

# Atelier base de données spatiales

## PostgreSQL/PostGIS

### 1 Créer une base de données spatiale

#### 1.1 Objectif

L'objectif de cet atelier est de vous faire découvrir l'interface de pgAdmin et les différents objets de base de données

Après cet exercice, vous serez capable de :

- Utiliser l'interface pgAdmin III
- Créer une nouvelle base de données PostgreSQL
- Créer l'extension PostGIS pour utiliser des données spatiales
- Créer une table et la remplir à l'aide de l'éditeur de requête
- Faire une requête simple sur la table


#### 1.2 Créer une base de données spatiale à partir de pgAdmin III

Un serveur PostgreSQL peut contenir plusieurs bases de données. Chaque base de données est conçue pour stocker les données relevant d'une même application.

Pour réaliser les exercices, nous allons créer une nouvelle base de données spatiale :


1. Ouvrir le dossier "Databases" et lancer pgAdmin III
2. Développer l'arborescence du serveur local (localhost:5432)
3. A l'aide d'un clic droit sur le dossier des bases de données, choisir la commande "Ajouter une base de données ..." ("New Database...")
4. Nommer la base de données "exercice", choisir le propriétaire "postgres" et valider la création de la base de données
5. Développer l'arborescence de la base de données "exercice", choisir le schéma "Public", et vérifiez qu'aucune table n'est présente sous l'onglet "Tables"

Il faut maintenant ajouter l'extension spatiale PostGIS à la base de données "exercice" :

6. Cliquer sur la base de données "exercice" et lancer l'éditeur de requête : 
7. Dans l'éditeur SQL taper la commande :

```
CREATE EXTENSION postgis;
```

Attention, il s'agit d'une commande, tous les caractères doivent être saisis à l'identique, y compris le ";" final qui indique au compilateur la fin d'une instruction. La casse n'a pas d'importance, cependant les instructions SQL sont souvent écrites en majuscule par souci de lisibilité.


8. Exécuter la commande à l'aide de la commande  ou de la touche F5

Le compte-rendu de l'exécution de la commande apparaît dans le cadre inférieur de l'éditeur. La requête ne retourne pas de données.

9. Fermer l'éditeur de requête et rafraîchir l'arborescence de la base de données "exercice" à l'aide du menu contextuel "Rafraîchir" ("Refresh") ou de la commande



10. Développer l'arborescence de la base de données "exercice", choisir le schéma "public" et vérifier la présence de la table "spatial\_ref\_sys". Cette table a été créée par PostGIS.

11. Vérifier le contenu de cette table, pour cela lancer l'éditeur de requête :  et taper la commande :

```
SELECT * FROM spatial_ref_sys;
```

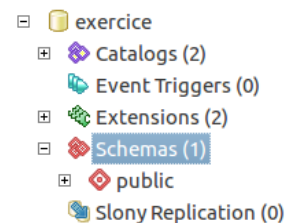
Exécutez la commande et observez le panneau de sortie (output panel)

- Combien de lignes contient cette table ?
- Combien de temps a duré l'exécution de la requête ?
- Que représentent ces données ?
- Que veut dire "srid" ?

### 1.3 Créer une table et la remplir avec des commandes SQL

Tout d'abord nous allons créer un nouveau schéma. Le schéma "public" est utilisé pour stocker les objets utilisés par le système comme nous l'avons vu dans l'exercice précédent. Il est recommandé de créer ses propres schémas, cela permet d'organiser les données de façon logique et ainsi d'y accéder plus facilement.

12. Créer un nouveau schéma : cliquer sur Schema et utiliser la commande "Ajouter un schéma ..." ("New schema ...")
13. Nommer le schéma "contexte", choisir le propriétaire "postgres" et valider la création du schéma



14. Rafraîchir l'arborescence de la base de données "exercice" et développer l'arborescence du nouveau schéma

Que contient le schéma "contexte" ?

Quelle différence avec le contenu du schéma "public" ?

