Guillaume Hutin

Rapport projet web sémantique

(lien github : https://github.com/Hemualliug/web_sem_project)

Présentation des données

Pour effectuer ce projet, j'ai choisi d'utiliser deux jeux de données :

- les chiffres clés des entreprises françaises en 2019
- les entreprises qui ont été radié en 2021

Ces données viennent de l'Open Data du site DataInfoGreffe. DataInfoGreffe est le leader de la donnée d'entreprise. Il nous permet d'accéder à des données fiables, contrôlées, certifiées et à jour.

Pour le jeu de données sur les chiffres clés des entreprises françaises en 2019, les lignes/individus sont les entreprises françaises existantes en 2019. Les colonnes sont des caractéristiques de ces entreprises telles que leur numéro SIREN qui les identifie de manière unique, leur chiffre d'affaires et leur résultat en 2019, leur code postal, leur date d'immatriculation, etc...

Pour le jeu de données sur les entreprises radiées en 2021, les lignes/individus sont les entreprises françaises qui ont été radiées en 2021. Les colonnes sont encore une fois des caractéristiques de ces entreprises. On peut retrouver la ville dans laquelle se trouve l'entreprise, le département, le secteur d'activité, la date d'immatriculation.

Le premier jeu de données fait 1300859 lignes pour 41 colonnes.
Le deuxième jeu de données fait 145466 lignes pour 29 colonnes.

Pré-traitement des données

Pour cette partie sur le pré-traitement des données, j'ai utilisé un notebook Jupyter et la librairie pandas.

Pour commencer, j'ai créé une variable "Radiée" qui sera une variable binaire, qui vaut 1 lorsque la société a été radiée en 2021, 0 sinon. J'ai créé cette variable seulement dans la table des données sur les sociétés radiées.

Une fois cette variable créée, un des premiers objectifs a été de regrouper les informations intéressantes des deux jeux de données au sein d'une même table. J'ai donc fusionné les deux tables en faisant une jointure à gauche sur la table des chiffres clés car nous voulons garder les entreprises qui n'ont pas été radiées en 2021.

Par rapport à la variable “Radiée”, les entreprises qui n’ont pas été radiées n’auront pas de valeur. Ce sera un noeud blanc.

Une fois cette jointure effectuée, j’ai dû choisir les variables/colonnes qui me semblaient les plus intéressantes pour construire des relations riches par la suite. J’ai donc choisi les colonnes suivantes :

- Dénomination (c’est le nom de la société)
- Siren (c’est le numéro unique d’une société qui servira d’identifiant)
- Code postal
- Ville
- Département
- Région
- Date d’immatriculation (correspond à la date de création de l’entreprise)
- CA 1
- Résultat 1
- Effectif 1
- CA 2
- Résultat 2
- Effectif 2
- CA 3
- Résultat 3
- Effectif 3

Les colonnes avec un 1 à la fin correspondent à des informations pour l'année 2019. Celles avec un 2, c'est l'année 2018, et celles avec un 3, l'année 2017.

Après ceci, j’ai décidé d’enlever les lignes où des informations importantes manquaient. En effet, des entreprises où le numéro Siren, la dénomination, le code postal, la région, le département et la date d'immatriculation ne sont pas présents ne sont pas très intéressantes pour notre future analyse.

La dernière modification effectuée est la transformation d’une variable. C’est la variable “date d’immatriculation”. En effet, de base cette variable était avec une structure YYYY/MM/dd, or par soucis de praticité, j’ai modifié cette variable pour ne retenir que l’année. Cela sera ensuite plus aisé de travailler avec.

Elaboration du squelette Openrefine

Nos données maintenant bien mises en forme, nous pouvons les charger dans OpenRefine afin de définir comment on va transformer nos données tabulaires en triplets RDF. On va donc définir un squelette RDF avec l’URI de base `http://mon_projet.org/entreprise#` : (code couleur : URI, noeud blanc, Litéral)

Siren URI de type :Entreprise → `rdfs:label` → Dénomination
→ `:siren` → Siren
→ `:adresse` → noeud blanc de type :Adresse → `:code_postal` → Code postal
→ `:ville` → Ville URI
→ `:immatriculation` → Date immatriculation
→ `:annee_2019` → noeud blanc de type :Resultat → `:ca` → CA 1
→ `:resultat` → Résultat 1
→ `:effectif` → Effectif 1
→ `:annee_2018` → noeud blanc de type :Resultat → `:ca` → CA 2
→ `:resultat` → Résultat 2

