|  |  |  |
| --- | --- | --- |
| **CEFS- UR0035 – Toulouse Comportement Ecologie de la Faune Sauvage** | **Protocole *nommer le protocole***  ***Auquel se rattache l’opération*** | **Réf :CEFS\_trame\_protocole\_V01 1 / 4 Date : 30/03/2020** |

****

**MODE OPERATOIRE   
Se connecter à une base de données distante**

|  |  |  |  |
| --- | --- | --- | --- |
| **version** | **rédacteur** | **relecteur** | **Changement par rapport à la version précédente** |
| **1** | **Yannick Chaval** |  |  |
| **2** | **Yannick Chaval** |  | Remplacement de RPostgreSQL pas Rpostgres |
| **3** | **Yannick Chaval** |  | Mise à jour de l’adresse serveur.  Ajout prise en compte du SSL dans les connexions Qgis.  Changement pgAdmin2Dbeaver. |

**Sommaire**

|  |  |
| --- | --- |
| **1. Descriptif de l’opération**  1.1. NATURE DE L’OPERATION--------------------------------------------------------------------------------------------------  1.2. DESCRIPTIF DE LA METHODE----------------------------------------------------------------------------------------------  1.3. MATERIEL - MODE D’ENREGISTREMENT – ORGANISATION DE L’EQUIPE | **1**  **2**  **3** |
| **2. ANNEXES** | **4** |
| **3. « PENSE-BETE » : LES QUESTIONS A SE POSER ET A RENSEIGNER LORS DE LA REDACTION D’UN PROTOCOLE** | **5** |

**2**

|  |  |  |
| --- | --- | --- |
| **CEFS- UR00335 – Toulouse Comportement Ecologie de la Faune Sauvage** | **Protocole *Nom du protocole*** | **Réf :CEFS\_trame\_protocole\_V01 2 / 4 Date : 07/09/2018** |
|  |  |  |

**1. Descriptif de l’opération**

**1.1. Nature de l’opération**

Les données du CEFS sont structurées au sein de base de données relationnelles client/serveur. Le Système de Gestion de Base de Données (SGBD) est Postgresql et pour les données spatialisées, l’extension dédiée Postgis. Les SGBD relationnelles (SGBDR) utilisent un langage, le Structured Query Language (SQL) qui possède quelques variantes d’un SGBDR à l’autre mais possède également de très fortes similitudes.

C’est un langage structuré qui permet de poser des questions à la base de données. La structure est toujours la même.

**SELECT** <noms\_colonnes> **FROM** <nom\_table> **WHERE** <condition>.

Ce qui en français donne sélectionne moi les colonnes dont les noms sont :<noms\_\_colonnes> de la table <nom\_table> ou les données suive la condition <Condition>.

Un peut ainsi sélectionner tout ou une partie des champs (colonnes) d’une relation (une table ou des tables) qui dont les enregistrements (les lignes) correspondent à la condition.

 Par défaut le SGBDR ira chercher la table dans le schéma par défaut qui se nomme public. Un schéma pour un SGBDR est l’équivalent d’un dossier dans windows. Si l’on veut chercher dans un schéma autre que public, il faut le spécifier de la façon suivant : env\_data.nom\_table dans le schéma env\_data, la table nom\_table

Pour plus d’info :<https://sql.sh/cours/select>

**1.2. Descriptif de l’opération**

Ce document a pour objet d’indiquer la démarche à suivre pour se connecter avec des clients (PqAdmin, R) et pour les données spatialisées (Qgis).

**1.3. MATERIEL - MODE D’ENREGISTREMENT – ORGANISATION DE L’EQUIPE – CONTRAINTES PARTICULIERES**

Posséder un accès au réseau internet et un compte d’accès aux bases de données du CEFS. Si vous ne possédez pas de droits d’accès, contacter [yannick.chaval@inrae.fr](mailto:yannick.chaval@inrae.fr) en indiquant les raisons de votre demande d’accès et la durée de l’accès demandé.

