3	10058	Hatley	0106000...	1	58
4	10059	Waterville	0106000...	1	59
5	20009	North Hatley	0106000...	2	9
6	20010	Hatley	0106000...	2	10
7	20048	Sherbrooke	0106000...	2	48
8	20051	Magog	0106000...	2	51
9	20058	Hatley	0106000...	2	58
10	20060	Sainte-Catherine-de-H...	0106000...	2	60
11	20062	Ascot Corner	0106000...	2	62



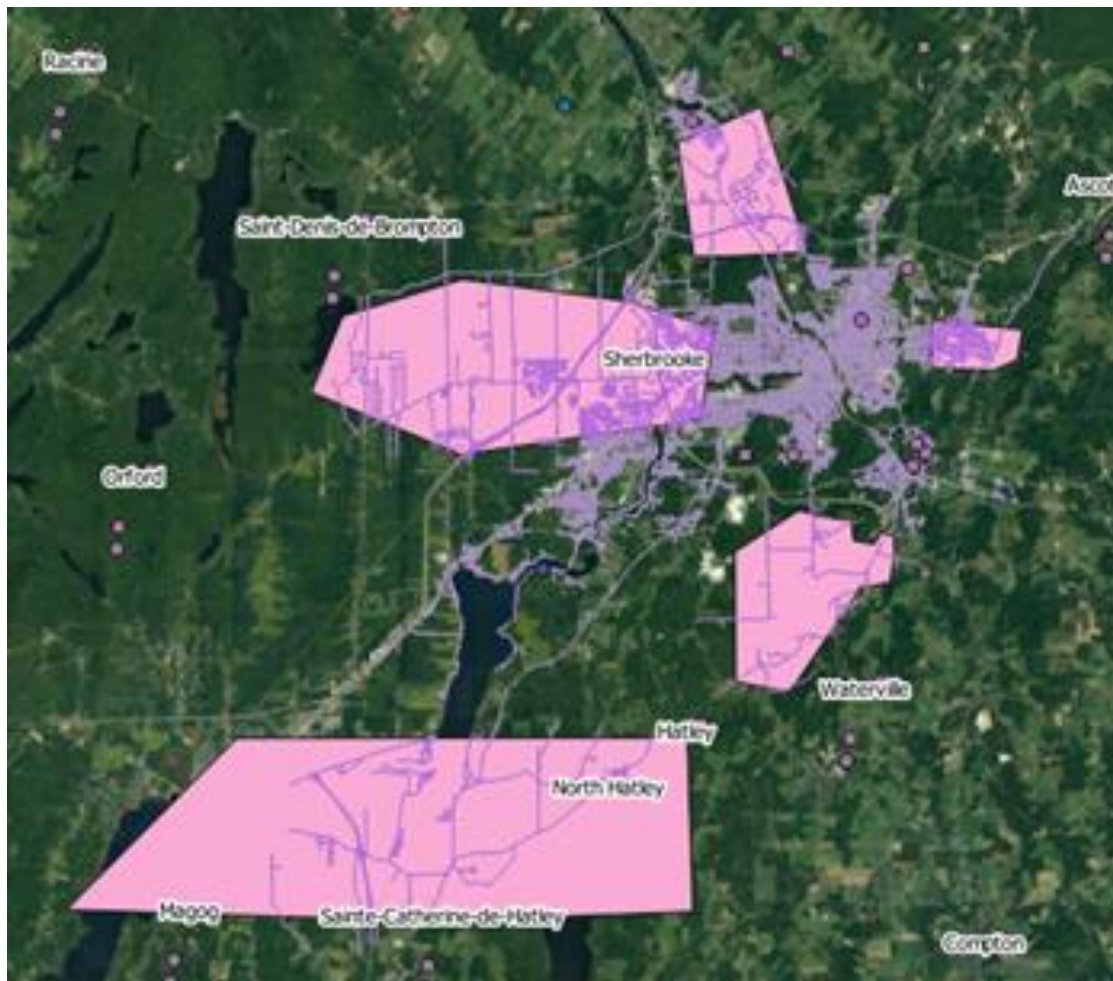
3. Quels sont les bâtiments affectés ?

```
1 DROP VIEW question3_batiments_affectes;  
2 CREATE VIEW question3_batiments_affectes AS  
3 SELECT distinct  
4     bat.id,  
5     bat.geom,  
6     ('batiment ' || bat.id || ' surface affectee ' || sa.id) nom  
7 FROM  
8     batiment_polygon bat,  
9     surface_affectee_polygon sa  
10 WHERE  
11     ST_INTERSECTS(sa.geom, bat.geom);
```



