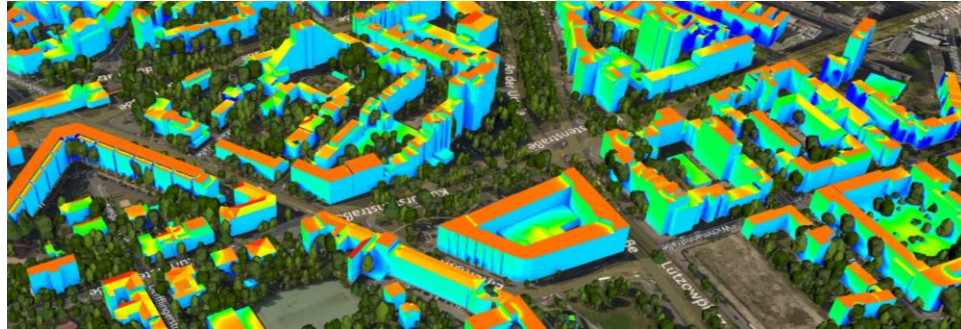


3D City Database 4.x (for PostgreSQL)

Quick installation guide for **Ubuntu**



License

This presentation is licensed under the [Creative Commons License CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/). According to CC BY-NC-SA 4.0 permission is granted to share this document, i.e. copy and redistribute the material in any medium or format, and to adapt it, i.e. remix, transform, and build upon the material under the following conditions:



- **Attribution:** You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **NonCommercial:** You may not use the material for commercial purposes.
- **ShareAlike:** If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- **No additional restrictions:** You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Overview

**Install required
software**

**Set up the
database**

**Connect to the
database via the
Importer/Exporter**

**Add additional
database schemas
(Optional)**

**Install ADE plug-ins
(Optional)**

Overview

**Install required
software**

**Set up the
database**

**Connect to the
database via the
Importer/Exporter**

**Add additional
database schemas
(Optional)**

**Install ADE plug-ins
(Optional)**

Software requirements

Software required

Software install

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

- **PostgreSQL**
 - See next slide for different download possibilities
- **PostGIS**
 - Often shipped as package together with your Linux distribution
 - https://postgis.net/documentation/getting_started/install_ubuntu/
- **pgAdmin**
 - <https://www.pgadmin.org/download/pgadmin-4-apt/>
- **Java 11 JDK or higher**
 - <https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>
- **CityGML 3D City Database Suite**
 - <https://github.com/3dcitydb/3dcitydb-suite/releases>
- **Google Earth Pro**
 - Optional, only if you want to export to KML/Collada
 - <https://www.google.com/earth/versions/#download-pro>
- **NodeJS**
 - Optional, needed only if you want to use the Web-map-client
 - <https://github.com/nodesource/distributions>

Software installation order

1) Install PostgreSQL (different ways available)

- Install **PostGIS**
- Install **pgAdmin** (optional, but highly recommended)

2) Install Java

- Required by the Importer/Exporter
- <https://www.oracle.com/java/technologies/javase/jdk11-archive-downloads.html>

3) Install the 3DCityDB Suite

- Execute the Java **.jar file**, it will start the installation process

4) Install Google Earth

- Optional, installation procedure not covered in these slides

5) Install NodeJS

- Optional, installation procedure not covered in these slides

Software required
Software install
Database setup
Imp/Exp connection
Additional schemas
ADE plug-ins
Further resources

PostgreSQL sources

You can install PostgreSQL on Ubuntu in different ways:

- **Using the packages available by default**

- Each Ubuntu version comes with a "fixed" edition, supported throughout the lifetime of that Ubuntu version
 - For example, Ubuntu 22.04 is shipped with PostgreSQL 14.x
- <https://www.postgresql.org/download/linux/ubuntu/>



- **PostgreSQL Apt Repository**

- Here, more recent versions of PostgreSQL are also available
- <https://wiki.postgresql.org/wiki/Apt>



Software required
Software install
Database setup
Imp/Exp connection
Additional schemas
ADE plug-ins
Further resources

