SHEVEK

Zeste de savoir

Explorons WikiData

11 décembre 2020

Table des matières

Vous vous êtes toujours demandé comment connaitre les villes les plus densément peuplées de France ou comment savoir dans quelle ville sont nés le plus de joueurs de football?

C'est aujourd'hui possible grâce à WikiData.

Je vais, dans ce tutoriel, vous guider dans l'exploration autour des données liées aux villes dans Wikidata.

Prérequis
Une connaissance basique du SQL est conseillée afin de comprendre plus facilement ce tuto.
Objectifs
Vous faire découvrir brièvement le fonctionnement du SPARQL et de Wikidata afin que vous ayez les bases pour créer vos propres requêtes.

1. WikiData késako ?

Wikidata est en quelque sorte la version "web sémantique" de Wikipedia, c'est une base de données libre et ouverte qui peut être éditées par des êtres humains et des machines. Le contenu est structuré sous forme de données liées entre elles.

Le **web sémantique** est défini par Tim Berners-Lee comme «une toile de données qui peuvent être traitées directement et indirectement par des machines pour aider leurs utilisateurs à créer de nouvelles connaissances». lien vers Wikipédia .

Le format de donnée utilisé est appelé le **RDF** *(Resource Description Framework)*, les données y sont stockées sous forme d'un triplet composé de trois éléments :

— le «**sujet**» représente la ressource à décrire, par exemple la ville de Paris.
— le «**prédicat**» représente un type de propriété applicable à cette ressource, par exemple un prédicat de type surnom.
— l' «**objet**» représente une donnée ou une autre ressource: c'est la valeur de la propriété, dans notre exemple la valeur "Ville-Lumière".

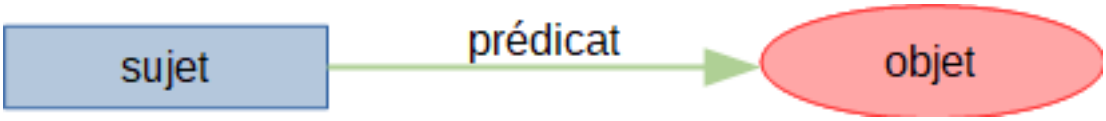

FIGURE 1.1. – Schéma de l'organisation des données

Pour explorer ces données, on utilise le langage de requête **SPARQL** , langage proche du **SQL** adapté pour traiter des données de format RDF.

2. Organisation des données de Wikidata

Pour retrouver les propriétés à utiliser lors de vos requêtes, vous pouvez directement vous rendre sur la page d'un élément de Wikidata et voir de quelles propriétés il est composé. Par exemple, avec la ville de Paris: https://www.wikidata.org/wiki/Q90 .