4. Quels sont les segments de rue affectés ?

```
1 DROP VIEW question4_rues_affectees;  
2 CREATE VIEW question4_rues_affectees AS  
3 SELECT DISTINCT  
4     sr.id,  
5     sr.geom,  
6     ('rue ' || sr.id || ' surface affectee ' || sa.id) nom  
7 FROM  
8     segment_rue_polyline sr,  
9     surface_affectee_polygon sa  
10 WHERE  
11     ST_INTERSECTS(sa.geom, sr.geom);
```



5. Quelles sont les conduits d'aqueduc affectés ?

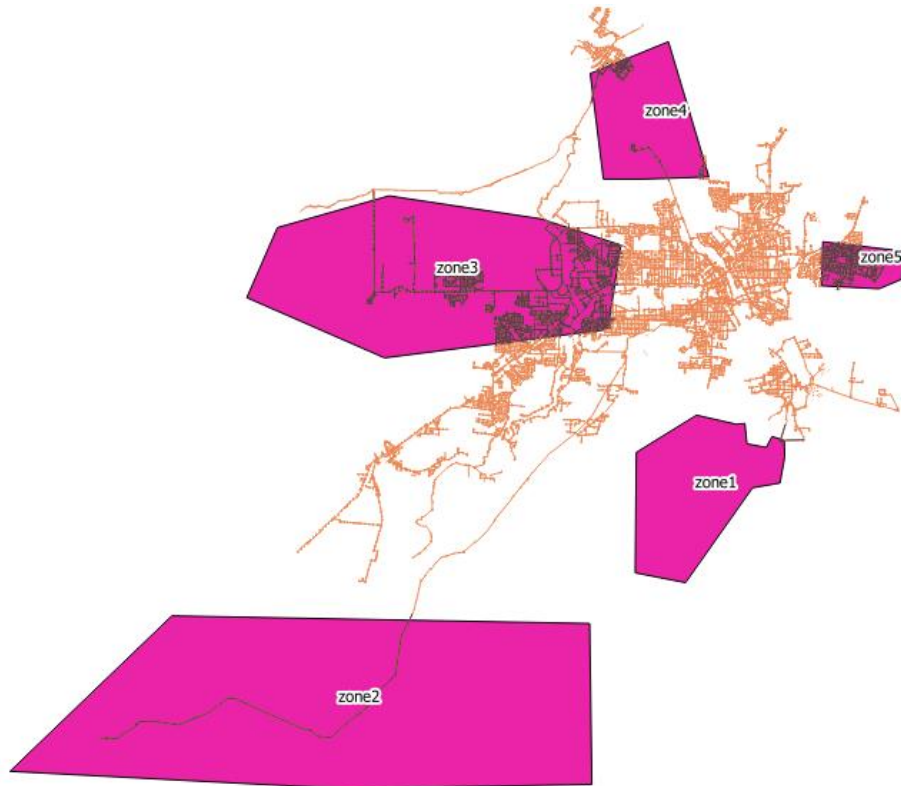
```

1 DROP VIEW question5_conduites_affectees;
2 CREATE VIEW question5_conduites_affectees AS
3 SELECT DISTINCT
4     ca.id,
5     ca.geom,
6     ('conduit aqueduc ' || ca.id || ' surface affectee ' || sa.id) nom
7 FROM
8     conduite_aqueduc_polyline ca,
9     surface_affectee_polygon sa
10 WHERE
11     ST_Intersects(sa.geom, ca.geom);

```

	id integer	geom geometry	nom text
1	3	0105000...	conduit aqueduc 3 surface affectee 3
2	4	0105000...	conduit aqueduc 4 surface affectee 3
3	5	0105000...	conduit aqueduc 5 surface affectee 3
4	6	0105000...	conduit aqueduc 6 surface affectee 3
5	9	0105000...	conduit aqueduc 9 surface affectee 3
6	10	0105000...	conduit aqueduc 10 surface affectee 3
7	12	0105000...	conduit aqueduc 12 surface affectee 3
8	45	0105000...	conduit aqueduc 45 surface affectee 3
9	46	0105000...	conduit aqueduc 46 surface affectee 3
10	47	0105000...	conduit aqueduc 47 surface affectee 3
11	48	0105000...	conduit aqueduc 48 surface affectee 3
12	49	0105000...	conduit aqueduc 49 surface affectee 3
13	58	0105000...	conduit aqueduc 58 surface affectee 2
14	61	0105000...	conduit aqueduc 61 surface affectee 3
15	62	0105000...	conduit aqueduc 62 surface affectee 3

... 6804 lignes retournées.



