



## arcOpole PRO PCRS

---

# Guide Installation

## Version 3.0.1

25/10/2024

## À propos du Guide

Les informations contenues dans ce document pourront faire l'objet de modifications sans préavis de la part de 1Spatial.

Sauf mention contraire, les sociétés, les noms et les données utilisés dans les exemples sont fictifs.

Aucune partie de ce document ne peut être reproduite ou transmise à quelque fin ou par quelque moyen que ce soit, électronique ou mécanique, sans la permission expresse et écrite de 1Spatial.

## Limitation de garantie et de responsabilité

1Spatial a réalisé tous les tests nécessaires et suffisants du Progiciel et a vérifié la conformité de son guide.

Le client reconnaît que dans l'état actuel de la technologie informatique, le fonctionnement du Progiciel est susceptible d'être interrompu ou affecté par des bogues. 1Spatial ne garantit pas que le Progiciel, le média d'installation, la clef ou la documentation livrés soient exempts d'erreurs, de bogues ou d'imperfections.

Ainsi, le client doit effectuer toutes les sauvegardes, prévoir et effectuer toutes les procédures de remplacement en vue d'éventuelles défaillances du Progiciel, prendre toutes les mesures appropriées pour se prémunir contre toute conséquence dommageable due à l'utilisation ou la non-utilisation du Progiciel.

Il est expressément convenu que 1Spatial ne sera en aucun cas responsable des dommages directs ou indirects dus à l'utilisation du Progiciel.

L'utilisation du Progiciel est soumise à la signature par le client du contrat de droit d'utilisation des logiciels de 1Spatial.

## Marques déposées

**Ce progiciel** est une marque déposée de 1Spatial.

**Ce progiciel**, développé par 1Spatial, est une marque déposée et est la propriété exclusive de 1Spatial. Toutes les autres marques citées appartiennent à leurs propriétaires respectifs.

Copyright © 2024, 1Spatial. Tous droits réservés.

## TABLE DES MATIÈRES

<b>1 PRÉSENTATION.....</b>	<b>5</b>
<b>2 PRÉREQUIS.....</b>	<b>6</b>
<b>3 COMPATIBILITÉ .....</b>	<b>7</b>
<b>4 CONFIGURATION PORTAL FOR ARCGIS® .....</b>	<b>8</b>
4.1 Création des groupes .....	8
4.2 Création de l'administrateur PCRS.....	9
4.3 Création de l'application PCRS.....	9
<b>5 COPIE DES FICHIERS DE L'APPLICATION.....</b>	<b>11</b>
<b>6 CRÉATION DE LA BASE .....</b>	<b>13</b>
6.1 Création de la base PCRS .....	13
6.1.1 Création de la base PCRS avec un script python.....	13
6.1.1.1 Modification du fichier CreateGDB.py.....	13
6.1.1.2 Modification du fichier ExecutePython.bat .....	14
6.1.2 Création de la base PCRS avec ArcGIS Pro .....	14
6.1.2.1 Création d'une géodatabase d'entreprise .....	15
6.1.2.2 Création d'un utilisateur de base de données .....	15
6.1.3 Vérification du traitement .....	16
6.2 Initialisation de la base de données de gestion de l'application.....	17
6.3 Initialisation de l'administrateur et des groupes Portal .....	18
6.4 Initialisation de la structure d'une initiative.....	19
6.5 Définition d'une initiative .....	20
6.6 Import/export de la classe Emprise de disponibilité .....	22
6.7 Extension du modèle PCRS .....	22
6.7.1 Utilisation des couches génériques PCRS .....	22
6.7.2 Définition de sous-classes dans le PCRS .....	23
6.7.2.1 Création des couches de données .....	23
6.7.2.2 Déclaration des couches de données et de la sous-classe .....	23
6.7.3 Extension du modèle pour gérer des couches n'appartenant pas au PCRS .....	24
6.7.4 Extension du modèle par l'ajout de nouvelles classes ou de nouveaux attributs .....	24
6.7.4.1 Les contraintes préliminaires.....	24
6.7.4.2 Import/Export GML .....	24
6.7.4.3 Import/Export Geodatabase et Geopackage .....	25
6.7.4.4 Import/Export shapefile .....	25
6.7.4.5 Import/Export DXF et DWG .....	26
6.7.4.6 Méthodologie générale .....	27
6.7.4.7 Rappels sur l'architecture des données.....	27
6.7.4.8 Limitations .....	28
6.7.4.9 Exemples .....	28
6.8 Création de sous-ensembles.....	33
<b>7 GESTION DES CARTES D'UNE INITIATIVE .....</b>	<b>35</b>
7.1 Publication des données.....	35
7.1.1 Création d'une carte dans ArcGIS PRO .....	35
7.1.1.1 Créer une nouvelle carte .....	35
7.1.1.2 Créer une carte à partir d'un modèle mapx .....	38

7.1.2	Modification de la légende de la carte PCRS .....	39
7.1.3	Publication de la source de données dans ArcGIS Server.....	42
7.1.4	Publication du service de carte.....	43
7.2	Création de la carte Web principale .....	45
7.3	Mise à jour de l'initiative .....	45
7.3.1	Modification de la carte de référence de l'initiative .....	45
7.3.2	Modification de la source RASTER de l'initiative .....	46
7.3.3	Ajout d'une emprise .....	46
7.4	Configuration de la popup de consultation .....	46
<b>8</b>	<b>INSTALLATION .....</b>	<b>48</b>
8.1	Ajout de librairies python .....	48
8.2	Configuration de l'application .....	48
8.2.1	arcopolepro.properties.....	48
8.2.2	backoffice.properties.....	49
8.2.3	pcrs.properties.....	49
8.2.4	process.properties .....	49
8.2.5	tasks.properties .....	49
8.3	Configuration du protocole et port d'écoute tomcat.....	50
8.4	Création du service .....	50
8.5	Configuration de déploiement APACHE.....	50
8.6	Démarrage .....	51
<b>9</b>	<b>CHARGEMENT DE LA BASE .....</b>	<b>52</b>
9.1	Import depuis une GDB fichier .....	52
9.2	Import depuis un fichier GML conforme CNIG .....	54
<b>10</b>	<b>CONDITIONS GÉNÉRALES D'UTILISATION .....</b>	<b>57</b>
<b>11</b>	<b>ANNEXES .....</b>	<b>58</b>
11.1	Installation FME .....	58
11.2	Installation Python.....	59
11.3	Gestion des conflits de versions python.....	59
11.4	Établir une connexion avec « pgAdmin » .....	59
11.5	Exécuter des commandes/scripts SQL à l'aide de pgAdmin .....	60
11.6	Réglages APACHE .....	61
11.7	Gestion des certificats .....	62
11.7.1	Récupération du certificat .....	62
11.7.2	Ajout du certificat au cacerts.....	63

## 1 PRÉSENTATION

Le produit arcOpole PRO PCRS permet de gérer la consolidation d'un Plan de Corps de Rue Simplifié. Il est composé de :

- ▶ Un serveur ;
- ▶ Une application cliente d'administration qui prend en charge :
  - La gestion des modèles d'export, d'extraction et d'import de données,
  - La gestion des utilisateurs et l'attribution des modèles qui leur sont accessibles,
  - Le suivi de l'activité,
  - La validation d'un import,
  - L'import et l'export gestionnaire.
- ▶ Une application cliente utilisateur pour :
  - Demander un export sur une zone,
  - Demander une extraction pour modification,
  - Importer des modifications,
  - Visualiser les erreurs d'imports,
  - Suivre les tâches en cours.
- ▶ Un ensemble d'outils annexes :
  - Scripts FME de conversion,
  - Outil AutoCAD pour la saisie des attributs obligatoires,
  - ...

Lors de la mise en œuvre, il est recommandé de placer le serveur derrière un reverse proxy en mode AJP (IIS, NGINX ou Apache 2.4) qui ne fonctionne qu'en mode HTTPS.

## 2 PRÉREQUIS

L'installation du serveur arcOpole PRO PCRS nécessite l'usage d'une installation ArcGIS Enterprise (pas forcément sur le même serveur) pour la définition des utilisateurs et des cartes affichées dans l'application.

- ▶ Version des produits ArcGIS® : 11.1 , réputé compatible pour 10.9.1
- ▶ ArcGIS® Server installé et configuré en https :
  - Le service System\PublishingTools doit être démarré,
  - Le service Utilities\Geometry doit être démarré.
- ▶ Portal for ArcGIS® installé et configuré en https ;
- ▶ ArcGIS© Pro Standard 3.1 pour la publication des services ;
- ▶ Pare-feu activé ;
- ▶ **Apache 2.4.46 minimum** en reverse proxy, avec un certificat SSL.  
**! Un accès vers internet est nécessaire depuis les postes des utilisateurs d'arcOpole PRO PCRS, pour avoir accès aux fonctionnalités de localisation du géocodage d'ESRI.**

Le serveur arcOpole PRO PCRS doit disposer localement des composants suivants :

- ▶ FME Form version 2023, réputé compatible FME 2021 (cf. chapitre 11.1, Installation FME);
- ▶ Environnement **Python 3.6 64 bits au minimum avec module « arcpy » et licence associée**. Cet environnement peut être celui installé en même temps que l'ArcGIS® Server utilisé pour la production de carte (s'il est sur le même serveur). Il peut aussi s'agir d'une installation d'un ArcGIS® Server non démarré, mais licencié sur le serveur où arcOpole PRO PCRS sera installé ;
- ▶ Version du module « arcpy » : 11.1.

Le serveur arcOpole PRO PCRS dispose d'un accès à une base de données PostgreSQL avec extension POSTGIS installée. La version requise dépend de la version du module « arcpy » installé sur le poste.

Pour l'installation, il faut :

- ▶ Un compte avec les droits administrateurs pour ArcGIS® Server ;
- ▶ Un compte avec les droits administrateurs pour Portal for ArcGIS® ;
- ▶ Les connexions au serveur Portal for ArcGIS®, des groupes d'utilisateurs Portal for ArcGIS® ;
- ▶ Les certificats pour les différents sites web.

### 3 COMPATIBILITÉ

Version du Produit arcOpole Pro PCRS	ArcGIS Serveur	FME	JAVA	Python	AutoCAD	Apache	Tomcat
arcOpole Pro PCRS 1.1	10.6 - 10.7.1	2019.0.2	JDK-11.0.1 - JDK-12.0.2	Python 2.7 (64 bits)	2019	> 2.4.46	9.0.20
arcOpole Pro PCRS 1.2	10.7.1 - 10.9.1	2021.2.0.1	JDK-12.0.2	Python 2.7 (64 bits)	2019-2020	> 2.4.46	9.0.20
arcOpole Pro PCRS 2.0	10.9.1	2021.2.0.1 ou plus	JDK-17.0.5	Python 3.6 à 3.7 (64 bits)	2018-2023	> 2.4.46	9.0.70
arcOpole Pro PCRS 3.0	11.1	2021.2.0.1 ou plus	JDK-17.0.5	Python 3.6 à 3.9 (64 bits)	2018-2024	> 2.4.46	9.0.70

## 4 CONFIGURATION PORTAL FOR ARCGIS®

Cette étape permet de définir les éléments Portal qui seront nécessaires pour l'installation de l'application, à savoir l'administrateur principal, les groupes d'utilisateurs et l'application de référence dans Portal.

Ces opérations s'effectuent par l'intermédiaire de l'interface Web de Portal for ArcGIS® en utilisant un identifiant disposant du rôle d'administration.

### 4.1 CRÉATION DES GROUPES

- ▶ Dans l'onglet « Groupes », activer la commande « Créer un groupe » ;
- ▶ Définir les informations du groupe ;
- ▶ Enregistrer le nouveau groupe ;
- ▶ Répéter l'opération deux fois pour créer les groupes « PCRS » et « PCRS\_ADMIN ».

Group Name	Owner	Last Modified	Visibility	Description
PCRS	portaladmin	23 nov. 2021	Tout le monde (public)	Groupe pour les utilisateurs standard de PCRS
PCRS_ADMIN	portaladmin	25 nov. 2021	Tout le monde (public)	Groupe des utilisateurs ayant accès à l'application d'administration PCRS

- ▶ Copier l'identifiant unique de chaque groupe. Ils seront utilisés plus tard lors de l'initialisation de la base. L'identifiant unique du groupe se trouve dans l'URL de consultation de celui-ci (élément souligné dans la copie d'écran ci-dessous) :

## 4.2 CRÉATION DE L'ADMINISTRATEUR PCRS

- Dans l'onglet « Organisation / Membres », activer la commande « Ajouter des membres » ;
- Définir le nouveau membre et ses caractéristiques :

Nouveau membre

Prénom	Nom
Administrateur	PCRS
Adresse électronique	Nom d'utilisateur
admin.pcrs@initiative.fr	pcrsadmin
Type d'utilisateur	Rôles compatibles 5 Licences additionnelles compatibles 8
Creator disponibilité : 45	
Rôle	Mot de passe
Administrateur	*****
Niveau de sécurité du mot de passe: Fort	
<a href="#">Ajouter un autre membre</a> <a href="#">Suivant</a> <a href="#">Annuler</a>	

Groupes

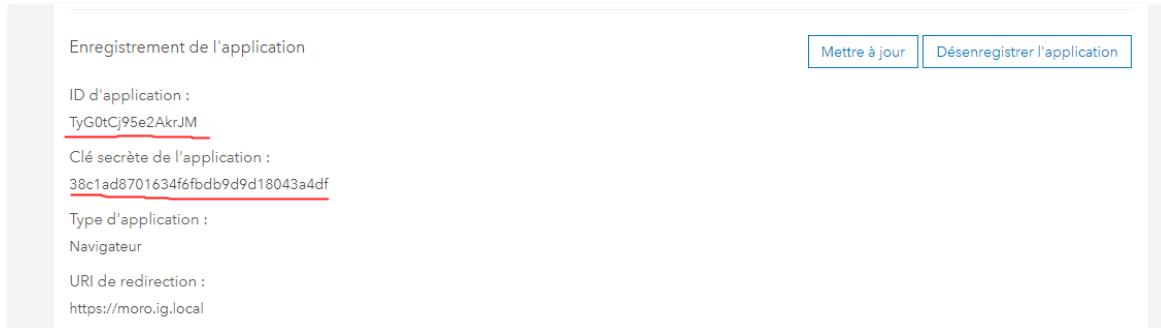
<a href="#">Gérer</a>
PCRS
PCRS_ADMIN

- Enregistrer le nouveau membre.

## 4.3 CRÉATION DE L'APPLICATION PCRS

- Dans l'onglet « Contenu », activer la commande « Nouvel élément » ;
- Choisir le type « Application » ;
- Sélectionner « Cartographie Web » et saisir l'URL de l'application PCRS (par exemple : <https://moro.ig.local/pcrs-server/public/client>);
- Dans la page suivante, définir les informations de l'application ;
- Enregistrer l'application ;
- Dans la page de l'application créée, sélectionner l'onglet « Paramètres » ;
- Activer la commande « Enregistrer » ;
- Sélectionner le type d'application « Navigateur » et définir l'URL de redirection (par exemple <https://moro.ig.local>);

- Copier l'identifiant de l'application et sa clé secrète qui seront utilisés plus tard, lors de la phase de configuration de l'application PCRS :



The screenshot shows a registration form for an application. At the top left is the text "Enregistrement de l'application". On the right are two buttons: "Mettre à jour" and "Déenregistrer l'application". The form contains the following fields:

- ID d'application : TyG0tCj95e2AkrJM
- Clé secrète de l'application : 38c1ad8701634f6fbdb9d9d18043a4df
- Type d'application : Navigateur
- URI de redirection : <https://moro.ig.local>

## 5 COPIE DES FICHIERS DE L'APPLICATION

Copier le contenu de la distribution (décompresser le contenu du fichier zip livré) sur le serveur dans le dossier d'installation.

Par exemple : « E:\1Spatial\pcrs ». Ce dossier sera référencé dans la suite de ce document par « <Dossier d'installation> ».

Ce dossier contient les éléments suivants :

- ▶ Install :
  - construction ;
  - migration ;
  - ressources : fichiers d'exemple pour aider à la configuration (stylx, mapx, ...) ;
- ▶ Documentation :
  - Ce dossier est livré vide. Vous devez y copier les documents d'installation livrés avec la distribution.
- ▶ archives : dossier de travail.
- ▶ config :
  - featureServices ;
  - gdal-ogr ;
  - libs ;
  - scripts : divers dossiers contenant scripts python, modèle FME, graine DWG, ... ;
  - arcopolepro.properties ;
  - backoffice.properties ;
  - pcrs.properties ;
  - process.properties ;
  - tasks.properties .
- ▶ Ressources :
  - configure\_webmap : dossier contenant le script de configuration de la carte PCRS ;
  - convert\_gdb : dossier contenant le script de conversion de la base PCRS en GDB fichier ;
  - plugin\_autocad : dossier contenant le plugin AutoCAD.
  - quadrillage\_pcrs
- ▶ Server :
  - apache-tomcat-9.0.70 ;
  - jdk-17.0.5.
- ▶ temp : dossier de travail.
  - processes ;
  - tasks ;
  - uploads ;

**! Si FME n'est pas installé dans le chemin par défaut : « c:\program files\FME\fme.exe », alors modifier le fichier « <Dossier d'installation>/config/scripts/pcrs\_ParamConfig.py ».**

Vers la ligne 35 il contient la ligne :

```
g_strPathFME_EXE = r"C:\Program Files\FME\fme.exe"
```

dans laquelle il faut remplacer le chemin par défaut, par le chemin réel du fichier fme.exe.

## 6 CRÉATION DE LA BASE

### 6.1 CRÉATION DE LA BASE PCRS

Il existe deux méthodes pour la mise en place de la base de données PCRS :

- ▶ soit à travers un script python si PostgreSQL est installé sur la même machine que la plateforme ESRI et si vous avez les droits d'exécuter le script sur cette machine
- ▶ soit en utilisant les outils standards ArcGIS Pro.

Il est conseillé de créer une base spécifique dans l'instance PostgreSQL pour le stockage des données du PCRS.

Il est obligatoire d'avoir l'extension POSTGIS installée sur la base.

De plus, bien que non utilisée dans la version actuelle du logiciel, il est conseillé d'en faire une géodatabase ESRI pour la prise en charge d'évolution future.