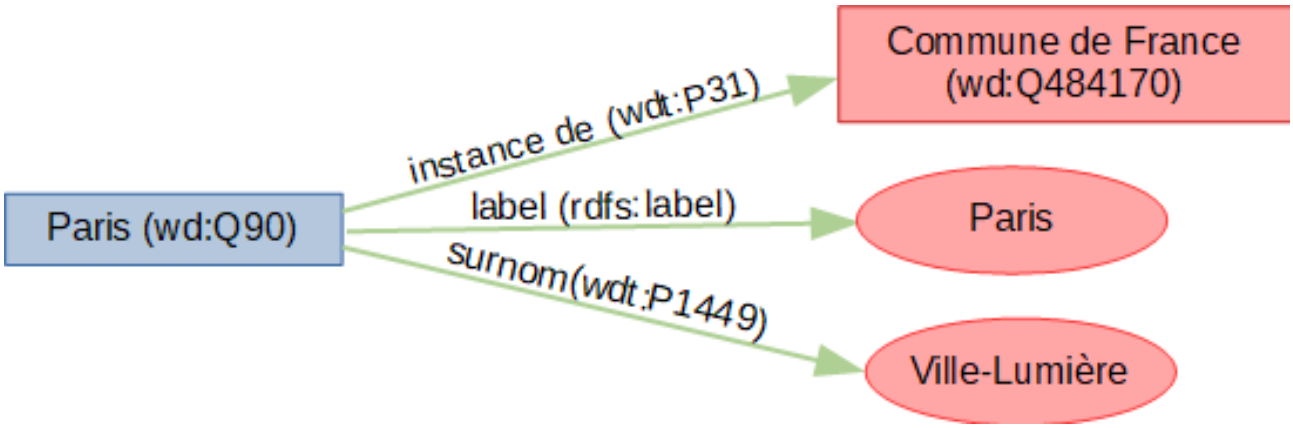

FIGURE 2.2. – Schéma de la structuration des données de Wikidata

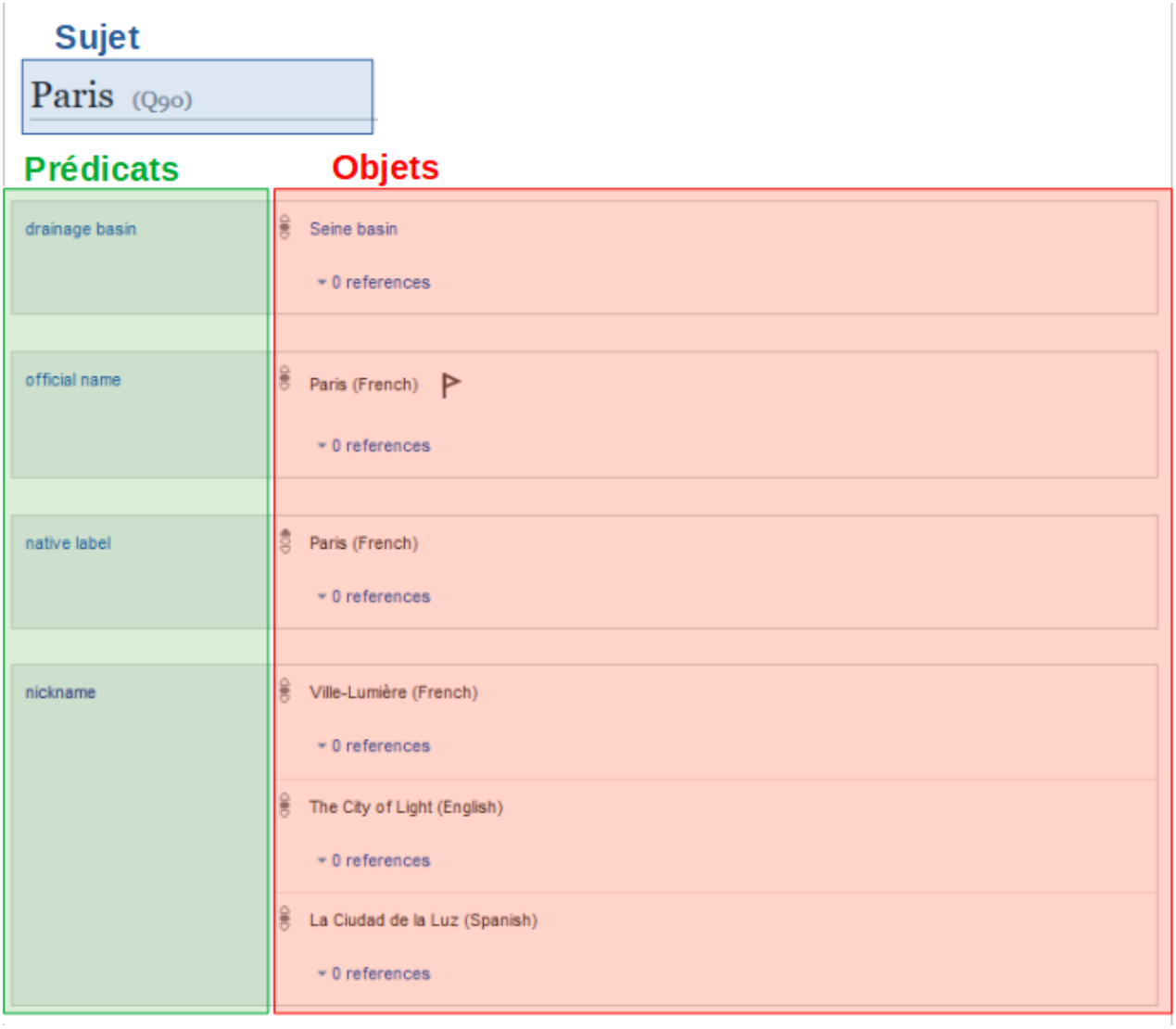

FIGURE 2.3. – Structure des données d'une page Wikidata

Pour récupérer le mot-clé à utiliser dans les requêtes qui vont suivre, il suffit de passer le curseur sur un élément:

FIGURE 2.4. – Récupération de l'identifiant du prédicat pour la ressource "instance of" : P31

FIGURE 2.5. – Récupération de l'identifiant de l'objet pour la ressource "commune of France" : Q484170

Wikidata, comme Wikipédia, possède des données sur beaucoup de sujets différents (histoire, lieux, sport, etc). Dans ce tutoriel, on va se concentrer sur une exploration des données liées aux villes où il y a beaucoup de requêtes différentes à découvrir.

Pour exécuter des requêtes sur Wikidata il suffit de se rendre à cette adresse: https://query.wikidata.org . Bien que Wikidata regroupe un nombre impressionnant de données, beaucoup de choses sont manquantes ou inexactes. En effet, comme Wikipédia, Wikidata est basé sur la participation bénévole de la communauté, à laquelle vous pouvez contribuer afin de l'améliorer .

3. Fonctionnement du SPARQL

Le SPARQL est un langage de requête inspiré du SQL permettant d'interroger, de créer et de modifier des données sous format RDF.

Dans le tutoriel nous verrons uniquement le fonctionnement des requêtes d'interrogation. Wikidata étant une base de données publique, il n'est pas permis d'insérer ou de modifier dans données via le SPARQL.

La requête se découpe en plusieurs parties:

- **SELECT**: Correspond aux noms des variables que l'on va chercher à récupérer.
- **WHERE**: Permet d'indiquer quelles données l'on souhaite récupérer et de filtrer ces données.
- **ORDER**: *[Optionnel]* Permets d'ordonner les données.
- **LIMIT**: *[Optionnel]* Nombre maximal de résultats récupérés.

i Les variables en SPARQL commencent par un `?`, ce qui nous permet de les distinguer des autres éléments.

La requête de base pour récupérer tous les contenus d'une base de données est celle-ci

```
SELECT ?sujet ?objet ?predicat
WHERE {
 ?sujet ?objet ?predicat
}
```

On récupère tous les sujets, les objets et les prédicats contenus dans la base donc toutes les données.

> Les requêtes s'écrivent toujours sous un format sujet, objets et prédicat qui peuvent être soit des variables, soit directement des valeurs.

Voici un exemple de requête schématique où l'on récupère tous les articles d'un site via le SPARQL.

```
SELECT ?mesArticle
WHERE {
 ?mesArticle type article
}
```

Dans ce cas, nous avons une variable appelée `mesArticle` qui a pour prédicat `type` et pour objet `article`. Donc on récupère tous les éléments de type `article`.

4. À la découverte des plus grande villes de France

Pour commencer, nous allons récupérer toutes les communes de France contenues dans la base de données, c'est-à-dire tous les éléments qui sont des **instances** (`wdt:P31`) de **commune de France** (`wd:Q484170`).

```
SELECT ?commune
WHERE
{
  ?commune wdt:P31 wd:Q484170
}
```