6. Quels sont les tronçons de voie ferrée affectés ?

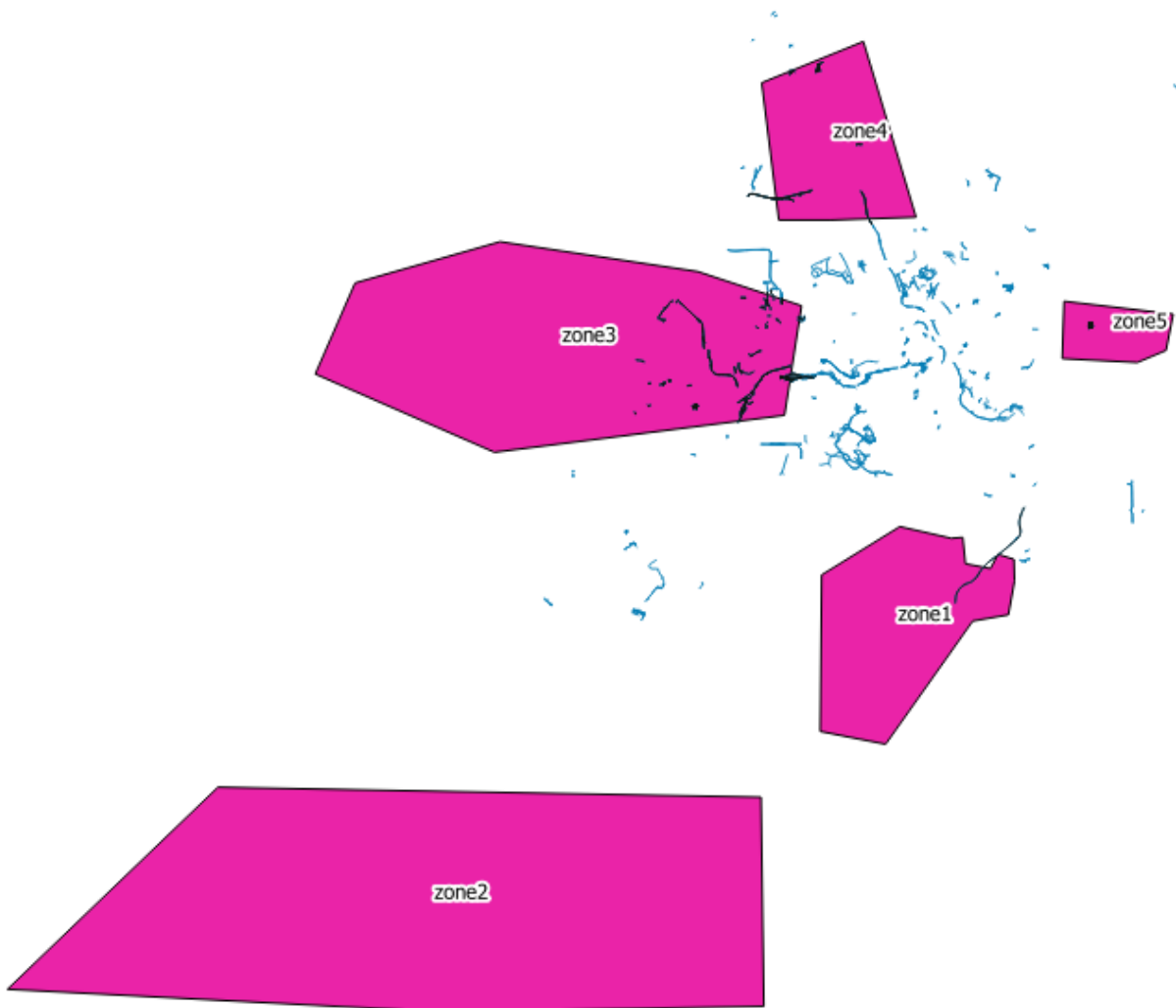
```

1 DROP VIEW question6_voie_ferrees_affectees;
2 CREATE VIEW question6_voie_ferrees_affectees AS
3 SELECT DISTINCT
4     vf.id id,
5     vf.geom,
6     ('voie ferree ' || vf.id || ' surface affectee ' || sa.id) nom
7 FROM
8     voie_ferree_polyline vf,
9     surface_affectee_polygon sa
10 WHERE
11     ST_Intersects(sa.geom,vf.geom);

```

	id integer	geom geometry	nom text
1	78	0105000...	voie ferree 78 surface affectee 3
2	80	0105000...	voie ferree 80 surface affectee 3
3	171	0105000...	voie ferree 171 surface affectee 3
4	172	0105000...	voie ferree 172 surface affectee 3
5	173	0105000...	voie ferree 173 surface affectee 3
6	174	0105000...	voie ferree 174 surface affectee 3
7	175	0105000...	voie ferree 175 surface affectee 3
8	176	0105000...	voie ferree 176 surface affectee 3
9	177	0105000...	voie ferree 177 surface affectee 3
10	178	0105000...	voie ferree 178 surface affectee 3
11	179	0105000...	voie ferree 179 surface affectee 3
12	180	0105000...	voie ferree 180 surface affectee 3
13	181	0105000...	voie ferree 181 surface affectee 3
14	182	0105000...	voie ferree 182 surface affectee 3
15	183	0105000...	voie ferree 183 surface affectee 3

... 317 lignes retournées.