#### 6.1.1 Crédation de la base PCRS avec un script python

Dans le dossier « <Dossier d'installation>/install/construction », vous trouverez les fichiers modèles permettant de créer une base avec les éléments nécessaires :

- ▶ Fichier « CreateGDB.py » ;
- ▶ Fichier « ExecutePython.bat » pour déclencher le script Python.

Ces deux fichiers modèles doivent être édités pour définir les informations de l'environnement d'exécution (Chemin, Adresse, Mot de passe, ...).

- ! Il faut que toutes les informations encodées ci-dessous soient en minuscules !**
- ! Sur les environnements basés sur une version ArcGIS Enterprise 10.9.1, il existe actuellement un bug qui empêche le chargement du module « arcpy » par l'interpréteur python. Pour pallier ce bug, il faut copier le fichier « <Dossier ArcGIS Server>\Support\Python\Server10.9.pth »**  
**Dans le dossier :**  
**« Votre\_Installation\_Python\ArcGIS\Pro\bin\Python\envs\arcgispro-py3\Lib\site-packages ».**  
**Ensuite, il faut l'édition et modifier les chemins s'y trouvant en fonction de votre installation.**

##### 6.1.1.1 Modification du fichier CreateGDB.py

Pour chaque paramètre dont la valeur est soulignée en jaune ci-dessous, encoder la valeur adéquate (en fonction de votre environnement) ou souhaitée (par exemple : nom de la base de données, mot de passe...).

```
import arcpy
import os
workFolder = sys.path[0] # "<Dossier de travail du script>"
pgFolder = "<Chemin d'installation de PostgreSQL>" 
licensePath = "<Chemin complet du fichier de licence ESRI pour la géodatabase (ecp)>"
```

```

pghost = "<FQDN du serveur de la base PostgreSQL>" 
pgport = "<Port de la base PostgreSQL>" 
pgtblspace = "<Nom du tablespace de stockage de la base PCRS>" 
pgbase = "<Nom de la base PCRS>" 
pguser = "<Identifiant administrateur PostgreSQL. Par défaut 'postgres'>" 
pgpwd = "<Mot de passe de l'administrateur de la base>" 
sdepswd = "<Mot de passe de l'administrateur de la géodatabase (sde)>" 
pcrsusr = "pcrs" //Schéma principal de la base : NE PAS MODIFIER 
pcrspwd = "<Mot de passe du propriétaire du schéma principal>" 

instance = pghost + "," + str(pgport) 
...

```

### 6.1.1.2 Modification du fichier ExecutePython.bat

Renseigner le chemin vers votre installation de python sur le serveur où le script va être exécuté.

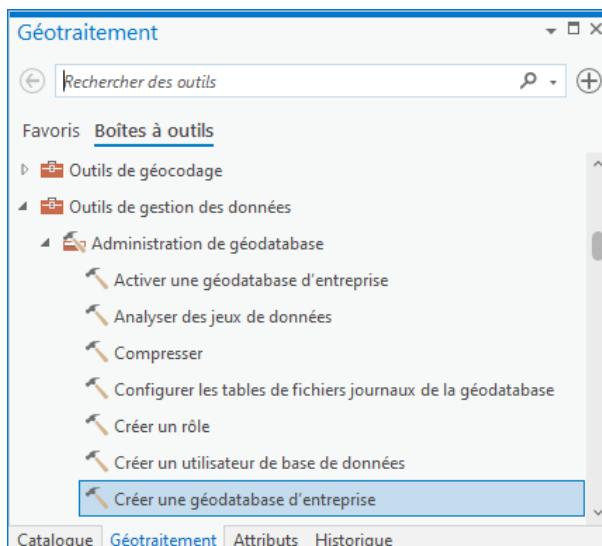
```

SET PYTHONPATH=<Chemin vers l'installation Python avec le module arcpy>
SET CURDIR=%~dp0
"%PYTHONPATH%\python.exe" "%CURDIR%CreateGDB.py"
pause

```

### 6.1.2 Crédit de la base PCRS avec ArcGIS Pro

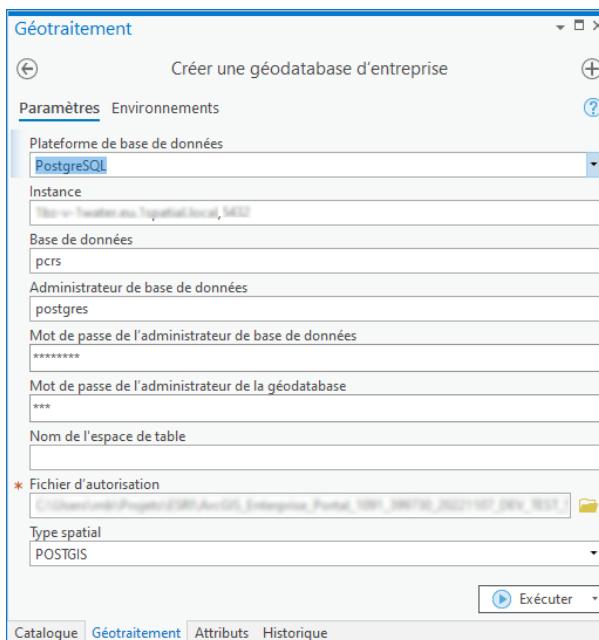
Dans cette partie, toutes les étapes de mise en place de la base de données se font à partir de la boîte à outils de gestion des données d'ArcGIS PRO.



### 6.1.2.1 Création d'une géodatabase d'entreprise

Pour ce faire, sélectionner l'outil « Crée une géodatabase d'entreprise », puis encoder les valeurs des paramètres demandés.

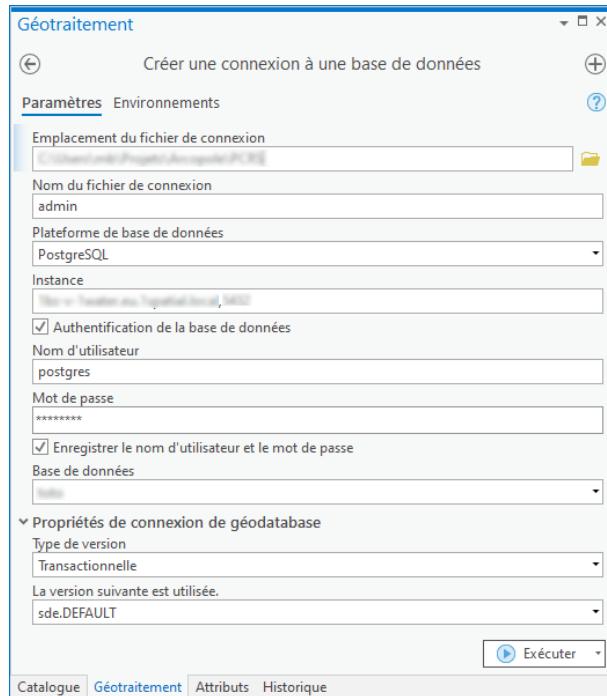
- ! Il faut s'assurer que la plateforme de base de données est « PostgreSQL » et que le type spatial est « POSTGIS ».**



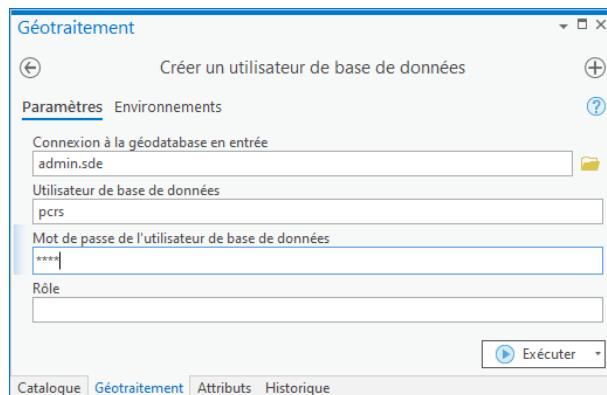
### 6.1.2.2 Création d'un utilisateur de base de données

La création d'un utilisateur de base de données à partir d'ArcGIS Pro se fait en deux étapes. Il faut d'abord créer un fichier de connexion « .sde » à la base de données. Puis, seulement on peut créer l'utilisateur qui se base sur ce fichier de connexion. Ces deux étapes vont être décrites ci-dessous.

- **Création d'un fichier de connexion à la base de données :** Pour ce faire, sélectionner l'outil « Créer une connexion à une base de données » qui permettra de créer le fichier de connexion « .sde » à la base de données avec un utilisateur administrateur des données. Puis, encoder les valeurs des paramètres demandées.



- **Création d'un utilisateur de la base de données :** Pour ce faire, sélectionner l'outil « Créer un utilisateur de base de données » qui permet de créer et assigner un nouvel utilisateur à la base de données. Puis encoder les valeurs des paramètres demandées.



### 6.1.3 Vérification du traitement

Afin de vérifier que la création de la base de données pour PCRS a été exécutée avec succès, utiliser « pgAdmin » pour vous connecter à la nouvelle base de données et accéder au schéma « pcrs » avec son utilisateur propriétaire.

- ! Pour se connecter à une base avec « pgAdmin », cf. chapitre 11.4, Établir une connexion avec « pgAdmin ».

## 6.2 INITIALISATION DE LA BASE DE DONNÉES DE GESTION DE L'APPLICATION

Dans le dossier « <Dossier d'installation>/install/construction », vous trouverez le modèle de script SQL « pcrs.sql » permettant d'initialiser la base de gestion de l'application.

Ce script doit être modifié avant d'être exécuté.

- ▶ Ouvrir ce fichier dans un éditeur de texte ;
- ▶ Par la commande rechercher/remplacer, remplacer la variable \${XXX} par sa valeur :
  - \${pcrs\_schema} par le nom du schéma principal « pcrs ».

```
SET client_encoding = 'UTF8';

CREATE SCHEMA ${pcrs_schema} AUTHORIZATION ${pcrs_schema};
```

qui devient :

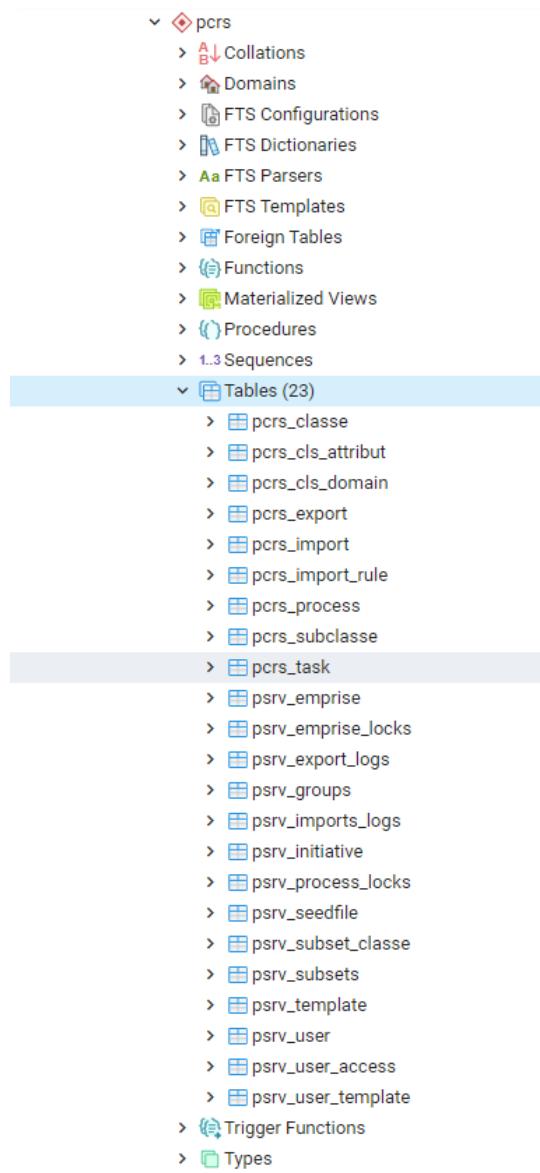
```
SET client_encoding = 'UTF8';

CREATE SCHEMA pcrs AUTHORIZATION pcrs
```

Il doit être exécuté dans le schéma « pcrs », avec l'utilisateur propriétaire de ce schéma (de même nom par défaut). Le mot de passe correspondant a été défini lors de la création de la base.

- ! Pour exécuter un script SQL, cf. chapitre 11.5 [Exécuter des commandes/scripts SQL à l'aide de pgAdmin](#).

Après l'exécution du script, le schéma PCRS doit contenir les éléments suivants :



## 6.3 INITIALISATION DE L'ADMINISTRATEUR ET DES GROUPES PORTAL

Un utilisateur administrateur est associé à une initiative. Une initiative a une identifiant numérique unique. Avec une seule initiative, cet identifiant sera 1.

Pour gérer plusieurs initiatives, il faudra définir un administrateur par initiative.

Dans le dossier « <Dossier d'installation>/install/construction », vous trouverez le modèle de script SQL « pcrs-load.sql » permettant de déclarer l'utilisateur administrateur.

- Ouvrir ce fichier dans un éditeur de texte ;
- Par la commande rechercher/remplacer, remplacer les variables \${XXX} par leurs valeurs :
  - \${pcrs\_schema} par le nom du schéma principal « pcrs »
  - \${initiative\_id} par l'identifiant de l'initiative (par défaut « 1 »)

- Définir les autres éléments surlignés en jaune ci-dessous :

```

SET client_encoding = 'UTF8';
SET search_path = ${pcrs_schema}, public, pg_catalog;

-- Users
insert into ${pcrs_schema}.psrv_user (initiative_id, email, username) values (
${initiative_id},
'<Adresse mail de l'administrateur PCRS>',
'<Identifiant de connexion portal de l'administrateur>');

-- Groupes
insert into ${pcrs_schema}.psrv_groups (role, group_id) values ('ADMIN',
'<Identifiant unique Portal du groupe PCRS ADMIN>');
insert into ${pcrs_schema}.psrv_groups (role, group_id) values ('PCRS',
'<Identifiant unique Portal du groupe PCRS>');

```

- ! L'application PCRS peut supporter plusieurs initiatives (ayant chacune un numéro unique de 1 à n).
- ! Il faut alors définir un Administrateur pour chaque initiative, et donc, dupliquer autant de fois que nécessaire la ligne « insert into pcrs.psrv\_user... » en indiquant à chaque fois l'identifiant de l'initiative.
- Exécuter le script obtenu dans le schéma « pcrs », avec l'utilisateur propriétaire de ce schéma (de même nom par défaut). Le mot de passe correspondant a été défini lors de la création de la base ;
- ! Pour exécuter un script SQL, cf. chapitre 11.5, Exécuter des commandes/scripts SQL à l'aide de pgAdmin.

Après l'exécution du script, les tables ont été chargées (« psrv\_user » : 1 enregistrement et « psrv\_groups » : 2 enregistrements).

- ! L'identifiant unique d'un groupe Portal peut être obtenu dans l'URL de consultation de ce groupe (par exemple : 'cb6bb51c5c5440e3a61830f6512195da').

## 6.4 INITIALISATION DE LA STRUCTURE D'UNE INITIATIVE

Une initiative a sa référence spatiale dans laquelle seront stockées ces données. Cette référence spatiale est définie par son SRID.

Si l'on gère plusieurs initiatives, il est préférable que toutes les initiatives aient la même référence spatiale.

Une initiative a besoin d'un schéma de stockage PostgreSQL pour ses données.

Le nom de ce schéma doit être une chaîne de caractères en minuscules, sans accents, ni caractères spéciaux, ni espaces, ni nombres.

Ce schéma doit être créé, ainsi que des couches et tables dans ce schéma.

Dans le dossier « <Dossier d'installation>/install/construction », vous trouverez le script SQL « initiative.sql » permettant de créer ce schéma, les couches et les tables qu'il doit contenir.

- Ouvrir ce fichier dans un éditeur de texte ;
- Par la commande rechercher/remplacer, remplacer les variables \${XXX} par leurs valeurs :
  - \${pcrs\_schema} par le nom du schéma principal « pcrs ».

- \${initiative\_schema} par le nom du schéma de l'initiative.
  - \${initiative\_srid} par le SRID de la référence spatiale des données de l'initiative.
- Exécuter le script obtenu dans le schéma « pcrs », avec l'utilisateur propriétaire de ce schéma (de même nom par défaut). Le mot de passe correspondant a été défini lors de la création de la base.
- ! Pour exécuter un script SQL, cf. chapitre 11.5, Exécuter des commandes/scripts SQL à l'aide de pgAdmin.**

À la fin de l'exécution de ce script, un nouveau schéma a été créé dans la base PCRS avec le nom affecté à la chaîne « \${initiative\_schema} »). Ce schéma contient toutes les tables obligatoires du standard CNIG PCRS.

Pour définir plusieurs initiatives, répéter cette étape pour chacune des autres initiatives.

## 6.5 DÉFINITION D'UNE INITIATIVE

Pour définir une initiative, on a besoin :

- De la géométrie de son emprise globale, exprimée dans la référence spatiale de l'application PCRS (par exemple SRID 2154) qui sera définie dans les fichiers de configuration de celle-ci (cf. chapitre 8.2.3, pcrs.properties).
- De définir la grille de l'initiative qui couvre son emprise.
 

**! La définition du système de cellules de la grille doit être réfléchie en amont, car ce dernier ne peut pas être modifié sans perdre l'historique de toutes les opérations sur l'initiative. Le quadrillage intégral représente un carré de 32768 cellules de côté. Selon les hauteur et largeur définis pour les cellules (par exemple 140 par 100), cela représente un espace considérable qu'il convient de centrer au mieux (en jouant sur les X et Y d'origine du quadrillage) sur l'espace de l'initiative pour éliminer le besoin ultérieur de modification en cas d'extension de celle-ci.**
- De l'identifiant de la carte Web Portal de cette initiative. Cette carte n'est pas encore définie à cette étape. Il faudra venir modifier cette valeur une fois la carte créée. Donc pour l'instant indiquer « XXX » pour l'identifiant unique de la carte Web portal. Cet identifiant sera à redéfinir une fois la carte de l'initiative créée (cf. chapitre 7.3.1, Modification de la carte de référence de l'initiative).

Dans le dossier « <Dossier d'installation>/install/construction », vous trouverez le modèle de script SQL « initiative-load.sql » permettant d'effectuer cette opération.