PostgreSQL via Ubuntu packages



- **RECOMMENDED!** Unless you *really* need the latest version of PostgreSQL, this is the most straightforward way of installing it on your Ubuntu machine.
- These slides are based on the notes available at
 - <https://www.postgresql.org/download/linux/ubuntu/>
 - <https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-22-04-quickstart>
- **Prerequisites:** you need a user with sudo privileges (or to login as root)
- The next steps must be carried out from a command console and **are highlighted in this way**. You can simply copy & paste them (minor adjustments may be required)
- In these slides, some commands require Midnight Commander (mc) to edit some text files. However, you can use any other text editor
- If not already available, you can install Midnight Commander directly from the command console as follows
`sudo apt install mc`

Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

PostgreSQL via Ubuntu packages



Step 1: Refresh your server's local package index

```
sudo apt update
```

Step 2: Install the PostgreSQL, PostGIS and all dependent packages

```
sudo apt install postgresql postgresql-contrib postgis
```

For example, the default PostgreSQL version is version **14** in Ubuntu 22.04

- To check the **installed version**, you can run for example

```
sudo updatedb
```

```
locate bin/postgres
```

You should see something like this in the console (here: **14** stands for the installed version)

```
/usr/lib/postgresql/14/bin/postgres
```

- Check the **server port** (if required, change **14** to your installed version)

```
grep "port = " /etc/postgresql/14/main/postgresql.conf
```

The default port number is **5432**. If it is different, take note as you'll need it later

Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

PostgreSQL via Ubuntu packages



Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Step 3: Start the PostgreSQL server

```
sudo systemctl start postgresql
```

Check whether the PostgreSQL server has been started correctly

```
sudo systemctl status postgresql
```

You should see something like this in the console:

```
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Tue 2024-01-30 17:12:23 CET; 18s ago
     Process: 3520 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 3520 (code=exited, status=0/SUCCESS)
      CPU: 1ms
```

Although not needed here, these are the complementary commands to stop the PostgreSQL server

```
sudo systemctl stop postgresql
```

and to disable the service (it won't start anymore automatically at boot)

```
sudo systemctl disable postgresql
```

Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Step 4: Change/set a secure password of user *postgres* that has been created by default upon installation and has database superuser privileges.

First, switch to the *postgres* user and load the psql client. **-p** is used for the **server port** (e.g. **5432**)

```
sudo -u postgres psql -p 5432
```

Then, from within psql, set the new password for user *postgres*

```
ALTER USER postgres PASSWORD 'write_here_your_new_robust_password';
```

Finally, exit psql

```
\q
```

Test whether you can connect as *postgres* user using the new password:

```
psql -U postgres -h localhost -p 5432
```

You should see something like this in the console:

```
psql (14.10 (Ubuntu 14.10-0ubuntu0.22.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=#
```

Again, exit psql:

```
\q
```

PostgreSQL via Ubuntu packages



Step 5: Enable the PostgreSQL service to start automatically at boot

```
sudo systemctl enable postgresql
```

Check that it has been correctly enabled

```
sudo systemctl is-enabled postgresql
```

NOTE BANE: The PostgreSQL server is now installed and running. There is at the moment only the default *postgres* user.

The additional (optional) steps to customise some server settings will be shown later on.

To add users and create databases, we will use the pgAdmin GUI (see later). Nevertheless, all these operations can be performed also directly from the console. Examples:

- <https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-22-04-quickstart>
- <https://www.postgresqltutorial.com/postgresql-getting-started/install-postgresql-linux/>

Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

PostgreSQL via Apt Repository



Software required

Software install

- PostgreSQL
- PostGIS

- pgAdmin
- Java

• 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

- **ALTERNATIVELY** PostgreSQL can be installed from the **PostgreSQL Apt repository**., which contains up to the latest available version of PostgreSQL (and PostGIS).
- These slides are based on the notes available at
 - <https://www.postgresql.org/download/linux/ubuntu/>
 - <https://wiki.postgresql.org/wiki/Apt>
- **Prerequisites:** you need a user with sudo privileges (or to login as root)
- The next steps must be carried out from a command console and **are highlighted in this way**. You can simply copy & paste them (minor adjustments may be required)
- In these slides, some commands require Midnight Commander (mc) to edit some text files. However, you can use any other text editor
- If not already available, you can install Midnight Commander directly from the command console as follows
sudo apt install mc