7. Trouver les bornes d'incendie qui se trouvent à 300 m ou moins de chaque surface affectée.

```

1 DROP VIEW question7_borne_incendie_moins_300m;
2 CREATE VIEW question7_borne_incendie_moins_300m AS
3 SELECT DISTINCT
4     bi.id,
5     bi.geom,
6     ('borne incendie ' || bi.id || ' surface affectee ' || sa.id) nom
7 FROM
8     borne_incendie_point bi,
9     surface_affectee_polygon sa
10 WHERE
11     ST_DISTANCE(bi.geom, sa.geom) <= 300;

```

	id integer	geom geometry	nom text
1	8	0104000...	borne incendie 8 surface affectee 3
2	9	0104000...	borne incendie 9 surface affectee 3
3	13	0104000...	borne incendie 13 surface affectee 3
4	15	0104000...	borne incendie 15 surface affectee 3
5	29	0104000...	borne incendie 29 surface affectee 3
6	30	0104000...	borne incendie 30 surface affectee 3
7	31	0104000...	borne incendie 31 surface affectee 3
8	33	0104000...	borne incendie 33 surface affectee 3
9	40	0104000...	borne incendie 40 surface affectee 3
10	41	0104000...	borne incendie 41 surface affectee 3
11	47	0104000...	borne incendie 47 surface affectee 3
12	50	0104000...	borne incendie 50 surface affectee 3
13	66	0104000...	borne incendie 66 surface affectee 3
14	102	0104000...	borne incendie 102 surface affectee 3
15	134	0104000...	borne incendie 134 surface affectee 4

... 1159 lignes retournées.



8. Quels sont les postes de police à contacter ? (au moins un par surface affectée. On doit indiquer, selon la distance, l'ordre dans lequel les postes de police doivent être contactés)
- a. Pour vérifier les trois postes de police les plus proche de leurs surfaces affectées:

```
SELECT
  (sa_outer.id * 100000 + ao.id) id,
  ao.nom,
  ao.geom,
  ao.id ao_id,
  sa_outer.id sa_outer_id,
  ST_Distance(sa_outer.geom, ao.geom) dist
FROM
  surface_affectee_polygon sa_outer,
  autorite_organisme_point ao,
  (SELECT 3 val) n
WHERE ao.id IN
(
  SELECT s1.ao_id
  FROM
    (
      SELECT
        sa_inner.id sa_id,
        ao.id ao_id,
        ST_Distance(sa_inner.geom, ao.geom) dist
      FROM
        autorite_organisme_point ao,
        type_autorite_organisme tao,
        surface_affectee_polygon sa_inner
      WHERE
        ao.id_type_au = tao.id
        AND tao.type = 'Police'
        AND sa_inner.id = sa_outer.id
      order by ST_Distance(sa_inner.geom, ao.geom)
      LIMIT n.val
    ) s1
  )
ORDER BY sa_outer_id, ST_Distance(sa_outer.geom, ao.geom), ao.id;
```