**1.3.1. MATERIEL**

Tout matériel informatique possédant un accès au web et pouvant héberger des application clientes.

**1.3.2. MODE D’ENREGISTREMENT (acquisition des données)**

Sofwares clients abordés ici :

R : <https://cran.r-project.org/>

pgAdmin IV : <https://www.pgadmin.org/download/pgadmin-4-windows/>

Dbeaver : <https://dbeaver.io/download/>

Nous avons choisi de travailler dorénavant avec Dbeaver, le mode de connexion est le même qu’avec pgAdmin IV

Qgis : <https://www.qgis.org/fr/site/forusers/download.html>

- Se connecter avec R :

Nous allons ici utiliser le package RPostgres qui est un client de R pour Postgresql :

<https://cran.r-project.org/web/packages/RPostgres/RPostgres.pdf>

La connexion se fait très simplement en créant un objet R con dans lequel on place les informations de connexion.

con <- dbConnect(Postgres(), host="bdd.cefs.inrae.fr", dbname="db\_name", user="user\_name", password="passwrd")

Il suffit alors de remplacer "db\_name" par la base de données à laquelle vous souhaitez vous connecter (db\_cefs, db\_gardouch, db\_bouquetins, db\_mov\_it) et de remplacer "user\_name" et "passwrd" respectivement par votre mot de passe et votre identifiant.

Une bonne pratique consiste à enregistrer les identifiants de connexion dans un fichier par exemple si vous vous appelez toto dans toto/Programmes/R/toto\_con\_2\_Postgresql.R et d’appeler se script depuis les autres scripts pour se connecter avec la fonction source() :

<https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/source>

Exemple : source(‘c:/users/toto/Programmes/R/toto\_con\_2\_Postgresql.R’). Lorsque vous exécuter cette commande dans R l’objet créé dans le script de connexion (ici con dans l’exemple plus haut) se crée.

Une fois la connexion établie, on peut envoyer des questions et recevoir des réponses de la base de données que l’on va placer dans un objet R à l’aide de la fonction dbGetQuery

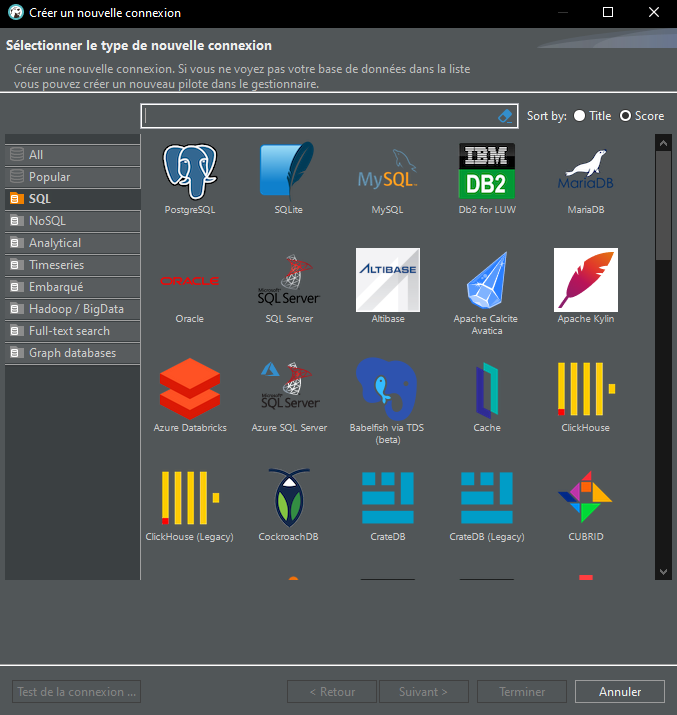
dat<- dbGetQuery(con, "SELECT \* FROM nom\_table")

Ici on affecte à l’objet data toutes les colonnes (\*) de la table nom.table sans condition (pas de clause WHERE).