Nous allons maintenant créer une table et la remplir en utilisant des commandes SQL. Il existe d'autres moyens de créer une table que nous verrons plus tard.

15. Cliquer sur la base de données "exercice" puis ouvrez l'éditeur de requête 

16. Dans l'éditeur SQL taper la commande :

```
CREATE TABLE contexte.test_ville (gid int4 primary key, nom_ville  
varchar(50), geom geometry(POINT,2154));
```

Cette commande crée la table "test\_ville" dans le schéma "contexte". Cette table a 3 colonnes :

- gid est l'identifiant, en tant que clé primaire, cette colonne devra toujours être renseignée et devra présenter des valeurs uniques.
- nom\_ville est une colonne de type texte qui supporte jusqu'à 50 caractères
- geom est une colonne de géométrie de type point

Remarquez que les noms de champ ne comportent jamais d'espace ni de caractères spéciaux.

17. Nous allons maintenant ajouter des données dans cette table. Dans l'éditeur SQL taper les commandes suivantes :

```
INSERT INTO contexte.test_ville (gid, nom_ville, geom) VALUES  
(1, 'Valence', ST_GeomFromText('POINT(849106.92 6426973.85)',2154));
```

```
INSERT INTO contexte.test_ville (gid, nom_ville, geom) VALUES  
(2, 'Romans', ST_GeomFromText('POINT(860560.84 6440404.16)',2154));
```

Ces commandes insèrent 2 lignes dans la table test\_ville, remarquez la façon de renseigner la colonne geom : la fonction ST\_GeomFromText(texte, srid) permet de transformer une chaîne de caractère en type géométrie avec le srid précisé, la fonction POINT(X Y) permet de créer un point à partir de ses coordonnées.


Pour vérifier le contenu de la table, nous allons utiliser l'interface de pgAdmin

18. Fermer l'éditeur de requête et rafraîchir la base de données "exercice"

19. Développez le schéma "contexte" et la liste des tables, puis cliquez sur la table "test\_ville"

Que voyez-vous dans le panneau SQL (SQL pane) situé en bas et à droite de l'écran ?

Vérifiez les colonnes en développant l'arborescence sous la table test\_ville

20. Pour voir le contenu de la table test\_ville, cliquez sur la table dans l'arborescence et utiliser la commande  de la barre d'outil.


Combien de lignes contient la table ?

Remarquez comment apparaissent les données de la colonne "geom"

Vérifiez que vous pouvez corriger le nom de la ville à travers cette interface.

## 1.4 Créer une requête simple

Nous allons écrire une requête simple en SQL pour présenter les données de façon plus lisible

21. Cliquez sur la base de données "exercice" et ouvrez l'éditeur de requête 

22. Dans l'éditeur SQL taper la commande :

```
SELECT nom_ville, ST_X(geom) As Longitude, ST_Y(geom) As Latitude  
FROM contexte.test_ville;
```

Exécutez la commande et observez le panneau de sortie (output panel)

Remarquez que le nom de la table est toujours précédé du nom du schéma

Les coordonnées sont présentées au format Lambert 93 (srid 2154)

## 2 Importer des données attributaires et des données spatiales

### 2.1 Objectif

L'objectif de cet atelier est d'importer des données attributaires dans postgresSQL à l'aide de l'interface de pgAdmin

Après cet exercice, vous serez capable de :

- Créer une table à partir de l'interface pgAdmin
- Importer des données à partir d'un fichier texte dans une table existante
- Importer des données à partir d'un fichier shape

### 2.2 Importer des données attributaires

Nous disposons d'informations sur les communes que nous souhaitons intégrer à la base de données. Ces données sont uniquement des données attributaires, elles sont disponibles au format csv (texte séparé par des virgules).

Le fichier commune\_detail.csv est présent sur la clé USB dans le dossier "exercice\_bd".

23. Ouvrez ce fichier et observez son contenu

Comment sont structurées les données dans ce fichier ?

Nous devons tout d'abord créer une table pour recevoir ces données, puis utiliser la commande COPY pour insérer les données dans la table.