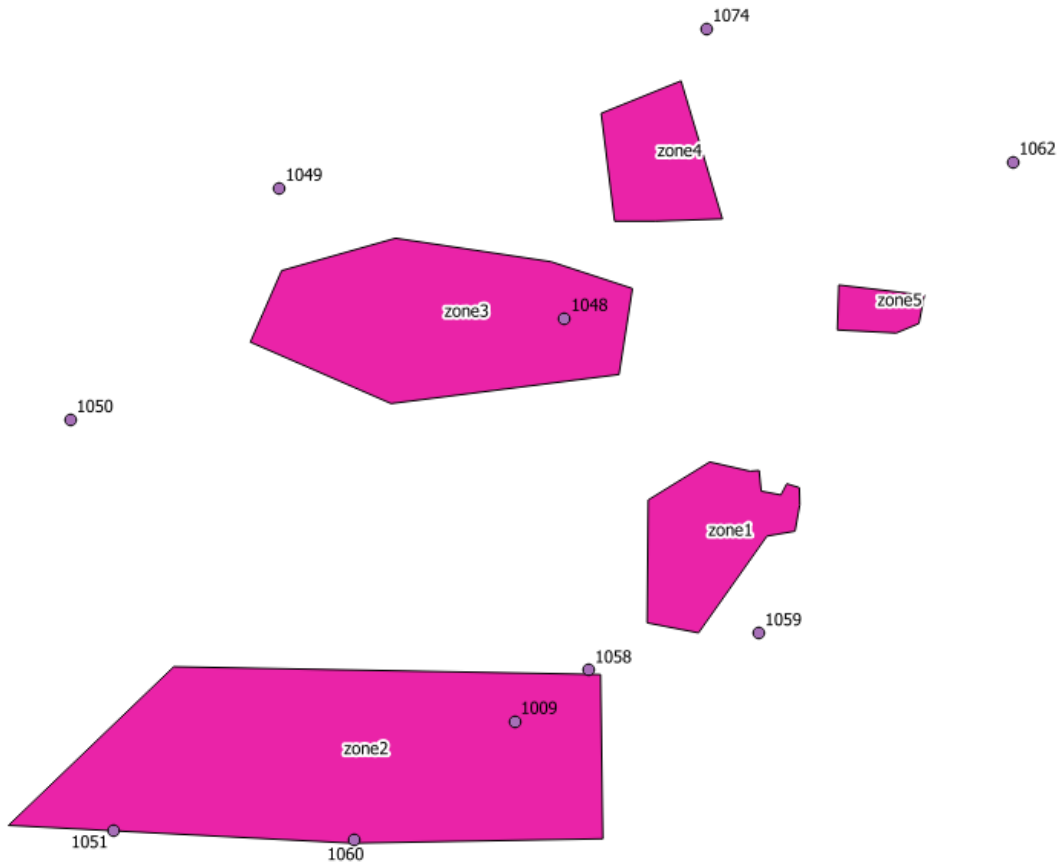

	id bigint	nom character varying (254)	geom geometry	ao_id bigint	sa_outer_id bigint	dist double precision
1	101059	poste_police59	0104000...	1059	1	1734.64751271627
2	101058	poste_police58	0104000...	1058	1	2654.47883108451
3	101009	poste_police9	0104000...	1009	1	5836.8549697005
4	201009	poste_police9	0104000...	1009	2	0
5	201051	poste_police51	0104000...	1051	2	0
6	201060	poste_police60	0104000...	1060	2	0
7	301048	poste_police48	0104000...	1048	3	0
8	301049	poste_police49	0104000...	1049	3	2831.45435644241
9	301050	poste_police50	0104000...	1050	3	6927.093937995
10	401074	poste_police74	0104000...	1074	4	2057.36778458905
11	401048	poste_police48	0104000...	1048	4	3853.06096881967
12	401062	poste_police62	0104000...	1062	4	10465.735498083
13	501062	poste_police62	0104000...	1062	5	5654.98002800787
14	501048	poste_police48	0104000...	1048	5	9638.92434038527
15	501074	poste_police74	0104000...	1074	5	10196.4409172697

b. La vue :

```

DROP VIEW question8_police_surface;
CREATE VIEW question8_police_surface AS
SELECT DISTINCT
  ao.id,
  ao.nom,
  ao.geom
FROM
  surface_affectee_polygon sa_outer,
  autorite_organisme_point ao,
  (SELECT 3 val) n
WHERE ao.id IN
(
  SELECT s1.ao_id
  FROM
    (
      SELECT
        sa_inner.id sa_id,
        ao.id ao_id,
        ST_Distance(sa_inner.geom, ao.geom) dist
      FROM
        autorite_organisme_point ao,
        type_autorite_organisme tao,
        surface_affectee_polygon sa_inner
      WHERE
        ao.id_type_au = tao.id
        AND tao.type = 'Police'
        AND sa_inner.id = sa_outer.id
      ORDER BY ST_Distance(sa_inner.geom, ao.geom)
      LIMIT n.val
    ) s1
  );

```



9. Quels sont les casernes de pompiers à contacter ? (au moins une par surface affectée. On doit indiquer, selon la distance, l'ordre dans lequel les postes de police doivent être contactés)
- a. Pour vérifier les trois casernes de pompier les plus proche de leurs surfaces affectées:

```
SELECT
  (sa_outer.id * 100000 + ao.id) id,
  ao.nom,
  ao.geom,
  ao.id ao_id,
  sa_outer.id sa_outer_id,
  ST_Distance(sa_outer.geom, ao.geom) dist
FROM
  surface_affectee_polygon sa_outer,
  autorite_organisme_point ao,
  (SELECT 3 val) n
WHERE ao.id IN
(
  SELECT s1.ao_id
  FROM
    (
      SELECT
        sa_inner.id sa_id,
        ao.id ao_id,
        ST_Distance(sa_inner.geom, ao.geom) dist
      FROM
        autorite_organisme_point ao,
        type_autorite_organisme tao,
        surface_affectee_polygon sa_inner
      WHERE
        ao.id_type_au = tao.id
        AND tao.type = 'Pompier'
        AND sa_inner.id = sa_outer.id
      ORDER BY ST_Distance(sa_inner.geom, ao.geom)
      LIMIT n.val
    ) s1
  )
ORDER BY sa_outer_id, ST_Distance(sa_outer.geom, ao.geom), ao.id;
```