Listing 1 – Requête sur Wikidata

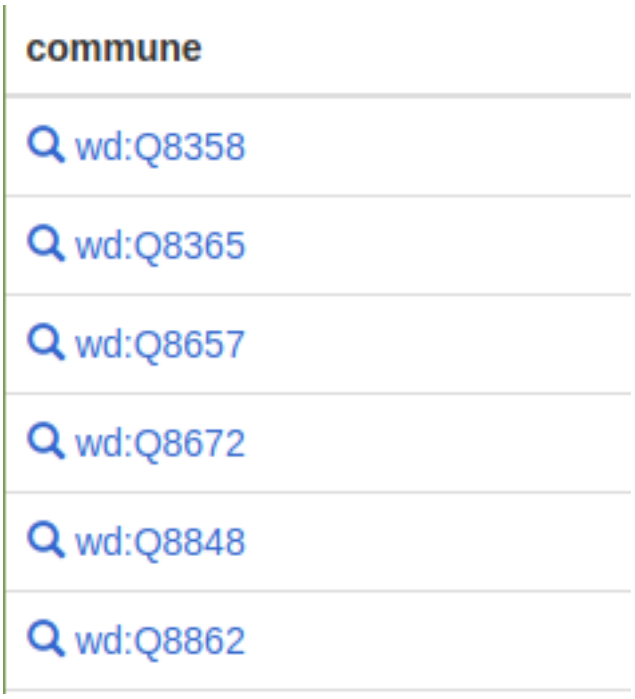

FIGURE 4.6. – Nos premiers résultats (sur plus de 40 000 résultats)

Comme vous pouvez le voir, les résultats ne sont pas très parlants. Nous allons donc chercher à afficher le label de la ville, ce qui correspond à son nom.

```
SELECT ?commune ?communeLabel
WHERE
{
  ?commune wdt:P31 wd:Q484170
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
```

Listing 2 – Requête sur Wikidata

Pour récupérer le label, nous définissons une ligne: `SERVICE wikibase:label ...` qui nous permet d'initialiser la récupération automatique des labels. Après cela, pour récupérer le label de la commune, il suffit d'ajouter une nouvelle variable dans la sélection nommée `?communeLabel` (nom de la variable + "Label"). Allez sur ce lien si vous souhaitez plus d'informations sur le fonctionnement de ce service .

commune	communeLabel
wd:Q90	Paris
wd:Q285	Cognac
wd:Q288	Tours
wd:Q342	Quimper
wd:Q343	La Flèche

FIGURE 4.7. – Les communes avec, à droite, leurs labels

5. Les villes les plus peuplées de France

Dans cette seconde étape, nous allons chercher à récupérer les 25 communes les plus peuplées de France.
Pour cela, nous allons afficher d'abord la population des communes:

FIGURE 5.8. – La propriété wdt:P1082 correspond à la population de la ville

```
SELECT ?commune ?communeLabel ?pop
WHERE
{
  ?commune wdt:P31 wd:Q484170 .
  ?commune wdt:P1082 ?pop
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
```

Le symbole `.` est indispensable à chaque fin de ligne si elle est suivie d'une ligne de requête, mais il n'est pas indispensable si la ligne suivant un service ou un filtre.

Il ne nous reste plus qu'à ordonner les données du plus grand au plus petit avec un `ORDER BY DESC` en fin de requête. Nous allons également limiter le nombre de résultats à 25 pour améliorer la vitesse de réponse de notre requête.

```
SELECT ?commune ?communeLabel ?pop
WHERE
{
  ?commune wdt:P31 wd:Q484170 .
  ?commune wdt:P1082 ?pop
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
ORDER BY DESC(?pop)
LIMIT 25
```

commune	communeLabel	pop
wd:Q90	Paris	2187526
wd:Q23482	Marseille	855393
wd:Q456	Lyon	506615
wd:Q7880	Toulouse	482738

FIGURE 5.9. – Le début de la liste des 25 villes les plus peuplées de France

Nous pouvons également afficher nos résultats sur une carte. Pour cela, il nous suffit de récupérer les coordonnées des villes et de choisir un affichage de résultat sous forme de carte.

FIGURE 5.10. – La propriété coordonnée

```
SELECT ?commune ?communeLabel ?pop ?coords
WHERE
{
  ?commune wdt:P31 wd:Q484170 .
  ?commune wdt:P1082 ?pop .
  ?commune wdt:P625 ?coords
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
ORDER BY DESC(?pop)
LIMIT 25
```

Listing 3 – Requête sur wikidata

Il nous faut choisir l'affichage sous forme de carte :