PostgreSQL via Apt Repository



Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Step 1: Set up the Apt repository

```
sudo apt install -y postgresql-common
```

```
sudo /usr/share/postgresql-common/pgdg/apt.postgresql.org.sh
```


This will add the Apt repository to your package index depending on your machine and Ubuntu distribution

Step 2: Refresh your server's local package index

```
sudo apt update
```

You should see something like this in the console:

```
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 https://deb.nodesource.com/node_14.x jammy InRelease
Hit:3 https://apt.postgresql.org/pub/repos/apt jammy-pgdg InRelease
Hit:4 http://it.archive.ubuntu.com/ubuntu jammy InRelease
Hit:5 http://it.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:6 http://it.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:7 https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/jammy pgadmin4 InRelease
```





Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Step 3: Install the PostgreSQL, PostGIS and the dependent packages.

```
sudo apt install postgresql-16 postgis postgresql-16-postgis-3
```

Here we install version **16**, but you can choose another one (e.g. 14, 15, ...) and edit the command accordingly.

- To check the **installed version**, you can run for example

```
sudo updatedb
```

```
locate bin/postgres
```

You should see something like this in the console (here: **16** stands for the installed version)

```
/usr/lib/postgresql/16/bin/postgres
```

- Check the **server port** (if required, change **16** to your installed version)

```
grep "port = " /etc/postgresql/16/main/postgresql.conf
```

The default port number is **5432**. If it is different, take note as you'll need it later

PostgreSQL via Ubuntu packages



Step 4: Start the PostgreSQL server

```
sudo systemctl start postgresql
```

Check whether the PostgreSQL server has been started correctly

```
sudo systemctl status postgresql
```

You should see something like this in the console:

```
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Wed 2024-01-31 09:41:06 CET; 49min ago
     Main PID: 1200 (code=exited, status=0/SUCCESS)
        CPU: 1ms
```

Although not needed here, these are the complementary commands to stop the PostgreSQL server

```
sudo systemctl stop postgresql
```

and to disable the service (it won't start anymore automatically at boot)

```
sudo systemctl disable postgresql
```

Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

PostgreSQL via Ubuntu packages



Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Step 5: Change/set a secure password of user *postgres* that has been created by default upon installation and has database superuser privileges.

First, switch to the *postgres* user and load the psql client. **-p** is used for the **server port** (e.g. **5432**)

```
sudo -u postgres psql -p 5432
```

Then, from within psql, set the new password for user *postgres*

```
ALTER USER postgres PASSWORD 'write_here_your_new_robust_password';
```

Finally, exit psql

```
\q
```

Test whether you can connect as *postgres* user using the new password

```
psql -U postgres -h localhost -p 5432
```

You should see something like this in the console:

```
psql (16.1 (Ubuntu 16.1-1.pgdg22.04+1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

postgres=#
```

Again, exit psql

```
\q
```

PostgreSQL via Ubuntu packages



Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Step 6: Enable the PostgreSQL service to start automatically at boot

```
sudo systemctl enable postgresql
```

Check that it has been correctly enabled

```
sudo systemctl is-enabled postgresql
```

NOTE BANE: The PostgreSQL server is now installed and running. There is at the moment only the default *postgres* user.

The additional (optional) steps to customise some server settings will be shown later on.

To add users and create databases, we will use the pgAdmin GUI (see later). Nevertheless, all these operations can be performed also directly from the console. Examples:

- <https://www.digitalocean.com/community/tutorials/how-to-install-postgresql-on-ubuntu-22-04-quickstart>
- <https://www.postgresqltutorial.com/postgresql-getting-started/install-postgresql-linux/>

Customising PostgreSQL



In these slides, the following optional aspects will be covered:

- 1) Changing the default **server port** of PostgreSQL
- 2) Changing the default **data directory** of PostgreSQL

The procedure is the same, no matter which installation strategy has been followed. However, the commands may have to be slightly adapted depending on the installed version of PostgreSQL.

Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Customising PostgreSQL: server port



The default server port of PostgreSQL is 5432. However, it can be changed. All most important server settings are in file **postgresql.conf**.