Dans le cas de plusieurs initiatives (chacune ayant son identifiant de 1 à N), il faudra répéter cette opération pour chaque initiative.

- Ouvrir ce fichier dans un éditeur de texte ;
- Par la commande rechercher/remplacer, remplacer les variables \${XXX} par leurs valeurs :
  - \${pcrs\_schema} : Nom du schéma métier (« pcrs »).
  - \${initiative\_schema} : Nom du schéma de l'initiative.
  - \${initiative\_id} : Identifiant de l'initiative (de 1 à N).

- \${initiative\_description} : Description de l'initiative (si cette description contient le caractère « ' », il faut le doubler. Exemple «Ville d'Alès» -> «Ville d"Alès»).
- \${initiative\_name} : Nom de l'initiative (si cette description contient le caractère « ' », il faut le doubler).
- \${initiative\_srid} : SRID de la référence spatiale des données de l'initiative.
- \${vertical\_srid} : SRID de la référence spatiale verticale (généralement : 5720).
- \${z\_min} : Altitude minimum des données (0 par exemple).
- \${z\_max} : Altitude maximum des données (4000 par exemple).
- \${map\_id} : Identifiant unique de la carte Web Portal à utiliser (pour l'instant indiquer « xxx »).
- \${subset\_name} : Nom du sous ensemble de données principal (par défaut : « pcrs »).
- \${subset\_label} : Libellé du sous-ensemble de données principal (par défaut : « pcrs »).
- \${x\_origin} : Origine X (sous forme d'un nombre entier) des cellules de la grille de l'initiative, dans la référence spatiale de l'initiative.
- \${y\_origin} : Origine Y des cellules (sous forme d'un nombre entier).
- \${cell\_width} : Largeur des cellules (sous forme d'un nombre entier), par exemple 140.
- \${cell\_height} : Hauteur des cellules (sous forme d'un nombre entier), par exemple 100.
- \${emprise\_id} : Identifiant de l'emprise principale (de 1 à N). Chaque emprise doit avoir un identifiant unique de 1 à N, toutes initiatives confondues.
- \${emprise\_name} : Nom de l'emprise principale de l'initiative (si ce nom contient le caractère « ' », il faut le doubler).
- \${geometry} : Géométrie de l'emprise de l'initiative, codée en geojson dans la référence spatiale principale **de l'application PCRS** (généralement 2154) qui sera définie dans les fichiers de configuration de celle-ci (cf. chapitre 8.2.3, pcrs.properties).

La géométrie de l'emprise est à fournir, sous la forme d'une chaîne json et de l'utilisation de fonction PostGIS.

Exemple :

```
st_setsrid(st_geomfromgeojson(' {"type":"Polygon","coordinates": [[[780000,6328000], [780000,6345000],[793000,6345000],[793000,6328000],[780000,6328000]]]}'),2154)
```

Il est souhaitable de caler l'emprise par rapport au système de cellules défini pour l'initiative.

**! Par défaut, le script comporte une seule emprise qui doit être l'enveloppe englobante de la zone de travail.**

**Pour définir plusieurs emprises pour une initiative, cf. chapitre 7.3.3, Ajout d'une emprise.**

- Exécuter le script obtenu dans le schéma « pcrs », avec l'utilisateur propriétaire de ce schéma (de même nom). Le mot de passe correspondant a été défini lors de la création de la base.
- ! **Pour exécuter un script SQL, cf. chapitre 11.5, Exécuter des commandes/scripts SQL à l'aide de pgAdmin.**

## 6.6 IMPORT/EXPORT DE LA CLASSE EMPRISE DE DISPONIBILITÉ

Si l'on souhaite pouvoir importer et exporter les zones d'emprise de disponibilité de PCRS, il faut déclarer des modèles spécifiques (pour chacune des initiatives).

Dans le dossier « <Dossier d'installation>/install/construction », vous trouverez le modèle de script SQL « pcrs\_declare\_template\_info.sql » permettant d'effectuer cette opération.

Dans le cas de plusieurs initiatives (chacune ayant son identifiant de 1 à N), il faudra répéter cette opération pour chaque initiative.

- ▶ Ouvrir ce fichier dans un éditeur de texte ;
  - ▶ Par la commande rechercher/remplacer, remplacer les variables \${XXX} par leurs valeurs :
    - \${pcrs\_schema} : Nom du schéma métier (« pcrs »).
    - \${initiative\_id} : Identifiant de l'initiative (de 1 à N).
    - \${initiative} : Nom de l'initiative.
  - ▶ Exécuter le script obtenu dans le schéma « pcrs », avec l'utilisateur propriétaire de ce schéma (de même nom). Le mot de passe correspondant a été défini lors de la création de la base.
- ! Pour exécuter un script SQL, cf. chapitre 11.5, Exécuter des commandes/scripts SQL à l'aide de pgAdmin.**

## 6.7 EXTENSION DU MODÈLE PCRS

En fonction des besoins, il existe plusieurs possibilités pour étendre le modèle d'une initiative PCRS.

### 6.7.1 Utilisation des couches génériques PCRS

Une des possibilités d'extension est basée sur l'utilisation des classes d'objets génériques (Point, ligne et polygones) du standard CNIG PCRS v2.0 pour stocker des données particulières, ne faisant pas partie des thématiques de base du standard CNIG. En effet, ces classes permettent de décrire un objet levé du PCRS dont la précision du géoréférencement est connue, sans qu'il ne soit rattaché à une autre classe d'objet du standard. Le champ [identification] prévu dans le standard, décrit le nom de la classe spécialisée, faisant partie en général des thématiques PCRC (Plan de Corps de Rue Complet) ou RTGE (Référentiel Topographique à Grande Échelle).

En utilisant le champ [identification] lors de la création de la BD PCRS et en modifiant les scripts FME d'import et d'export vers DWG, on pourra obtenir des layers spécifiques dans le fichier résultat et gérer ces éléments de plan tout en les conservant dans le PCRS.

Le principal avantage de cette méthode est de ne pas changer le modèle de données PCRS dans la base de données centrale et de faire tout le filtrage à partir des scripts FME. Toutefois, cette méthode peut être limitée pour les conversions de formats des données n'utilisant pas FME (le format GDB utilise les outils ESRI et le format GPKG utilise GDAL. Ces deux formats n'utilisent donc pas FME) ; de plus, elle nécessite des paramétrages spécifiques des scripts FME d'import/export au moment du déploiement de l'application.

## 6.7.2 Définition de sous-classes dans le PCRS

Il peut être avantageux de découper une classe du PCRS en fonction d'un attribut et d'une valeur. Cela permettra de rester dans le cadre du PCRS, tout en pouvant cibler les opérations de mise à jour (import ou import gestionnaire) sur une partie seulement de la classe.

Un exemple de ce type de fonctionnement est défini dans le script standard pour la gestion des affleurants du gestionnaire ENEDIS : seul le gestionnaire ENEDIS a le droit de mettre à jour ses affleurants et il le fait annuellement par la fourniture d'un fichier GML contenant l'intégralité de l'initiative.

Cette règle de gestion se traduit de la manière suivante :

- ▶ Création des couches de données
- ▶ Déclaration des couches de données et déclaration d'une sous-classe « Affleurant\_ENEDIS » filtrée par la requête « gestionnaire='Enedis' » ;
- ▶ Définition d'un modèle d'import gestionnaire ENEDIS ayant le droit de modifier la sous-classe ;
- ▶ Aucun autre modèle d'import ne doit permettre la modification de la sous-classe ;

Les deux premières étapes sont décrites par les chapitres suivants.

### 6.7.2.1 Crédation des couches de données

Dans le dossier « <Dossier d'installation>/install/construction », vous trouverez le script SQL « initiative-enedis.sql » permettant de créer les couches nécessaires.

- ▶ Ouvrir ce fichier dans un éditeur de texte ;
- ▶ Par la commande rechercher/remplacer, remplacer les variables \${XXX} par leurs valeurs :
  - \${pcrs\_schema} par le nom du schéma principal « pcrs ».
  - \${initiative\_schema} par le nom du schéma de l'initiative.
  - \${initiative\_srid} par le SRID de la référence spatiale des données de l'initiative.
- ▶ Exécuter le script obtenu dans le schéma « pcrs », avec l'utilisateur propriétaire de ce schéma (de même nom par défaut). Le mot de passe correspondant a été défini lors de la création de la base ;
  - ! Pour exécuter un script SQL, cf. chapitre 11.5, Exécuter des commandes/scripts SQL à l'aide de pgAdmin.**

### 6.7.2.2 Déclaration des couches de données et de la sous-classe

Dans le dossier « <Dossier d'installation>/install/construction », vous trouverez le script SQL « initiative-load-enedis.sql » permettant de déclarer les couches nécessaires dans l'initiative.

- ▶ Ouvrir ce fichier dans un éditeur de texte ;
- ▶ Par la commande rechercher/remplacer, remplacer les variables \${XXX} par leurs valeurs :
  - \${pcrs\_schema} par le nom du schéma principal « pcrs ».
  - \${initiative\_schema} par le nom du schéma de l'initiative.
  - \${initiative\_id} par le numéro de l'initiative
  - \${subset\_name} par le nom du sous-ensemble de données (par défaut : pcrs)

- Exécuter le script obtenu dans le schéma « pcrs », avec l'utilisateur propriétaire de ce schéma (de même nom par défaut). Le mot de passe correspondant a été défini lors de la création de la base ;
  - ! Pour exécuter un script SQL, cf. chapitre 11.5, Exécuter des commandes/scripts SQL à l'aide de pgAdmin.**

### 6.7.3 Extension du modèle pour gérer des couches n'appartenant pas au PCRS

Le principe est similaire au cas précédent, sans la déclaration en tant que sous-classe (création des couches et description). En revanche, ces couches ne devront pas faire partie du sous-ensemble « pcrs », ni des modèles d'import au format GML PCRS, car hors standard CNIG. Cela nécessitera aussi de faire un autre sous-ensemble de données pour y définir des modèles d'import-export les prenant en charge.

### 6.7.4 Extension du modèle par l'ajout de nouvelles classes ou de nouveaux attributs

Les explications ci-dessous décrivent la mise en place d'un script définissant une extension du modèle PCRS permettant :

- d'ajouter de nouvelles classes,
- d'ajouter des champs (attributs complémentaires) à des classes du standard CNIG PCRS v2.0.

#### 6.7.4.1 Les contraintes préliminaires

- Ne pas modifier les champs existants des classes du standard CNIG PCRS v2.0
- Ne pas changer le nom des classes du standard CNIG PCRS v2.0
- Ne pas supprimer des classes du standard CNIG PCRS v2.0
- Ne pas changer le nom de champs existants du standard CNIG PCRS v2.0
- Ne pas supprimer de champs existants du standard CNIG PCRS v2.0
- Ne pas supprimer des nomenclatures du standard CNIG PCRS v2.0
- Ne pas supprimer des valeurs des nomenclatures du standard CNIG PCRS v2.0
- ! Toutes les initiatives doivent avoir la même structures (tables et champs).**

**Ce qui est autorisé :**

- Ajouter des champs dans des classes existantes.
- Ajouter de nouvelles classes.
- Ajouter de nouvelles nomenclatures.
- Ajouter des valeurs à des nomenclatures existantes du standard CNIG PCRS v2.0.

#### 6.7.4.2 Import/Export GML

Si l'on ajoute des champs ou des classes, celles-ci ne seront pas gérées par l'import-export GML : cet export doit rester conforme au standard CNIG PCRS v2.0.

En export, les classes et champs supplémentaires ne seront pas exportés.

En import, puisque le GML ne peut pas contenir ces classes et champs supplémentaires, ils ne seront pas importés.

Donc, si un export GML, puis un réimport GML sont effectués sur une zone géographique, les données des champs et classes supplémentaires seront perdues.

- ! En conclusion : si l'on souhaite étendre le MDD par des nouveaux champs et nouvelles classes, le format d'échange GML doit être abandonné pour l'export/import (à moins de définir un nouveau format d'échange GML, mais qui ne sera plus conforme au standard CNIG PCRS v2.0 ce qui est fortement déconseillé !).**

#### 6.7.4.3 Import/Export Geodatabase et Geopackage

Les classes et champs supplémentaires seront exportés et réimportés sans perte de données.

#### 6.7.4.4 Import/Export shapefile

L'import/export shapefile supporte automatiquement les champs et classes supplémentaires, mais lors de l'export en shapefile, ces noms de champs seront tronqués à 10 caractères (c'est une limitation du format shapefile). Il y a donc des contraintes sur ces noms de champs pour qu'ils résistent à l'export shapefile.

Dans le script SQL qui déclare les champs des classes d'une initiative (initiativel\_load.sql), les noms de champs lors de l'export, sont contenus dans la colonne 'pcrs\_nom' de la table pcrs\_cls\_attribut (Cf. chapitre 6.5, Définition d'une initiative, script SQL « initiative-load.sql »).

Pour le nommage des champs d'export, il faut donc respecter quelques règles. Si ces règles ne sont pas respectées, l'export en shapefile sera refusé (pour ne pas risquer de perdre des données lors du réimport).

Que cela soit pour ajouter des champs à des classes existantes, ou pour déclarer de nouvelles classes, les noms des champs (supplémentaires ou non) lors de leur exportation, doivent respecter les règles suivantes :

- Soit porter un nom de moins de 10 caractères,
- soit porter un nom de plus de 10 caractères, mais correspondant à un des noms de champ du standard CNIG PCRS V2.0

Nom dans la géodatabase d'export	Nom dans les shapefiles
'anglerotation'	'anglerotat'
'dateacquisition'	'dateacquis'
'datecreation'	'datecreati'
'daterecette'	'daterecett'
'descriptionelementsqualite'	'descriptio'
'gestionnaire'	'gestionnai'
'id_affleurantsymbole'	'id_affleur'
'id_habillagesymbole'	'id_habilla'
'identification'	'identifica'
'idhabillage'	'idhabillag'

Nom dans la géodatabase d'export	Nom dans les shapefiles
'idnomvoirie'	'idnomvoiri'
'idnumerovoarie'	'idnumerovo'
'immatriculation'	'immatricul'
'justification'	'justificat'
'longueur_mm'	'longueur_m'
'numeropoint'	'numeropoin'
'positionnement'	'positionne'
'precisionaltimetrique'	'precisiona'
'precisionplanimetrique'	'precisionp'
'precisionplanimetrique'	'precisionp'
'precisionplanimetriquecorpsderue'	'precisionp'
'precisionplanimetriquezonenaturelle'	'precisionp'
'precisionxy'	'precisionx'
'qualitecategorisation'	'qualitecat'
'transparence'	'transparen'
'fournisseur'	'fournisseu'
'datepublication'	'datepublic'

- soit porter un nom de 10 caractères exactement, mais qui ne rentre pas en conflit avec un des noms de champ du standard CNIG PCRS V2.0, quand ils sont tronqués à 10 caractères. Donc **il ne faut pas utiliser** un des noms suivants :

'anglerotat'	'idhabillag'	'precisiona'
'dateacquis'	'idnomvoiri'	'precisionp'
'datecreati'	'idnumerovo'	'precisionx'
'daterecett'	'immatricul'	'qualitecat'
'descriptio'	'justifica'	'transparen'
'gestionnai'	'longueur_m'	'fournisseu'
'id_affleur'	'numeropoin'	'datepublic'
'id_habilla'	'positionne'	
'identifica'	'precisio_1'	

#### 6.7.4.5 Import/Export DXF et DWG

Pour gérer l'import/export de nouvelles classes et nouveaux champs en DXF et DWG, il faut modifier les scripts FME mis en place pour réaliser ces imports/export, ainsi que la configuration du plugin PCRS pour AutoCAD.

#### 6.7.4.6 Méthodologie générale

Pour ajouter des classes ou des champs sur des classes du standard CNIG PCRS v2.0, il faut écrire un script SQL qui va :

- ▶ ajouter des champs sur les classes existantes,
- ▶ créer de nouvelles classes,
- ▶ déclarer les nouveaux champs des classes existantes,
- ▶ déclarer les nouvelles classes et leurs champs,
- ▶ définir les règles d'import.

**! Ces ajouts de champs et classes peuvent se faire sur une base de données déjà existantes et contenant des données, mais il ne faut pas qu'il y ait des données exportées en attente de réimport (sinon les réimports seront très probablement rejetés).**

#### 6.7.4.7 Rappels sur l'architecture des données

- ▶ Chaque initiative a son propre schéma, contenant les couches de données (les classes).
- ▶ Pour chaque couche de données de ce schéma, il doit y avoir une couche "ARCH\_<couche>" qui contiendra une sauvegarde des données de la couche avant un import.

Cette couche doit avoir exactement la même structure attributaires que la couche originale, plus un attribut [processid] :

```
processid character varying(100) NOT NULL
```

- ▶ Les couches et les classes doivent être décrites dans des tables du schéma principal (nommé par défaut "pcrs").
- ▶ Les classes sont décrites dans la table pcrs.pcrs\_classe.
- ▶ Les champs des classes sont décrits dans la table pcrs.pcrs\_cls\_attribut.
- ▶ Les nomenclatures de certains champs sont décrites dans la table pcrs.pcrs\_cls\_domain. Les nomenclatures sont communes à toutes les initiatives.

Si l'on veut créer une nouvelle nomenclature, il faut lui donner un nom unique qui n'existe pas déjà dans le standard CNIG PCRS v2.0.

**! Un champ soumis à une nomenclature est forcément un champ de type caractères sur 10 caractères.**

**Toute déclaration incorrecte de classes, champs, nomenclature et règles de contrôle des imports, peut entraîner des comportements erronés ou des plantages de l'application.**

Avant tout ajout de classes ou de champs, il faut s'assurer que les informations qu'ils vont contenir, n'existent pas déjà sous un autre nom dans des classes existantes.

Il faut donc bien analyser l'existant dans le standard CNIG PCRS v2.0 et son implémentation dans arcOpole PRO PCRS.

#### 6.7.4.8 Limitations

Les couches des classes sont forcément d'un des types géométriques suivants (attention, ils doivent toujours avoir des Z, même si les Z ne sont pas forcément renseignés dans les données).

Type PCRS	Type postGIS
Ligne	LineStringZ
Polyligne	MultiLineStringZ
Point	PointZ
Multipoints	MultiPointZ
Polygone	PolygonZ
Multipolygone	MultiPolygonZ

Les noms des classes sont toujours en minuscules.