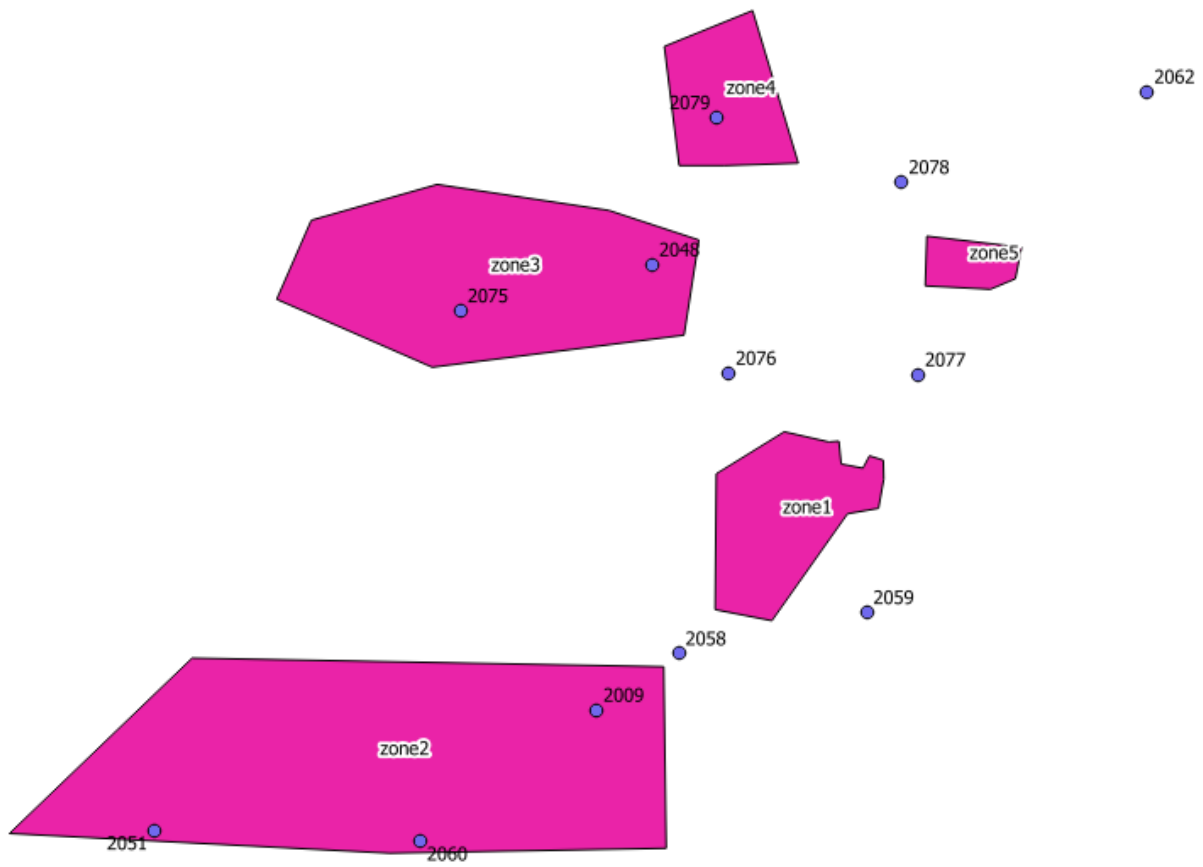
	id bigint	nom character varying (254)	geom geometry	ao_id bigint	sa_outer_id bigint	dist double precision
1	102058	poste_pompier58	0104000...	2058	1	1819.6203558483
2	102059	poste_pompier59	0104000...	2059	1	2312.05753005004
3	102076	poste_pompier76	0104000...	2076	1	2533.321659347
4	202009	poste_pompier9	0104000...	2009	2	0
5	202051	poste_pompier51	0104000...	2051	2	0
6	202060	poste_pompier60	0104000...	2060	2	0
7	302048	poste_pompier48	0104000...	2048	3	0
8	302075	poste_pompier75	0104000...	2075	3	0
9	302076	poste_pompier76	0104000...	2076	3	1873.08850072629
10	402079	poste_pompier79	0104000...	2079	4	0
11	402048	poste_pompier48	0104000...	2048	4	3272.55225931341
12	402078	poste_pompier78	0104000...	2078	4	3371.18398566312
13	502078	poste_pompier78	0104000...	2078	5	1907.95229971182
14	502077	poste_pompier77	0104000...	2077	5	2838.34567743438
15	502062	poste_pompier62	0104000...	2062	5	6405.75667575672