→ :effectif → Effectif 2
→ :annee_2017 → noeud blanc de type :Resultat → :ca → CA 3
→ :resultat → Résultat 3
→ :effectif → Effectif 3
→ :radiee → Radiée

Ville URI de type :Ville → rdfs:label → Ville
→ :part_of → Département URI
Département URI de type Departement → rdfs:label → Département
→ part_of → Région URI
Région URI de type :Region → rdfs:label → Région

Schéma RDF

Après avoir défini ce squelette, nous avons pu définir le schéma RDF :

```
@prefix :      <http://127.0.0.1:3333/> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .

:Lieu a rdfs:Class.

:Entreprise a rdfs:Class.

:Resultat a rdfs:Class.

:Adresse a rdfs:Class.

:Ville a rdfs:Class;
rdfs:subClassOf :Lieu.

:Departement a rdfs:Class;
rdfs:subClassOf :Lieu.

:Region a rdfs:Class;
rdfs:subClassOf :Lieu.

:siren a rdf:Property;
rdfs:domain :Entreprise;
rdfs:range rdfs:Literal.

:adresse a rdf:Property;
rdfs:domain :Entreprise;
rdfs:range :Adresse.

:code_postal a rdf:Property;
rdfs:domain :Adresse
rdfs:range rdfs:Literal.

:ville a rdf:Property;
rdfs:domain :Adresse
rdfs:range :Ville.

:immatriculation a
rdf:Property;
rdfs:domain :Entreprise;
rdfs:range rdfs:Literal.

:annee_2019 a rdf:Property;
rdfs:domain :Entreprise;
rdfs:range :Resultat.

:annee_2018 a rdf:Property;
rdfs:domain :Entreprise;
rdfs:range :Resultat.

:annee_2017 a rdf:Property;
rdfs:domain :Entreprise;
rdfs:range :Resultat.

:ca a rdf:Property;
rdfs:domain :Resultat;
rdfs:range rdfs:Literal.

:resultat a rdf:Property;
rdfs:domain :Resultat;
rdfs:range rdfs:Literal.

:effectif a rdf:Property;
rdfs:domain :Resultat;
rdfs:range rdfs:Literal.

:part_of a rdf:Property;
rdfs:domain :Lieu;
rdfs:range :Lieu
```

Ensuite, afin de nettoyer et bien mettre en forme nos données, nous avons utilisé cette application web : [RDF Validator and Converter]

Voici un extrait du fichier Turtle renvoyé :

```
:421394834  a              :Entreprise ;
        rdfs:label         "SARL LES KORRIGANS" ;
        :adresse           [ a             :Adresse ;
                             :code_postal  "24000.0" ;
                             :ville        :PERIGUEUX
                           ] ;
        :annee_2017        [ a           :Resultat ;
                             :ca         "132110.0" ;
                             :resultat   "520.0"
                           ] ;
        :annee_2018        [ a           :Resultat ;
                             :ca         "154660.0" ;
                             :effectif   "2.0" ;
                             :resultat   "80.0"
                           ] ;
        :annee_2019        [ a           :Resultat ;
                             :ca         "136782.0" ;
                             :resultat   "-11871.0"
                           ] ;
        :immatriculation   "1999" ;
        :siren             "421394834" .
```