Les couches des classes ont forcément les attributs :

- **id** de type postgreSQL **serial** (il sert de clef unique numérique)
- **shape** de type geometry (voir pour des exemples plus loin)

Les champs des classes peuvent être des types suivants :

Type PCRS	Type postgresql
Caractères	varchar(255)
Caractères soumis à une nomenclature	varchar(10)
Entier	integer
Double	numeric
Date	timestamp without time zone

Les noms de champs sont toujours en minuscules.

#### 6.7.4.9 Exemples

Des exemples d'extension du modèle de données sont fournis dans le répertoire Install\construction\extension-sample.sql.

Dans les exemples suivants, on utilise les codes suivants. Ces codes sont à remplacer par leur valeur avant d'exécuter le script SQL (recherche/remplacer par Notepad++ par exemple) :

- \${pcrs\_schema} : nom du schéma principal (par défaut utiliser le nom : pcrs)
- \${initiative\_schema} : nom du schéma de l'initiative

- \${initiative\_id} : id de l'initiative (de 1 à N)
- \${initiative\_srid} : SRID de la référence spatiale des données de l'initiative

► Crédit d'une classe ou ajout de champs sur une classe existante :

Exemple de création d'une classe de polygones **TEST** avec des champs de chacun des types possibles (ainsi que de sa couche de sauvegarde **ARCH\_TEST**) :

```
DROP TABLE IF EXISTS ${initiative_schema}.test CASCADE;
CREATE TABLE ${initiative_schema}.test (
    id serial,
    texte_nomenclature varchar(10),
    texte_libre varchar(255),
    valeur_entiere integer,
    valeur_entiere_nomenclature integer,
    valeur_decimale numeric,
    date_test timestamp without time zone,
    shape geometry(PolygonZ,${initiative_srid})
);

ALTER TABLE ${initiative_schema}.test OWNER TO ${pcrs_schema};
CREATE INDEX idx_oid_test ON ${initiative_schema}.test USING btree (id);
CREATE INDEX idx_shp_test ON ${initiative_schema}.test USING gist (shape);

DROP TABLE IF EXISTS ${initiative_schema}.arch_test CASCADE;
CREATE TABLE ${initiative_schema}.Arch_test (
    processid character varying(100) NOT NULL,
    id integer NOT NULL,
    texte_nomenclature varchar(10),
    texte_libre varchar(255),
    valeur_entiere integer,
    valeur_entiere_nomenclature integer,
    valeur_decimale numeric,
    date_test timestamp without time zone,
    shape geometry(PolygonZ,${initiative_srid})
);

ALTER TABLE ${initiative_schema}.arch_test OWNER TO ${pcrs_schema};
CREATE INDEX idx_id_arch_test ON ${initiative_schema}.arch_test USING btree
(processid);
```

Exemple d'ajout d'un champ **NATURE** sur une classe standard PCRS existante **pointleve**

```
ALTER TABLE IF EXISTS ${initiative_schema}.pointleve
    ADD COLUMN nature character varying(10);

ALTER TABLE IF EXISTS ${initiative_schema}.arch_pointleve
    ADD COLUMN nature character varying(10);
```

► Déclaration des classes et de leurs champs dans les tables systèmes de PCRS :

Rappel : chaque initiative à un identifiant unique (\${initiative\_id}).

Chaque classe doit avoir un n° unique (un entier) qui permet de la déclarer dans la BD d'arcOpole PRO PCRS. Ce numéro doit être unique dans la table pcrs\_classe, toutes initiatives confondues.

Exemple :

Déclaration d'une nouvelle classe **test** dont le n° unique sera **150** :

```
insert into ${pcrs_schema}.pcrs_classe
(id, initiative_id, geometrie, nom, pcrs_nom, calque, editable)
values
(150, ${initiative_id}, 0, '${initiative_schema}.test', 'testPCRS', 'PCRS_TEST', 1),
```

**Signification des attributs de pcrs\_classe :**

id	Identifiant unique de la classe (de 1 à N)
initiative_id	Identifiant unique de l'initiative
geometrie	Code du type de géométrie de la couche :  0 : point 1 : ligne 2 : polygone 3 : multipoints 4 : multilignes 5 : multipolygones
nom	Le nom de la couche (incluant le nom du schéma de l'initiative)
pcrs_nom	Le nom de la couche dans les exports bruts (géodatabase, geopackage...)
calque	Le nom du calque correspondant à la couche dans les exports DWG ou DXF
editable	0 : les données sont non éditables -> elles sont exportées mais non réimportées 1 : les données sont éditables (elles seront importables)

- Déclaration des champs de la classe **test** dans **pcrs\_cls\_attribut** (remarque : le champ **id** ne doit pas être déclaré dans pcrs\_cls\_attribut).

```
Insert into ${pcrs_schema}.pcrs_cls_attribut
(class_name, nom, pcrs_nom, datatype, domain, defvalue, nonnulle, related)
values
('${initiative_schema}.test', 'texte_nomenclature', 'texte_nomenclature', 'texte', '1', '1', '1', '-1),
('${initiative_schema}.test', 'texte_libre', 'texte_libre', 'texte', '1', '1', '1', '-1),
('${initiative_schema}.test', 'valeur_entiere', 'valeur_entiere', 'integer', '5', '1', '1', '-1),
('${initiative_schema}.test', 'valeur_entiere_nomenclature', 'valeur_entiere_nomenclature', 'texte', '1', '1', '1', '-1),
('${initiative_schema}.test', 'valeur_decimale', 'valeur_decimale', 'real', '2', '1', '1', '-1),
('${initiative_schema}.test', 'date_test', 'date_test', 'date', '4', '1', '1', '-1),
```

**Signification des attributs de pcrs\_cls\_attribut :**

class_name	Nom de la couche du schéma de l'initiative
------------	--

nom	Nom du champ dans la couche (32 caractères maximum)
pcrs_nom	Nom du champ lorsqu'il est exporté (dans une gdb par exemple)
datatype	Code du type de champ : 0 : texte libre 1 : entier 2 : double 4 : date 5 : entier soumis à une nomenclature 6 : texte soumis à une nomenclature
domain	Le nom de la nomenclature à laquelle est soumis ce champ -> " si pas de nomenclature Les nomenclatures sont déclarées dans la table <b>pcrs_cls_domain</b>
defvalue	Valeur par défaut du champ sous forme de texte (même pour une valeur numérique)
nonnulle	0 : la valeur a le droit d'être null 1 : la valeur ne peut pas être null
related	Pour définir des relations entre cette classe et une autre -1 : ce champ ne définit pas une relation avec une autre classe Sinon c'est que ce champ est une relation vers une autre classe, et related doit avoir pour valeur l'identifiant de la classe (tel qu'il est déclaré dans pcrs_classe pour la classe liée)

► Déclaration d'une nouvelle nomenclature :

Par exemple, pour une nomenclature 'PCRSX\_NATUREOUVRAGEAERIENS' d'un champ numérique pouvant prendre les valeurs 0, 1, 2 ou 3 :

```
insert into pcrs.pcrs_cls_domain
(nom,                               code, valeur)
values
('PCRSX_NATUREOUVRAGEAERIENS', '0', 'autre'),
('PCRSX_NATUREOUVRAGEAERIENS', '1', 'bande transporteuse'),
('PCRSX_NATUREOUVRAGEAERIENS', '2', 'conduite aerienne'),
('PCRSX_NATUREOUVRAGEAERIENS', '3', 'supports ouvrages aeriens');
```

**Signification des attributs de pcrs\_cls\_domain :**

nom	Nom de la nomenclature  Ce nom est le code que l'on retrouve dans le champ pcrs_cls_attribut.domain
-----	---

code	Code de la valeur de cette nomenclature (10 caractères maximum)  Ce code est toujours une chaîne, mais pour un champ numérique, il ne doit contenir que des valeurs numériques entières
valeur	Libellé de la valeur de cette nomenclature

► Déclaration du subset auquel appartient la classe :

```
insert into pcrs.psrv_subset_classe
(classe_id, subset_name)
values
(150, 'nom du subset');
```

**Signification des attributs de *psrv\_subset\_classe* :**

classe_id	Nom de la couche du schéma de l'initiative
subset_name	Nom du subset auquel appartient cette classe  Les subsets sont déclarés dans la table <b>psrv_subsets</b>

► Déclaration des règles de contrôle des données d'une classe :

Ci-dessous un exemple de déclaration d'une règle, mais beaucoup d'autres choses peuvent être définies. Des compléments d'explications pourront être donnés par les équipes 1Spatial.

```
insert into pcrs.pcrs_import_rule
(initiative_id,
name,
message,
code,
classe,
payload,
status,
priorite,
valtype)
Values
(${initiative_id} ,
'Vérification des attributs obligatoires (Ouvrages aériens)' ,
'Valeur non définie',
8,
null,
'{"levelError" : 2,"classe" : 150,"fields" :
["thematique","qualitecategorisation","precisionplaniometrique","producteur","dateleve","nature","code","idhabillagesymbole"]}'::jsonb,
1 ,
8 ,
2);
```

**Signification des attributs de *pcrs\_import\_rule* :**

initiative_id	Identifiant unique de l'initiative
name	Nom de la règle
message	Message qui sera indiqué si la règle n'est pas respectée

code	Code numérique du type de règle d'import  -1 : règle externe (...)  0 : vérification de l'existence des classes  1 : vérification de l'existence des tables système  2 : vérification de la structure de tables système (RelationPCRS et EmpriseEchangePCRS)  3 : vérification du schéma des classes à importer  4 : vérification de l'emprise  5 : vérification de l'inclusion des données à importer dans l'emprise  6 : vérification des valeurs des domaines  7 : vérification des Z (altitude)  8 : vérification des attributs obligatoires  98 : vérification des relations entre affleurements et symboles  99 : vérification des relations entre classes
classe	Utilisation interne : mettre toujours à null
payload	Il s'agit d'une structure JSON qui définit des attributs spécifiques à chaque type de règle.
Status	0 : La règle inactive  1 : La règle arrête le processus uniquement dans le cas d'une erreur fatale (exception imprévue)  2 : La règle arrête le processus uniquement dans le cas d'une erreur bloquante (error)  3 : La règle arrête le processus uniquement dans le cas d'une erreur non bloquante (warning)
priorite	Utilisation interne : mettre toujours à null
valtype	0 : la règle s'applique sur les imports gestionnaires et les imports partenaire  3 : la règle s'applique uniquement sur les imports gestionnaire

## 6.8 CRÉATION DE SOUS-ENSEMBLES

Au préalable, distinguons une sous-classe d'un sous-ensemble : une sous-classe est une sorte de classe qui est générée à partir d'un filtre sur une classe «parente» (cf. chapitre 6.7.2, Définition de sous-classes dans le PCRS), alors qu'un sous-ensemble de classes est une liste de classes (pour n'exporter que certaines classes).

Par défaut, le sous-ensemble « pcrs » contient toutes les couches du MDD PCRS CNIG v2.0.

Il est possible de créer des nouveaux sous-ensembles pour permettre de n'exporter/importer que certaines classes seulement et ce, uniquement pour les formats DWG, GDB, SHP, GPKG, DXF.

Des sous-ensembles ne peuvent pas être créés pour le format GML du standard CNIG PCRS v2.0.

Voici un exemple de manipulation à réaliser dans pgAdmin pour créer un sous-ensemble nommé ici « pcrs\_CUD » :

- Au préalable, ouvrir la table pcrs\_classe ; noter les numéros des classes à gérer dans le sous-ensemble ; par exemple, ici les façades et les arbres :

Query Editor							
Data Output							
	<b>id</b> [PK] integer	<b>initiative_id</b> integer	<b>geometrie</b> integer	<b>nom</b> character varying (255)	<b>pcrs_nom</b> character varying (255)	<b>calque</b> character varying (255)	<b>editable</b> integer
1	102	1	0	initpcrs.pointgenerique	ObjetGeneriquePCRS_Point	PCRS_OBJET_GENERIQUE	1
2	103	1	4	initpcrs.ligne generique	ObjetGeneriquePCRS_Ligne	PCRS_OBJET_GENERIQUE	1
3	104	1	5	initpcrs.surfacegenerique	ObjetGeneriquePCRS_Surfa...	PCRS_OBJET_GENERIQUE	1
4	105	1	0	initpcrs.pointcanevas	PointCanevasPCRS	PCRS_POINT_CANEVAS	1
5	106	1	1	initpcrs.seuil	SeuilPCRS	PCRS_SEUIL	1
6	107	1	1	initpcrs.marcheescalier	MarcheEscalierPCRS	PCRS_MARCHE_ESCALIER	1
7	108	1	1	initpcrs.facade	FacadePCRS	PCRS_FACADE	1
8	109	1	1	initpcrs.proeminencebati	ProeminenceBatiPCRS	PCRS_PROEMINENCE_BATI	1
9	110	1	1	initpcrs.limitevoirie	LimiteVoiriePCRS	PCRS_LIMITE_VOIRIE	1
10	111	1	1	initpcrs.qualifluviomaritime	QualifluvioMaritimePCRS	PCRS_QUAL_FLUVIOMARITI...	1
11	112	1	1	initpcrs.rail	RailPCRS	PCRS_RAIL	1
12	113	1	1	initpcrs.qualrail	QualRailPCRS	PCRS_QUAL_RAIL	1
13	114	1	1	initpcrs.mur	MurPCRS	PCRS_MUR	1
14	115	1	0	initpcrs.pilierregulier	PilierRegulierPCRS	PCRS_PILIER_REGULIER	1
15	116	1	1	initpcrs.pilierparticulier	PilierParticulierPCRS	PCRS_PILIER_PARTICULIER	1
16	117	1	0	initpcrs.arbre	ArbrePCRS	PCRS_ARBRE	1
17							

- Création du jeu de classe pcrs\_CUD :

```
insert into pcrs.psrv_subsets
(initiative_id,      label,          name)
values
(1,                  'pcrs_CUD', 'pcrs_CUD');
```

- Rajout des classes : ici, par exemple, les façades et les arbres

```
insert into
pcrs.psrv_subset_classe
(classe_id,    subset_name)
values
(108,          'pcrs_CUD');

insert into
pcrs.psrv_subset_classe
(classe_id,    subset_name)
values
(117,          'pcrs_CUD');
```

- L'administrateur d'arcOpole PRO PCRS pourra alors mettre en place un modèle d'import/export ne contenant que certaines couches seulement basé sur ce nouveau sous-ensemble.

## 7 GESTION DES CARTES D'UNE INITIATIVE

### 7.1 PUBLICATION DES DONNÉES

Pour publier le service d'accès aux données d'une initiative PCRS, il faut :

- ▶ Créer une carte dans ArcGIS PRO ;
- ▶ Publier sa source de données ;
- ▶ La publier en tant que service de carte (Couche Web, Map Service, Map Image layer).

#### 7.1.1 Crédit d'une carte dans ArcGIS PRO

Deux solutions sont possibles. Soit créer une nouvelle carte et définir les représentations des couches, soit créer une carte en se basant sur le fichier install\resources\PCRS.mapx et utiliser les représentations prédéfinies.

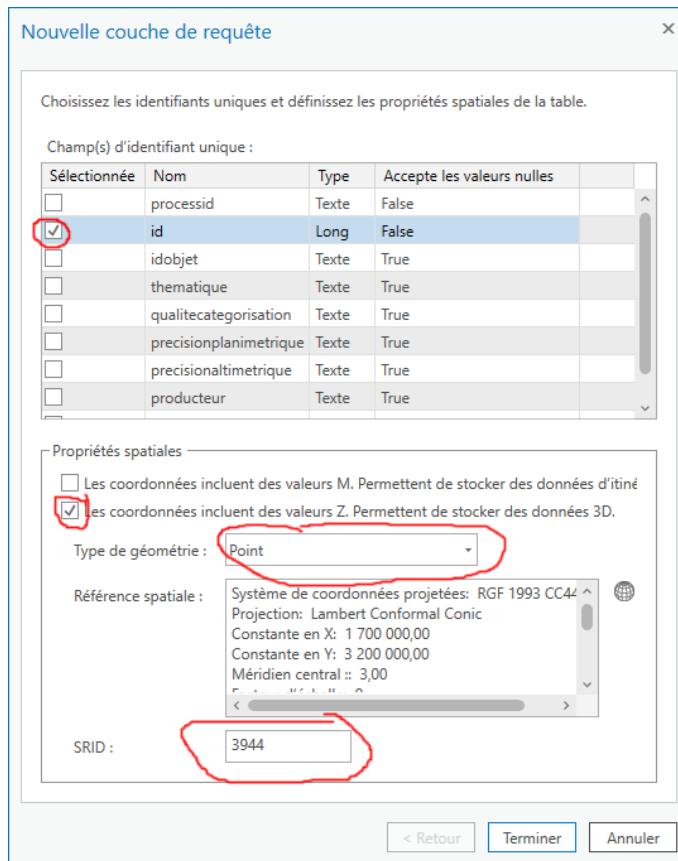
- ▶ Ouvrir ArcGIS PRO et créer un nouveau projet.
  - ! Il est fortement conseillé de conserver ce projet pour pouvoir simplement effectuer des modifications du service publié, sans avoir à tout refaire.**
- ▶ Définir une connexion vers la base PCRS à l'aide de l'utilisateur « pcrs » et de son mot de passe. Renommer cette connexion en « initiative.sde » par exemple ;

##### 7.1.1.1 Crédit une nouvelle carte

- ▶ Créer une nouvelle carte sans fond de plan : nous vous conseillons de numérotter manuellement les identifiants des couches ;
- ▶ Définir la référence spatiale (système de coordonnées) de la carte (la même que celle de l'initiative) ;
- ▶ Ajouter les couches du PCRS **une par une**, à partir du schéma de l'initiative, et définir leurs représentations.

Les couches du PCRS ne sont pas enregistrées dans la géodatabase ESRI. Il sera donc essentiel de définir les informations nécessaires dans l'interface présentée par ArcGIS Pro, lors de l'ajout de

chaque couche (si la couche ne contient pas de données, il faudra indiquer manuellement ses caractéristiques) :



- Les éléments surlignés sont à définir ;
- Le type de géométrie dépend de la couche (voir ci-dessous) ;
- La référence spatiale est celle choisie pour l'initiative.