Step 1: Stop the PostgreSQL server

```
sudo systemctl stop postgresql
```

Step 2: (If necessary) check the installed version of PostgreSQL

```
sudo updatedb
```

```
locate bin/postgres
```

You should see something like this in the console (here: **14** stands for the installed version)

```
/usr/lib/postgresql/14/bin/postgres
```

Step 3: (If necessary) locate the **postgresql.conf** file

```
locate main/postgresql.conf
```

You should see something like this in the console (again, **14** stands for the installed version):

```
/etc/postgresql/14/main/postgresql.conf
```

Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Customising PostgreSQL: server port



Step 4: Open and edit the **postgresql.conf** file. You can use whatever editor you prefer. For example, the Midnight Commander editor (mc). The **14** stands for the PostgreSQL version.

```
sudo mc -e /etc/postgresql/14/main/postgresql.conf
```

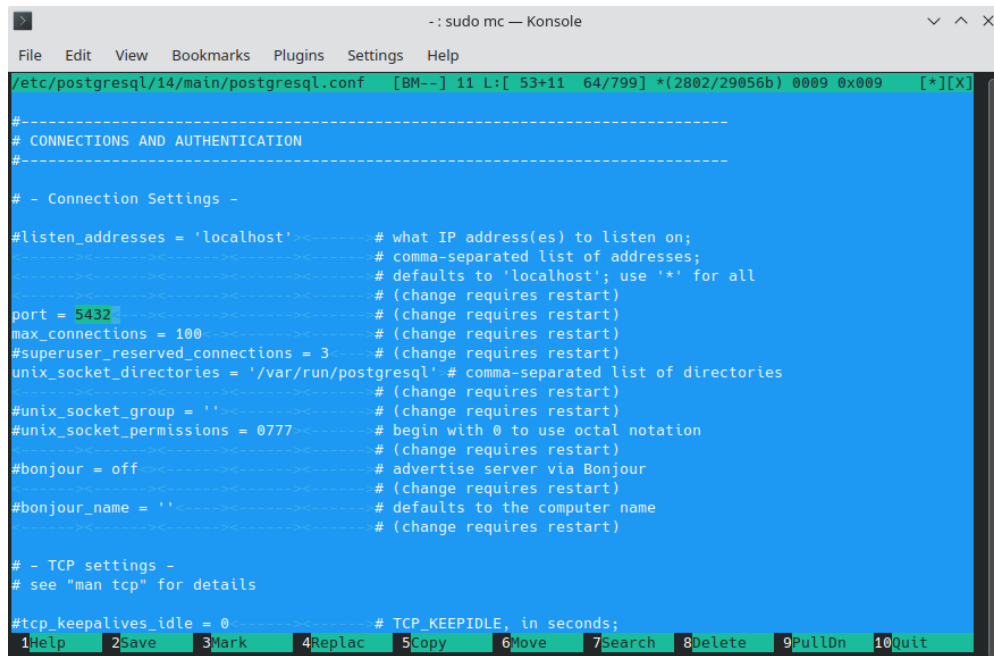
Scroll to the **Connection and Authentication** section.

You can edit the port number.

When you are done save and exit.

Step 5: Restart the PostgreSQL server

```
sudo systemctl start postgresql
```



```
-- sudo mc — Konsole
File Edit View Bookmarks Plugins Settings Help

/etc/postgresql/14/main/postgresql.conf [BM--] 11 L: 53+11 64/799 *(2802/29056b) 0009 0x009 [*][X]

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

#listen_addresses = 'localhost'          # what IP address(es) to listen on;
#                                     # comma-separated list of addresses;
#                                     # defaults to 'localhost'; use '*' for all
#                                     # (change requires restart)
port = 5432                             # (change requires restart)
max_connections = 100                    # (change requires restart)
#superuser_reserved_connections = 3      # (change requires restart)
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories
#                                     # (change requires restart)
#unix_socket_group = ''                  # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
#                                     # (change requires restart)
#bonjour = off                           # advertise server via Bonjour
#                                     # (change requires restart)
#bonjour_name = ''                       # defaults to the computer name
#                                     # (change requires restart)

# - TCP settings -
# see "man tcp" for details

#tcp_keepalives_idle = 0                  # TCP_KEEPIDL, in seconds;

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

PostgreSQL stores all data in a default directory. However, it can be changed. All most important server settings are in the file **postgresql.conf**

Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Step 1: Stop the PostgreSQL server

