CONCEVEZ ET ANALYSEZ UNE BASE DE DONNÉES NOSQL

Projet Data Engineer – KELLENI Antoine

lien GitHub

OPENCLASSROOMS

mongoDB

ROADMAP DE LA MISSION

1. **Contexte & objectifs de mission**
2. Exploration des données
3. Requête simple en CLI
4. Requête complexe PyMongo/Polar
5. Visualisation dashboard Power BI
6. Import données Lyon
7. Réplication avec ReplicatSet
8. Distribuer les données avec Sharding

Contexte & objectifs de mission

L’association **NetCités**, gestionnaire de plateformes de locations courte durée, a **subi un crash de sa base SQL**.
Face à la croissance des données et à la diversité des informations (annonces, hôtes, disponibilités), elle souhaite migrer vers MongoDB, une base NoSQL plus flexible et scalable.

Objectif :
Importer et analyser les données de Paris, intégrer celles de Lyon par fusion, puis assurer la résilience par réplication et la performance par un sharding permettant de filtrer par ville.
Pipeline : Import → contrôle qualité → fusion → réplication → sharding.

ROADMAP DE LA MISSION

1. Contexte & objectifs de mission
2. **Exploration des données**
3. Requête simple en CLI
4. Requête complexe PyMongo/Polar
5. Visualisation dashboard Power BI
6. Import données Lyon
7. Réplication avec ReplicatSet
8. Distribuer les données avec Sharding

Exploration des données

Dataset :

Dictionnaire de données → lien dictionnaire de données

Données Paris → lien donnée CSV

Données Lyon → lien donnée CSV

shape: (5, 75)

id	listing_url	scrape_id	last_scraped	source	name	description	neighborhood_overview	picture_url	host_id	host_url	host_na
i64	str	i64	str	str	str	str	str	str	i64	str	
80260	"https://www.airbnb.com/rooms/8...	20240610195007	"2024-06-13"	"previous scrape"	"Nice studio in Jourdain's vill...	null	null	"https://a0.muscache.com/pictur...	333548	"https://www.airbnb.com/users/s...	"Charlo
3109	"https://www.airbnb.com/rooms/3...	20240610195007	"2024-06-12"	"city scrape"	"zen and calm"	"Lovely Appartment with one bed...	"Good restaurants very clo...	"https://a0.muscache.com/pictur...	3631	"https://www.airbnb.com/users/s...	"Ar
80301	"https://www.airbnb.com/rooms/8...	20240610195007	"2024-06-13"	"city scrape"	"toits de Paris"	"On the top,sharing my space,yo...	"SAFE neighborhood,late bus and...	"https://a0.muscache.com/pictur...	433758	"https://www.airbnb.com/users/s...	"Geneviè
5396	"https://www.airbnb.com/rooms/5...	20240610195007	"2024-06-13"	"city scrape"	"Your perfect Paris studio on l...	"NEW SOFA-BED SINCE JUNE 2023, ...	"You are within walking distanc...	"https://a0.muscache.com/pictur...	7903	"https://www.airbnb.com/users/s...	"Borz
7397	"https://www.airbnb.com/rooms/7...	20240610195007	"2024-06-13"	"city scrape"	"MARAIS - 2ROOMS APT - 2/4 PEOP...	"VERY CONVENIENT, WITH THE BEST...	null	"https://a0.muscache.com/pictur...	2626	"https://www.airbnb.com/users/s...	"Fran

Aperçu des données

Aperçu des colonnes

```
['id',
 'listing_url',
 'scrape_id',
 'last_scraped',
 'source',
 'name',
 'description',
 'neighborhood_overview',
 'picture_url',
 'host_id',
 'host_url',
 'host_name',
 'host_since',
 'host_location',
 'host_about',
 'host_response_time',
 'host_response_rate',
 'host_acceptance_rate',
 'host_is_superhost',
 'host_thumbnail_url',
 'host_picture_url',
 'host_neighbourhood',
 'host_listings_count',
 'host_total_listings_count',
 'host_verifications',
...
 'calculated_host_listings_count',
 'calculated_host_listings_count_entire_homes',
 'calculated_host_listings_count_private_rooms',
 'calculated_host_listings_count_shared_rooms',
 'reviews_per_month']
```

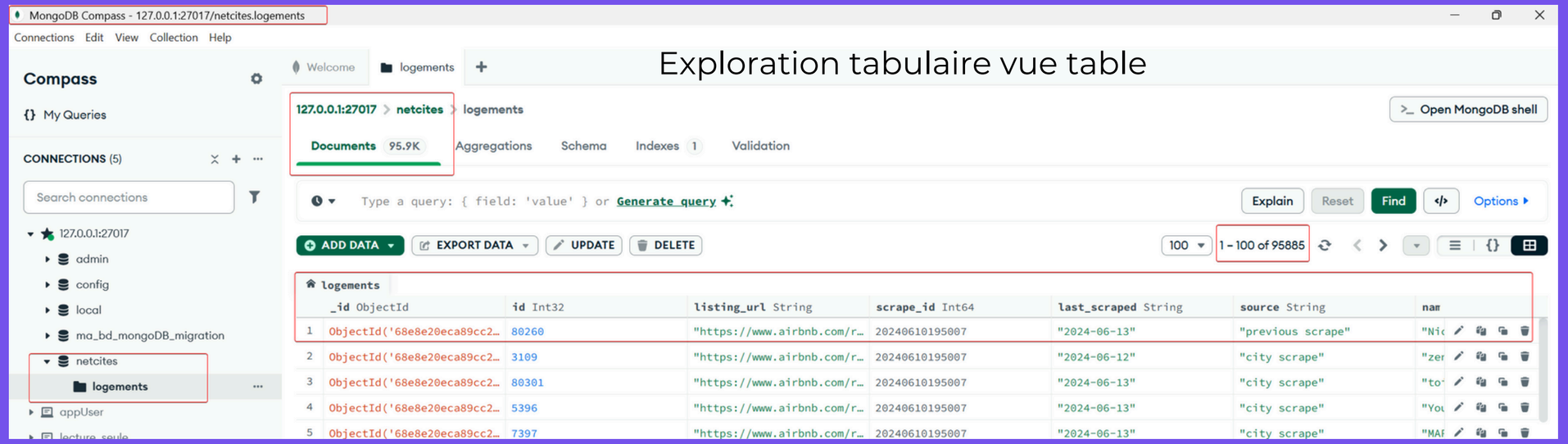
Aperçu des description

Data Dictionary for listings.csv detailed file

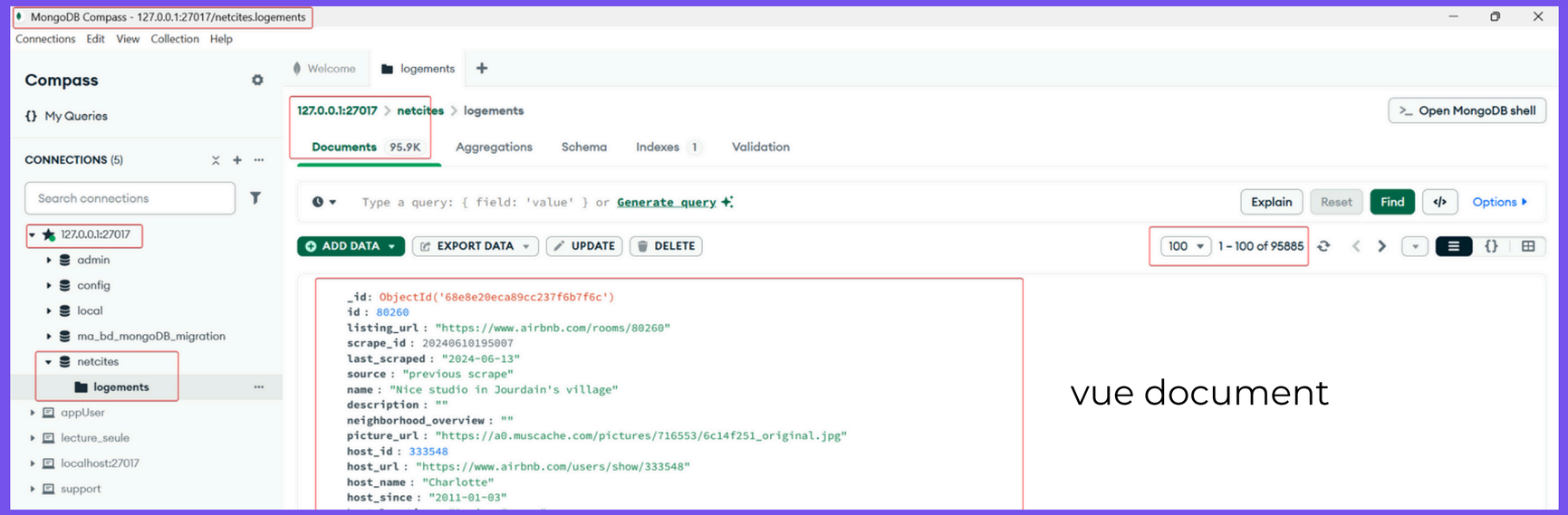
File Name	listings.csv
Version	4.3
Date Introduced	août. 2022
Description	

Field	Type	Calculated	Description
id	integer		Unique identifier for the listing
listing_url	text	y	
scrape_id	bigint	y	Our "Scrape" reference this was part of
last_scraped	datetime	y	UTC. The date and time this listing was
source	text		One of "neighbourhood search" or "previous
name	text		Name of the listing
description	text		Detailed description of the listing
neighborhood_overview	text		Host's description of the neighbourhood
picture_url	text		URL to the home rental website hosted regular
host_id	integer		Unique identifier for the host/user
host_url	text	y	The home rental website page for the host
host_name	text		Name of the host. Usually just the first
host_since	date		The date the host/user was created. For hosts
host_location	text		The host's self reported location
host_about	text		Description about the host
host_response_time			
host_response_rate			
host_acceptance_rate			That rate at which a host accepts booking
host_is_superhost	boolean [t=true; f=false]		
host_thumbnail_url	text		
host_picture_url	text		
host_neighbourhood	text		
host_listings_count	text		The number of listings the host has (per the
host_total_listings_count	text		The number of listings the host has (per the
host_verifications			
host_has_profile_pic	boolean [t=true; f=false]		
host_identity_verified	boolean [t=true; f=false]		
neighbourhood	text		
neighbourhood_cleansed	text	y	The neighbourhood as geocoded using the
neighbourhood_group_clear	text	y	The neighbourhood group as geocoded using
latitude	numeric		Uses the World Geodetic System (WGS84)
longitude	numeric		Uses the World Geodetic System (WGS84)
property_type	text		Self selected property type. Hotels and Bed
room_type	text		[Entire home/apt|Private room|Shared
accommodates	integer		The maximum capacity of the listing
bathrooms	numeric		The number of bathrooms in the listing
bathrooms_text	string		The number of bathrooms in the listing.
bedrooms	integer		The number of bedrooms
beds	integer		The number of bed(s)
amenities	json		
price	currency		daily price in local currency.
minimum_nights	integer		minimum number of night stay for the listing
maximum_nights	integer		maximum number of night stay for the listing