24. Dans pgAdmin, développer l'arborescence de la base de données "exercice" et le schéma "contexte"

25. A l'aide d'un clic droit sur le dossier des tables, choisir la commande "Ajouter une table ..." ("New table ...")

26. Nommer la table "commune\_detail" et choisir le propriétaire "postgres"

27. Ouvrir l'onglet "Colonnes" ("Columns") et utiliser la commande "Ajouter" ("Add") pour créer les 3 colonnes décrites dans le tableau suivant :

Nom (name)	Type de données (datatype)	Longueur (length)
no_insee	character varying	5
intercommunalite	character varying	200
population	integer	

28. Ouvrir l'onglet "Contraintes" ("Constraints") et ajouter une nouvelle contrainte de clé primaire (primary key) à l'aide de la commande située en bas de l'écran




Primary Key Add Remove

29. Dans l'écran de saisie de la nouvelle clé primaire saisir le nom de la clé primaire : PK\_commune\_detail puis ouvrir l'onglet "Colonnes" ("Columns") et utiliser la commande "Ajouter" ("Add") pour ajouter la colonne "no\_insee"



30. Valider la nouvelle contrainte, puis valider la création de la table  
31. Vérifier la présence de la table "commune\_detail" dans le schéma "contexte"

Nous allons maintenant insérer les données dans cette table :

32. Cliquer sur la base de données "exercice" et ouvrir l'éditeur de requête 

33. Dans l'éditeur SQL taper la commande :

```
COPY contexte.commune_detail FROM  
'/home/user/SNIEAU/exercice_bd/commune_detail.csv'  
encoding 'latin1' delimiter ';' ;
```

Exécutez la commande et observez le panneau de sortie (output panel)

Combien de lignes ont été insérées dans la table contexte.commune\_detail ?

Fermer l'éditeur de requête, vérifier le contenu de la table contexte.commune\_detail


L'option "encoding 'latin1'" de la commande COPY permet de reconnaître les caractères accentués

## 2.3 Importer des données spatiales

Nous allons importer les données spatiales des communes à partir d'un fichier shape existant. Il existe plusieurs méthodes pour importer des données spatiales.

Nous utiliserons ici l'utilitaire shp2pg en ligne de commande.

ATTENTION ! les utilitaires d'import de données dans PostGIS n'acceptent pas les espaces dans le chemin d'accès aux fichiers.

34. Ouvrir l'éditeur LXTerminal à l'aide de la commande  (3<sup>ème</sup> en bas de l'écran)

35. Taper la commande :

```
shp2pgsql -I -s 2154 /home/user/SNIEAU/exercice_bd/commune.shp  
contexte.commune | psql -U postgres -d exercice
```

Exécutez la commande, vous devez voir les résultats des commandes d'insertion

36. Dans pgAdmin, rafraîchir l'arborescence de la base de données exercice et vérifier le contenu de la table contexte.commune

### 3 Créer des requêtes et des vues


#### 3.1 Objectif

L'objectif de cet atelier est d'aborder le langage SQL en créant des requêtes simples et de comprendre l'utilisation des vues.

Après cet exercice, vous serez capable de :

- Utiliser le constructeur graphique de pgAdmin
- Créer des requêtes simples en utilisant la jointure et l'agrégation
- Utiliser des fonctions spatiales dans les requêtes
- Rendre ces requêtes permanentes en créant des vues

#### 3.2 Requête avec jointure et critère

37. Cliquer sur la base de données "exercice" et ouvrir l'éditeur de requête 

38. Choisir l'onglet "Constructeur graphique de requête" ("Graphical Query Builder")

39. Développer l'arborescence des schémas à gauche de l'écran et double cliquer sur les tables "commune" et "commune\_detail"

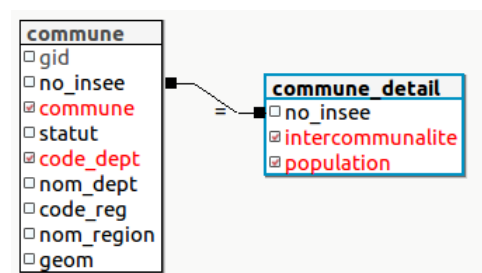
La requête est basée sur 2 tables, il s'agit d'une requête de jointure. Il faudra préciser les champs de jointure pour obtenir un résultat cohérent.