Triple Store et requêtes SPARQL

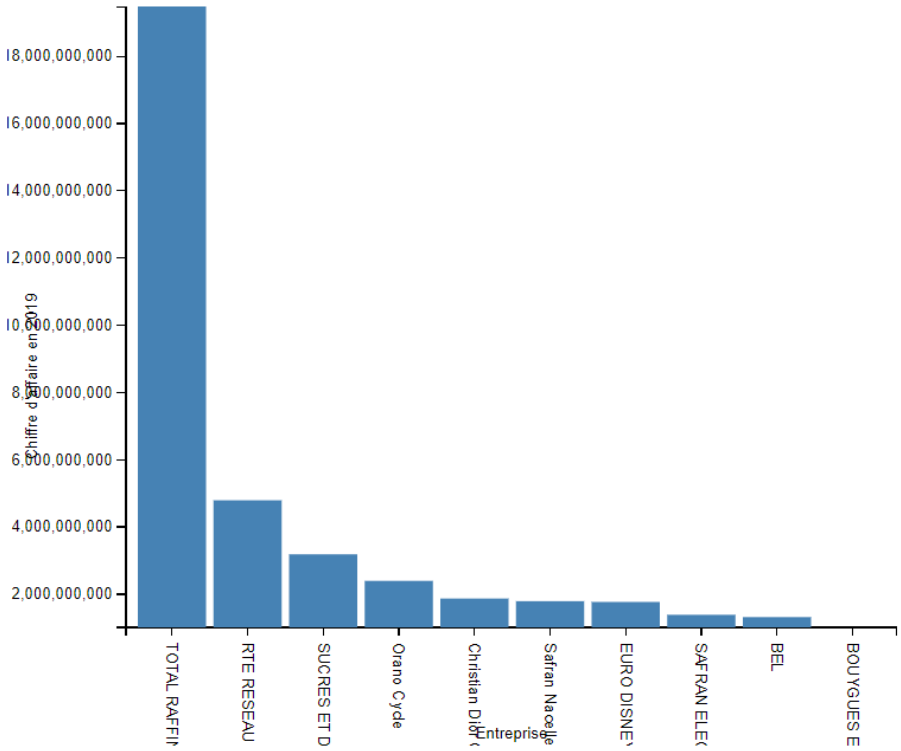
La prochaine étape a été de stocker nos triplets RDF dans un triple store. Pour ce projet nous avons choisi d'utiliser Apache Fuseki. Nous avons juste eu à importer notre fichier Turtle obtenu en sortie du convertisseur RDF. Dans le fichier Turtle, on a donc tous les triplets de notre table de base ainsi que le schéma RDF. Dernière étape du projet, c'est le requêtage du triple store à l'aide du langage SPARQL et l'affichage des résultats dans une page web. Nous avons utilisé [D3sparql].

Nos trois premières requêtes fonctionnent ensemble pour faire un résultat dynamique sur la page web. En effet, on peut visualiser avec un barchart quelles sont les 10 premières entreprises en fonction de leur chiffre d'affaires en fonction de 3 années possibles (2019, 2019 ou 2017)

1ère requête :

```
PREFIX : <http://mon_projet.org/entreprise#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT DISTINCT ?nom ?ca
WHERE {
    ?entreprise :annee_2019 [:ca ?ca].
    ?entreprise rdfs:label ?nom.
}
ORDER BY DESC(xsd:float(?ca))
LIMIT 10
```

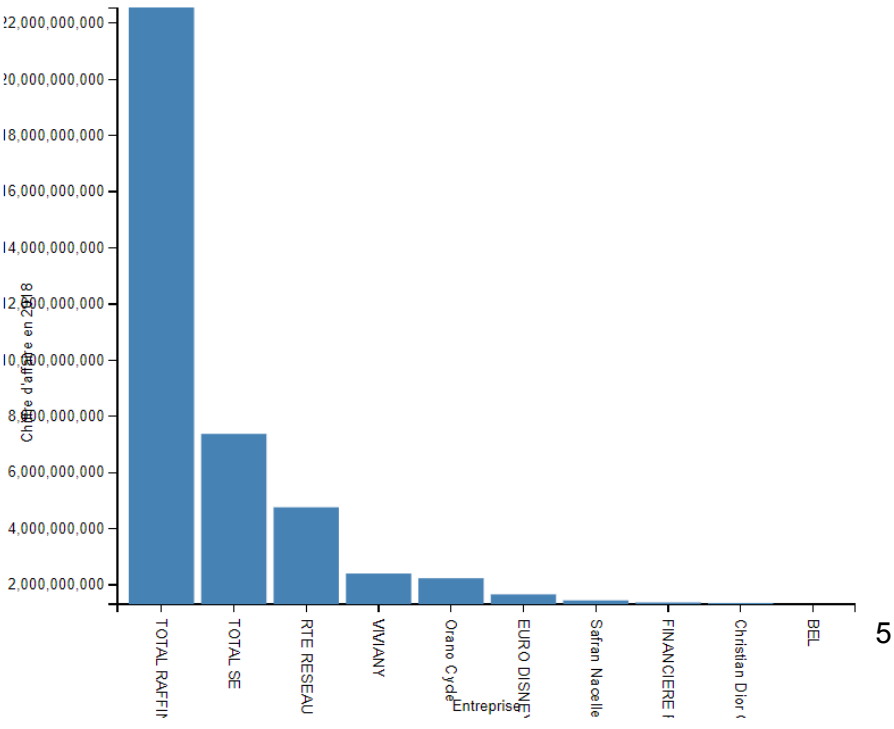
Résultat :