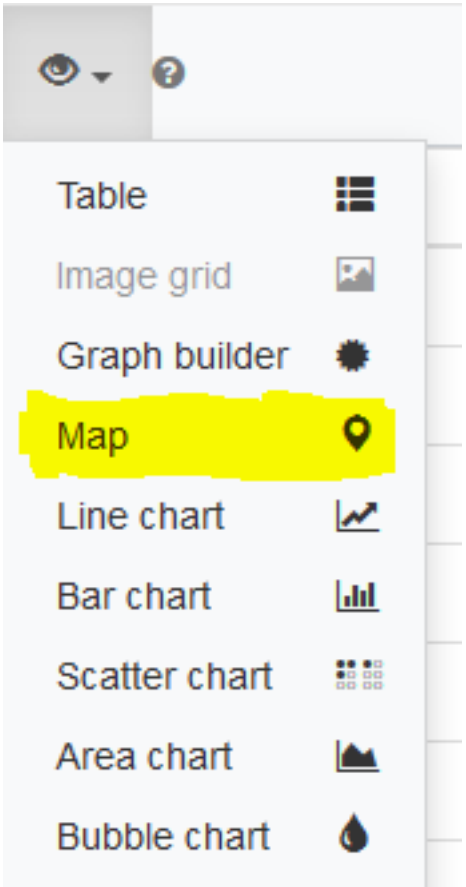

FIGURE 5.11. – On obtient une jolie carte

Nous allons maintenant essayer des requêtes un peu plus fantaisistes. 🍊

6. La ville de plus de 100 000 habitants la moins peuplée

Pour obtenir la ville la moins peuplée mais ayant plus 100 000 habitants, nous allons premièrement filtrer les villes ayant plus de 100 000 habitants grâce à la propriété FILTER:

```
SELECT ?commune ?communeLabel ?pop
WHERE
{
  ?commune wdt:P31 wd:Q484170 .
  ?commune wdt:P1082 ?pop
  FILTER (?pop > 100000)
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
ORDER BY DESC(?pop)
```

Les filtres s'écrivent comme ceci `FILTER(condition)`. Ils nous permettent de récupérer uniquement les éléments qui valident la ou les conditions contenues dans la parenthèse. On peut appliquer des conditions sur des nombres, des dates, des textes, etc.
Il nous faut récupérer la dernière de ces villes. Pour cela, on inverse l'ordre de tri et, pour avoir une seule ville, on limite le résultat retourné à 1.

```
SELECT ?commune ?communeLabel ?pop
WHERE
{
  ?commune wdt:P31 wd:Q484170 .
  ?commune wdt:P1082 ?pop
  FILTER (?pop > 100000)
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
ORDER BY(?pop)
LIMIT 1
```

Listing 4 – Requête sur wikidata

commune	communeLabel	pop
wd:Q40898	Nancy	104286

FIGURE 6.12. – On obtient la ville de Nancy

Si vous lisez ce tutoriel longtemps après son écriture, il est possible que le résultat ne soit plus le même. En effet, les populations des villes sont en constante évolution.

7. Tri des villes par densité de population

L'objectif de cette partie va être de trier les villes de plus de 100 000 habitants par leur densité de population.
Pour commencer, nous allons faire une requête avec les communes de 100 000 habitants triées de la plus peuplée à la moins peuplée. Nous allons également y ajouter la superficie.

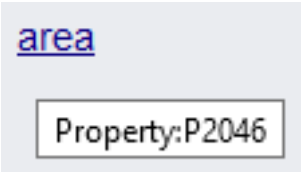

FIGURE 7.13. – Propriété superficie

```
SELECT ?commune ?communeLabel ?pop ?area
WHERE
{
  ?commune wdt:P31 wd:Q484170 .
  ?commune wdt:P1082 ?pop .
  ?commune wdt:P2046 ?area
  FILTER (?pop > 100000)
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
```

```
10 ORDER BY DESC(?pop)
```

Pour obtenir la densité, il va nous falloir diviser la population par la superficie. Pour faire ce calcul, nous allons utiliser la propriété `bind`.
Il ne nous restera plus qu'ordonner les résultats par rapport à la densité calculée.

```
1  SELECT ?commune ?communeLabel ?pop ?area ?density
2  WHERE
3  {
4    ?commune wdt:P31 wd:Q484170 .
5    ?commune wdt:P1082 ?pop .
6    ?commune wdt:P2046 ?area
7    FILTER (?pop > 100000)
8    BIND( (?pop / ?area) as ?density )
9    SERVICE wikibase:label { bd:serviceParam wikibase:language
         "[AUTO_LANGUAGE],fr,en". }
10 }
11 ORDER BY DESC(?density)
```