Interface graphique MongoDB Compass

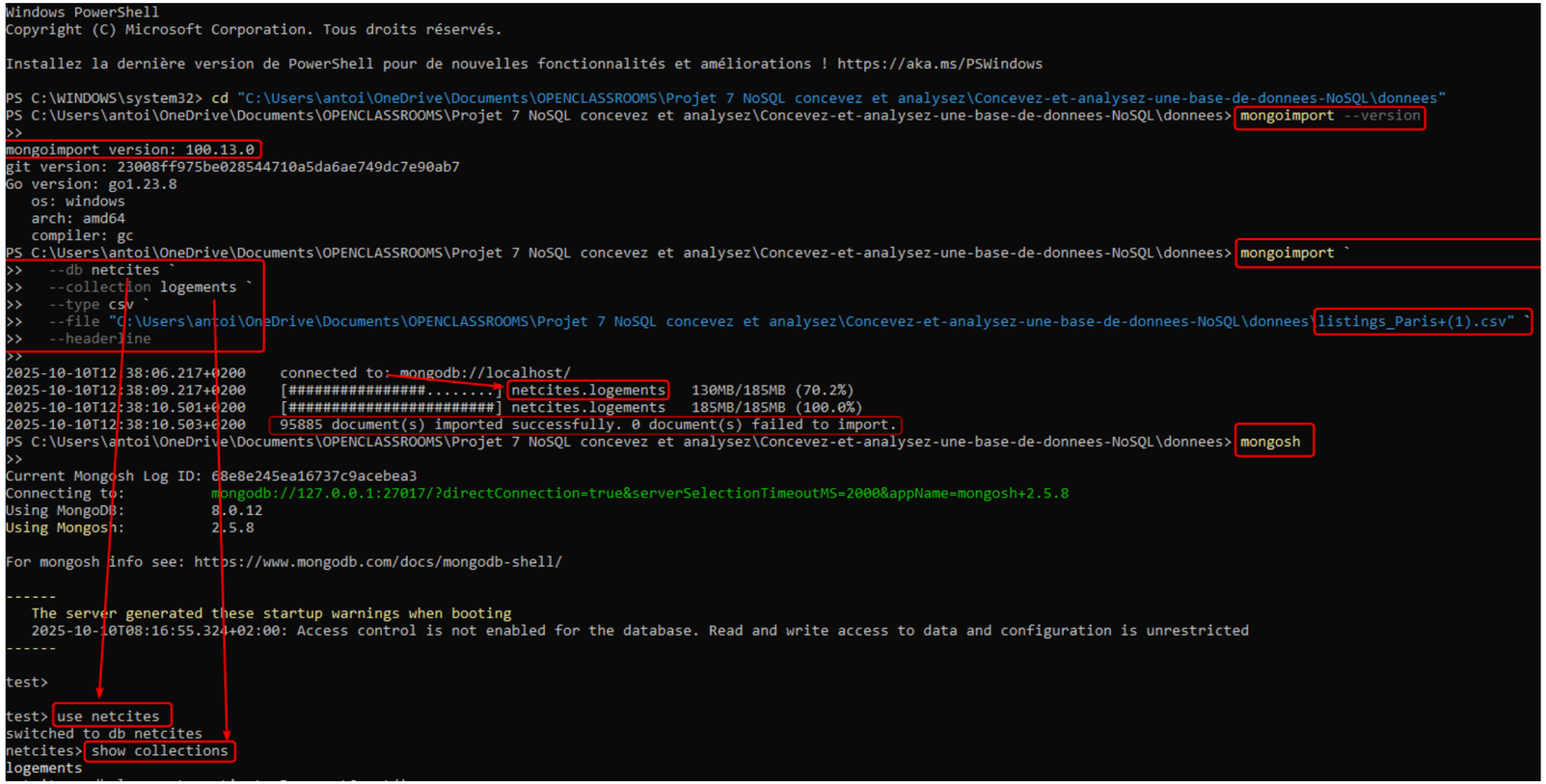


Exploration tabulaire vue table



vue document

Import des données (Paris)



- Import CSV Paris via mongoimport dans netcites.logements.
- 95 885 documents importés, 0 erreur.
- Connexion mongosh et vérification de la collection.

L'ingestion brute de Paris est réussie et traçable en CLI.

Échantillon & volumétrie

- findOne() pour valider la structure et les champs clés.
- countDocuments() = 95 885 (cohérent avec l'import).

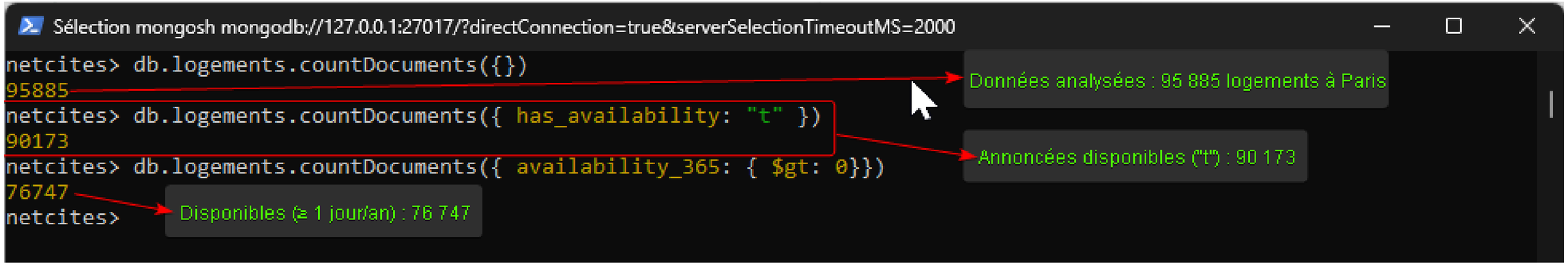
```
netcites> db.logements.countDocuments({})
95885
```

```
netcites> db.logements.findOne()
{
  _id: ObjectId('68e8e20eca89cc237f6b7f6c'),
  id: 80260,
  listing_url: 'https://www.airbnb.com/rooms/80260',
  scrape_id: Long('20240610195007'),
  last_scraped: '2024-06-13',
  source: 'previous scrape',
  name: "Nice studio in Jourdain's village",
  description: '',
  neighborhood_overview: '',
  picture_url: 'https://a0.muscache.com/pictures/716553/6c14f251_original.jpg',
  host_id: 333548,
  host_url: 'https://www.airbnb.com/users/show/333548',
  host_name: 'Charlotte',
  host_since: '2011-01-03',
  host_location: 'Paris, France',
  host_about: "My name is Charlotte, I'm 26 and I study cinema production. I'll
  host_response_time: 'N/A',
  host_response_rate: 'N/A',
  host_acceptance_rate: 'N/A',
  host_is_superhost: 'f',
  host_thumbnail_url: 'https://a0.muscache.com/im/users/333548/profile_pic/1350
  host_picture_url: 'https://a0.muscache.com/im/users/333548/profile_pic/135032
  host_neighbourhood: 'Buttes-Chaumont - Belleville',
  host_listings_count: 1,
  host_total_listings_count: 1,
  host_verifications: "['email', 'phone']",
  host_has_profile_pic: 't',
  host_identity_verified: 't',
  neighbourhood: '',
  neighbourhood_cleansed: 'Ménilmontant',
  neighbourhood_group_cleansed: '',
  latitude: 48.87131,
  longitude: 2.38848,
  property_type: 'Entire rental unit',
  room_type: 'Entire home/apt',
  accommodates: 3,
  bathrooms: '',
  bathrooms_text: '1 bath',
  bedrooms: 1,
  beds: '',
  amenities: '["Hangers", "Essentials", "Wifi", "Dishes and silverware", "Dedic
  price: '',
  minimum_nights: 2,
  maximum_nights: 730,
  minimum_minimum_nights: 2,
  maximum_minimum_nights: 2,
  minimum_maximum_nights: 730,
  maximum_maximum_nights: 730,
  minimum_nights_avg_ntm: 2,
  maximum_nights_avg_ntm: 730,
  calendar_updated: '',
  has_availability: 't',
  availability_30: 0,
  availability_60: 0,
  availability_90: 0,
  availability_365: 0,
  calendar_last_scraped: '2024-06-13',
  number_of_reviews: 206,
  number_of_reviews_ltm: 0,
  number_of_reviews_l30d: 0,
  first_review: '2011-04-18',
  last_review: '2021-10-03',
  review_scores_rating: 4.63,
  review_scores_accuracy: 4.61,
  review_scores_cleanliness: 4.75,
  review_scores_checkin: 4.85,
  review_scores_communication: 4.78,
  review_scores_location: 4.61,
  review_scores_value: 4.64,
  license: Long('7512005340473'),
  instant_bookable: 'f',
  calculated_host_listings_count: 1,
  calculated_host_listings_count_entire_homes: 1,
  calculated_host_listings_count_private_rooms: 0,
  calculated_host_listings_count_shared_rooms: 0,
  reviews_per_month: 1.29
}
```

ROADMAP DE LA MISSION

1. Contexte & objectifs de mission
2. Exploration des données
3. **Requête simple en CLI**
4. Requête complexe PyMongo/Polar
5. Visualisation dashboard Power BI
6. Import données Lyon
7. Réplication avec ReplicatSet
8. Distribuer les données avec Sharding

Disponibilité (KPIs de base)



- Annonces avec disponibilité (has_availability:"t") → 95 885 logements à Paris.
- Annonces avec availability_365 > 0 → 76 747 logements .

~95% ont un statut dispo((90 173/95 885)*100) et ~80% ont au moins 1 jour dispo/an ((76 747/95 885)*100)

Répartition par type de logement

Agrégat $sortByCount: "$property_type".

Top catégories :

1. Entire rental unit (80.5k),
2. Private room in rental unit (5.9k),

L'offre est massivement pour des logements entiers.

```
netcites> db.logements.countDocuments()
95885
netcites> db.logements.aggregate([{$sortByCount: "$property_type"}])
[
  { _id: 'Entire rental unit', count: 80516 },
  { _id: 'Private room in rental unit', count: 5980 },
  { _id: 'Entire condo', count: 2679 },
  { _id: 'Room in boutique hotel', count: 1256 },
  { _id: 'Entire loft', count: 1007 },
  { _id: 'Room in hotel', count: 1003 },
  { _id: 'Entire home', count: 696 },
  { _id: 'Private room in bed and breakfast', count: 590 },
  { _id: 'Private room in condo', count: 344 },
  { _id: 'Entire serviced apartment', count: 277 },
  { _id: 'Shared room in rental unit', count: 274 },
  { _id: 'Entire townhouse', count: 268 },
  { _id: 'Private room in home', count: 122 },
  { _id: 'Private room in townhouse', count: 86 },
  { _id: 'Private room in loft', count: 73 },
  { _id: 'Room in aparthotel', count: 73 },
  { _id: 'Shared room in hostel', count: 54 },
  { _id: 'Private room in hostel', count: 53 },
  { _id: 'Entire place', count: 46 },
  { _id: 'Private room in guesthouse', count: 45 }
]
Type "it" for more
netcites>
```

Annonces avec les plus d’avis (Top 5)

- Filtre number_of_reviews existant, tri décroissant, top 5.
- Affichage ciblé : name, host_name, property_type.

```
netcites> db.logements.find(
...   { number_of_reviews: { $exists: true } }, // filtre les documents ayant ce champ
...   { name: 1, host_name: 1, number_of_reviews: 1, property_type: 1} // affiche ces champs
... ).sort({ number_of_reviews: -1 }) // tri décroissant
... .limit(5)
[
  {
    _id: ObjectId('68e8e20eca89cc237f6bb707'),
    name: 'Sweet & cosy room next to Canal Saint Martin 💚',
    host_name: 'Alexandra',
    property_type: 'Room in boutique hotel',
    number_of_reviews: 3067
  },
  {
    _id: ObjectId('68e8e20fca89cc237f6bcfe1'),
    name: 'Double/Twin Room, close to Opera and the Louvre with breakfast included',
    host_name: 'David',
    property_type: 'Room in boutique hotel',
    number_of_reviews: 2620
  },
  {
    _id: ObjectId('68e8e20fca89cc237f6bf481'),
    name: 'Bed in Dorm of 8 Beds "The Big One" in Paris',
    host_name: 'The People',
    property_type: 'Shared room in hostel',
    number_of_reviews: 2294
  },
  {
    _id: ObjectId('68e8e20fca89cc237f6bf269'),
    name: 'Comfortable bed in shared rooms of 8 in Paris 12e',
    host_name: 'The People Paris Bercy',
    property_type: 'Shared room in hostel',
    number_of_reviews: 2105
  },
  {
    _id: ObjectId('68e8e20fca89cc237f6be3e0'),
    name: 'Nice Room for 2 people',
    host_name: 'Hotel De L’Aqueduc',
    property_type: 'Room in boutique hotel',
    number_of_reviews: 2048
  }
]
netcites>
```

Nombre d’hôtes uniques

```
netcites> db.logements.distinct("host_id").length
71979
```

distinct("host_id").length = 71 979 hôtes uniques.