2ème requête : (je ne remet plus les préfixes par praticité)

```
SELECT DISTINCT ?nom ?ca
WHERE {
    ?entreprise :annee_2018 [:ca ?ca].
    ?entreprise rdfs:label ?nom.
}
ORDER BY DESC(xsd:float(?ca))
LIMIT 10
```

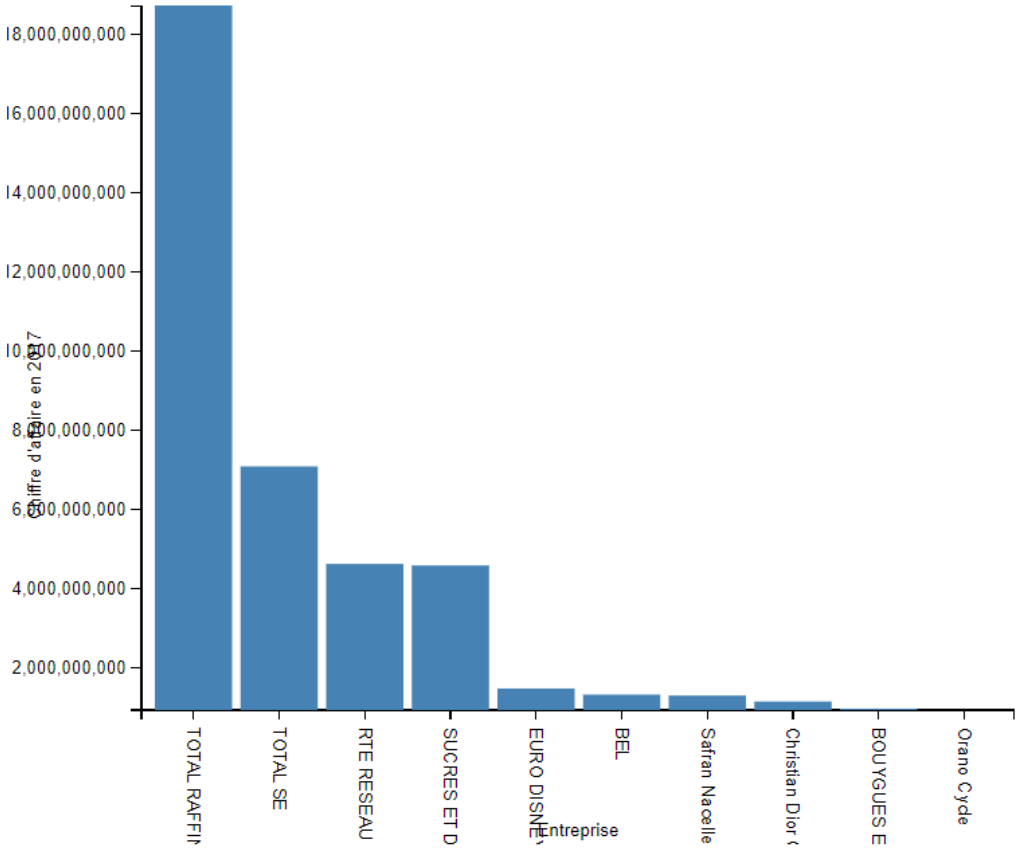
Résultat :



3ème requête :

```
SELECT DISTINCT ?nom ?ca
WHERE {
    ?entreprise :annee_2017 [:ca ?ca].
    ?entreprise rdfs:label ?nom.
}
ORDER BY DESC(xsd:float(?ca))
LIMIT 10
```

Résultat :



Pour mes deux prochaines requêtes, étant donné qu'on avait des informations sur la localisation des entreprises, j'ai voulu voir comment les entreprises se répartissaient en France, c'est-à-dire si beaucoup d'entreprises étaient regroupées au même endroit.