Listing 5 – Requête sur wikidata

i Le mot-clé `BIND` permet de créer une nouvelle variable à partir d'opérations effectuées sur d'autres variables. On peut y faire des opérations de calcul, mais également des opérations sur des chaines de caractères.

commune	communeLabel	pop	area	density
wd:Q90	Paris	2187526	105.4	20754.516129032258064516 13
wd:Q172455	Boulogne-Billancourt	120071	6.17	19460.453808752025931928 69
wd:Q193370	Montreuil	109897	8.92	12320.291479820627802690 58
wd:Q456	Lyon	506615	47.87	10583.141842490077292667 64
wd:Q582	Villeurbanne	147712	14.52	10173.002754820936639118 46
wd:Q160506	Saint-Denis	111135	12.36	8991.5048543689320388349 5
wd:Q1289	Grenoble	158454	18.13	8739.878654164368450082 74

FIGURE 7.14. – Notre résultat

8. Exploration avancée

Dans cette dernière section, nous allons chercher à trier les villes de plus de 100000 habitants par le nombre de personnes y étant nées contenu dans Wikidata.
Comme première étape, nous récupérons toutes les personnes nées dans une commune de plus de 100000 habitants.

FIGURE 8.15. – La propriété P19 correspond au lieu de naissance d'une personne

```
SELECT ?commune ?communeLabel ?person ?personLabel
WHERE
{
  ?commune wdt:P31 wd:Q484170 .
  ?commune wdt:P1082 ?pop .
  ?person wdt:P19 ?commune .
  ?person wdt:P31 wd:Q5
  FILTER (?pop > 100000)
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
LIMIT 100
```

Pour avoir le nombre de personnes nées dans chaque ville, il va falloir grouper nos données afin d'avoir une ligne par ville.

Ville	Personne
Paris	Roger Dupont
Paris	Émilie Riel
Paris	Julie Lépicier
Lyon	Claudette Beaudoin
Lyon	Raymond Desnoyer

FIGURE 8.16. – Notre visualisation actuelle

Ville	Nombre de personnes
Paris	3
Lyon	2

FIGURE 8.17. – Ce que l'on souhaite obtenir

Pour cela, on utilise un `GROUP BY` sur les propriétés `commune` `communeLabel` et on ajoute dans le `SELECT` un décompte du nombre de personnes grâce à la propriété `count()`.

```
SELECT ?commune ?communeLabel (count(distinct ?person) as
    ?nbPerson)
WHERE
{
  ?commune wdt:P31 wd:Q484170 .
  ?commune wdt:P1082 ?pop .
  ?person wdt:P19 ?commune .
  ?person wdt:P31 wd:Q5
  FILTER (?pop > 100000)
```

```
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
GROUP BY ?commune ?communeLabel
ORDER BY DESC(?nbPerson)
```

Listing 6 – Requête sur wikidata

i Le mot-clé GROUP BY fonctionne comme en SQL, il est généralement utilisé avec des fonctions agrégations (COUNT, MAX, MIN, SUM, AVG) pour grouper les résultats d'une ou plusieurs colonnes.

commune	communeLabel	nbPerson
wd:Q90	Paris	23689
wd:Q456	Lyon	3099
wd:Q23482	Marseille	2697
wd:Q6602	Strasbourg	2017
wd:Q1479	Bordeaux	1731
wd:Q7880	Toulouse	1676

FIGURE 8.18. – Notre résultat final

9. Petits défis

Je vous propose quelques requêtes supplémentaires à réaliser par vous-même:
Les 10 communes de France de plus de 100 000 habitants dans lesquelles sont nés le plus de joueurs de football.

Contenu masqué n°1

Liste des lieux où sont nés les Présidents de la République française.

Contenu masqué n°2

Personnes dont les deux parents sont nés à Paris.

Contenu masqué n°3

Autres nationalités des personnes de nationalité française, groupées par nationalités et ordonnées par nombre de personnes (nationalité = *country of citizenship*).