Réservable instantanément (part %)

```
netcites> // Nombre total d'annonces
... var total = db.logements.estimatedDocumentCount();
...
... // Nombre de logements réservables instantanément ("t")
... var instant = db.logements.countDocuments({ instant_bookable: "t" });
...
... // Calcul du pourcentage
... var proportion = (instant / total * 100).toFixed(2);
...
... // Affichage formaté
... print("Total d'annonces :", total);
... print("Réservables instantanément :", instant);
... print("Proportion :", proportion + "%");
Total d'annonces : 95885
Réservables instantanément : 22094
Proportion : 23.04%
```

- Total : 95 885 ; instant_bookable:"t" : 22 094.
- Proportion ≈ 23,04 %.
- (22 094/95 885)*100 = 23,04%

Environ un quart des annonces sont bookables instantanément.

Multi-listings (hôtes avec ≥100 annonces)

- Agrégat par host_id, seuil ≥100 annonces.

Exemples :

1. Blueground (730),
2. Veeve (497),
3. Pierre De WeHost (420),

Il y a une présence d'opérateurs professionnels à large inventaire.

```
netcites> db.logements.aggregate([
...   { $group: { _id: "$host_id", nombre_annonces: { $sum: 1 }, nom_hote: { $first: "$host_name" } } },
...   { $match: { nombre_annonces: { $gt: 100 } } },
...   { $sort: { nombre_annonces: -1 } }
... ])
[
  { _id: 314994947, nombre_annonces: 730, nom_hote: 'Blueground' },
  { _id: 33889201, nombre_annonces: 497, nom_hote: 'Veeve' },
  { _id: 50502817, nombre_annonces: 426, nom_hote: 'Pierre De WeHost' },
  { _id: 50978178, nombre_annonces: 307, nom_hote: 'Sébastien' },
  {
    _id: 26981054,
    nombre_annonces: 274,
    nom_hote: 'Cédric De ClickYourFlat'
  },
  { _id: 460047164, nombre_annonces: 232, nom_hote: 'FlexLiving' },
  { _id: 7642792, nombre_annonces: 211, nom_hote: 'Ludovic' },
  { _id: 436103373, nombre_annonces: 210, nom_hote: 'David Et Warren' },
  { _id: 528015349, nombre_annonces: 189, nom_hote: 'Checkmyguest' },
  { _id: 125797498, nombre_annonces: 170, nom_hote: 'Welkeys' },
  { _id: 335998296, nombre_annonces: 154, nom_hote: 'Studioprestige' },
  { _id: 51567288, nombre_annonces: 149, nom_hote: 'Sweet Inn' },
  { _id: 1112584, nombre_annonces: 145, nom_hote: 'IntoParis' },
  { _id: 564251645, nombre_annonces: 137, nom_hote: 'Rusard' },
  { _id: 440295601, nombre_annonces: 132, nom_hote: 'Parisian Home' },
  { _id: 517515174, nombre_annonces: 122, nom_hote: 'Barnes' },
  { _id: 99040006, nombre_annonces: 120, nom_hote: 'Jérémy' },
  {
    _id: 28313443,
    nombre_annonces: 119,
    nom_hote: 'Michael & Johanna'
  },
  { _id: 24495283, nombre_annonces: 111, nom_hote: 'Giacomo' },
  { _id: 506389460, nombre_annonces: 109, nom_hote: 'Check My Guest' }
]
Type "it" for more
netcites> it
[
  { _id: 512746089, nombre_annonces: 104, nom_hote: 'Checkmyguest' },
  { _id: 374552379, nombre_annonces: 104, nom_hote: 'Checkmyguest' },
  { _id: 499962530, nombre_annonces: 103, nom_hote: 'Check My Guest' },
  { _id: 21630783, nombre_annonces: 102, nom_hote: 'Pierre' }
]
netcites>
```

Multi-listings (hôtes avec ≥100 annonces) en % sur le total

```
netcites> // 1. Nombre total d'hôtes uniques
... var total_hotes = db.logements.distinct("host_id").length
... print("Nombre total d'hôtes :", total_hotes)
...
... // 2. Nombre d'hôtes avec plus de 100 annonces
... var hotes_100 = db.logements.aggregate([
...   { $group: { _id: "$host_id", count: { $sum: 1 } } },
...   { $match: { count: { $gt: 100 } } }
... ]).toArray().length
... print("Nombre d'hôtes avec plus de 100 annonces :", hotes_100)
...
... // 3. Calcul du pourcentage
... var pourcentage = (hotes_100 / total_hotes * 100).toFixed(2)
... print("Pourcentage d'hôtes avec plus de 100 annonces :", pourcentage + "%")
...
Nombre total d'hôtes : 71979
Nombre d'hôtes avec plus de 100 annonces : 24
Pourcentage d'hôtes avec plus de 100 annonces : 0.03%
```

- Hôtes uniques : 71 979.
- Hôtes avec ≥ 100 annonces : 24.
- Part : 0,03 % des hôtes.

Une hyper-concentration marginale (quelques opérateurs très professionnels). Minoritaire en nombre mais impact potentiellement fort sur l'offre et les prix.

Super hôtes (qualité de l’offre)

```
netcites> // 1. Nombre total d'hôtes distincts
... var total_hotes = db.logements.distinct("host_id").length
... print("Nombre total d'hôtes :", total_hotes)
...
... // 2. Nombre de super hôtes distincts
... var super_hotes = db.logements.distinct("host_id", { host_is_superhost: "t" }).length
... print("Nombre de super hôtes :", super_hotes)
...
... // 3. Calcul du pourcentage
... var pourcentage_super = (super_hotes / total_hotes * 100).toFixed(2)
... print("Pourcentage de super hôtes :", pourcentage_super + "%")
...
Nombre total d'hôtes : 71979
Nombre de super hôtes : 10027
Pourcentage de super hôtes : 13.93%
```

- Hôtes uniques : 71 979.
- Super hôtes : 10 027.
- Part : 13,93 %.

À retenir : une base de qualité significative (~14 % de super hôtes), gage d’expérience fiable.

ROADMAP DE LA MISSION

1. Contexte & objectifs de mission
2. Exploration des données
3. Requête simple en CLI
4. **Requête complexe PyMongo/Polar**
5. Visualisation dashboard Power BI
6. Import données Lyon
7. Réplication avec ReplicatSet
8. Distribuer les données avec Sharding

le taux de réservation moyen par mois par type de logement

Par type de chambre

```
(df.with_columns(((30 - pl.col("availability_30").cast(pl.Float64,strict=False))/30).alias("taux"))
   .group_by("room_type")
   .agg((pl.col("taux").mean()*100).round(2).alias("taux_moyen_mensuel_%"))
   .sort("taux_moyen_mensuel_%", descending=True))
```

shape: (4, 2)

room_type	taux_moyen_mensuel_%
str	f64
"Entire home/apt"	71.29
"Private room"	70.29
"Shared room"	60.72
"Hotel room"	53.53

Par type de propriété

```
(df.with_columns(((30 - pl.col("availability_30").cast(pl.Float64,strict=False))/30).alias("taux"))
   .group_by("property_type")
   .agg((pl.col("taux").mean()*100).round(2).alias("taux_moyen_mensuel_%"))
   .sort("taux_moyen_mensuel_%", descending=True))
```

shape: (70, 2)

property_type	taux_moyen_mensuel_%
str	f64
"Entire bungalow"	100.0
"Shared room in ice dome"	100.0
"Cave"	100.0
"Dome"	100.0
"Private room in cave"	100.0
...	...
"Shipping container"	3.33
"Tower"	3.33
"Entire bed and breakfast"	0.0
"Barn"	0.0
"Castle"	0.0

La médiane des nombres d’avis pour tous les logements

```python
df.select(pl.col("number_of_reviews").cast(pl.Float64,strict=False).median()).item()
print("Médiane du nombre d'avis pour tous les logements:", df.select(pl.col("number_of_reviews").cast(pl.Float64,strict=False).median()).item())
```

Python

```
Médiane du nombre d'avis pour tous les logements: 3.0
```

La médiane des nombres d’avis par catégorie d’hôte

```python
print("Médiane du nombre d'avis par catégorie d'hôte :")

(df.with_columns(pl.col("number_of_reviews").cast(pl.Float64,strict=False))
   .group_by("host_is_superhost")
   .agg(pl.col("number_of_reviews").median().alias("median_number_of_reviews"))
   .sort("median_number_of_reviews", descending=True))
```

```
Médiane du nombre d'avis par catégorie d'hôte :
```

shape: (3, 2)

host_is_superhost	median_number_of_reviews
str	f64
"t"	24.0
null	12.5
"f"	2.0

La densité de logements par quartier de Paris

```
total = df.height
(df.filter(pl.col("neighbourhood_cleansed").is_not_null())
   .group_by("neighbourhood_cleansed")
   .agg(pl.count().alias("nombre_logements"))
   .with_columns((pl.col("nombre_logements")/total*100).round(2).alias("part_%"))
   .sort("nombre_logements", descending=True))
```

shape: (20, 3)

neighbourhood_cleansed	nombre_logements	part_%
str	u32	f64
"Buttes-Montmartre"	10555	11.01
"Popincourt"	8430	8.79
"Vaugirard"	7802	8.14
"Batignolles-Monceau"	6857	7.15
"Entrepôt"	6558	6.84
…	…	…
"Élysée"	2898	3.02
"Hôtel-de-Ville"	2821	2.94
"Palais-Bourbon"	2740	2.86
"Luxembourg"	2701	2.82
"Louvre"	2026	2.11

Les quartiers avec le + fort taux de réservations par mois

```
(df.with_columns(((30 - pl.col("availability_30").cast(pl.Float64,strict=False))/30).alias("taux"))
   .filter(pl.col("neighbourhood_cleansed").is_not_null() & pl.col("taux").is_not_null())
   .group_by("neighbourhood_cleansed")
   .agg((pl.col("taux").mean()*100).round(2).alias("taux_moyen_mensuel_%"))
   .sort("taux_moyen_mensuel_%", descending=True))
```

shape: (20, 2)

neighbourhood_cleansed	taux_moyen_mensuel_%
str	f64
"Ménilmontant"	75.42
"Entrepôt"	74.81
"Popincourt"	74.78
"Buttes-Chaumont"	74.13
"Panthéon"	73.14
…	…
"Palais-Bourbon"	68.87
"Bourse"	68.69
"Luxembourg"	66.45
"Élysée"	62.45
"Passy"	62.3

ROADMAP DE LA MISSION

1. Contexte & objectifs de mission
2. Exploration des données
3. Requête simple en CLI
4. Requête complexe PyMongo/Polar
5. **Visualisation dashboard Power BI**
6. Import données Lyon
7. Réplication avec ReplicatSet
8. Distribuer les données avec Sharding