40. Relier avec la souris les champs no\_insee de chacune des 2 tables afin de spécifier les champs de jointure

Vérifier la jointure dans l'onglet "Jointure" ("Joins") en bas de l'écran

	Source Column	Join Type	Destination Column
1	commune.no_insee	=	commune_detail.no_insee

41. Sélectionner les champs à afficher :  
cocher les champs "commune" et "code\_dept"  
de la table "commune"  
et les champs  
"intercommunalite" et "population"  
de la table commune\_detail



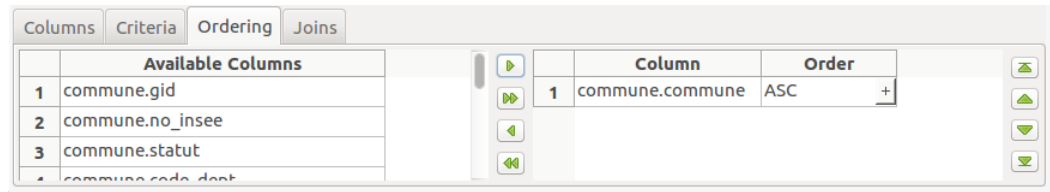
42. Ajouter un filtre : dans l'onglet "Critère" ("Criteria") utiliser la commande 

Choisir le champ commune.code\_dept, et la valeur 26 pour filtrer uniquement les communes du département de la Drôme

	Restricted Value	Operator	Value	Connector
1	commune.code_dept	=	'26'	AND

ATTENTION ! la colonne code\_dept est de type texte, il faut donc écrire la valeur entre simples quotes '26'

43. Trier les données sur le champ "commune" en le sélectionnant dans l'onglet "Tri" ("Order") au bas de l'écran



44. Cliquer sur l'onglet de l'éditeur SQL et vérifier la syntaxe de la commande

45. Exécuter la requête

Combien de communes sont dans le département de la Drôme ?

### 3.3 Requête d'agrégation

Les requêtes d'agrégation permettent de faire des sommes, des comptes, des moyennes, ... sur un ensemble de données

46. Créer une nouvelle requête et taper la commande suivante :

```
SELECT intercommunalite, Sum(population)
FROM contexte.commune c INNER JOIN contexte.commune_detail cd
ON c.no_insee = cd.no_insee
GROUP BY intercommunalite
```

Exécutez la requête

A quoi correspond le résultat ?

Combien y a-t-il d'intercommunalités dans la zone d'étude ?

47. Ajouter le compte du nombre de commune par intercommunalité dans la clause SELECT :

```
SELECT intercommunalite, sum(population), count(no_insee)
...
```

Exécutez la requête

ATTENTION ! dans une requête avec agrégation (clause GROUP BY) vous ne pouvez afficher que les critères de groupement (champs de la clause GROUP BY) et des fonctions d'agrégation (sum, count, avg, max, min, ...)

### 3.4 Fonction spatiale

A partir de la dernière requête, il serait intéressant de pouvoir regrouper les polygones des communes afin d'obtenir les polygones des intercommunalités. Pour cela nous pouvons utiliser la fonction d'union : ST\_Union

48. Compléter la clause SELECT de la requête précédente :

```
SELECT intercommunalite, sum(population), count(c.no_insee),
st_union(geom) as geom
```

...

Exécutez la requête

### 3.5 Création d'une vue

Pour cartographier les intercommunalités, nous allons créer une requête afin de pouvoir utiliser le résultat de la requête précédente de façon permanente.