Contenu masqué n°4

10. Pour aller plus loin

Cette exploration est maintenant terminée, vous y avez découvert les bases du SPARQL et de l'exploration de Wikidata, mais il vous reste encore beaucoup à apprendre… 🍊
En effet, le SPARQL est un langage très complet qui permet de faire beaucoup de choses. Si vous désirez approfondir SPARQL et/ou Wikidata, vous trouverez ci-dessous une petite liste de ressources à consulter:

— Vous pouvez continuer à explorer les données de Wikidata avec d'autres types de requêtes disponibles dans le menu `Exemples` de Wikidata.

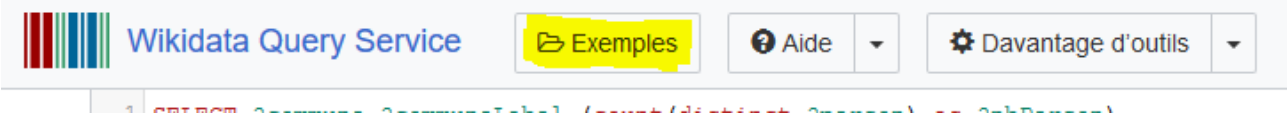

FIGURE 10.19. – Le menu Exemples de Wikidata

— Vous pouvez également lire ce tutoriel en anglais qui vous présente toutes les fonctionnalités du SPARQL sur Wikidata.
— Pour créer votre propre base de données RDF, je vous conseille d'utiliser Apache Jena Fuseki .
— Pour aller plus loin avec le langage SPARQL, je vous conseille ce tutoriel du site Developpez qui introduit aux principales fonctionnalités du langage, mais aussi le livre *Learning SPARQL* (en anglais), ouvrage très complet qui vous offre un tour d'horizon des fonctionnalités du langage SPARQL.

Je vous remercie d'avoir lu ce tutoriel jusqu'au bout, et j'espère qu'il vous a plu.
Merci particulièrement à @Vanadiae, @QuentinC et @Arius pour leurs relectures et leurs conseils d'amélioration.
Pour ceux que ça intéresse, j'ai créé un outil permettant de créer des requêtes sur Wikidata à partir de dessins de graphe. Plus d'informations sur le sujet du forum .
N'hésitez pas à venir partager vos requêtes dans les commentaires et remonter vos éventuels suggestions et points à améliorer.

Contenu masqué

Contenu masqué n°1

```
SELECT ?commune ?communeLabel (count(distinct ?person) as
    ?nbPerson)
WHERE
{
  ?commune wdt:P31 wd:Q484170 .
  ?commune wdt:P1082 ?pop .
  ?person wdt:P19 ?commune .
  ?person wdt:P31 wd:Q5 .
  ?person wdt:P106 wd:Q937857
  filter (?pop > 100000)
```

Contenu masqué

```
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
GROUP BY ?commune ?communeLabel
ORDER BY DESC(?nbPerson)
LIMIT 10
```

Retourner au texte.

Contenu masqué n°2

```
SELECT ?commune ?communeLabel (count(distinct ?person) as
    ?nbPerson)
WHERE
{
  ?person wdt:P39 wd:Q191954 .
  ?person wdt:P31 wd:Q5 .
  ?commune wdt:P31 wd:Q484170 .
  ?person wdt:P19 ?commune
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
GROUP BY ?commune ?communeLabel
ORDER BY DESC(?nbPerson)
```

Retourner au texte.

Contenu masqué n°3

```
SELECT ?person ?personLabel
WHERE
{
  ?person wdt:P22 ?father .
  ?person wdt:P25 ?mother .
  ?father wdt:P19 wd:Q90 .
  ?mother wdt:P19 wd:Q90
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
```

Retourner au texte.

Contenu masqué n°4

```
SELECT ?otherNation ?otherNationLabel (count(distinct ?person) as
    ?nbPerson)
WHERE
{
  ?person wdt:P27 wd:Q142  .
  ?person wdt:P27 ?otherNation
  FILTER (?otherNation != wd:Q142)
  SERVICE wikibase:label { bd:serviceParam wikibase:language
      "[AUTO_LANGUAGE],fr,en". }
}
GROUP BY ?otherNation ?otherNationLabel
ORDER BY DESC(?nbPerson)
```

Retourner au texte.