Ci-dessous la liste des sources de données et le nom qu'il faut leur donner dans la carte (ce nom sera important lors de l'usage du script `configure_webmap.bat` qui permet de configurer les popups de la Carte Web (Map Service, Web Map) (cf. 7.4, Configuration de la popup de consultation)).

Source dans le schéma de l'initiative	Type de géométrie	Avec des Z ?	Nom de la couche dans la carte
arbre	Point	Oui	Arbre
commune	Polygon	Oui	Communes
facade	Polyline	Oui	Facade
ligneaffleurant	Polyline	Oui	Ligne Affleurant
lignegenerique	Polyline	Oui	Ligne Generique
lignehabillage	Polyline	Oui	Ligne Habilage

limitevoirie	Polyline	Oui	Limite de voirie
marcheescalier	Polyline	Oui	Marche
mur	Polyline	Oui	Mur
nomvoirie	Point	Oui	Nom de voirie
numerovoirie	Point	Oui	Numéro de voirie
pilepont	Polygon	Oui	Pile de pont
pilierparticulier	Polyline	Oui	Pilier particulier
pilierregulier	Point	Oui	Pilier Regulier
pointaffleurant	Point	Oui	Affleurant Point
pointcanevas	Point	Oui	Point Canevas
pointgenerique	Point	Oui	Point Générique
pointleve	Point	Oui	Point levé
pointshabillage	Multipoint	Oui	Points Habillage
proeminencebati	Polyline	Oui	Proeminence
quadrillage	Polygon	<b>Non</b>	Quadrillage
quaifluviomaritime	Polyline	Oui	Quai fluvio-maritime
quairail	Polyline	Oui	Quai Rail
rail	Polyline	Oui	Rail
seuil	Polyline	Oui	Seuil
surfaceaffleurant	Polygon	Oui	Surface Affleurant
surfacegenerique	Polygon	Oui	Surface générique
surfacehabillage	Polygon	Oui	Surface habillage
symboleaffleurant	Point	Oui	Symbole affleurant
symbolehabillage	Point	Oui	Symbole Habillage
textehabillage	Point	Oui	Texte Habillage
tronconvoirie	Polyline	Oui	Tronçon de voirie

**Couche des entreprises :** Pour cette couche, il faut l'ajouter en tant que query layer à partir du schéma pcrs, avec la requête suivante :

```
SELECT emprise.id,emprise.shape,emprise.name FROM pcrs.psrv_emprise AS emprise
INNER JOIN pcrs.psrv_initiative AS initiative ON emprise.id = initiative.emprise_id
WHERE initiative.id=${id_initiative}
```

Dans cet exemple, on considère que le schéma système se nomme pcrs et il faut remplacer \${id\_initiative} par l'identifiant de l'initiative de la carte (par exemple 1).

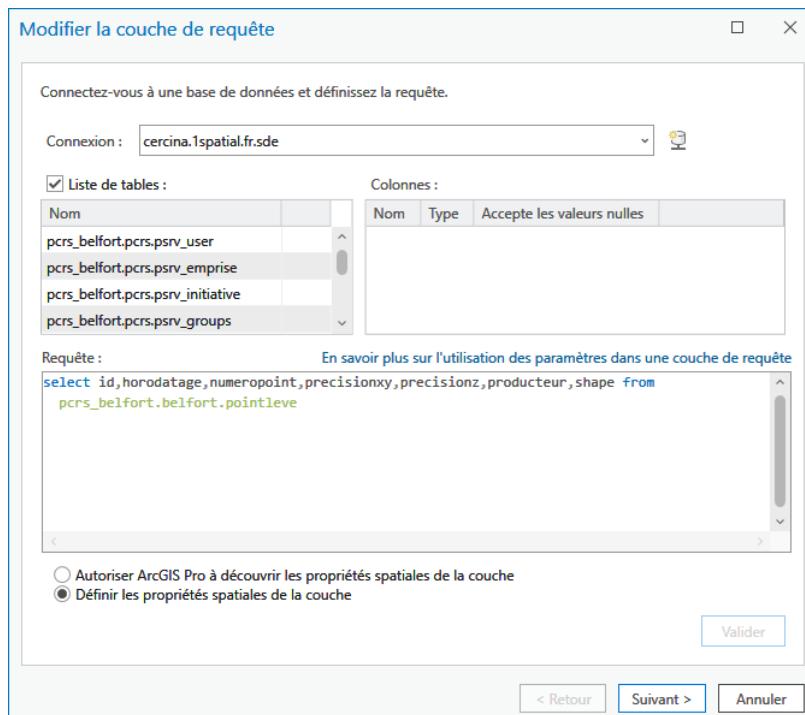
Source dans le schéma système (pcrs)	Type de géométrie	Avec des Z ?	Nom de la couche dans la carte
psrv_emprise	Polygon	Non	Emprise

- ▶ Définir la représentation de chaque couche ajoutée (échelle de visibilité, symbolologie, ...). Un fichier de style ArcGIS PRO est disponible dans les ressources d'installation du produit : « <Dossier d'installation>/install/resources/pcrs.stylx » ;
- ▶ Sauver le projet.

### 7.1.1.2 Créer une carte à partir d'un modèle mapx

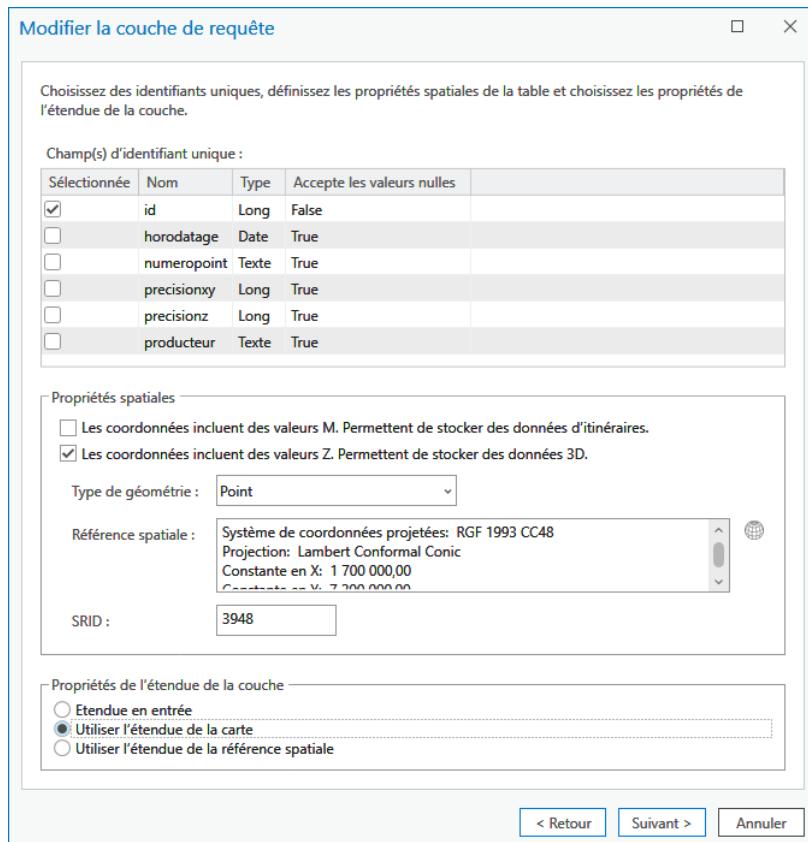
- ▶ Importer une carte à partir du modèle Install\resources\PCRS.mapx ;
- ▶Modifier la référence spatiale de la carte en utilisant celle de l'initiative ;
- ▶ Pour chaque couche, la reconnecter à la source de données de l'initiative (propriété de la couche, Source, requête -> bouton à droite de la requête) ;

Pour cela, se référer au tableau des couches indiqué précédemment.



- ▶ Choisir la connexion .sde
- ▶ Rechoisir la source de données dans le schéma de l'initiative

- Cocher "définir les propriétés spatiales de la couche" puis Valider, puis Suivant



- Suivant puis Terminer.

Répéter l'opération pour chaque couche

- Sauver le projet.

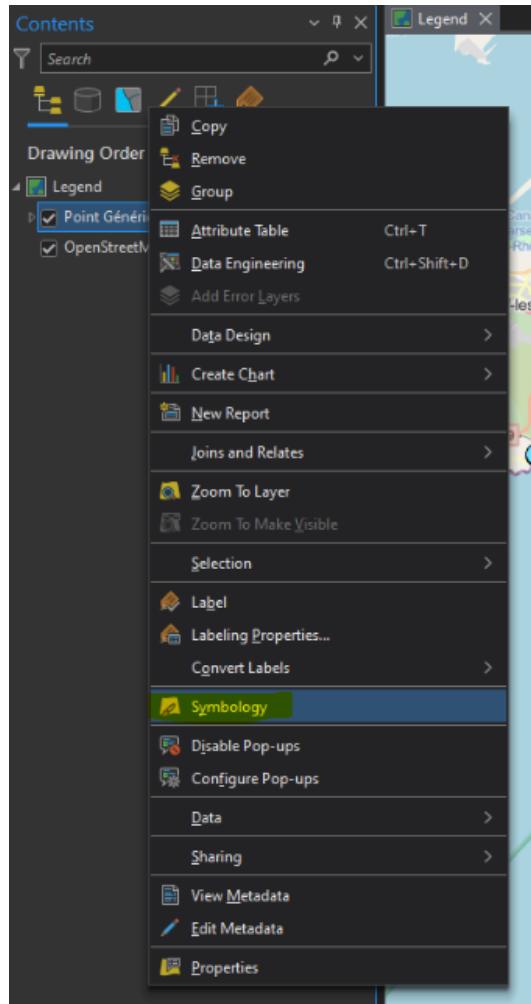
### 7.1.2 Modification de la légende de la carte PCRS

Afin d'appliquer une symbologie selon une analyse thématique sur une ou plusieurs couches, il est possible de modifier la légende de la carte PCRS.

En utilisant ArcGIS Pro, il faut choisir les paramétrages, les styles et les symbolologies des couches souhaitées sur la carte PCRS avant qu'elle ne soit publiée en tant que couche Web (map service, Web Map) sur le portal.

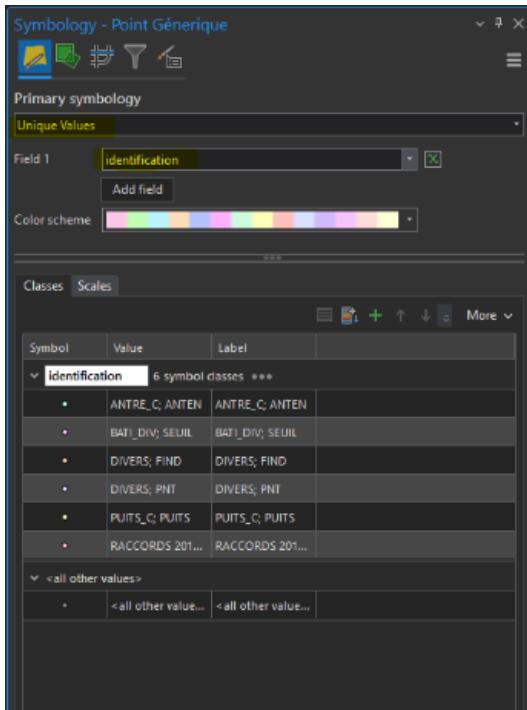
- Par exemple, ci-dessous la méthodologie utilisée pour appliquer une symbologie par valeur unique sur la couche "Point Générique" ;

- Après l'import des données à partir de la base SDE, effectuer un clic droit sur la couche afin d'accéder à l'interface de symbologie :

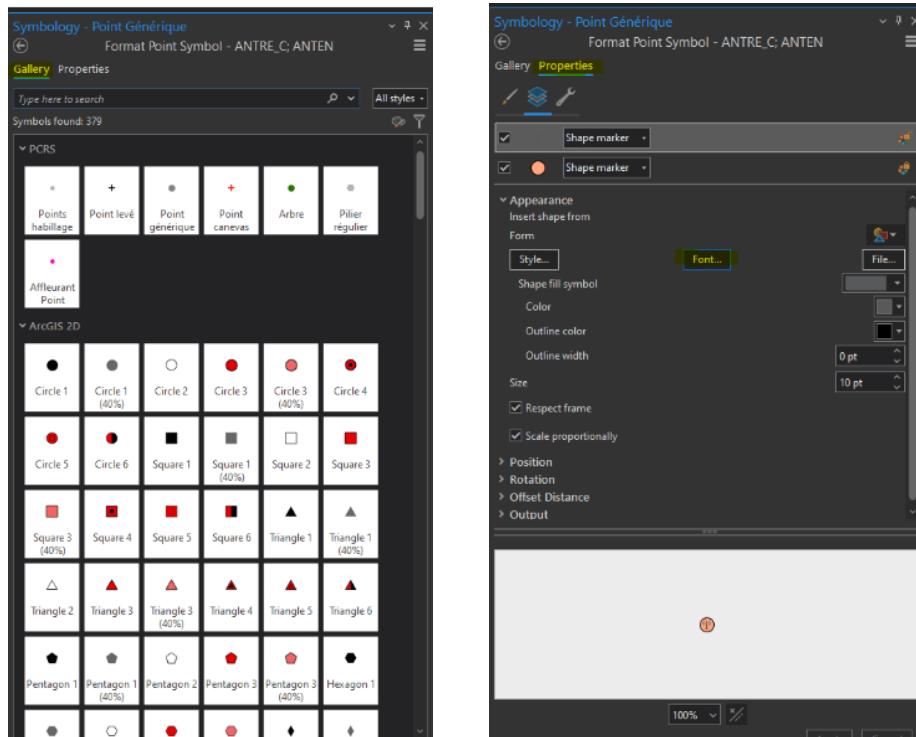


- Dans l'interface de symbologie, choisir le type de symbologie "Valeur Unique" et le champ à utiliser pour classifier les données. Ici, dans l'exemple, utilisation du champ "identification" ;

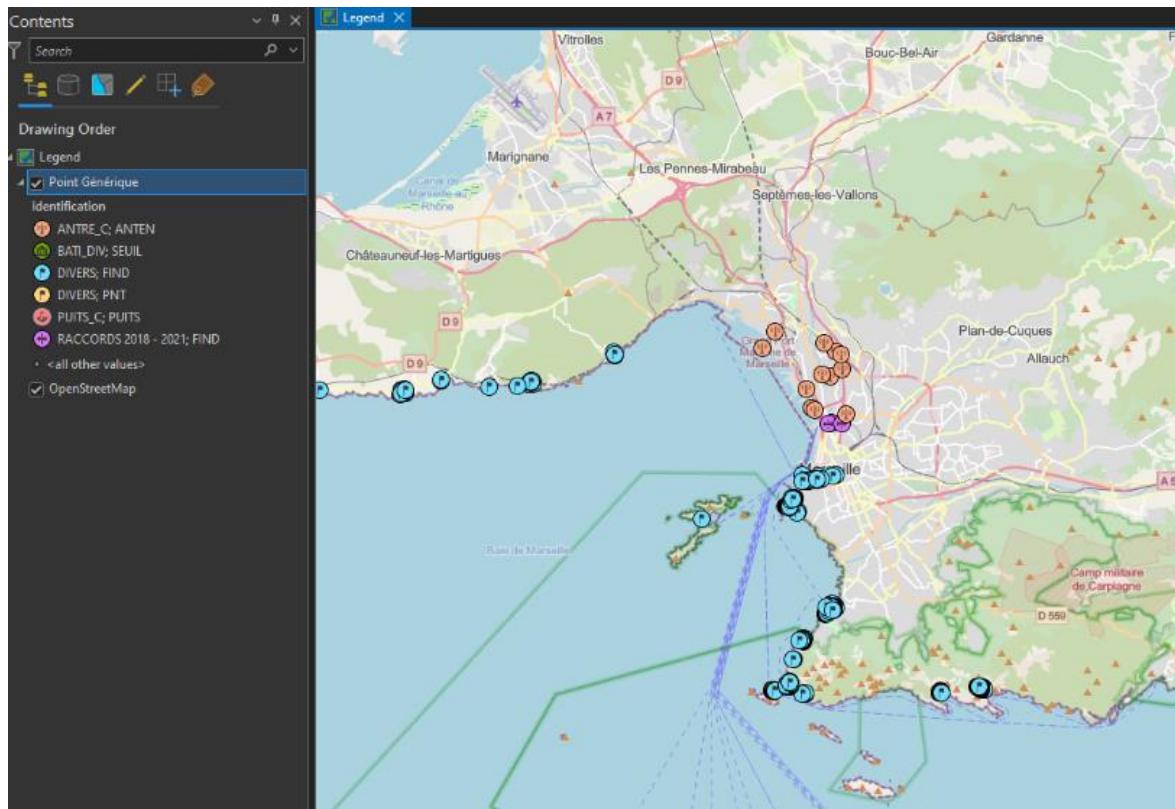
► Définir la symbologie souhaitée :



- Il est possible de choisir un symbole à partir de l'onglet galerie d'ArcGIS Pro ou d'importer une nouvelle police de caractères à partir de l'onglet propriétés et le bouton Polices. 1Spatial peut mettre en place cette police supplémentaire, définie selon le gabarit DAO de l'APLC.
- Il est également possible de changer la couleur, la taille, la transparence, etc, dans l'onglet propriétés :



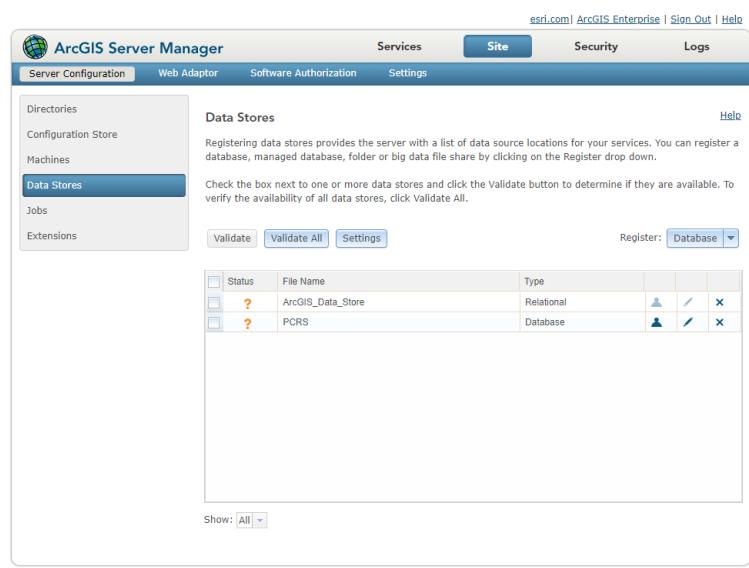
- Après la validation de la symbologie, la nouvelle classification s'affiche sur la carte :



- Il faut ensuite publier la carte sur le portal selon les indications des chapitres suivants.