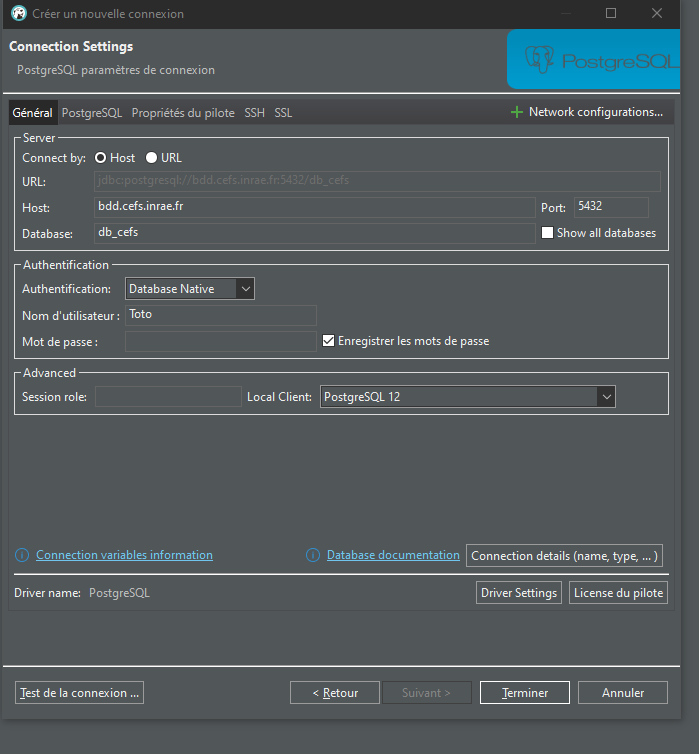
-Se connecter avec Dbeaver : <https://dbeaver.io/download/>

Installer et ouvrir Dbeaver.

Puis cliquer sur la prise (1) une fenêtre « Créer une nouvelle connexion » va s’ouvrir



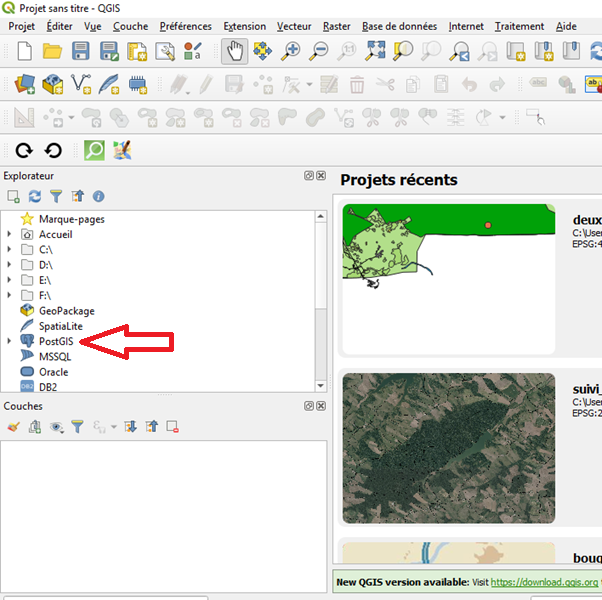
Cliquer sur la tête d’éléphant, PostgreSQL puis cliquer sur suivant.

 Entrer le nom du serveur (Host: bdd.cefs.inrae.fr), le nom de la base de données à laquelle vous souhaitez vous connecter au choix et suivant vos autorisations: db\_cefs, db\_gardouch, db\_bouquetins, db\_mov\_it, db\_cerf.

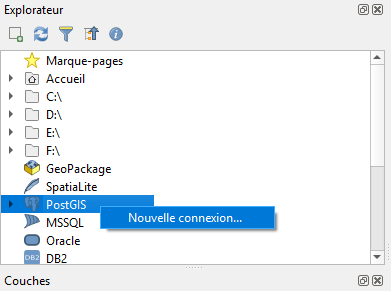
RenseigneZ votre nom d’utilisateur et votre mot de passe.