b. La vue :

```

DROP VIEW question9_pompier_surface;
CREATE VIEW question9_pompier_surface AS
SELECT DISTINCT
  ao.id,
  ao.geom,
  ao.nom
FROM
  surface_affectee_polygon sa_outer,
  autorite_organisme_point ao,
  (SELECT 3 val) n
WHERE ao.id IN
(
  SELECT s1.ao_id
  FROM
    (
      SELECT
        sa_inner.id sa_id,
        ao.id ao_id,
        ST_Distance(sa_inner.geom, ao.geom) dist
      FROM
        autorite_organisme_point ao,
        type_autorite_organisme tao,
        surface_affectee_polygon sa_inner
      WHERE
        ao.id_type_au = tao.id
        AND tao.type = 'Pompier'
        AND sa_inner.id = sa_outer.id
      ORDER BY ST_Distance(sa_inner.geom, ao.geom)
      LIMIT n.val
    ) s1
  )
);

```



10. Quels sont les centres de santé à contacter ? (au moins un par surface affectée.
On doit indiquer, selon la distance, l'ordre dans lequel les postes de police doivent être contactés)

a. Pour vérifier les trois centres de santé les plus proche de leurs surfaces affectées:

```
SELECT
    (sa_outer.id * 100000 + ao.id) id,
    ao.nom,
    ao.geom,
    ao.id ao_id,
    sa_outer.id sa_outer_id,
    ST_Distance(sa_outer.geom, ao.geom) dist
FROM
    surface_affectee_polygon sa_outer,
    autorite_organisme_point ao,
    (SELECT 3 val) n
WHERE ao.id IN
(
    SELECT s1.ao_id
    FROM
        (
            SELECT
                sa_inner.id sa_id,
                ao.id ao_id,
                ST_Distance(sa_inner.geom, ao.geom) dist
            FROM
                autorite_organisme_point ao,
                type_autorite_organisme tao,
                surface_affectee_polygon sa_inner
            WHERE
                ao.id_type_au = tao.id
                AND (tao.type = 'Sante' OR tao.type = 'Hopital')
                AND sa_inner.id = sa_outer.id
            ORDER BY ST_Distance(sa_inner.geom, ao.geom)
            LIMIT n.val
        ) s1
    )
ORDER BY sa_outer_id, ST_Distance(sa_outer.geom, ao.geom), ao.id;
```