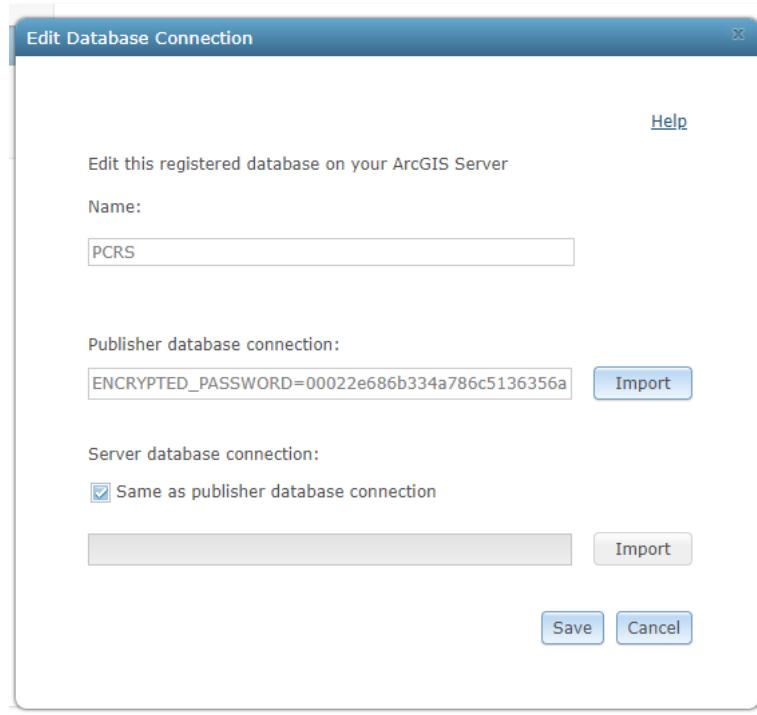
### 7.1.3 Publication de la source de données dans ArcGIS Server

- Ouvrir l'application Web ArcGIS Server Manager dans un navigateur ;
- Sélectionner l'onglet « Site / Data Stores » :



- Activer la commande « Register Database » ;
- Activer la commande « Import » ;

- ▶ Sélectionner le fichier « .sde » créé précédemment (initiative.sde) ;
- ▶ Valider :

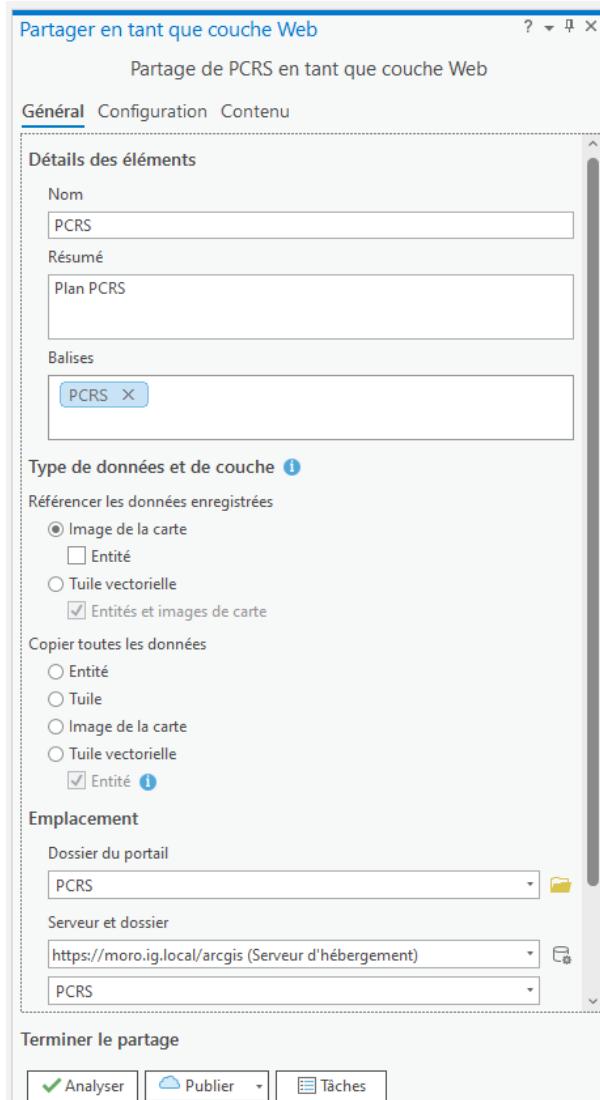


- ▶ Donner un nom au DataStore et valider l'ajout (Save).

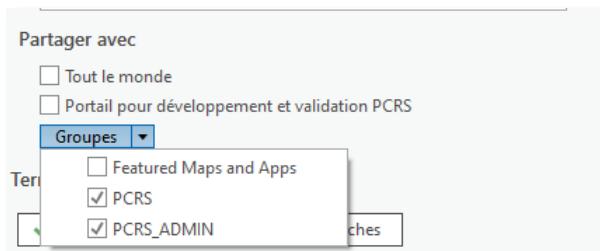
#### 7.1.4 Publication du service de carte

- ▶ Dans ArcGIS PRO, avec la carte construite précédemment, vérifier que le Portal courant est bien celui sur lequel publier le service ;
- ▶ Activer la commande « Partager / Couche Web / Publier la couche Web » ;

► Configurer la couche Web :



► Définir le partage de la couche Web avec les groupes PCRS :



► Activer la commande « Publier ».

## 7.2 CRÉATION DE LA CARTE WEB PRINCIPALE

L'application utilise deux définitions de carte Web (qui peuvent être différentes ou identiques). Ces deux cartes sont présentées dans deux parties de l'application :

- ▶ Lors de la sélection de zone à exporter ou à extraire pour modification ;
- ▶ Lors de la présentation d'erreur de données à importer.

Ces deux cartes sont définies dans la référence spatiale WGS-84 (3857) pour s'affranchir des références spatiales définies pour l'initiative.

La création de la ou des cartes Web de l'application suit le principe défini ci-dessous à l'aide de Portal for ArcGIS :

- ▶ Ouvrir l'application Web Portal for ArcGIS dans un navigateur ;
- ▶ Créer une nouvelle carte Web (Web Map);
- ▶ Définir le fond de plan par défaut de la carte. Ce fond de plan détermine la référence spatiale de la carte et doit donc être en WGS-84. Il peut s'agir par exemple de la carte « Open Street Map » ou d'une Ortho photo ;
- ▶ Ajouter la couche Web des données du PCRS de l'initiative (Map Service, Map Image Layer);
- ▶ Sauver la carte ;
- ▶ Partager la carte avec les groupes du PCRS (PCRS, PCRS\_ADMIN) ;
- ▶ Relever l'identifiant de la carte (dans l'url du navigateur qui l'affiche). Il sera utilisé pour la mise à jour de l'initiative au chapitre suivant.

## 7.3 MISE À JOUR DE L'INITIATIVE

Certaines modifications de la configuration de l'initiative ne sont pas prises en charge par l'application d'administration comme la gestion des emprises ou la définition des cartes de l'application. Ces opérations se font au travers de « PG-Admin » par l'exécution de requête SQL.

**! Pour exécuter des commandes SQL, cf. chapitre 11.5, Exécuter des commandes/scripts SQL à l'aide de pgAdmin.**

### 7.3.1 Modification de la carte de référence de l'initiative

```
UPDATE ${pcrs_schema}.psrv_initiative
SET map_id='<Identifiant de la carte Web>'
WHERE id=${initiative_id};
```

Avant d'exécuter ce code SQL par pgAdmin, remplacer :

- ▶ \${pcrs\_schema} par le nom du schéma principal « pcrs » ;
- ▶ \${initiative\_id} par l'identifiant de l'initiative.

L'identifiant de la carte Web est une chaîne de caractères obtenue dans l'url de consultation de la carte dans Portal.

**Par exemple : '99939c53f78644f0b321215b8e033f6c'**

### 7.3.2 Modification de la source RASTER de l'initiative

L'URL du service RASTER va permettre de fournir des extraits de l'ortho photo lors de l'export ou l'extraction de données d'une ou plusieurs cellules.

- ! Le service d'image RASTER doit obligatoirement être défini dans la même référence spatiale que celle de l'initiative.**

```
UPDATE ${pcrs_schema}.psrv_initiative
SET raster_dataset='<Url du service RASTER>'
WHERE id=${initiative_id};
```

Avant d'exécuter ce code SQL par pgAdmin, remplacer :

- ▶ \${pcrs\_schema} par le nom du schéma principal « pcrs » ;
- ▶ \${initiative\_id} par l'identifiant de l'initiative.

### 7.3.3 Ajout d'une emprise

Pour déclarer une emprise supplémentaire pour une initiative :

```
insert into ${pcrs_schema}.psrv_emprise (id, shape, name)
values (
${initiative_id},
'${geometry}',
'${entreprise_name}'
);
```

Avant d'exécuter ce code SQL par pgAdmin, remplacer :

- ▶ \${pcrs\_schema} par le nom du schéma principal « pcrs » ;
- ▶ \${initiative\_id} par l'identifiant de l'initiative ;
- ▶ \${entreprise\_name} : Nom de l'emprise de l'initiative (si ce nom contient le caractère « ' », il faut le doubler).
- ▶ \${geometry} : Géométrie de l'emprise de l'initiative, codée en geojson dans la référence spatiale principale **de l'application PCRS** (généralement 2154) qui sera définie dans les fichiers de configuration de celle-ci (cf. chapitre 8.2.3, pcrs.properties).

La géométrie de l'emprise est à fournir, sous la forme d'une chaîne json et de l'utilisation de fonction PostGIS. Exemple :

```
st_setsrid(st_geomfromgeojson('{"type":"Polygon","coordinates": [[[780000,6328000],
[780000,6345000],[793000,6345000],[793000,6328000],[780000,6328000]]]}}'),2154)
```

## 7.4 CONFIGURATION DE LA POPUP DE CONSULTATION

Après avoir publié le service de carte PCRS, il faut exécuter le script de configuration du popup de consultation qui se trouve sous le dossier « ressources\configure\_webmap » et qui permet de configurer le titre du popup, l'affichage des champs, les alias, l'ordre et les domaines.

- ▶ Ce dossier contient les fichiers suivants :
  - configure\_webmap.bat : le fichier qui permet l'exécution du traitement et qui contient les paramètres d'environnement et de connexion

- configure\_webmap.json : le fichier qui contient la liste de configuration des couches PCRS
- configure\_webmap.py : le script python principal.
- Pcrs\_messages.py
- Licence\_arcpy\_arcgisserver.bat
- README.txt

► Les prérequis :

- Python 3 d'ArcGIS (doit contenir la librairie Arcpy)
- Service de carte PCRS publié sur le portal ArcGIS

► Paramétrage du fichier « configure\_webmap.bat » : Dans le fichier batch, il faut changer les paramètres suivants :

- portal\_url : L'URL du portal qui contient le service de carte
- portal\_username : l'identifiant de l'administrateur du portal
- portal\_password : le mot de passe de l'administrateur du portal
- service\_name : le nom de la couche Web PCRS (Map Service, Map Image Layer)
- python\_path : Dossier du python 3 avec Arcpy installé.

► Exécuter le fichier configure\_webmap.bat

**! Attention ce fichier se base sur les noms des couches de la carte qui a permis de publier la couche Web. Si un des noms de couche ne correspond pas, la couche ne sera pas traitée et un message l'indiquera (et la couche n'aura pas sa définition de popup modifiée).**

## 8 INSTALLATION

- ! L'installation de FME et Python, l'accès à la base de données et les réglages APACHE sont décrits au chapitre 11, Annexes. À lire avant de démarrer l'installation.**
- L'installation du produit suppose un certificat SSL valide : cf. chapitre 11.7, Gestion des certificats.**

### 8.1 AJOUT DE LIBRAIRIES PYTHON

L'installation Python 3.6 64 bits au minimum incluant le module « arcpy » disponible sur le poste, doit être complétée avec des librairies supplémentaires nécessaires à l'application. Ces librairies sont les suivantes :

- ▶ LXML ;
- ▶ PSYCOPG2.

Pour installer ces deux librairies :

- ▶ Ouvrir une invite de commande DOS ;
- ▶ Se positionner dans le dossier d'installation Python où se trouve l'application « python.exe ». Par exemple :  
C:\Program Files\ArcGIS\Server\framework\runtime\ArcGIS\bin\Python\envs\arcgispro-py3 ;
- ▶ Installer LXML en tapant la commande :

```
python -m pip install lxml
```

- ▶ Installer PSYCOPG2 en tapant la commande :

```
python -m pip install psycopg2
```

### 8.2 CONFIGURATION DE L'APPLICATION

Les cinq fichiers de configuration de l'application se trouvent dans le dossier « <Dossier d'installation>/config ». Les éléments surlignés doivent être mis à jour.

#### 8.2.1 arcopolepro.properties

```
application.url=https://<nom du server PCRS>/pcrs-server
application.online.secret=<clé d'accès à ArcGIS Online>
application.online.clientId=<identifiant client ArcGIS Online>
application.portal.url=<Url du Portal for ArcGIS>
application.portal.clientId=<Identifiant de l'application PCRS sous Portal>
application.geometryServiceUrl=<Url ArcGIS
Server>/rest/services/Utilities/Geometry/GeometryServer
application.printServiceUrl=<Url ArcGIS
Server>/rest/services/Utilities/PrintingTools/GPSServer/Export Web Map Task
```

- ! Les identifiants « Online » sont facultatifs.**

## 8.2.2 backoffice.properties

```
backoffice.portal.user=<Identifiant portal de l'administrateur PCRS>
backoffice.portal.password=<Mot de passe de l'administrateur PCRS>
```

## 8.2.3 pcrs.properties

```
pcrs.geometry.mode=pg_geometry
pcrs.geometry.srid=<Référence spatiale de l'application PCRS : par défaut 2154>
pcrs.schema=<schéma principal de la base : pcrs par défaut>
pcrs.validation-map-id=<identifiant de la carte Web de validation d'import>
pcrs.validation-srid=<Référence spatiale de la carte : par défaut 3857>
pcrs.database.driverclassname=org.postgresql.Driver
pcrs.database.url=jdbc:postgresql://<hostname_database_server de la base
postgreSQL PCRS>:<port>/<base_name>
pcrs.database.username=<Identifiant du propriétaire du schéma PCRS : pcrs par
défaut>
pcrs.database.password=<Mot de passe du propriétaire du schéma PCRS>
pcrs.database.maximumPoolSize=8
```

## 8.2.4 process.properties

```
process.uploads-directory=<Dossier d'installation>\\temp\\uploads
process.workspaces-directory=<Dossier d'installation>\\temp\\processes
process.archives-directory=<Dossier d'installation>\\archives
process.keep-workspaces=false
```

La balise « process.keep-workspaces » affectée à « true » permet de conserver les dossiers de traitement pour recherche de bug.

## 8.2.5 tasks.properties

```
tasks.python.home=<Dossier d'installation du python avec arcpy>
tasks.python.exe=python.exe
tasks.python.scripts-directory=${arcopolepro.home}\\scripts
tasks.python.env[0]=arcopolepro.home=${arcopolepro.home}

tasks.java.home=<Dossier d'installation>\\server\\jdk-17.0.5
tasks.java.classpath=${arcopolepro.home}\\libs
tasks.java.options=-Xms256m -Xmx512m -Dfile.encoding=UTF-8 -
Darcopolepro.home=${arcopolepro.home}
tasks.java.libs-directory=${arcopolepro.home}\\libs
tasks.java.env[0]=GDAL_DATA=${arcopolepro.home}\\gdal-ogr\\bin\\gdal-data
tasks.java.env[1]=PROJ_LIB=${arcopolepro.home}\\gdal-ogr\\bin\\proj6\\share
tasks.java.env[2]=GDAL_DRIVER_PATH=${arcopolepro.home}\\gdal-
ogr\\bin\\gdal\\plugins
tasks.java.env[3]=Path+=;${arcopolepro.home}\\gdal-ogr\\bin

tasks.java.jdwp-port=7896

tasks.tasks-directory=<Dossier d'installation>\\temp\\tasks
tasks.libs-directory=${arcopolepro.home}\\libs
tasks.concurrency=2
tasks.keep-directories=true
tasks.timeout=90

...
```

**!** La suite du fichier de configuration décrit les différentes tâches de traitement. La configuration n'est utile que dans le cadre du développement ou de la recherche d'erreur. Elle ne doit pas être modifiée pour l'installation.

## 8.3 CONFIGURATION DU PROTOCOLE ET PORT D'ÉCOUTE TOMCAT

La configuration du protocole et du port d'écoute Tomcat pour l'application PCRS est fournie dans le fichier « <Dossier d'installation>/server/apache-tomcat-9.0.70/conf/server.xml ».

Dans le fichier par défaut, cette configuration est définie à la ligne 116 :

```
<Connector port="8049" protocol="AJP/1.3" redirectPort="8443"
           executor="tomcatThreadPool" keepAliveTimeout="20000"
           asyncTimeout="300000" connectionTimeout="300000"/>
```

Le protocole AJP est fortement recommandé. Le port utilisé (8049) est une convention par défaut pour éviter l'usage d'un même port par les produits ESRI et les autres applications de la gamme arcOpole PRO. Il est bien sur possible de le modifier si une autre application l'utilise déjà (attention, dans ce cas, à la configuration APACHE qui devra être adaptée).