Connexion BI : principe

Power BI

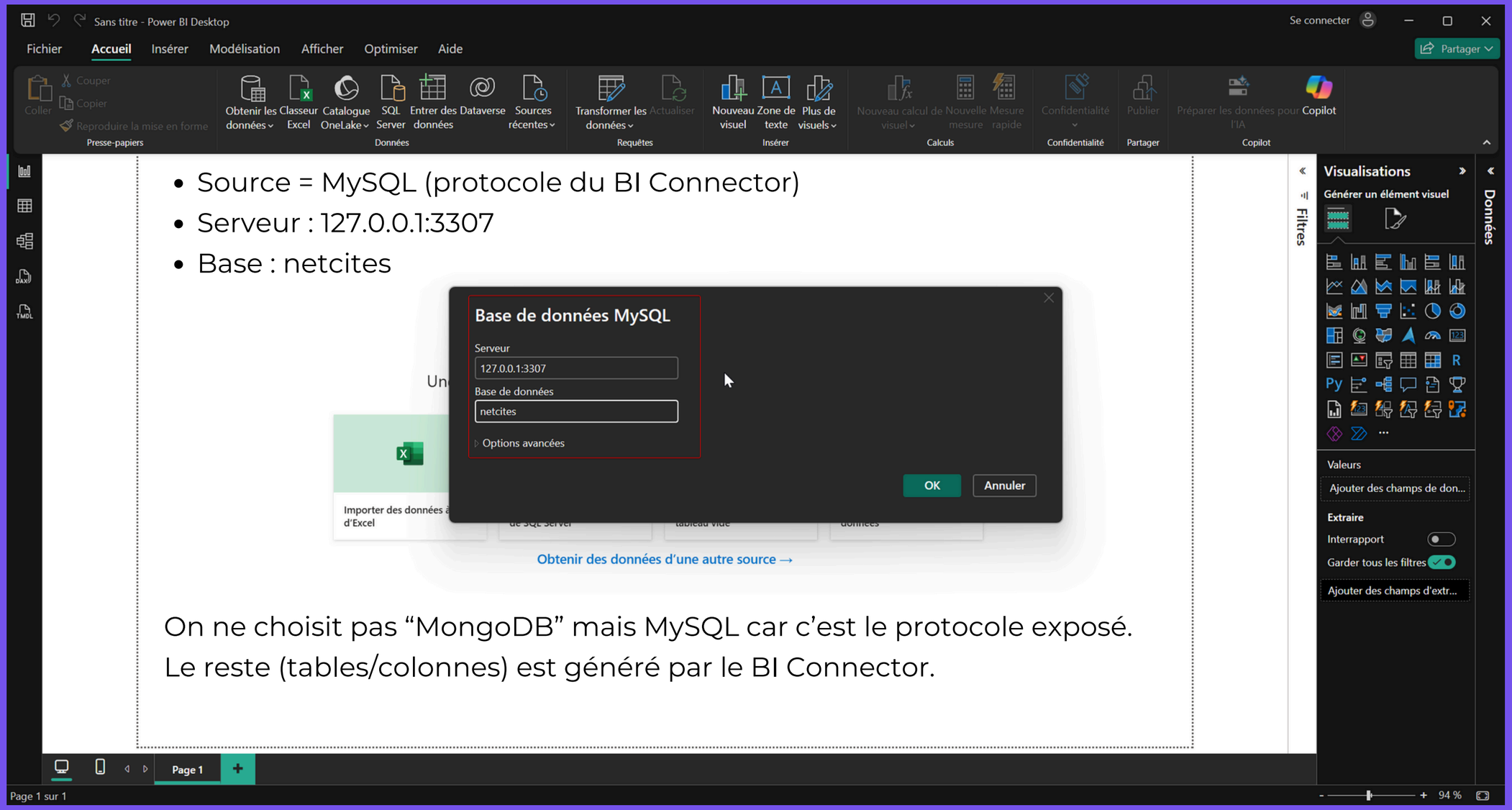
```
PS C:\Program Files\MongoDB\Connector for BI\2.14\bin> .\mongosqld.exe --mongo-uri "mongodb://localhost:27017" --addr 127.0.0.1:3307
>>
2025-10-14T14:19:28.122+0200 I CONTROL    [initandlisten] mongosqld starting: version=v2.14.25 pid=19320 host=ADMIN
2025-10-14T14:19:28.134+0200 I CONTROL    [initandlisten] git version: 8e65b98675bb632dae9ab88ef6ea3f0268237187
2025-10-14T14:19:28.134+0200 I CONTROL    [initandlisten] OpenSSL version OpenSSL 3.5.0 8 Apr 2025 (built with OpenSSL 3.5.0 8 Apr 2025)
2025-10-14T14:19:28.134+0200 I CONTROL    [initandlisten] options: {}
2025-10-14T14:19:28.134+0200 I CONTROL    [initandlisten] ** WARNING: Access control is not enabled for mongosqld.
2025-10-14T14:19:28.134+0200 I CONTROL    [initandlisten]
2025-10-14T14:19:28.136+0200 I NETWORK    [initandlisten] waiting for connections at 127.0.0.1:3307
2025-10-14T14:19:28.149+0200 I SCHEMA     [sampler] sampling MongoDB for schema...
2025-10-14T14:19:28.873+0200 I SCHEMA     [sampler] mapped schema for 3 namespaces: "ma_bd_mongoDB_migration" (2): ["medical", "test"]; "netcites" (1): ["logements"]
2025-10-14T14:27:55.091+0200 I NETWORK    [conn1] connection accepted from 127.0.0.1:49319 #1 (1 connection now open)
2025-10-14T14:27:55.101+0200 E NETWORK    [conn1] handshake error: ERROR 1043 (08S01): recv handshake response error: EOF
2025-10-14T14:27:55.102+0200 I NETWORK    [conn1] end connection 127.0.0.1:49319 (0 connections now open)
2025-10-14T14:28:02.910+0200 I NETWORK    [conn2] connection accepted from 127.0.0.1:49324 #2 (1 connection now open)
```

- Démarrage du MongoDB Connector for BI (mongosqld)
- Mapping automatique des schémas → netcites.logements exposé en SQL
- Écoute locale sur 127.0.0.1:3307

Le BI Connector traduit Mongo en SQL (MySQL wire protocol) pour les outils BI.

Power BI : paramétrage de la source

Power BI



- Source = MySQL (protocole du BI Connector)
- Serveur : 127.0.0.1:3307
- Base : netcites

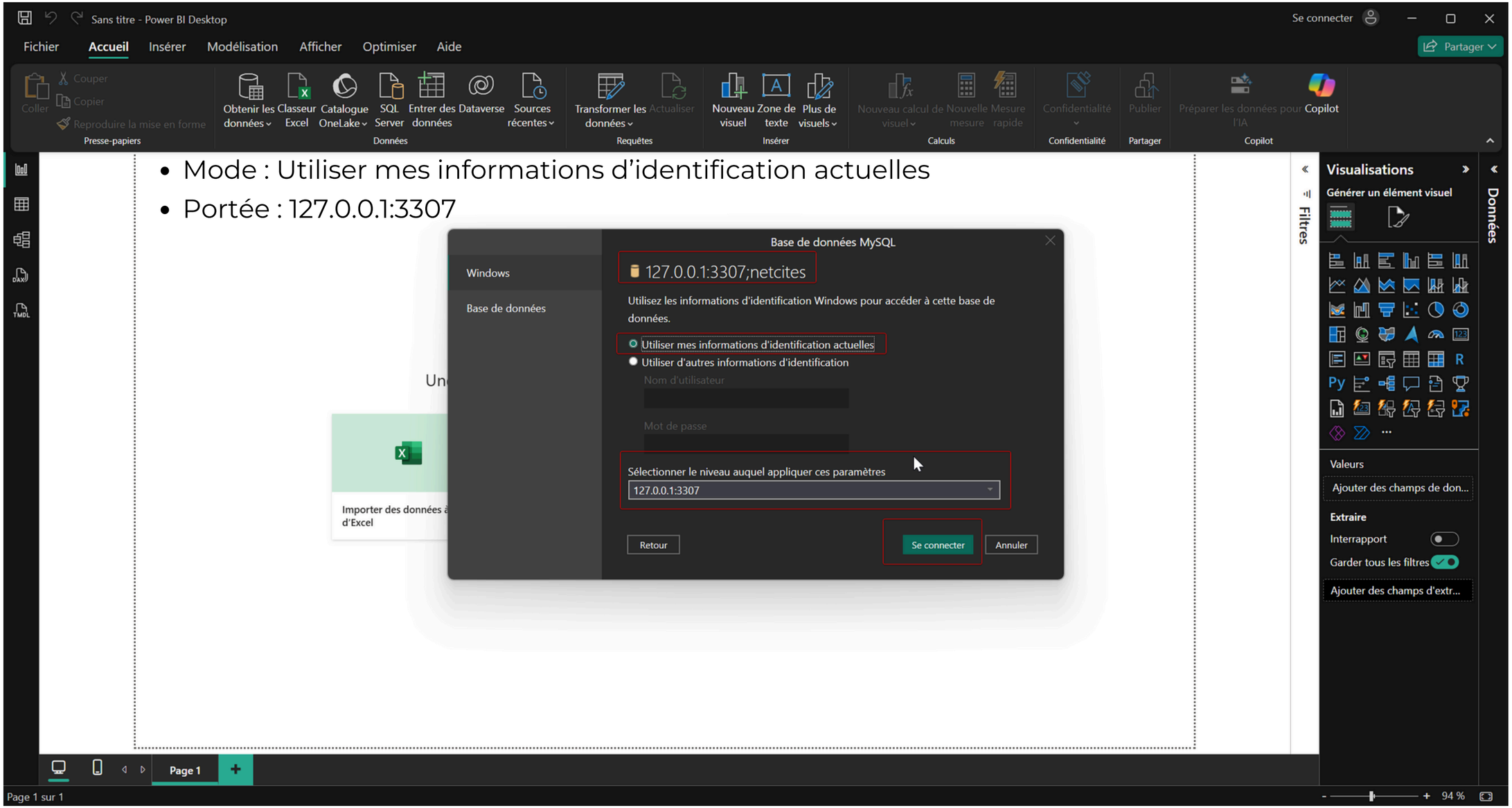
On ne choisit pas “MongoDB” mais MySQL car c’est le protocole exposé.
Le reste (tables/colonnes) est généré par le BI Connector.

OPENCLASSROOMS

mongoDB

Authentification Power BI

Power BI



- Mode : Utiliser mes informations d'identification actuelles
- Portée : 127.0.0.1:3307