49. Copier la requête que nous venons de faire dans le presse-papier
50. Revenir sur l'écran principal de pgAdmin et développer l'arborescence de la base de données "exercice" et du schéma "contexte"
51. A l'aide d'un clic droit sur le dossier des vues, choisir la commande "Ajouter une vue ..." ("New view ...")
52. Nommer la vue "intercommunalite" et choisir le propriétaire "postgres"
53. Dans l'onglet "Définition", coller le texte de la requête

Le client SIG a besoin d'un identifiant pour utiliser une requête. Pour cela nous allons ajouter le gid maximum à la requête précédente, nous allons aussi donner des alias aux noms de colonnes afin que la vue soit lisible.

54. Corriger la clause SELECT de la requête comme suit :


```
SELECT max(gid) as gid, intercommunalite,  
sum(population) as population, count(c.no_insee) as nb_commune,  
st_union(geom) as geom  
FROM contexte.commune c INNER JOIN contexte.commune_detail cd  
ON c.no_insee = cd.no_insee  
GROUP BY intercommunalite
```

55. Dans l'onglet "Définition", coller le texte de la requête
56. Valider la création de la vue
57. Rafraîchir la liste des vues et cliquer sur la vue intercommunalite  
Observer la syntaxe de la commande de création de la vue  
Afficher les données de la vue
58. Fermer pgAdmin



## 4 Créer des cartes à partir de données PostGIS

Les tables et requêtes de PostGIS sont utilisables directement comme des couches dans QGIS. Pour cela il faut configurer une connexion à la base de données depuis QGIS

59. Lancer QGIS, et cliquer sur la commande "Ajouter une couche PostGIS"  à gauche de l'écran

La première fois que l'on accède à une base de données spatiale, il faut configurer une connexion :

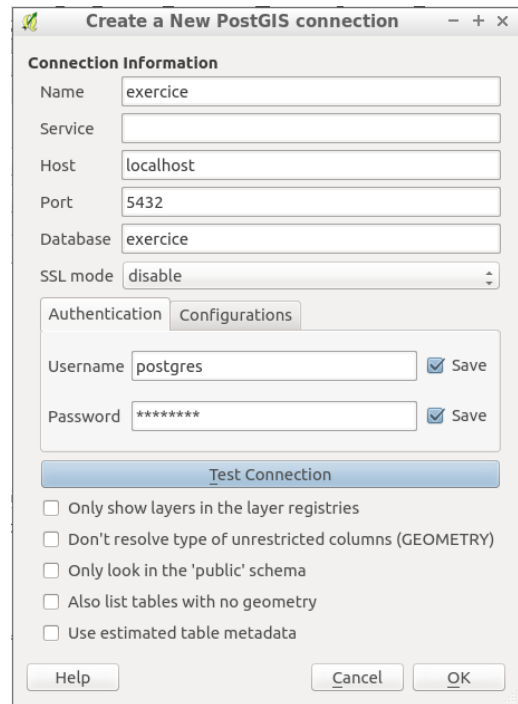
60. Cliquer sur la commande "Nouveau" ("New") pour créer une connexion

61. Configurer la connexion comme présenté ci-contre.

le nom de la connexion servira à retrouver la connexion dans la liste, le mot de passe de l'utilisateur postgres est : "postgres" cocher les 2 options : sauvegarder le nom d'utilisateur et le mot de passe afin de ne pas avoir à le ressaisir ensuite.

62. Tester la connexion

63. Si le test est réussi, cliquer sur OK, la nouvelle connexion apparaîtra dans la liste.



64. Cliquer sur la commande "Connecter", la liste des schémas apparaît

65. Développer le schéma "contexte" pour voir les tables et requêtes que l'on peut ajouter comme couche dans QGIS

66. Ajouter la table des communes, valider

67. Vérifier l'affichage des communes dans QGIS

68. Cliquer de nouveau sur la commande "Ajouter une couche PostGIS" 

69. Choisir la connexion "exercice" dans la liste

70. Sur la requête "intercommunalite", dans la colonne "id de l'entité" ("Feature id") cocher le champ "gid", puis sélectionner et ouvrir la couche

Vous pouvez maintenant construire vos cartes comme avec des fichiers shape