- Se connecter avec Qgis :

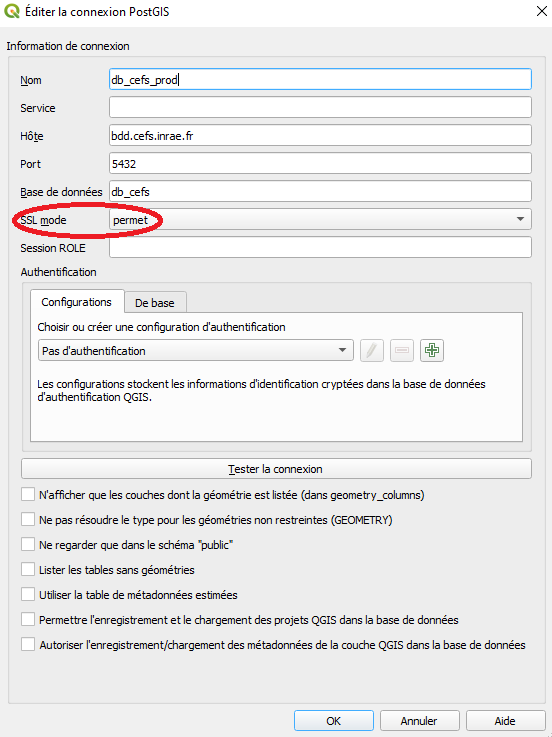
Lorsqu’on ouvre Qgis dans la fenêtre « Explorateur » apparaît Postgis qui est l’extension spatiale de postgresql. Attention seules les tables spatiales de postgresql (i.e. celles qui possède un champ de type geometry) peuvent être affichiées.



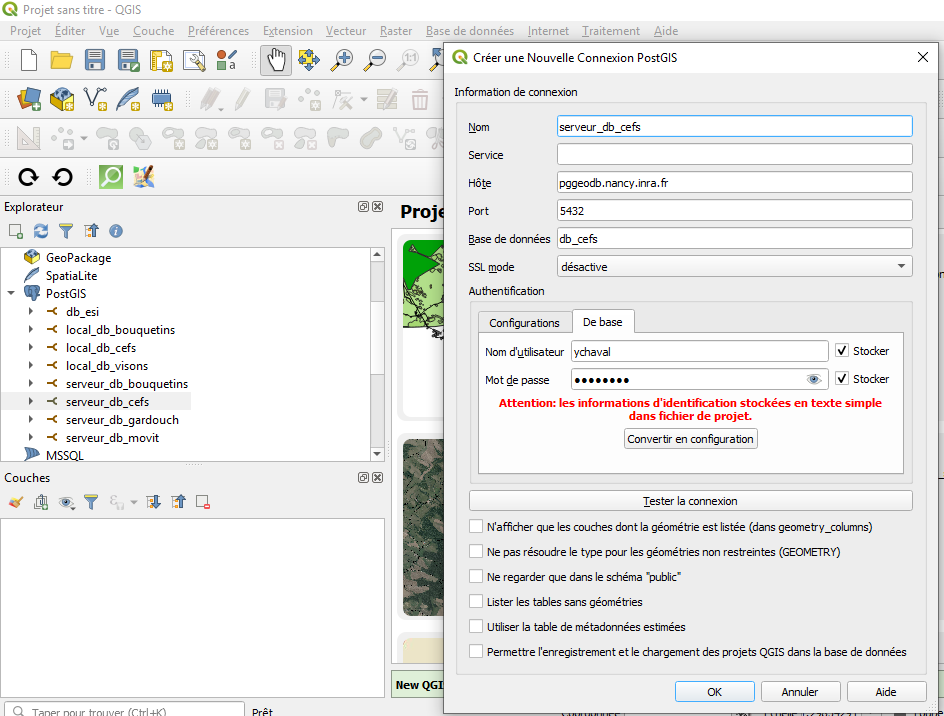
Faire un clic droit dessus et nouvelle connexion.