```
sudo systemctl stop postgresql
```

Step 2: (If necessary) check the installed version of PostgreSQL

```
sudo updatedb
```

```
locate bin/postgres
```

You should see something like this in the console (here: **14** stands for the installed version)

```
/usr/lib/postgresql/14/bin/postgres
```

Step 3: (If necessary) locate the **postgresql.conf** file

```
locate main/postgresql.conf
```

You should see something like this in the console (again, **14** stands for the installed version):

```
/etc/postgresql/14/main/postgresql.conf
```

Customising PostgreSQL: data directory



Step 4: Create and set up the new data directory. For example, in **/home/pg_data**

```
sudo mkdir /home/pg_data
```

```
sudo chown postgres:postgres /home/pg_data
```

```
sudo chmod 700 /home/pg_data
```

Step 5: Create a new database cluster in that directory. The **14** stands for the PostgreSQL version.

```
sudo -u postgres /usr/lib/postgresql/14/bin/initdb -D /home/pg_data
```

Step 6: Open and edit the **postgresql.conf** file. You can use whatever editor you prefer. For example, the Midnight Commander editor (mc).

The **14** stands for the PostgreSQL version.

```
sudo mc -e /etc/postgresql/14/main/postgresql.conf
```

Scroll to the **File Locations** section.

You can edit the data directory.

When you are done save and exit. →

Step 7: Restart the PostgreSQL server

```
sudo systemctl start postgresql
```

```

-: sudo mc — Konsole
File Edit View Bookmarks Plugins Settings Help
/etc/postgresql/14/main/postgresql.conf [BM--] 32 L: [ 35+ 7 42/799] *(1875/29042b) 0009 0x009 [*][X]
# FILE LOCATIONS
#-----
# The default values of these variables are driven from the -D command-line
# option or PGDATA environment variable, represented here as ConfigDir.
data_directory = '/home/pg_data' # use data in another directory
                                # (change requires restart)
hba_file = '/etc/postgresql/14/main/pg_hba.conf' # host-based authentication file
                                # (change requires restart)
ident_file = '/etc/postgresql/14/main/pg_ident.conf' # ident configuration file
                                # (change requires restart)
# If external_pid_file is not explicitly set, no extra PID file is written.
external_pid_file = '/var/run/postgresql/14-main.pid' # write an extra PID file
                                # (change requires restart)
#-----
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit

```

Software required

Software install

- PostgreSQL
- PostGIS
- **pgAdmin**
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Step 1: Setup the repository, Install the public key for the repository (if not done previously):

```
curl -fsS https://www.pgadmin.org/static/packages_pgadmin_org.pub | sudo gpg --dearmor -o  
/usr/share/keyrings/packages-pgadmin-org.gpg
```

Step 2: Create the repository configuration file:

```
sudo sh -c 'echo "deb [signed-by=/usr/share/keyrings/packages-pgadmin-org.gpg]  
https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/$(lsb_release -cs) pgadmin4 main" >  
/etc/apt/sources.list.d/pgadmin4.list && apt update'
```

Step 3: Install pgAdmin4, only for desktop, only web mode, or both (choose one of the three)

```
sudo apt install pgadmin4-desktop
```

```
sudo apt install pgadmin4-web
```

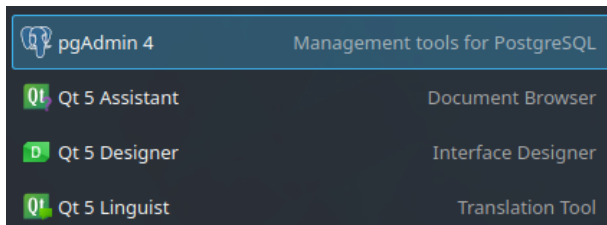
```
sudo apt install pgadmin4
```

Step 4: if you installed pgadmin4-web, configure the webserver, :