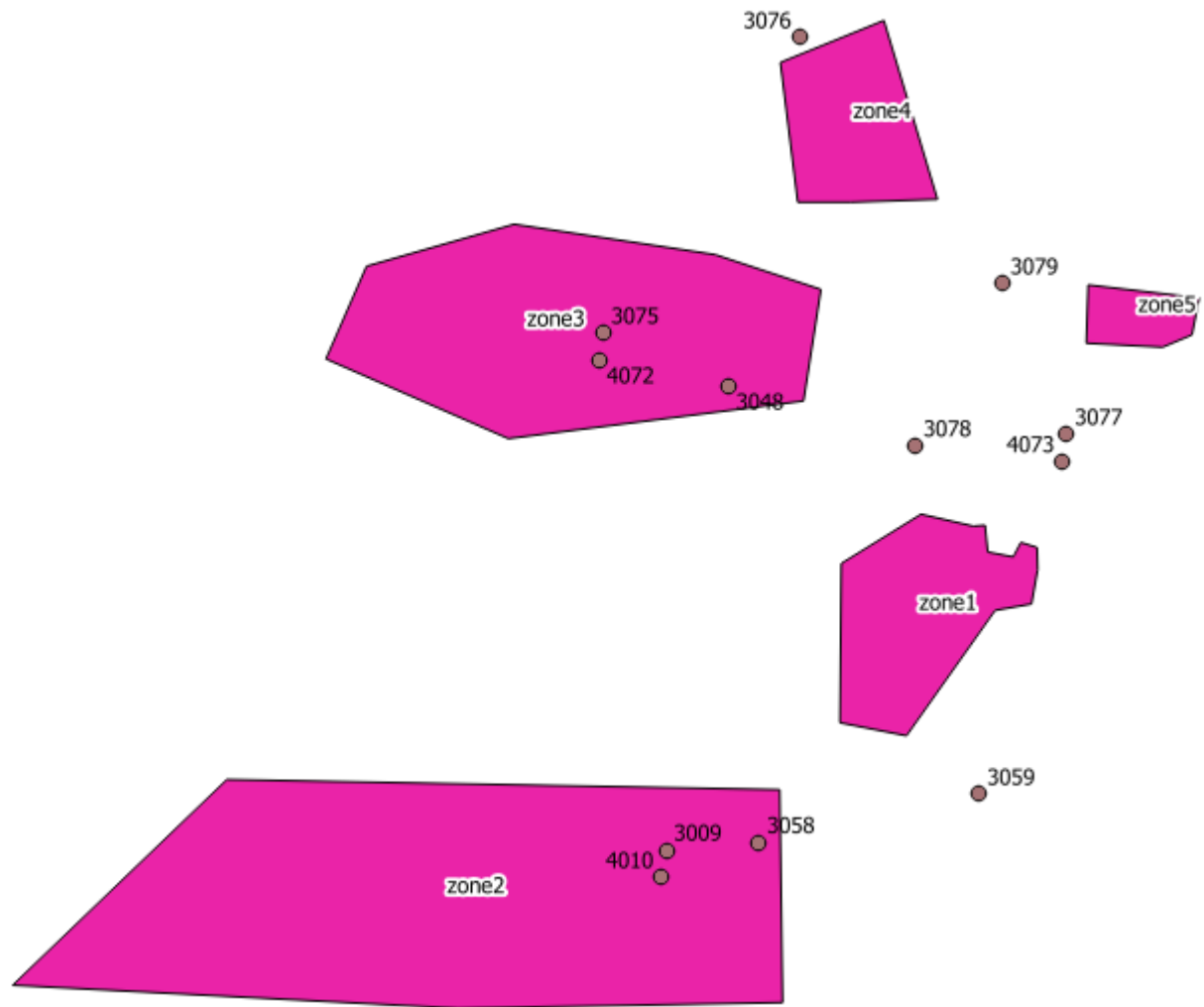
	id bigint	nom character varying (254)	geom geometry	ao_id bigint	sa_outer_id bigint	dist double precision
1	103078	dsc78	0104000...	3078	1	1870.31030062735
2	104073	hopital73	0104000...	4073	1	2439.57697418319
3	103059	dsc59	0104000...	3059	1	2547.90796932707
4	203009	dsc9	0104000...	3009	2	0
5	203058	dsc58	0104000...	3058	2	0
6	204010	hopital10	0104000...	4010	2	0
7	303048	dsc48	0104000...	3048	3	0
8	303075	dsc75	0104000...	3075	3	0
9	304072	hopital72	0104000...	4072	3	0
10	403076	dsc76	0104000...	3076	4	435.84956514833
11	403079	dsc79	0104000...	3079	4	2905.12035286943
12	403048	dsc48	0104000...	3048	4	5375.01393750384
13	503079	dsc79	0104000...	3079	5	2335.65434513701
14	503077	dsc77	0104000...	3077	5	2540.63161198601
15	504073	hopital73	0104000...	4073	5	3297.65292451126

b. La vue :

```

DROP VIEW question10_sante_surface;
CREATE VIEW question10_sante_surface AS
SELECT DISTINCT
    ao.id,
    ao.geom,
    ao.nom
FROM
    surface_affectee_polygon sa_outer,
    autorite_organisme_point ao,
    (SELECT 3 val) n
WHERE ao.id IN
(
    SELECT s1.ao_id
    FROM
        (
            SELECT
                sa_inner.id sa_id,
                ao.id ao_id,
                ST_Distance(sa_inner.geom, ao.geom) dist
            FROM
                autorite_organisme_point ao,
                type_autorite_organisme tao,
                surface_affectee_polygon sa_inner
            WHERE
                ao.id_type_au = tao.id
                AND (tao.type = 'Sante' OR tao.type = 'Hopital')
                AND sa_inner.id = sa_outer.id
            ORDER BY ST_Distance(sa_inner.geom, ao.geom)
            LIMIT n.val
        ) s1
    );

```

Conclusion globale sur les connaissances acquises durant le cours

Ce cours nous a permis de prendre connaissance des règles de modélisation des bases de données relationnelles incluant des références spatiales. Nous avons aussi pu comprendre comment les géométries spatiales vectorielles peuvent être stockées dans une base de données PostgreSQL avec l'extension PostGIS et manipulées avec le langage SQL. Enfin nous avons pu expérimenter la visualisation (vue) et la validation des requêtes spatiales provenant de la base de données PostgreSQL dans le logiciel SIG QGIS.