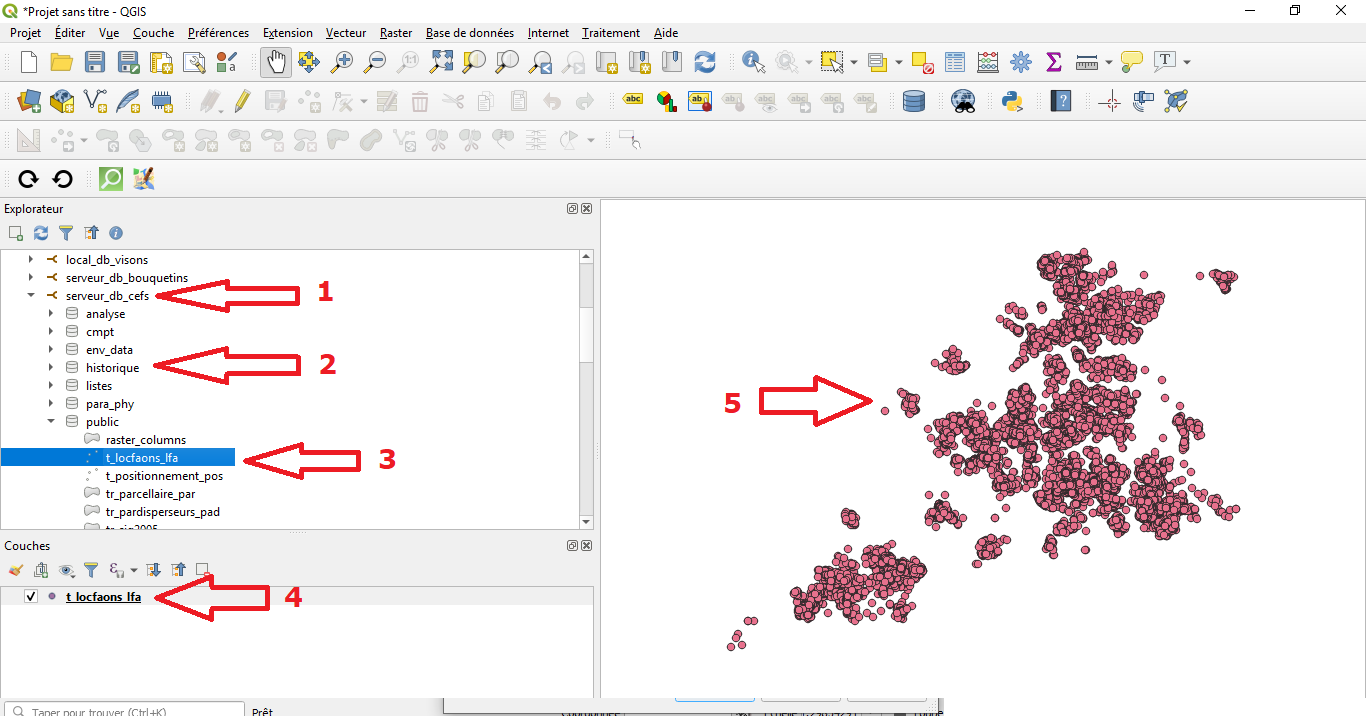
Apparait alors une fenêtre en tout point semblable à celle de Dbeaver. Remplissez les champs en autorisant les connexion ssh. C’est connexions permettent un chiffrement des données entre logiciel client et serveur.