```
sudo /usr/pgadmin4/bin/setup-web.sh
```


Step 5a: Run pgAdmin directly from the Ubuntu GUI.

pgAdmin is reachable via the desktop menu as a normal application.



Step 5b: Run pgAdmin from the console

- Check where the **pgAdmin executable** has been installed

`sudo updatedb`

`locate bin/pgadmin`

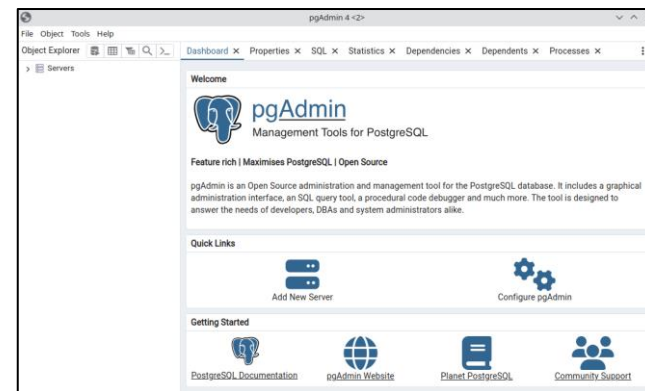
You should see something like this in the console:

```
/usr/pgadmin4/bin/pgadmin4
```

- Then, pgAdmin can be run directly from the command console

`/usr/pgadmin4/bin/pgadmin4 &`

Please note: the **&** is optional. It will allow you to keep working also on the console.



Set up PostgreSQL from pgAdmin

Software required

Software install

- PostgreSQL
- PostGIS
- **pgAdmin**
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

pgAdmin can be used to set up PostgreSQL, e.g. to

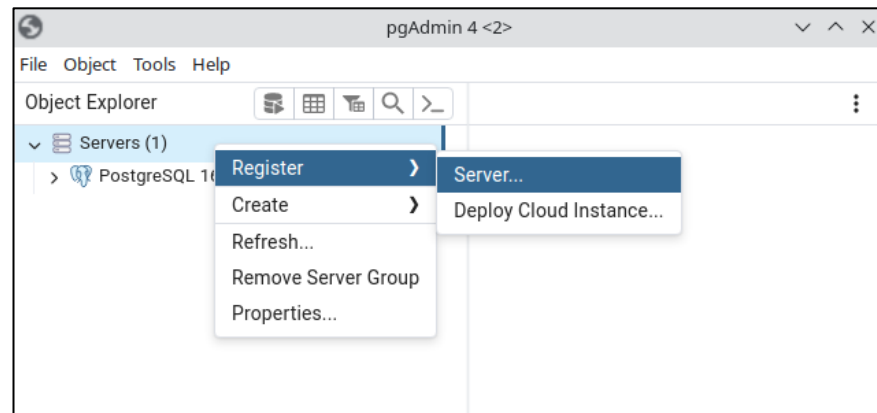
- Add/create database connections
- Add/create new database users