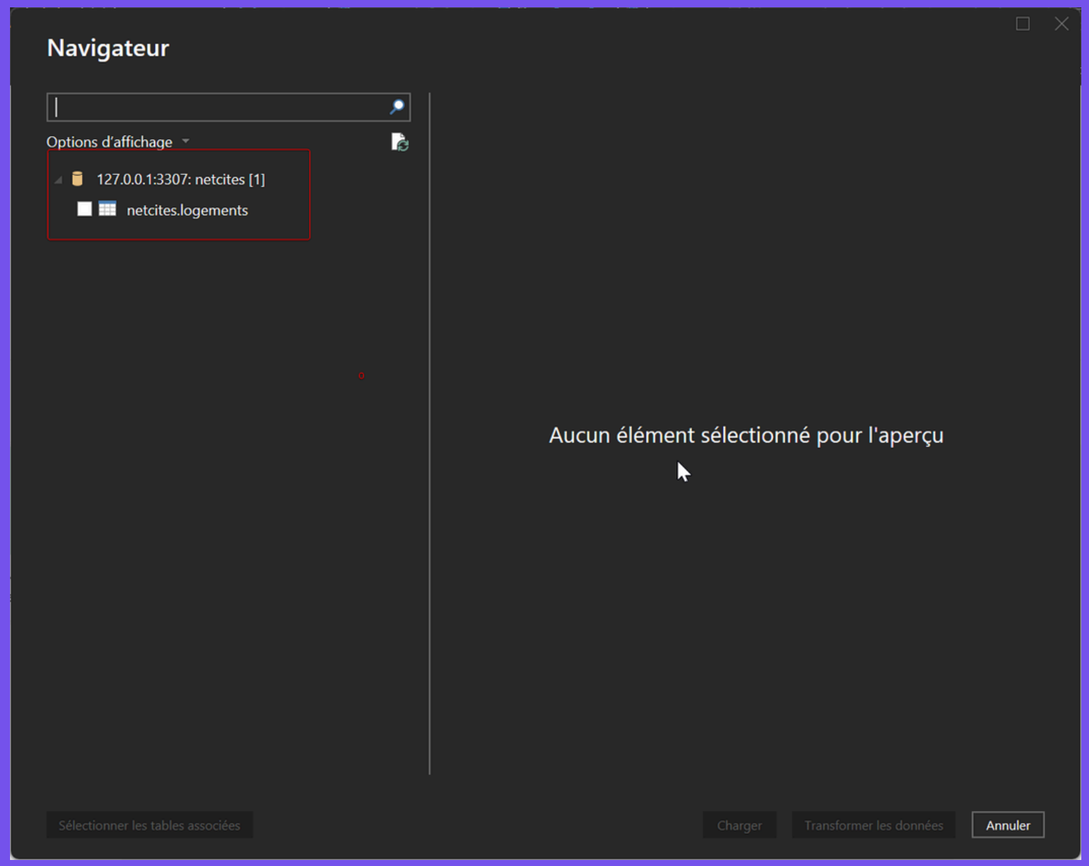
OPENCLASSROOMS

mongoDB

Détection de la table

Power BI

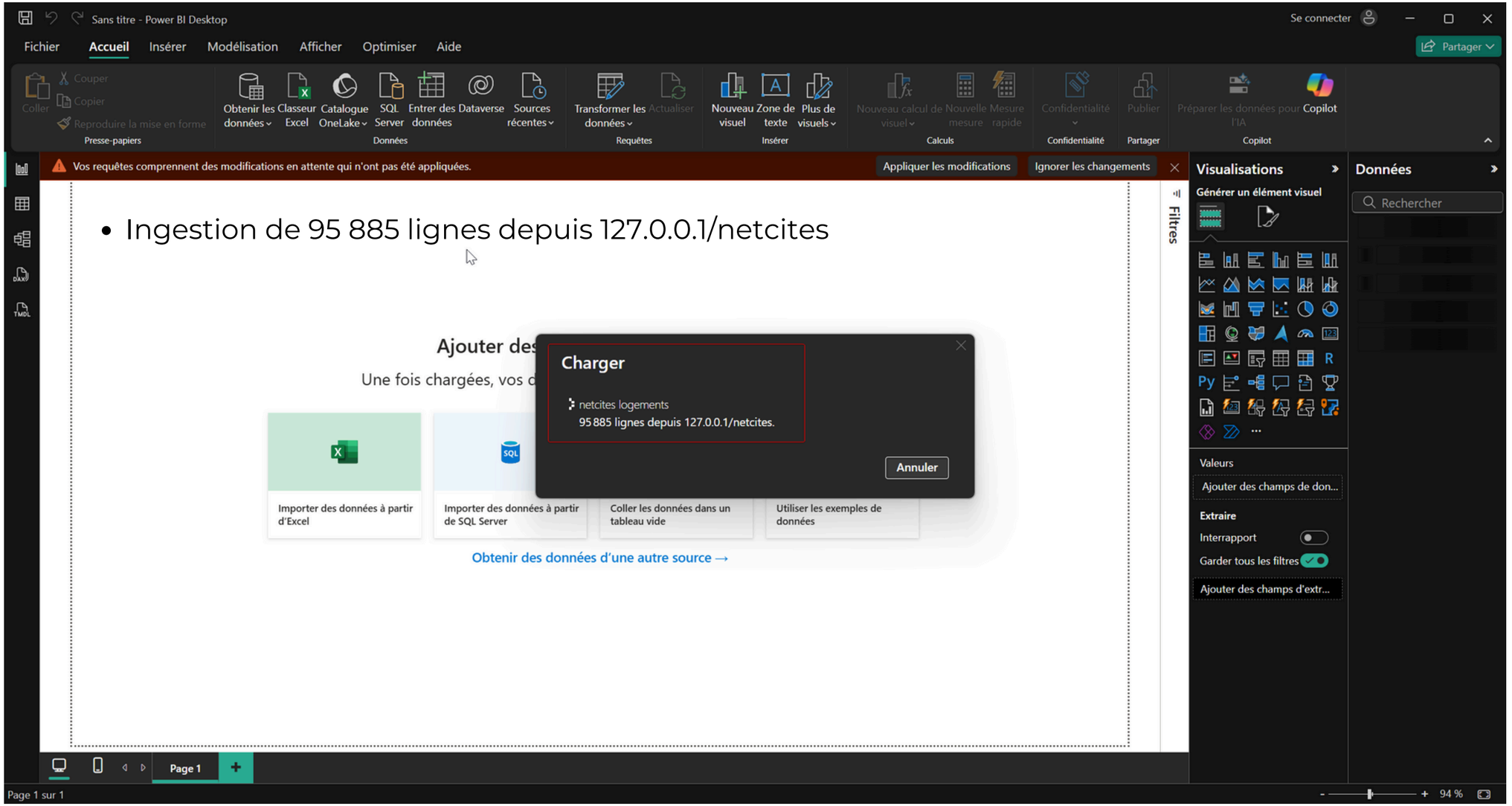
- Découverte du dataset : netcites.logements
- Sélection pour chargement ou transformation (Power Query)



Le BI Connector met à plat les documents (colonnes), prêt pour un modèle tabulaire.
Possibilité de filtrer/typer dans Power Query avant chargement.

Chargement des données

Power BI



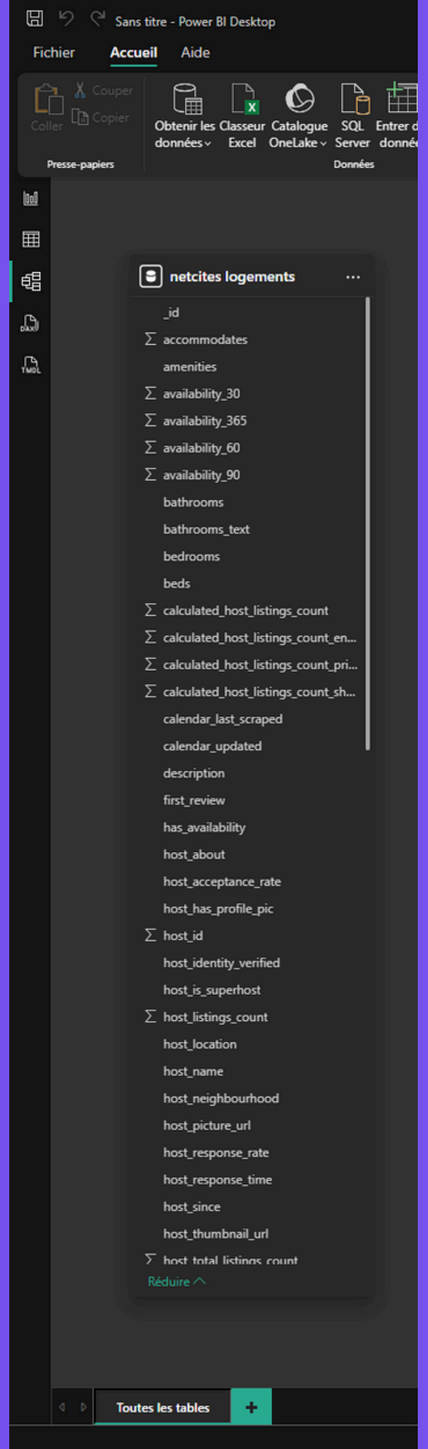
- Ingestion de 95 885 lignes depuis 127.0.0.1/netcites

OPENCLASSROOMS

mongoDB

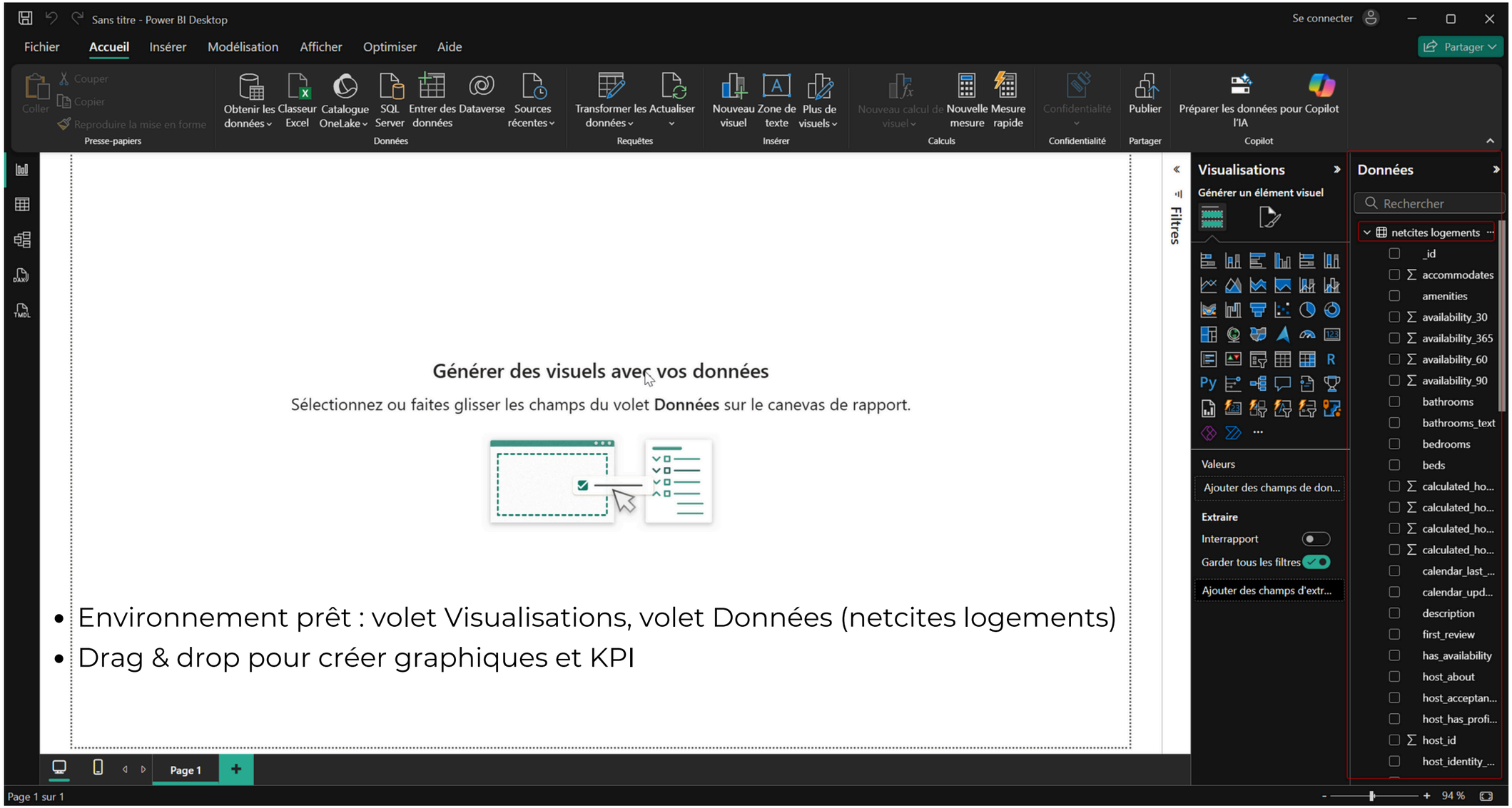
Champs disponibles

- Ensemble des colonnes exposées
- Types numériques/texte prêts à utiliser dans les visuels



Interface prête pour la dataviz

Power BI



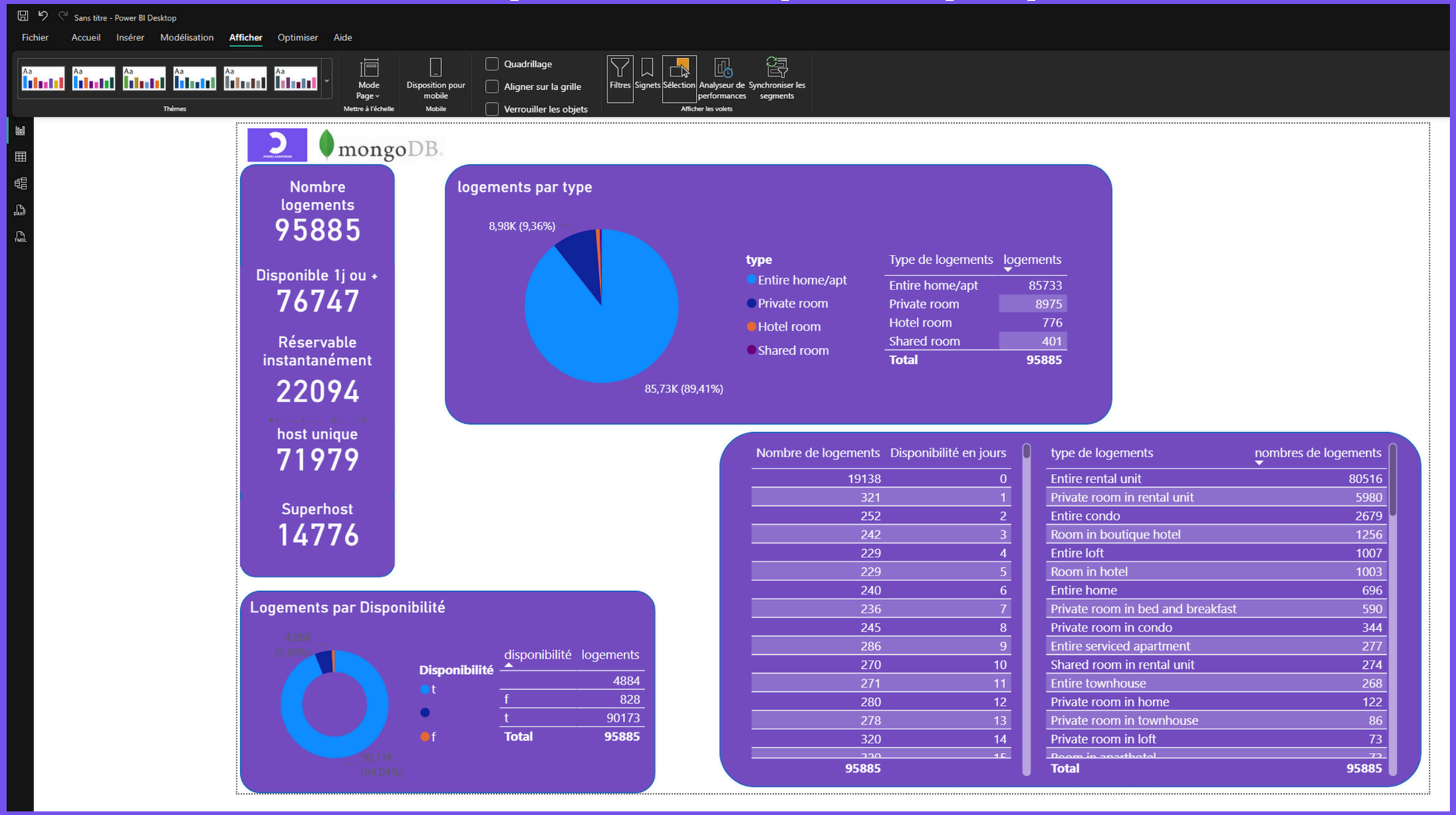
- Environnement prêt : volet Visualisations, volet Données (netcites logements)
- Drag & drop pour créer graphiques et KPI