On remplit les champs et on crée la connexion



Une fois la couche crée, elle apparait dans l’explorateur 1 (attention dans la capture d’écran ci-dessus j’ai fait apparaitre une connexion existante c’est pour cela qu’elle apparait déjà dans l’explorateur). Quand on clique sur la connexion choisie, l’arborescence des schémas de la base de données 2 (équivalent aux dossiers de windows). On clique sur le schéma choisi, ici « public » dans l’exemple ci-dessous et les tables du schéma s’affiche alors avec leur tpe de geométrie (des points pour un type point, un patatoïde pour un type polygone et une ligne brisée pour le type ligne). Lorsqu’on double clique sur la couche choisie (3), elle apparait dans la fenêtre couches (4) et sur l’écran d’affichage (5)



**1.3.3. ORGANISATION DE L’EQUIPE**

Si vous rencontrez un problème, après avoir murement réfléchi à celui-ci contacter yannick.chaval@inrae.fr

**1.3.4 CONTRAINTES PARTICULIERES**

- Avec DBeaver :

dans les options de Dbeaver il faut cocher Disable date/time formating (Préférences/Résultats/Formats de données) sinon les données entre 2 et 3 heures du matin le jour du changement d’heure d’été seront mal converties.

Avec cette case non coché (option par défaut lors de l’installation) et dbeaver installé en français :

**select** '2003-03-30 02:01:47'::**timestamp**

2003-03-30 03:01:47

**select** '2003-03-30 02:01:47'::**timestamptz**

2003-03-30 03:01:47

**Ce qui est faux en UTC il n’y a pas de changement d’heure. Une fois la case cochée on a**

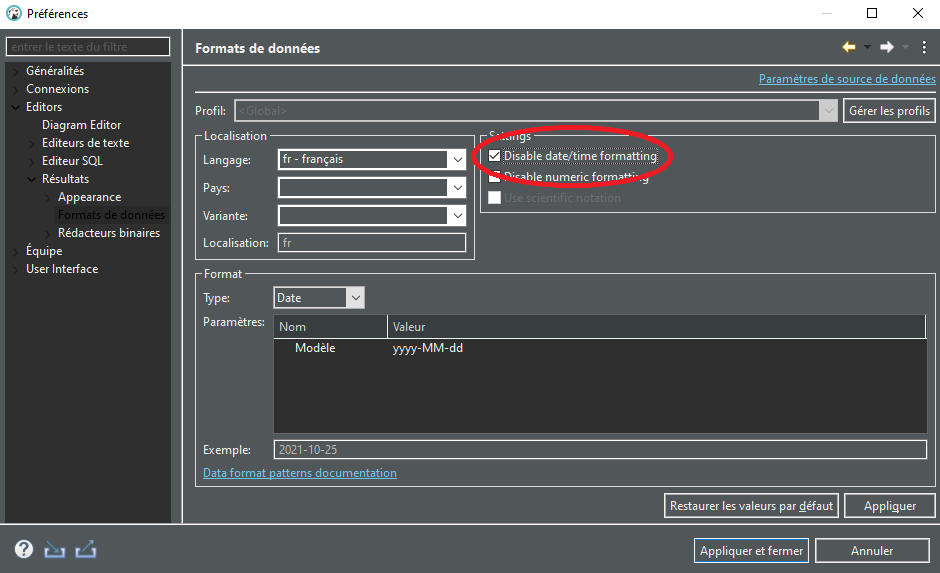
**select** '2003-03-30 02:01:47'::**timestamp**

2003-03-30 02:01:47

**select** '2003-03-30 02:01:47'::**timestamptz**

2003-03-30 03:01:47+02

**Ce qui est juste, Postgresql considère que les heures entre 2h00 et 3h00 du matin n’existe pas puisqu’on change d’heure.**



|  |  |  |
| --- | --- | --- |
| **CEFS- UR0035 – Toulouse Comportement Ecologie de la Faune Sauvage** | **Protocole *Nom du protocole*** | **Réf :CEFS\_trame\_protocole\_V01 4 / 4 Date : 07/09/2018** |
|  |  |  |

**3. « PENSE-BETE » :**

**LES QUESTIONS A SE POSER ET A RENSEIGNER**