## 8.4 CRÉATION DU SERVICE

- Ouvrir le fichier <Dossier d'installation>\server\apache-tomcat-9.0.70\bin\service.bat pour modifier ces variables d'environnement principales. Ces variables se trouvent au début du fichier (lignes 40 et 41) :

```
...
setlocal

set JAVA_HOME=<Dossier d'installation>\server\jdk-17.0.5
set ARCOPELPRO_HOME=<Dossier d'installation>\config

set "SELF=%~dp0%service.bat"
...
```

- Ouvrir une invite de commande en tant qu'administrateur du serveur ;
- Aller dans le dossier « <Dossier d'installation>\server\apache-tomcat-9.0.70\bin » ;
- Exécuter la commande suivante :
 

```
service.bat install
```
- Vérifier la présence du nouveau service PCRS dans la console des services du serveur ;
- Terminer l'installation avant de démarrer le serveur.

## 8.5 CONFIGURATION DE DÉPLOIEMENT APACHE

- Ouvrir le fichier de configuration de l'installation APACHE « <Dossier d'installation APACHE>/conf/extrahtdp-ssl.conf » et définir les paramètres suivants :

```
# REWRITE
RewriteEngine On

#Rewrite PCRS
RewriteRule ^/pcrs-server/public/(client|backoffice) /(?:[a-zA-Z0-9]+)(?:/[a-zA-Z0-9]+)*$ /pcrs-server/public/$1/ [R]
RewriteRule ^/pcrs-server/public/(client|backoffice) /$ ajp://localhost:8049/pcrs-server/public/$1/index.html [P]
```

```
#Proxy pour arcOpole PRO PCRS
ProxyPass /pcrs-server/ ajp://localhost:8049/pcrs-server/ ttl=60 timeout=120
ProxyPassReverse /pcrs-server/ ajp://localhost:8049/pcrs-server/
```

- ! **Il est à noter que pour pouvoir utiliser le protocole « ajp », il faut que la ligne « LoadModule proxy\_ajp\_module modules/mod\_proxy\_ajp.so » soit décommentée (pas de '#' dans httpd.conf).**

## 8.6 DÉMARRAGE

- ▶ Ouvrir la console de gestion des services Windows du serveur où aOP PCRS est installée ;
- ▶ Sélectionner le service « ArcOpole PRO PCRS Server » ;
- ▶ Ouvrir les propriétés du service pour configurer le démarrage automatique ;
- ▶ Cliquer sur Démarrer.
  - ! **Lors du premier démarrage du serveur, il est recommandé de vérifier le log de celui-ci (<Dossier d'installation>/server/apache-tomcat-9.0.70/logs/arcopolepropcrsserver-stderr <date>.log) pour voir si aucune anomalie n'est relevée lors du démarrage.**

## 9 CHARGEMENT DE LA BASE

Le chargement de la base va dépendre des sources de données disponibles. On pourra, par exemple, disposer de certaines couches dans une GDB Fichier ESRI (par exemple, les communes ou les tronçons de voirie). D'autres données pourront être fournies par un gestionnaire externe (par exemple, ses affleureurs ou un fond de plan PCRS déjà existant) au format PCRS (GML).

L'application d'administration va permettre le chargement de ces différents types de données dans la base de l'application en définissant des modèles d'import « gestionnaire », puis en les utilisant pour importer les données.

- ! La fonction d'import « gestionnaire » remplace (par défaut), l'intégralité des données des couches indiquées comme modifiables par le modèle.**

### 9.1 IMPORT DEPUIS UNE GDB FICHIER

L'installateur dispose des couches « Commune » et « TronconVoirie » de son emprise sous la forme de couches de données d'une GDB Fichier ESRI.

- ! La base doit comporter la définition intégrale d'une GDB Fichier d'échange PCRS pour être considérée comme une source d'import valide.**
- Tous les attributs des couches doivent être initialisés.**

Commune X						
Champ :	Sélection :	En surbrillance :				
SHAPE *	OBJECTID *	id	codeinsee	nom	reference	SHAPE_Length SHAPE_Area
Cliquez pour ajouter une nouvelle ligne.						

TronconVoirie X						
Champ :	Sélection :	En surbrillance :				
SHAPE *	OBJECTID *	id	propriete	source	reference	SHAPE_Length
Cliquez pour ajouter une nouvelle ligne.						

Pour l'importer dans sa base, l'installateur va effectuer les opérations suivantes :

- Ouvrir l'application d'administration PCRS avec l'URL suivante :  
« <https://<server>/pcrs-server/public/backoffice/> »
- S'authentifier avec l'identifiant portal déclaré comme administrateur de l'application PCRS ;
- Sur le panneau de gauche, sélectionner l'entrée « Modèles » ;

- Activer la commande  en haut à droite du panneau de droite.

 arcOpole **PRO** Administration  
PCRS

Administrateur PCRS 

Initiatives

Utilisateurs

**Modèles**

▼ Plans

Tous

Attendus

En retard

À valider

Carte

Gestionnaire

▼ Imports

Tous

Gestionnaire

► Activité

Initiative \*

Nom du modèle \*

Import GDB Fichier

Description \*

Import commune et tronçon de voirie

Emprise

Format \*

GDB

Joindre un fond image

Validation manuelle obligatoire

Gestionnaire

Cumuler les données existantes

Jeu de classes \*

pcrs

Gabarit

Échanges DWG/DXF seulement

Annuler

**Créer**

Copyright : © 2019-2024 - 1spatial SAS. Tous droits réservés.



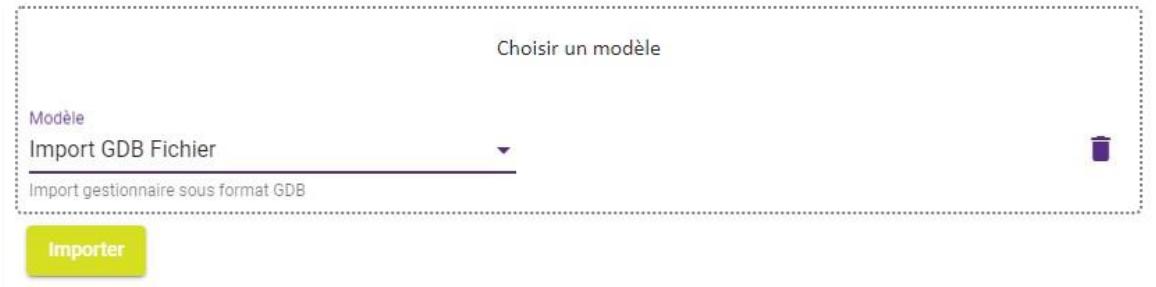
- ▶ Définir les paramètres du modèle d'import (cocher gestionnaire) ;
  - ▶ Activer la commande « Créer » :

Modèle		
Initiative	Alès	Format
Nom	Import GDB Fichier	Racine
Description	Import commune et tronçon de voirie	Sous-ensemble
Emprise		Raster
Utilisateurs		Validation
		Gestionnaire
		Lecture seule

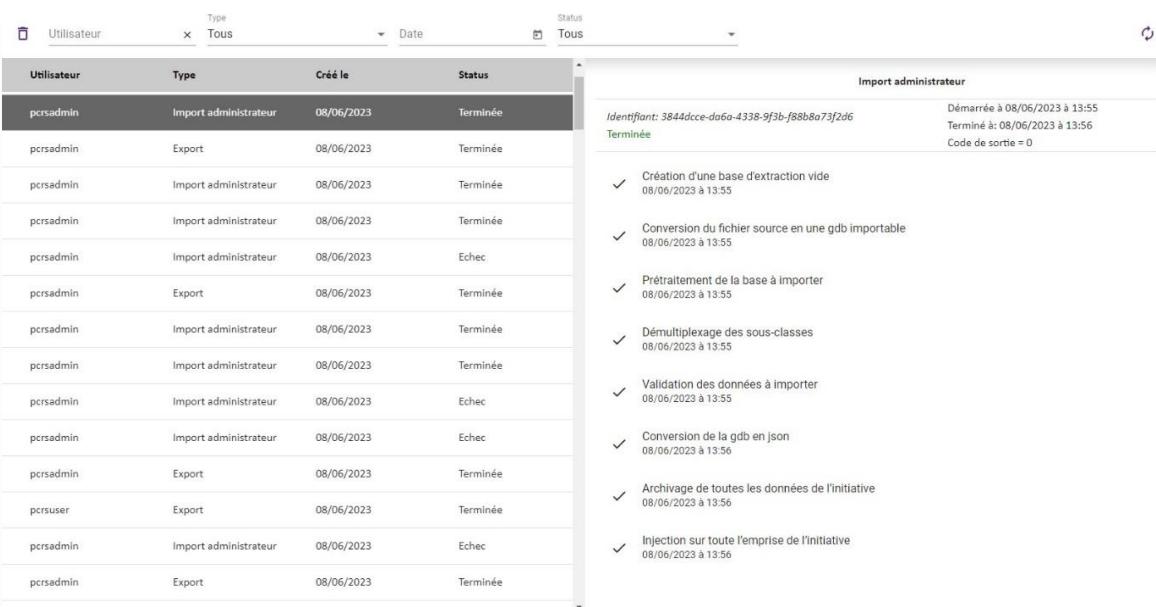
- ▶ Activer la commande « Modifier » pour définir les utilisateurs du modèle ;
  - ▶ Définir l'administrateur comme utilisateur ;
  - ▶ Activer la commande « Modifier » des couches en lecture seule ;
  - ▶ Cocher les classe « Commune » et « TronconVoirie » :

- Dans le panneau de gauche, déplier la zone « Imports » et sélectionner « Gestionnaire » ;
  - Glisser-Déposer un fichier zip contenant la base source.  
**!** La base GDB Fichier doit être renommée en « import.gdb » avant d'être compressée dans un fichier zip. C'est le dossier « import.gdb » qui doit être compressé et pas seulement son contenu.

- Définir le modèle d'import :



- Activer la commande « Importer ». La page d'avancement du traitement s'affiche :



Utilisateur	Type	Créé le	Status
pcrsadmin	Import administrateur	08/06/2023	Terminée
pcrsadmin	Export	08/06/2023	Terminée
pcrsadmin	Import administrateur	08/06/2023	Terminée
pcrsadmin	Import administrateur	08/06/2023	Echec
pcrsadmin	Export	08/06/2023	Terminée
pcrsadmin	Import administrateur	08/06/2023	Terminée
pcrsadmin	Import administrateur	08/06/2023	Terminée
pcrsadmin	Import administrateur	08/06/2023	Echec
pcrsadmin	Import administrateur	08/06/2023	Echec
pcrsadmin	Export	08/06/2023	Terminée
pcruser	Export	08/06/2023	Terminée
pcrsadmin	Import administrateur	08/06/2023	Echec
pcrsadmin	Export	08/06/2023	Terminée

**Import administrateur**

Identifiant: 3844dcce-da6a-4338-9f3b-f88b8a73f2d6  
Terminée

Démarrée à 08/06/2023 à 13:55  
Terminé à: 08/06/2023 à 13:56  
Code de sortie = 0

- ✓ Création d'une base d'extraction vide  
08/06/2023 à 13:55
- ✓ Conversion du fichier source en une gdb importable  
08/06/2023 à 13:55
- ✓ Prétraitement de la base à importer  
08/06/2023 à 13:55
- ✓ Démultiplexage des sous-classes  
08/06/2023 à 13:55
- ✓ Validation des données à importer  
08/06/2023 à 13:55
- ✓ Conversion de la gdb en json  
08/06/2023 à 13:56
- ✓ Archivage de toutes les données de l'initiative  
08/06/2023 à 13:56
- ✓ Injection sur toute l'emprise de l'initiative  
08/06/2023 à 13:56

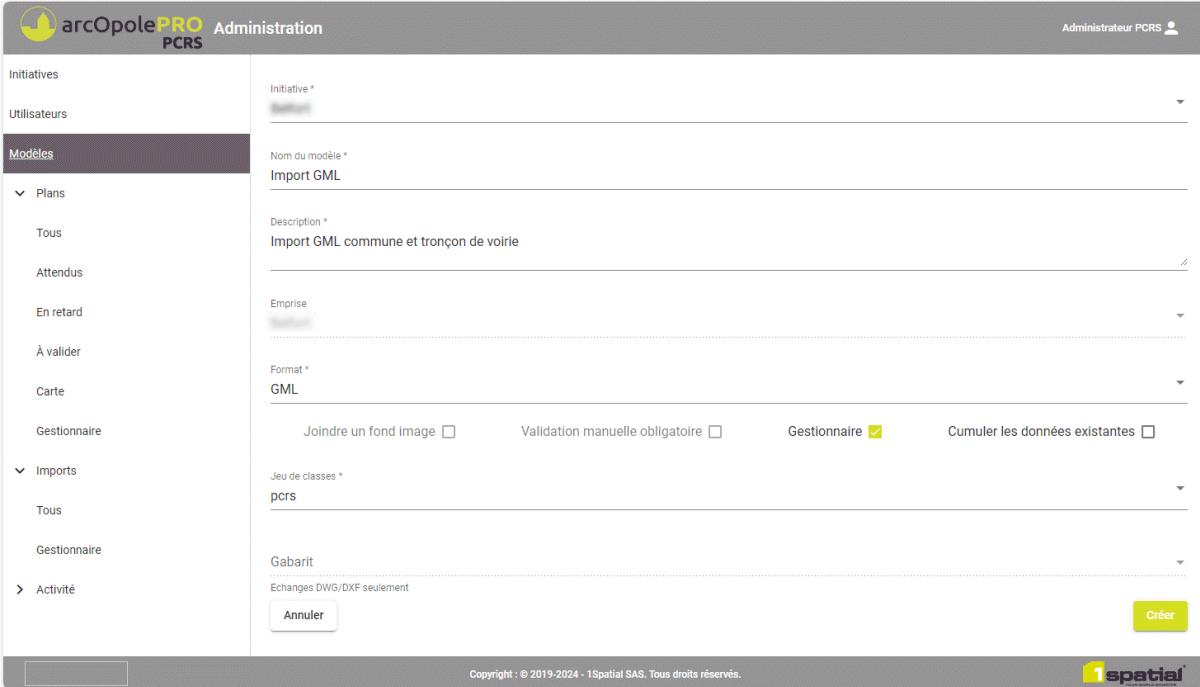
## 9.2 IMPORT DEPUIS UN FICHIER GML CONFORME CNIG

L'installateur dispose d'un fichier GML contenant toutes les données au format PCRS.

Pour l'importer dans sa base, il va effectuer les opérations suivantes :

- Ouvrir l'application d'administration PCRS avec l'URL suivante :  
**« <https://<server>/pcrs-server/public/backoffice/> »**
- S'authentifier avec l'identifiant portal déclaré comme administrateur de l'application PCRS ;
- Sur le panneau de gauche, sélectionner l'entrée « Modèles » ;

- Activer la commande  en haut à droite du panneau de droite :



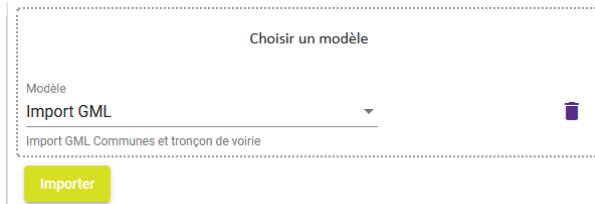
The screenshot shows the 'Administration' screen of the software. On the left, there's a sidebar with categories like 'Initiatives', 'Utilisateurs', 'Modèles' (which is selected and highlighted in dark grey), 'Plans', 'Tous', 'Attendus', 'En retard', 'À valider', 'Carte', 'Gestionnaire', 'Imports', 'Tous', 'Gestionnaire', and 'Activité'. The main panel on the right has fields for 'Initiative' (set to 'Belfort'), 'Nom du modèle' (set to 'Import GML'), 'Description' (set to 'Import GML commune et tronçon de voirie'), 'Format' (set to 'GML'), 'Jeu de classes' (set to 'pcrs'), and 'Gabarit' (set to 'Echanges DWG/DXF seulement'). There are also checkboxes for 'Validation manuelle obligatoire', 'Gestionnaire' (which is checked), and 'Cumuler les données existantes'. At the bottom right of the panel is a yellow 'Créer' (Create) button.

- Définir les paramètres du modèle d'import (cocher Gestionnaire) ;  
► Activer la commande « Crée » ;  
► Activer la commande « Modifier » pour définir les utilisateurs du modèle ;  
► Définir l'administrateur comme utilisateur ;  
► Activer la commande « Modifier » des couches en lecture seule ;  
► Cocher toutes les classes « Commune » et « TronconVoirie » :

Modèle	
<i>Initiative</i>	Belfort
<i>Nom</i>	Import GML
<i>Description</i>	Import GML Communes et tronçon de voirie
<i>Emprise</i>	
<i>Utilisateurs</i>	
pcradmin	
	
<i>Format</i>	GML
<i>Gabarit</i>	
<i>Sous-ensemble</i>	pcrs
<i>Raster</i>	Non
<i>Validation</i>	Non
<i>Gestionnaire</i>	Oui
<i>Classes éditable (2)</i>	
	
Commune	
TronconVoirie	
	

- Dans le panneau de gauche, déplier la zone « Imports » et sélectionner « Gestionnaire » ;  
► Glisser-Déposer le fichier zip contenant le fichier gml.  
! **Le fichier GML doit être renommé en « import.gml » avant d'être compressé dans un fichier zip.**

► Définir le modèle d'import :



► Activer la commande « Importer ». La page d'avancement du traitement s'affiche.

Utilisateur	Type	Créé le	Status	Import administrateur
pcrsadmin	Import administrateur	08/06/2023	Terminée	Identifiant: 3844dcce-da6a-4338-9f3b-f88b8a73f2d6 Terminée
pcrsadmin	Export	08/06/2023	Terminée	Démarrée à 08/06/2023 à 13:55 Terminé à: 08/06/2023 à 13:56 Code de sortie = 0
pcrsadmin	Import administrateur	08/06/2023	Terminée	✓ Création d'une base d'extraction vide 08/06/2023 à 13:55
pcrsadmin	Import administrateur	08/06/2023	Echec	✓ Conversion du fichier source en une gdb importable 08/06/2023 à 13:55
pcrsadmin	Export	08/06/2023	Terminée	✓ Prétraitement de la base à importer 08/06/2023 à 13:55
pcrsadmin	Import administrateur	08/06/2023	Terminée	✓ Démultiplexage des sous-classes 08/06/2023 à 13:55
pcrsadmin	Import administrateur	08/06/2023	Echec	✓ Validation des données à importer 08/06/2023 à 13:55
pcrsadmin	Import administrateur	08/06/2023	Echec	✓ Conversion de la gdb en json 08/06/2023 à 13:56
pcrsadmin	Export	08/06/2023	Terminée	✓ Archivage de toutes les données de l'initiative 08/06/2023 à 13:56
pcrsadmin	Import administrateur	08/06/2023	Echec	✓ Injection sur toute l'emprise de l'initiative 08/06/2023 à 13:56
pcrsadmin	Export	08/06/2023	Terminée	

## 10 CONDITIONS GÉNÉRALES D'UTILISATION

Les conditions générales d'utilisation présentées dans l'application au moment de l'extraction de données PCRS doivent être mises à jour selon les règles propres à chaque Autorité Publique Locale Compétente (APLC) au moment de déploiement de l'application.