OPENCLASSROOMS

mongoDB

Tableau de bord d'exploration (exemple)

Power BI



OPENCLASSROOMS

mongoDB

ROADMAP DE LA MISSION

1. Contexte & objectifs de mission
2. Exploration des données
3. Requête simple en CLI
4. Requête complexe PyMongo/Polar
5. Visualisation dashboard Power BI
6. **Import données Lyon**
7. Réplication avec ReplicatSet
8. Distribuer les données avec Sharding

Import données de Lyon en staging

```
netcites> db.logements_staging.countDocuments()
9973
netcites> db.logements_staging.findOne()
{
  _id: ObjectId('68ee7446e89437c672001dfd'),
  id: 77104,
  listing_url: 'https://www.airbnb.com/rooms/77104',
  scrape_id: Long('20240618214414'),
  last_scraped: '2024-06-19',
  source: 'city scrape',
  name: 'Nice Flat in the center of Lyon',
  description: '',
  neighborhood_overview: '',
  picture_url: 'https://a0.muscache.com/pictures/518769/186679f7_original.jpg',
  host_id: 411285,
  host_url: 'https://www.airbnb.com/users/show/411285',
  host_name: 'Clementine',
  host_since: '2011-02-26',
  host_location: 'Lyon, France',
  host_about: "Je travaille dans le cinéma et je voyage beaucoup. J'aime tout particuliè
    "Je parle anglais et français et j'ai des notions d'espagnol et d'italien.\n",
  host_response_time: 'within a few hours',
  host_response_rate: '100%',
  host_acceptance_rate: '67%',
```

- Import du CSV Lyon dans netcites.logements_staging.
- 9 973 documents importés, 0 échec.
- Même schéma que Paris pour faciliter la fusion.

Commande utilisée :

mongoimport --db netcites --collection logements_staging --type csv --headerline --file listings_Lyon+(1).csv.

Ajout du champ city = "Lyon"

```
netcites> db.logements_staging.updateMany({}, { $set: { city: "Lyon" } })
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 9973,
  modifiedCount: 9973,
  upsertedCount: 0
}
```

Contrôle du contenu (staging Lyon)

```
netcites> db.logements_staging.aggregate([
...   { $group: { _id: "$city", n: { $sum: 1 } } }
... ])
[ { _id: 'Lyon', n: 9973 } ]
netcites>
```

Ajout du champ city = "Paris"

```
netcites> db.logements.updateMany({}, { $set: { city: "Paris" } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 95885,
  modifiedCount: 95885,
  upsertedCount: 0
}
netcites> db.logements.aggregate([
...   { $group: { _id: "$city", n: { $sum: 1 } } }
... ])
[ { _id: 'Paris', n: 95885 } ]
netcites>
```

Fusion Lyon dans la collection finale

```
netcites> db.logements.createIndex({ id: 1 }, { unique: true })
id_1
netcites> db.logements_staging.aggregate([
...   { $addFields: { city: "Lyon" } },
...   { $merge: { into: "logements", on: "id", whenMatched: "merge", whenNotMatched: "insert" } }
... ])

netcites> db.logements.countDocuments()
... db.logements.aggregate([{ $group: { _id: "$city", n: { $sum: 1 } } }, { $sort: { n: -1 } }])
[ { _id: 'Paris', n: 95885 }, { _id: 'Lyon', n: 9973 } ]
```

- Index métier unique sur la finale : { id: 1 }.
- Fusion $merge de logements_staging → logements (on: "id", matched: merge, notMatched: insert).

Vérification du marquage dans la collection finale

```
netcites> db.logements.countDocuments()
105858
netcites> db.logements.aggregate([
...   { $group: { _id: "$city", n: { $sum: 1 } } },
...   { $sort: { n: -1 } }
... ])
[ { _id: 'Paris', n: 95885 }, { _id: 'Lyon', n: 9973 } ]
netcites>
```

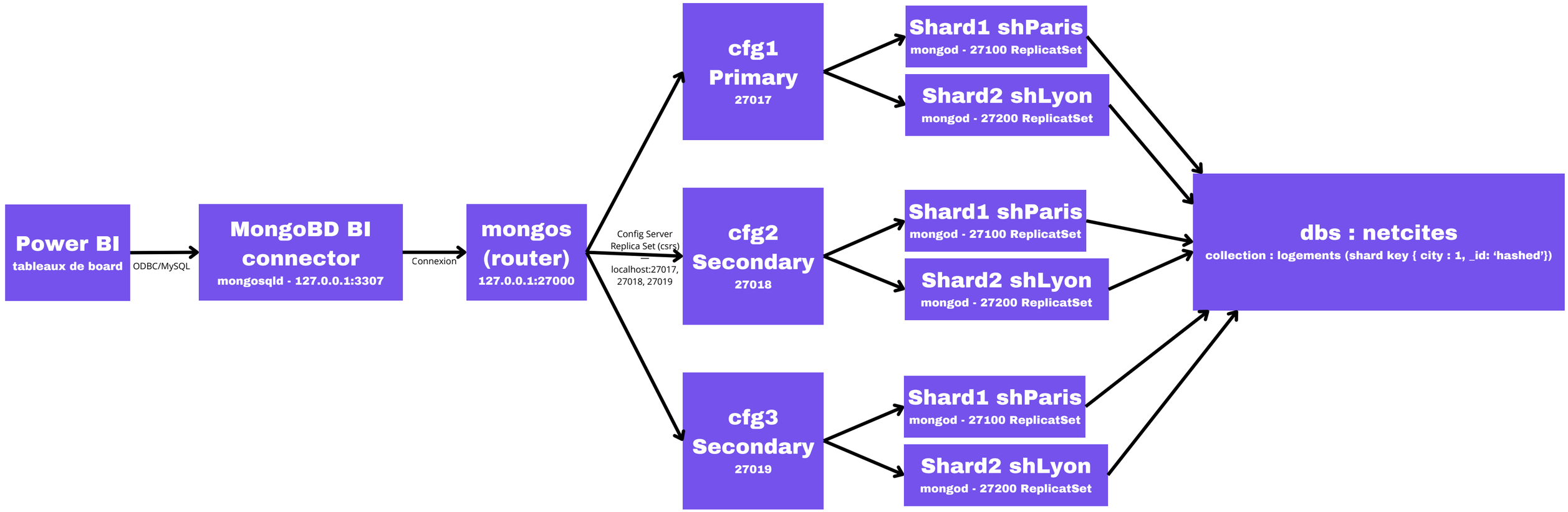
S’assurer que tous les docs ont bien le city (Paris 95 885 et Lyon 9973 = 105 858 au total)

Notre collection est prête pour la réplication avec les données de Paris et Lyon totalement différiable avec la colonne city et des id uniques.

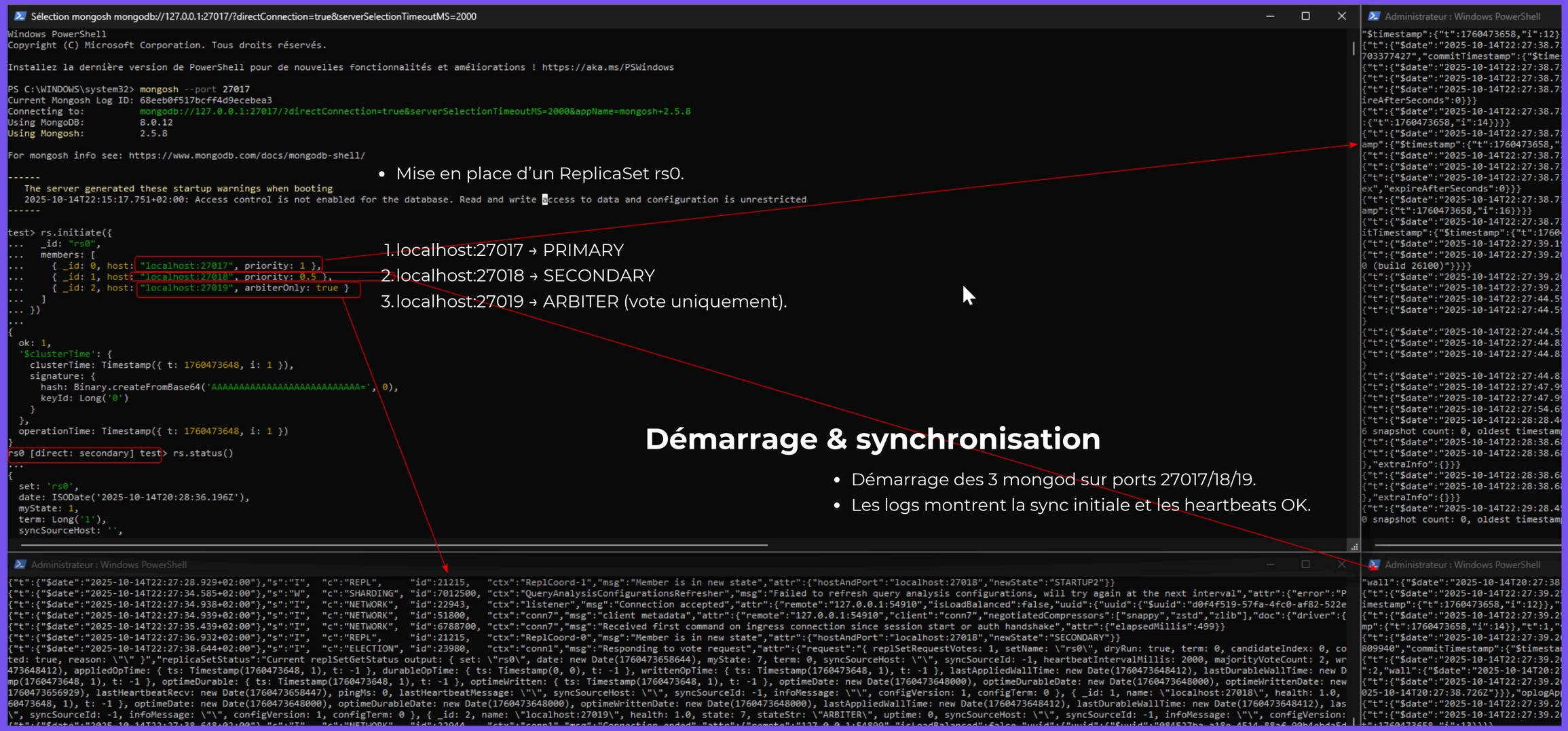
ROADMAP DE LA MISSION

1. Contexte & objectifs de mission
2. Exploration des données
3. Requête simple en CLI
4. Requête complexe PyMongo/Polar
5. Visualisation dashboard Power BI
6. Import données Lyon
7. **Réplication avec ReplicatSet**
8. Distribuer les données avec Sharding