4ème requête :

```
SELECT (?nom_region as ?Region) (COUNT(?entreprise) as ?Nombre_d_entreprise )
WHERE {
  ?entreprise :adresse [:ville ?ville].
  ?ville rdfs:label ?nom_ville.
  ?ville :part_of ?departement.
  ?departement rdfs:label ?nom_departement.
  ?departement :part_of ?region.
  ?region rdfs:label ?nom_region.

}
GROUP BY ?nom_region
```

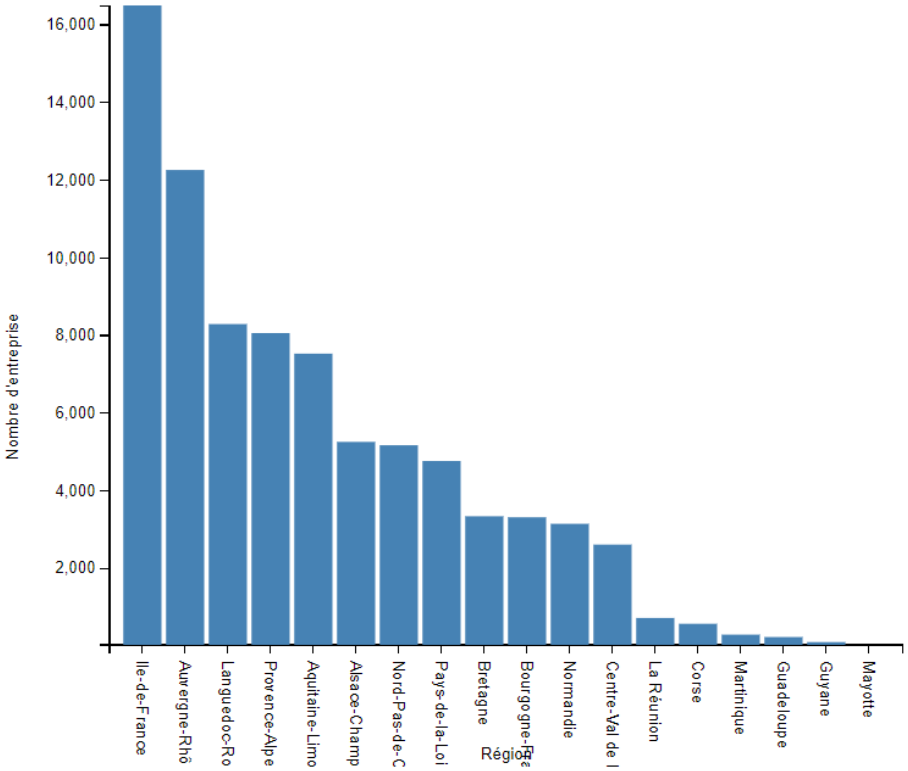
Résultat :

Region	Nombre_d_entreprise
Normandie	3148
Aquitaine-Limousin-Poitou-Charentes	7532
Provence-Alpes-Côte d'Azur	8062
Martinique	292
Bretagne	3342
Pays-de-la-Loire	4766
Ile-de-France	16504
Languedoc-Roussillon-Midi-Pyrénées	8294
Bourgogne-Franche-Comté	3316
Guadeloupe	230
Centre-Val de Loire	2612
Mayotte	28
Auvergne-Rhône-Alpes	12264
Alsace-Champagne-Ardenne-Lorraine	5258
Corse	572
Nord-Pas-de-Calais-Picardie	5172
Guyane	100
La Réunion	722

5ème requête :

```
SELECT (?nom_region as ?Region) (COUNT(?entreprise) as ?Nombre_d_entreprise )
WHERE {
    ?entreprise :adresse [:ville ?ville].
    ?ville rdfs:label ?nom_ville.
    ?ville :part_of ?departement.
    ?departement rdfs:label ?nom_departement.
    ?departement :part_of ?region.
    ?region rdfs:label ?nom_region.
}
GROUP BY ?nom_region
ORDER BY DESC(?Nombre_d_entreprise)
```

Résultat :



Pour mes deux dernières requêtes, j'ai utilisé la variable "Radiée" qui permet de savoir si une entreprise a été radiée en 2021. Je vais donc croiser cette information avec sa localisation pour voir où est ce qu'il y a le plus d'entreprises radiées.