Il suffit de remplacer le texte standard fourni par défaut avec l'application par celui de l'APLC. Le texte initial est un fichier au format HTML, mis en place dans le dossier de l'application :

« <Dossier d'installation>\server\apache-tomcat-9.0.70\webapps\pcrs-server\WEB-INF\classes\public\client\assets ».

## 11 ANNEXES

### 11.1 INSTALLATION FME

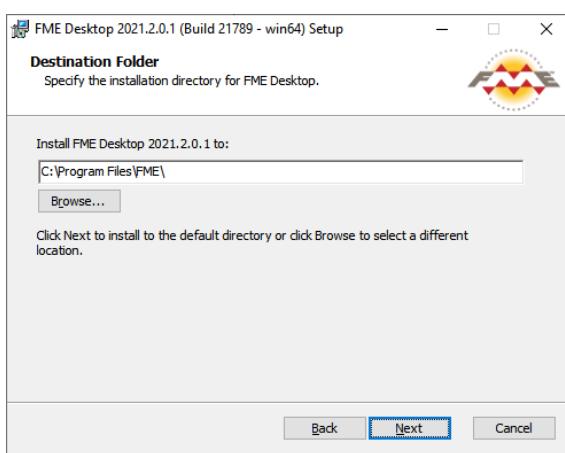
- Activer l'installateur du produit par un double-clic « fme-desktop-2021.2.0.1-b21789-win-x64.exe » :



- Activer la commande « Install » :



- Définir le dossier d'installation :



- Enregistrer la licence du produit.

**! Si FME n'a pas été installé dans le chemin par défaut c:\program files\FME\fme.exe, il faudra modifier le fichier : « <Dossier d'installation>/config/scripts/pcrs\_ParamConfig.py ».**

**Il contient la ligne :**

```
g_strPathFME_EXE = r"C:\Program Files\FME\fme.exe"
```

**dans laquelle il faut remplacer le chemin par défaut, par le chemin réel du fichier fme.exe.**

## 11.2 INSTALLATION PYTHON

L'installation de Python 3.6 64 bits au minimum et du module « arcpy » correspondant est réalisée lors de l'installation standard d'ArcGIS Server. Se reporter à la documentation correspondante pour plus d'informations.

Une fois l'installation terminée, si ArcGIS Server n'a été mis en œuvre que pour l'usage de Python, il est recommandé d'arrêter le service « ArcGIS Server » et d'en désactiver le démarrage automatique dans la console Windows d'administration des services.

## 11.3 GESTION DES CONFLITS DE VERSIONS PYTHON

Il peut arriver d'avoir des conflits de versions de python et du module « arcpy » lors de l'exécution des traitements si plusieurs produits ESRI différents sont installés sur le serveur.

- ArcGIS Server (jusqu'en 10.9.1) installe deux versions de python (une 2.7 en 64 bits et un 3.x en 64 bits). Ces deux versions cohabitent normalement sans problème ;
- ArcMAP installe une version python 2.7 en 32 bits ;
- ArcMAP peut être complété par une version python 2.7 en 64 bits ;
- ArcGIS PRO installe une version 3.x en 64 bits.

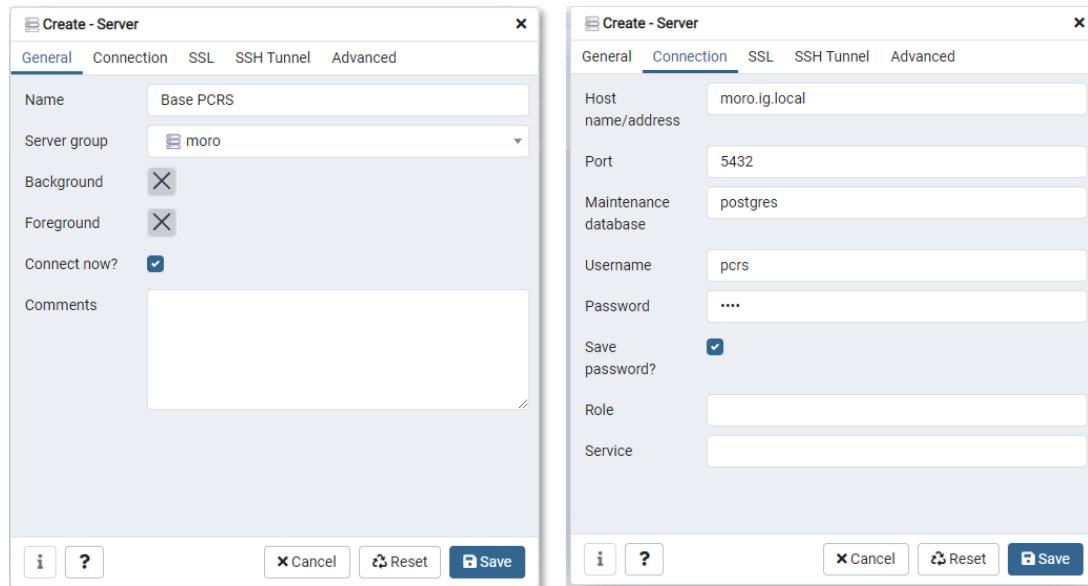
Ce type de problème peut provenir de différentes sources :

- Présence d'une variable d'environnement système « PYTHONPATH » qui indique une installation de python précise ;
- Présence d'information sur une installation python particulière dans le chemin de recherche du système ou, s'il y en a plusieurs, l'ordre dans lequel ces chemins sont définis ;
- Association de type de fichier avec l'exécutable leur correspondant ;
- Problème d'installation d'ArcGIS Server.

## 11.4 ÉTABLIR UNE CONNEXION AVEC « PGADMIN »

- Démarrer « pgAdmin » ;
- Activer la commande « Create Server » ;

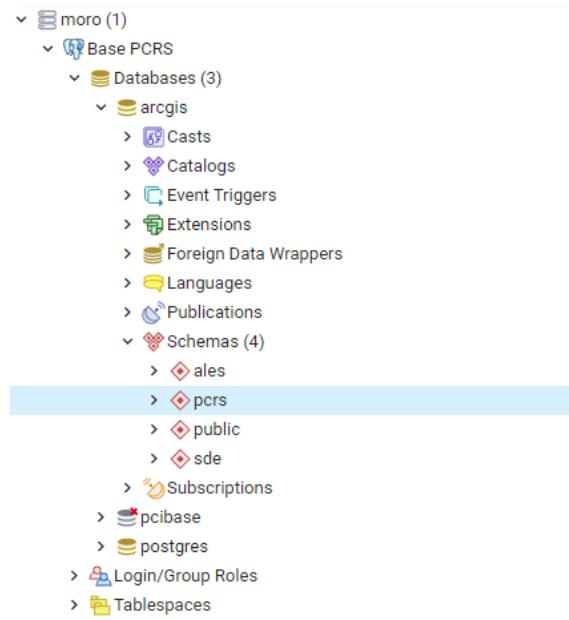
- Définir les informations de connexion dans les onglets « General » et « Connection » du dialogue ci-dessous :



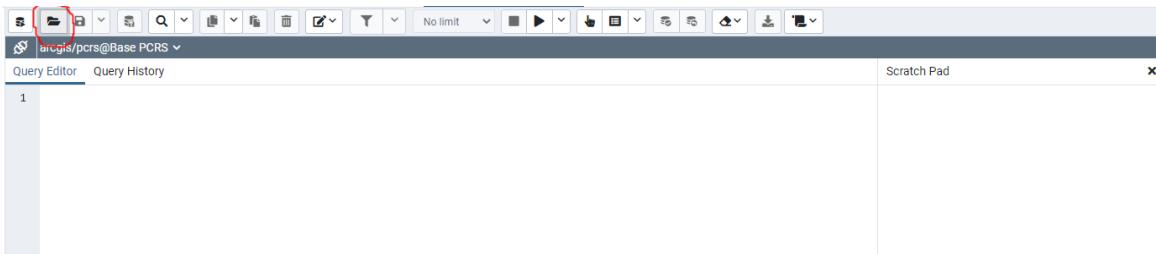
- Sauver la définition.

## 11.5 EXÉCUTER DES COMMANDES/SCRIPTS SQL À L'AIDE DE PGADMIN

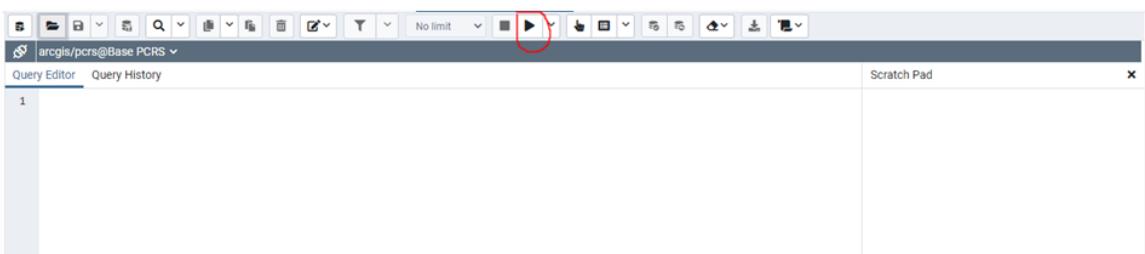
- Ouvrir la connexion à la base (voir le chapitre précédent pour la définir) ;  
 ► Parcourir l'arborescence et sélectionner le schéma souhaité :



- Faire apparaître le menu contextuel (clic droit) et activer la commande « Query Tool » ;  
 ► Pour exécuter un script sql, dans le bandeau de commande de la nouvelle fenêtre, activer la commande « Open File » et sélectionner le fichier à exécuter. Celui-ci apparaît dans la zone de saisie de script de la fenêtre :



- ▶ Pour exécuter des commandes, taper directement les commandes SQL dans cette fenêtre ;
- ▶ Activer la commande « Execute / Refresh » :



- ▶ Vérifier le résultat dans la console, en dessous de la zone de saisie de script.

## 11.6 RÉGLAGES APACHE

Certains réglages sont recommandés dans le cas d'une application en production :

- ▶ Ouvrir le fichier de configuration de l'installation APACHE « <Dossier d'installation APACHE>/conf/httpd.conf » et définir les paramètres suivants :

```
EnableMMAP off
EnableSendfile on

AcceptFilter http none
AcceptFilter https none

KeepAlive On
KeepAliveTimeout 20
```

- ▶ Ouvrir le fichier de configuration de l'installation APACHE « <Dossier d'installation APACHE>/conf/extra/httpd-ssl.conf » et définir les paramètres suivants :

```
<Location "/pcrs-server/">
    # Gestion de la compression
    SetOutputFilter DEFLATE

    AddOutputFilterByType DEFLATE text/html text/plain text/xml text/x-js
    text/javascript text/css
    AddOutputFilterByType DEFLATE application/xml application/xhtml+xml
    application/x-javascript application/javascript
    AddOutputFilterByType DEFLATE application/json

    SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-vary
    SetEnvIfNoCase Request_URI \.(?:exe|t?gz|zip|bz2|sit|rar)$ no-gzip dont-vary
    SetEnvIfNoCase Request_URI \.pdf$ no-gzip dont-vary

</Location>

# REWRITE
RewriteEngine On
```

```
#Rewrite PCRS
RewriteRule ^/pcrs-server/public/(client|backoffice) /(?:[a-zA-Z0-9]+)(?:/[a-zA-Z0-9]+)*$ /pcrs-server/public/$1/ [R]
RewriteRule ^/pcrs-server/public/(client|backoffice) /$ ajp://localhost:8049/pcrs-server/public/$1/index.html [P]

#Proxy pour arcOpole PRO PCRS
ProxyPass /pcrs-server/ ajp://localhost:8049/pcrs-server/ ttl=60 timeout=120
ProxyPassReverse /pcrs-server/ ajp://localhost:8049/pcrs-server/
```

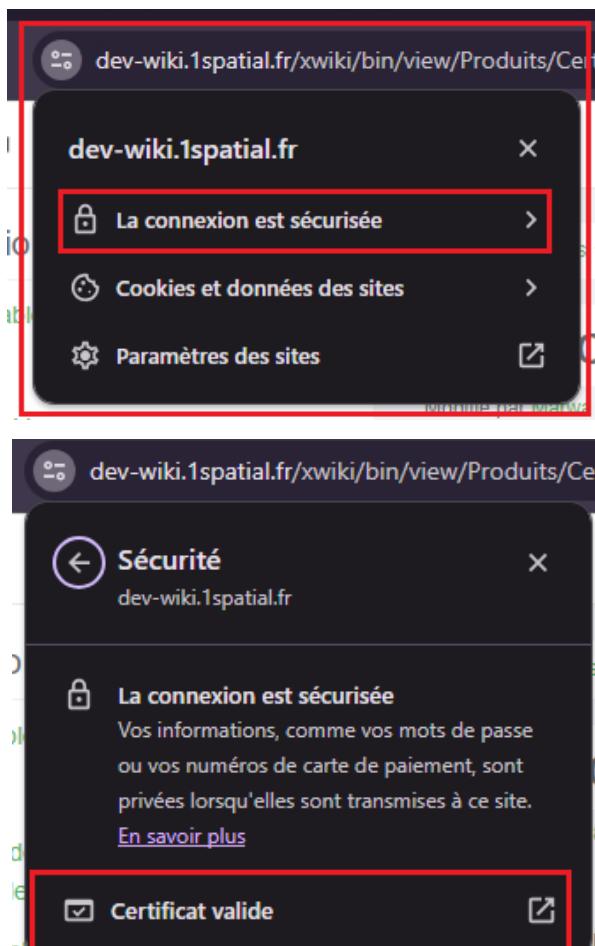
- ! Il est à noter que pour pouvoir utiliser les fonctionnalités « AddOutputFilterByType », il faut que « LoadModule filter\_module modules/mod\_filter.so » soit décommenté (pas de '#' dans httpd.conf).

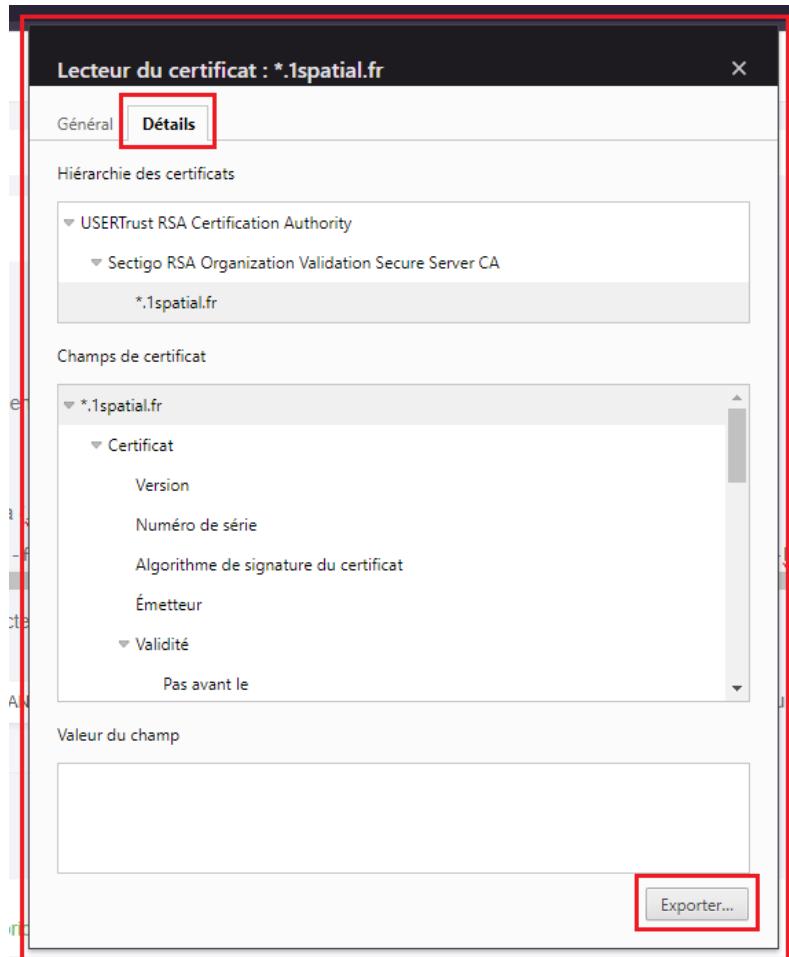
## 11.7 GESTION DES CERTIFICATS

### 11.7.1 Récupération du certificat

Le certificat est récupérable :

- Soit par l'équipe IT
- Soit en l'exportant à partir du navigateur (pour le navigateur Chrome) :





### 11.7.2 Ajout du certificat au cacerts

Ouvrir une invite de commande dans le répertoire bin de votre java (<Dossier d'installation>\server\jdk-17.0.5\bin).

Copier et modifier cette commande :

```
keytool -importcert -trustcacerts -alias "%votre_alias%" -file "%path_certif%" -storepass %password% -keystore <Dossier d'installation>\server\jdk-17.0.5\lib\security\cacerts
```

Les valeurs à modifier sont :

- l'alias : -alias "%votre\_alias%"
- Le path du certificat : -file "%path\_certif%"
- le mot de passe : -storepass %password% (**attention à ne pas oublier ce mot de passe**)
- Le path du cacerts de votre Java -keystore <Dossier d'installation>\server\jdk-17.0.5\lib\security\cacerts

Ce qui donne par exemple :

```
keytool -importcert -trustcacerts -alias "1Spatial" -file "D:\Temp\SSL Certificate - 1spatial.fr.crt" -storepass changeit -keystore <Dossier d'installation>\server\jdk-17.0.5\lib\security\cacerts
```