Schéma de la BDD incluant la connexion Power BI

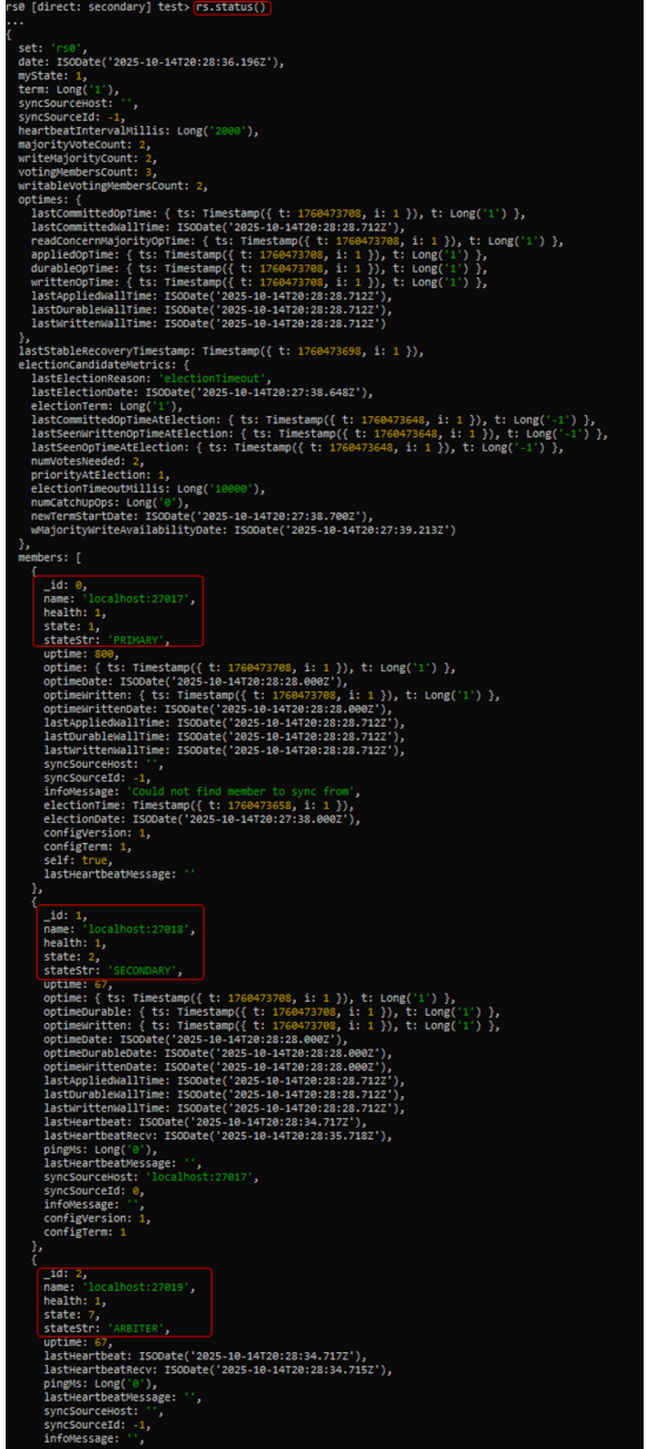


Topologie de réplication/ReplicatSet

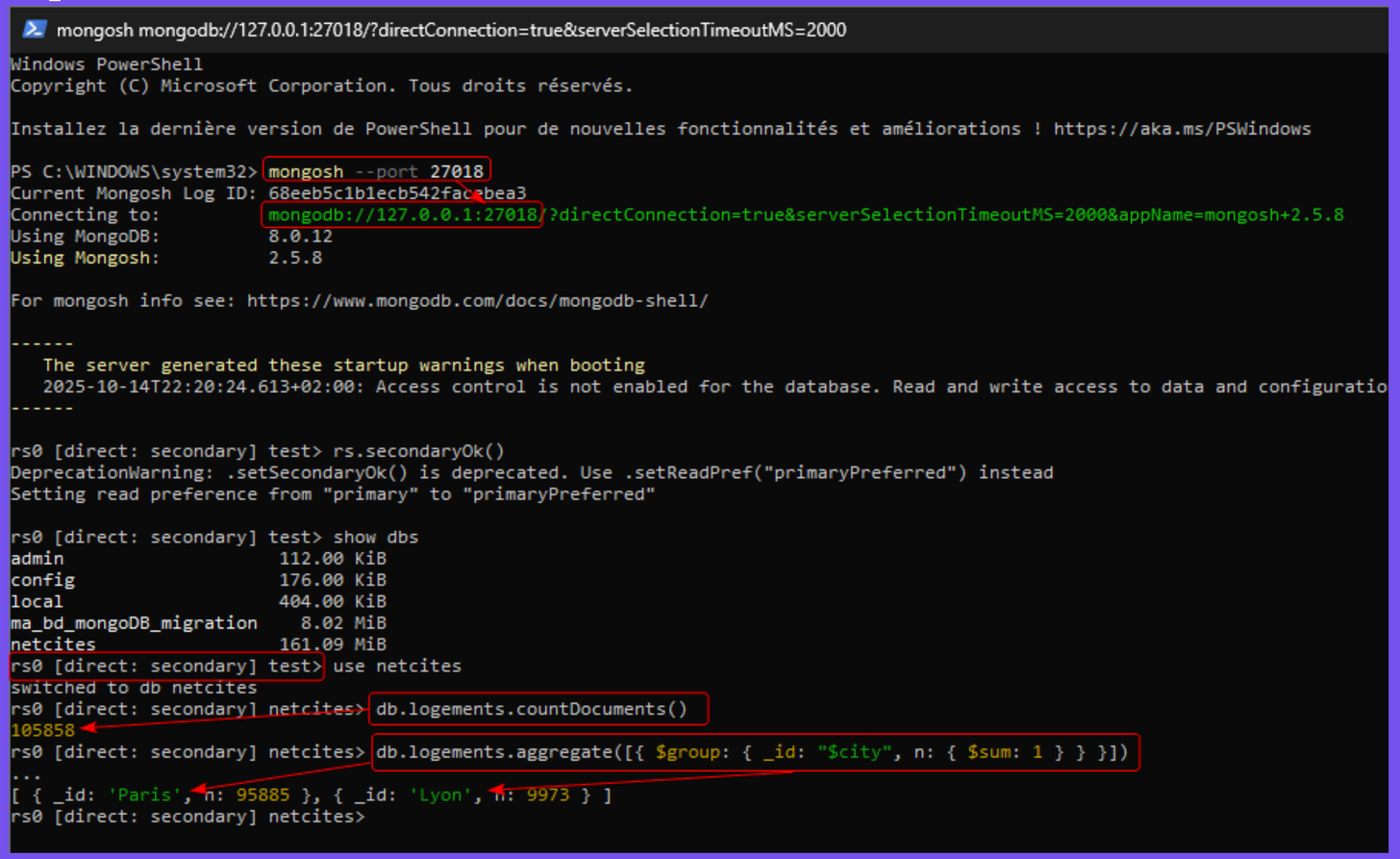


Vérification ReplicaSet

- PRIMARY : localhost:27017 (stateStr: PRIMARY).
- SECONDARY : localhost:27018 (stateStr: SECONDARY).
- ARBITER : localhost:27019 (stateStr: ARBITER).



Lectures depuis le SECONDARY



Connexion à 27018

Contrôles : countDocuments() = 105 858. Par ville : Paris 95 885 et Lyon 9 973.

Les données sont répliquées et cohérentes.

ROADMAP DE LA MISSION

1. Contexte & objectifs de mission
2. Exploration des données
3. Requête simple en CLI
4. Requête complexe PyMongo/Polar
5. Visualisation dashboard Power BI
6. Import données Lyon
7. Réplication avec ReplicatSet
8. **Distribuer les données avec Sharding**

Démarrage des shards (Paris & Lyon)

```
PS C:\WINDOWS\system32> Start-Process "C:\Program Files\MongoDB\Server\8.0\bin\mongod.exe" -ArgumentList @("--shardsvr","--replSet","shParis","--port","27100","--dbpath","C:\Users\antoi\mongo\shards\paris\27100","--bind_ip","127.0.0.1")
PS C:\WINDOWS\system32> Start-Process "C:\Program Files\MongoDB\Server\8.0\bin\mongod.exe" -ArgumentList @("--shardsvr","--replSet","shLyon","--port","27200","--dbpath","C:\Users\antoi\mongo\shards\lyon\27200","--bind_ip","127.0.0.1")
```

Deux instances mongod lancées en mode --shardsvr :

- shParis → localhost:27100
- shLyon → localhost:27200.

Initialisation des ReplicaSets des shards

```
PS C:\WINDOWS\system32> mongosh --port 27100 --eval "rs.initiate({_id:'shParis', members:[{_id:0, host:'localhost:27100'}]})"
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1760477487, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1760477487, i: 1 })
}
PS C:\WINDOWS\system32> mongosh --port 27200 --eval "rs.initiate({_id:'shLyon', members:[{_id:0, host:'localhost:27200'}]})"
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1760477488, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1760477488, i: 1 })
}
```

rs.initiate() exécuté sur chaque shard.

Retour ok:1.