6ème requête :

```
                SELECT (?nom_region as ?Region) (COUNT(?entreprise) as
?Nombre_d_entreprise_radiée )
                WHERE {
                ?entreprise :adresse [:ville ?ville].
                ?ville rdfs:label ?nom_ville.
                ?ville :part_of ?departement.
                ?departement rdfs:label ?nom_departement.
                ?departement :part_of ?region.
                ?region rdfs:label ?nom_region.
                ?entreprise :radiee ?radiee
                FILTER(?radiee = "1")
                }
                GROUP BY ?nom_region
```

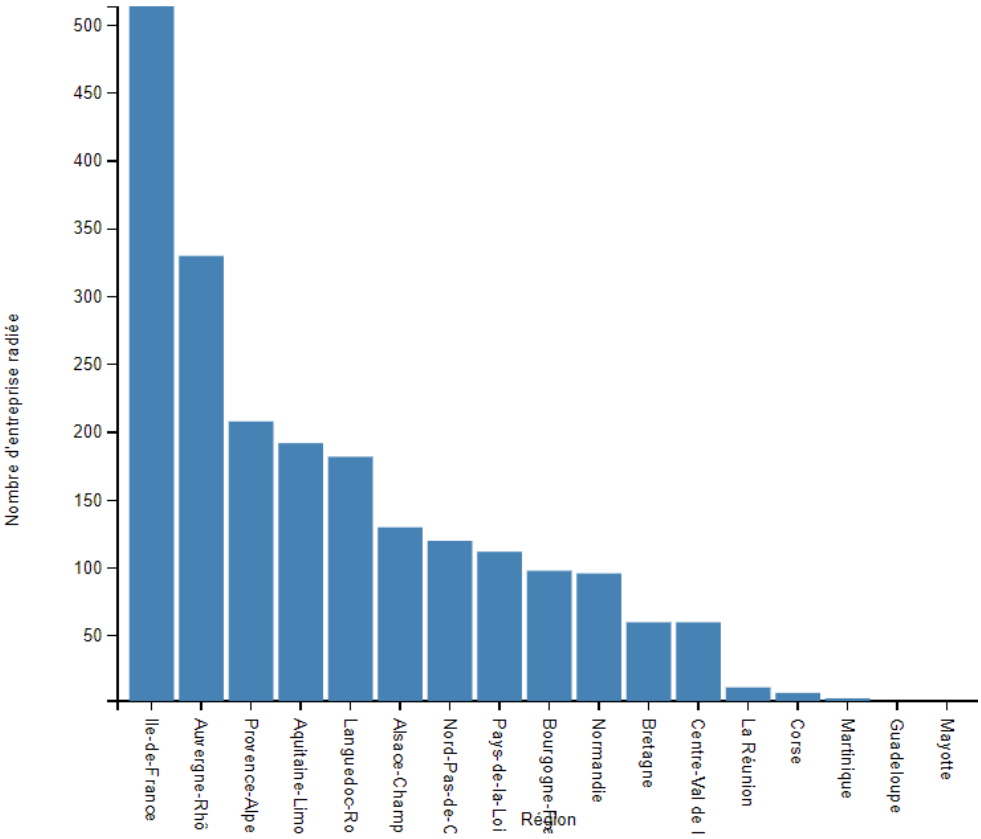
Résultat :

Region	Nombre_d_entreprise_radiée
Normandie	96
Aquitaine-Limousin-Poitou-Charentes	192
Provence-Alpes-Côte d'Azur	208
Martinique	4
Pays-de-la-Loire	112
Bretagne	60
Ile-de-France	514
Languedoc-Roussillon-Midi-Pyrénées	182
Bourgogne-Franche-Comté	98
Guadeloupe	2
Centre-Val de Loire	60
Mayotte	2
Auvergne-Rhône-Alpes	330
Alsace-Champagne-Ardenne-Lorraine	130
Corse	8
Nord-Pas-de-Calais-Picardie	120
La Réunion	12

7ème requête :

```
SELECT (?nom_region as ?Region) (COUNT(?entreprise) as ?Nombre_d_entreprise_radiée )
                WHERE {
                ?entreprise :adresse [:ville ?ville].
                ?ville rdfs:label ?nom_ville.
                ?ville :part_of ?departement.
                ?departement rdfs:label ?nom_departement.
                ?departement :part_of ?region.
                ?region rdfs:label ?nom_region.
                ?entreprise :radiee ?radiee
                FILTER(?radiee = "1")
                }
                GROUP BY ?nom_region
                ORDER BY DESC(?Nombre_d_entreprise_radiée)
```

Résultat :


Nombre d'entreprises radiée
500
450
400
350
300
250
200
150
100
50
Ile-de-France
Auvergne-Rhô
Provence-Alpe
Aquitaine-Limo
Languedoc-Ro
Alsace-Champ
Nord-Pas-de-C
Pays-de-la-Loi
Bourgogne-Fran
Normandie
Bretagne
Centre-Val de l
La Réunion
Corse
Martinique
Guadeloupe
Mayotte
Région