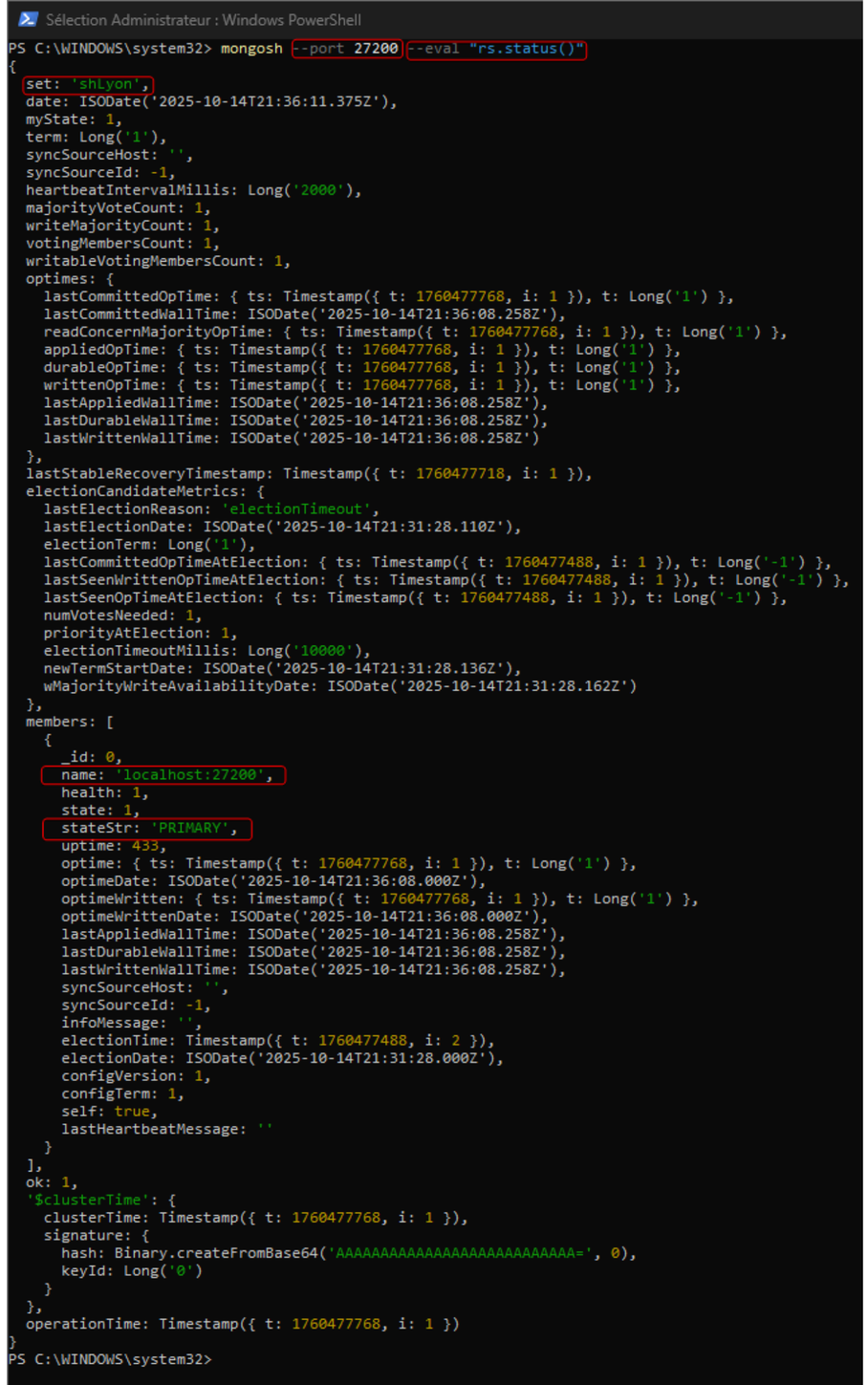
Vérification shard Paris

rs.status() → membre localhost:27100 en PRIMARY

```
PS C:\WINDOWS\system32> mongosh --port 27100 --eval "rs.status()"
{
  set: 'shParis',
  date: ISODate('2025-10-14T21:36:11.070Z'),
  myState: 1,
  term: Long('1'),
  syncSourceHost: '',
  syncSourceId: -1,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 1,
  writeMajorityCount: 1,
  votingMembersCount: 1,
  writableVotingMembersCount: 1,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1760477767, i: 1 }), t: Long('1') },
    lastCommittedWallTime: ISODate('2025-10-14T21:36:07.849Z'),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1760477767, i: 1 }), t: Long('1') },
    appliedOpTime: { ts: Timestamp({ t: 1760477767, i: 1 }), t: Long('1') },
    durableOpTime: { ts: Timestamp({ t: 1760477767, i: 1 }), t: Long('1') },
    writtenOpTime: { ts: Timestamp({ t: 1760477767, i: 1 }), t: Long('1') },
    lastAppliedWallTime: ISODate('2025-10-14T21:36:07.849Z'),
    lastDurableWallTime: ISODate('2025-10-14T21:36:07.849Z'),
    lastWrittenWallTime: ISODate('2025-10-14T21:36:07.849Z')
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1760477717, i: 1 }),
  electionCandidateMetrics: {
    lastElectionReason: 'electionTimeout',
    lastElectionDate: ISODate('2025-10-14T21:31:27.698Z'),
    electionTerm: Long('1'),
    lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1760477487, i: 1 }), t: Long('-1') },
    lastSeenWrittenOpTimeAtElection: { ts: Timestamp({ t: 1760477487, i: 1 }), t: Long('-1') },
    lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1760477487, i: 1 }), t: Long('-1') },
    numVotesNeeded: 1,
    priorityAtElection: 1,
    electionTimeoutMillis: Long('10000'),
    newTermStartDate: ISODate('2025-10-14T21:31:27.717Z'),
    wMajorityWriteAvailabilityDate: ISODate('2025-10-14T21:31:27.741Z')
  },
  members: [
    {
      id: 0,
      name: 'localhost:27100',
      health: 1,
      state: 1,
      stateStr: 'PRIMARY',
      uptime: 443,
      optime: { ts: Timestamp({ t: 1760477767, i: 1 }), t: Long('1') },
      optimeDate: ISODate('2025-10-14T21:36:07.000Z'),
      optimeWritten: { ts: Timestamp({ t: 1760477767, i: 1 }), t: Long('1') },
      optimeWrittenDate: ISODate('2025-10-14T21:36:07.000Z'),
      lastAppliedWallTime: ISODate('2025-10-14T21:36:07.849Z'),
      lastDurableWallTime: ISODate('2025-10-14T21:36:07.849Z'),
      lastWrittenWallTime: ISODate('2025-10-14T21:36:07.849Z'),
      syncSourceHost: '',
      syncSourceId: -1,
      infoMessage: '',
      electionTime: Timestamp({ t: 1760477487, i: 2 }),
      electionDate: ISODate('2025-10-14T21:31:27.000Z'),
      configVersion: 1,
      configTerm: 1,
      self: true,
      lastHeartbeatMessage: ''
    }
  ],
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1760477767, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1760477767, i: 1 })
}
```

Vérification shard Lyon

rs.status() → membre localhost:27200 en PRIMARY



Démarrage du routeur mongos

```
PS C:\WINDOWS\system32> & "C:\Program Files\MongoDB\Server\8.0\bin\mongos.exe" `
>>   --configdb "csrs/localhost:27050,localhost:27051,localhost:27052" `
>>   --port 27000 `
>>   --bind_ip 127.0.0.1
{"t":{"$date":"2025-10-14T23:58:30.064+02:00"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 speci
{"t":{"$date":"2025-10-14T23:58:30.066+02:00"},"s":"I",  "c":"NETWORK",  "id":4648601, "svc":"-", "ctx":"thread1","msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen
{"t":{"$date":"2025-10-14T23:58:30.067+02:00"},"s":"I",  "c":"HEALTH",   "id":5936503, "svc":"-", "ctx":"thread1","msg":"Fault manager changed state ","attr":{"state":"Star
{"t":{"$date":"2025-10-14T23:58:30.068+02:00"},"s":"I",  "c":"NETWORK",  "id":4915701, "svc":"-", "ctx":"thread1","msg":"Initialized wire specification","attr":{"spec":{"in
{"t":{"$date":"2025-10-14T23:58:30.068+02:00"},"s":"W",  "c":"CONTROL",  "id":22120,   "svc":"-", "ctx":"thread1","msg":"Access control is not enabled for the database. Rea
{"t":{"$date":"2025-10-14T23:58:30.069+02:00"},"s":"I",  "c":"CONTROL",  "id":23403,   "svc":"R", "ctx":"mongosMain","msg":"Build Info","attr":{"buildInfo":{"version":"8.0.
{"t":{"$date":"2025-10-14T23:58:30.069+02:00"},"s":"I",  "c":"CONTROL",  "id":51765,   "svc":"R", "ctx":"mongosMain","msg":"Operating System","attr":{"os":{"name":"Microsof
{"t":{"$date":"2025-10-14T23:58:30.069+02:00"},"s":"I",  "c":"CONTROL",  "id":21951,   "svc":"R", "ctx":"mongosMain","msg":"Options set by command line","attr":{"options":{
{"t":{"$date":"2025-10-14T23:58:30.070+02:00"},"s":"I",  "c":"NETWORK",  "id":4603701, "svc":"R", "ctx":"mongosMain","msg":"Starting Replica Set Monitor","attr":{"protocol"
{"t":{"$date":"2025-10-14T23:58:30.071+02:00"},"s":"I",  "c":"-",        "id":4333223, "svc":"R", "ctx":"mongosMain","msg":"RSM now monitoring replica set","attr":{"replica
{"t":{"$date":"2025-10-14T23:58:30.071+02:00"},"s":"I",  "c":"-",        "id":4333226, "svc":"R", "ctx":"mongosMain","msg":"RSM host was added to the topology","attr":{"rep
{"t":{"$date":"2025-10-14T23:58:30.071+02:00"},"s":"I",  "c":"-",        "id":4333226, "svc":"R", "ctx":"mongosMain","msg":"RSM host was added to the topology","attr":{"rep
```

mongos --configdb csrs/localhost:27050,27051,27052 --port 27000 --bind_ip 127.0.0.1

Contrôles depuis mongos

```
PS C:\WINDOWS\system32> & "C:\Program Files\MongoDB\Tools\100\bin\mongodump.exe" --host "localhost" --port 27017 --db netcites --collection logements --archive="C:\Users\antoi\mongo\netcites_logements_vers_sharded.gz" --gzip
2025-10-15T00:28:08.332+0200    writing netcites.logements to archive 'C:\Users\antoi\mongo\netcites_logements_vers_sharded.gz'
2025-10-15T00:28:11.319+0200    [##########..........]  netcites.logements  51185/105858  (48.4%)
2025-10-15T00:28:14.319+0200    [####################.]  netcites.logements  102520/105858  (96.8%)
2025-10-15T00:28:14.566+0200    [########################]  netcites.logements  105858/105858  (100.0%)
2025-10-15T00:28:14.674+0200    done dumping netcites.logements (105858 documents)
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32> & "C:\Program Files\MongoDB\Tools\100\bin\mongorestore.exe" --host "localhost" --port 27000 --archive="C:\Users\antoi\mongo\netcites_logements_vers_sharded.gz" --gzip
2025-10-15T00:28:39.717+0200    preparing collections to restore from
2025-10-15T00:28:39.731+0200    reading metadata for netcites.logements from archive 'C:\Users\antoi\mongo\netcites_logements_vers_sharded.gz'
2025-10-15T00:28:39.737+0200    restoring to existing collection netcites.logements without dropping
2025-10-15T00:28:39.737+0200    restoring netcites.logements from archive 'C:\Users\antoi\mongo\netcites_logements_vers_sharded.gz'
2025-10-15T00:28:42.711+0200    netcites.logements  125MB
2025-10-15T00:28:45.711+0200    netcites.logements  252MB
2025-10-15T00:28:48.417+0200    netcites.logements  361MB
2025-10-15T00:28:48.418+0200    finished restoring netcites.logements (105858 documents, 0 failures)
2025-10-15T00:28:48.418+0200    restoring indexes for collection netcites.logements from metadata
2025-10-15T00:28:48.418+0200    index: &idx.IndexDocument{Options:primitive.M{"name":"property_type_1", "v":2}, Key:primitive.D{primitive.E{Key:"property_type", Value:1}}, PartialFilterExpression:primitive.D(nil)}
2025-10-15T00:28:48.418+0200    index: &idx.IndexDocument{Options:primitive.M{"name":"id_1", "unique":true, "v":2}, Key:primitive.D{primitive.E{Key:"id", Value:1}}, PartialFilterExpression:primitive.D(nil)}
2025-10-15T00:28:48.424+0200    Failed: netcites.logements: error creating indexes for netcites.logements: createIndex error: (CannotCreateIndex) Index build failed: 94ea56c5-6de2-4309-8e0c-c32e017bc715: Collection netcites.logements ( 7f
2025-10-15T00:28:48.424+0200    105858 document(s) restored successfully. 0 document(s) failed to restore.
PS C:\WINDOWS\system32> mongosh --port 27000
Current Mongosh Log ID: 68eecec9ef696f8e6ecebea3
Connecting to:          mongodb://127.0.0.1:27000/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.8
Using MongoDB:          8.0.12
Using Mongosh:          2.5.8

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-10-14T23:50:14.695+02:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

[direct: mongos] test> use netcites
switched to db netcites
[direct: mongos] netcites> db.logements.countDocuments()
105858
[direct: mongos] netcites> db.logements.getShardDistribution()
Shard shParis at shParis/localhost:27100
{
  data: '327.42MiB',
  docs: 95885,
  chunks: 4,
  'estimated data per chunk': '81.85MiB',
  'estimated docs per chunk': 23971
}
---
Shard shLyon at shLyon/localhost:27200
{
  data: '33.23MiB',
  docs: 9973,
  chunks: 1,
  'estimated data per chunk': '33.23MiB',
  'estimated docs per chunk': 9973
}
---
Totals
{
  data: '360.66MiB',
  docs: 105858,
  chunks: 5,
  'Shard shParis': [
    '90.78 % data',
    '90.57 % docs in cluster',
    '3KiB avg obj size on shard'
  ],
  'Shard shLyon': [
    '9.21 % data',
    '9.42 % docs in cluster',
    '3KiB avg obj size on shard'
  ]
}
[direct: mongos] netcites> db.logements.aggregate([{ $group: { _id: "$city", n: { $sum: 1 } } }, { $sort: { n: -1 } }])
[ { _id: 'Paris', n: 95885 }, { _id: 'Lyon', n: 9973 } ]
```

Vérifications côté mongos (stats/distribution après import & sharding).

MERCI