- Add/create new databases

However, first of all, a connection to the default "postgres" database must be established.

Therefore:

Step 1: Run pgAdmin

Step 2: With the mouse, right click on the "Servers" (see screenshot) and select Register\Server...



Set up PostgreSQL from pgAdmin

Step 3: Fill out the required fields using the connection parameters of user *postgres* to connect to the default database "postgres". The screenshots offer an example.

Software required

Software install

- PostgreSQL
- PostGIS
- **pgAdmin**
- Java
- 3DCityDB Suite

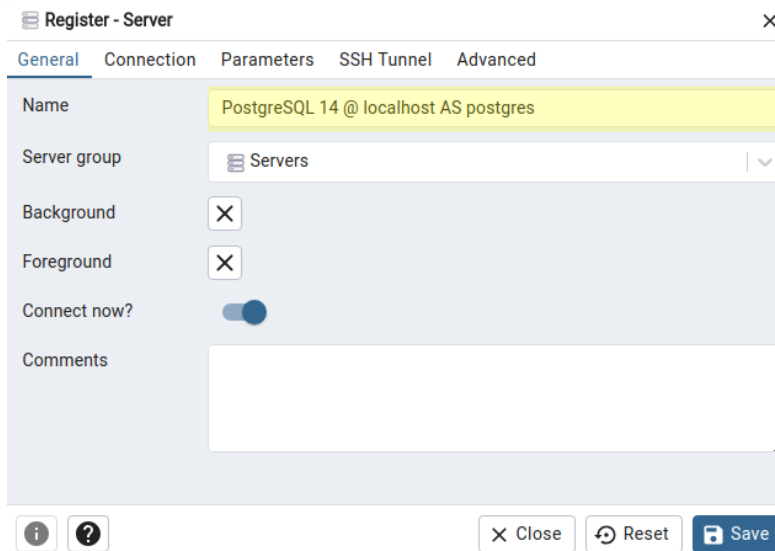
Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources



Register - Server

General | Connection | Parameters | SSH Tunnel | Advanced

Name: PostgreSQL 14 @ localhost AS postgres

Server group: Servers

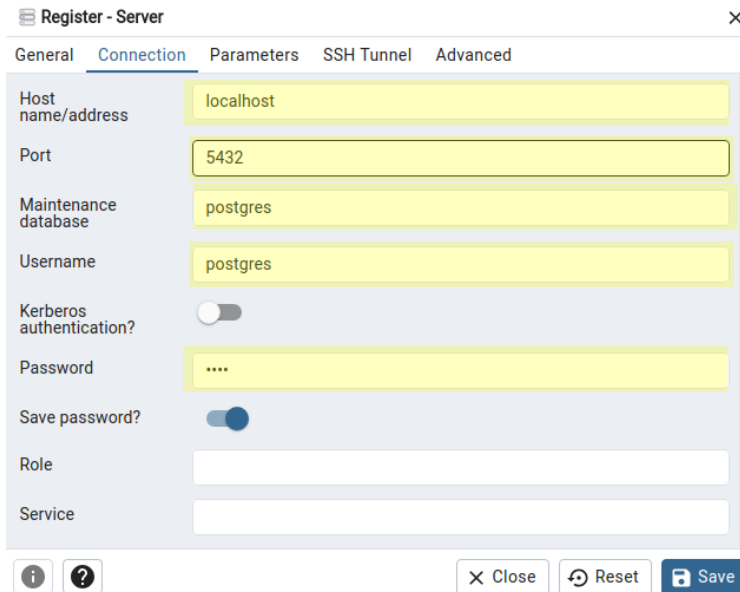
Background: ☐

Foreground: ☐

Connect now?: ☒

Comments:

Close Reset Save



Register - Server

General | Connection | Parameters | SSH Tunnel | Advanced

Host name/address: localhost

Port: 5432

Maintenance database: postgres

Username: postgres

Kerberos authentication?: ☐

Password:

Save password?: ☒

Role:

Service:

Close Reset Save

Set up PostgreSQL from pgAdmin

Step 4: From the newly created connection, you will be able to access the default "postgres" database. You can now add new users and create new databases (see next slides).

Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

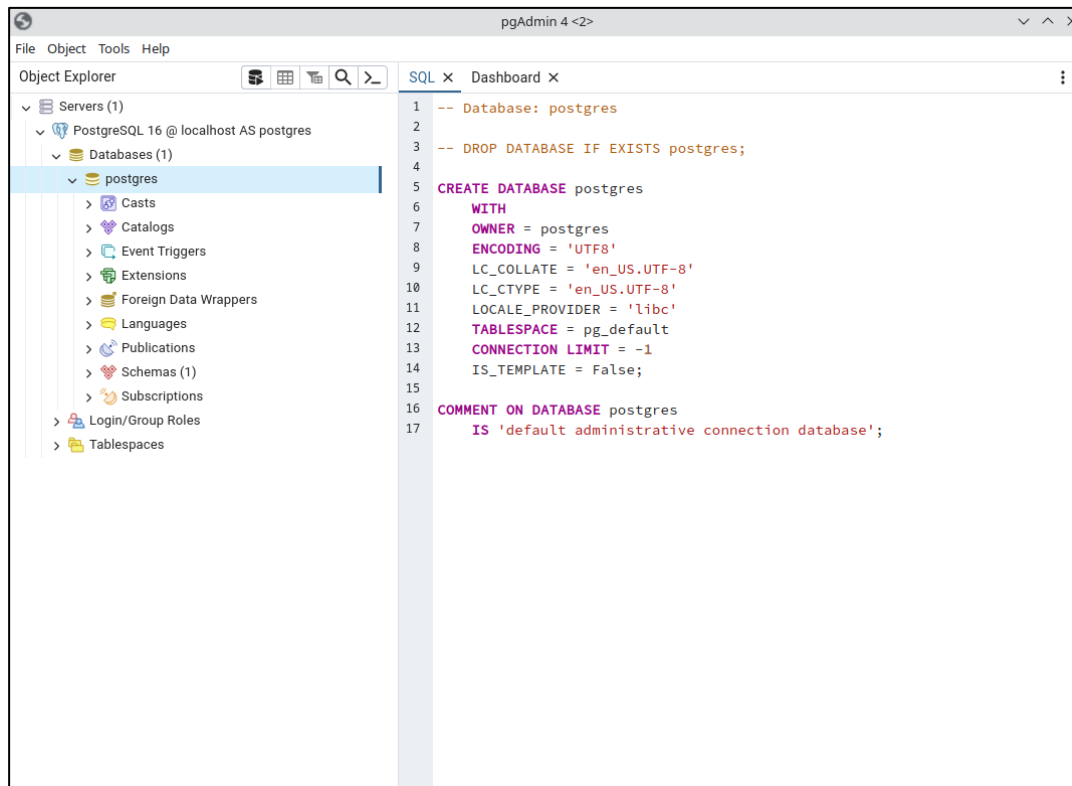
Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources



Set up PostgreSQL from pgAdmin

Software required

Software install

- PostgreSQL
- PostGIS
- **pgAdmin**
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

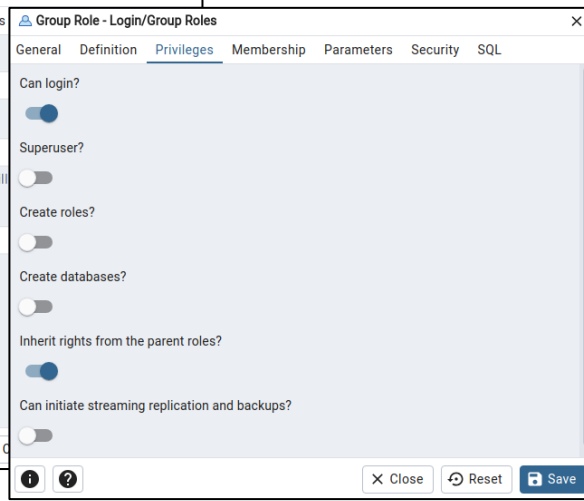
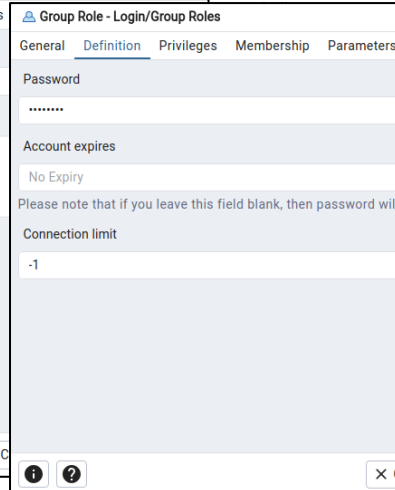
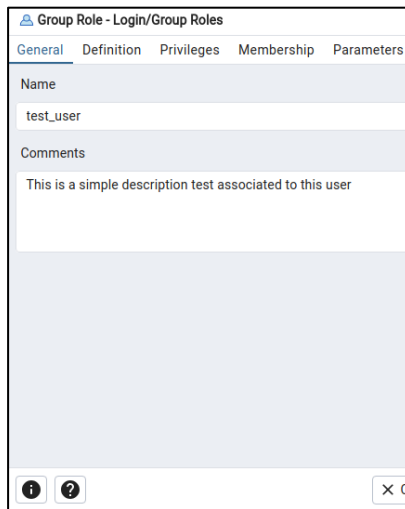
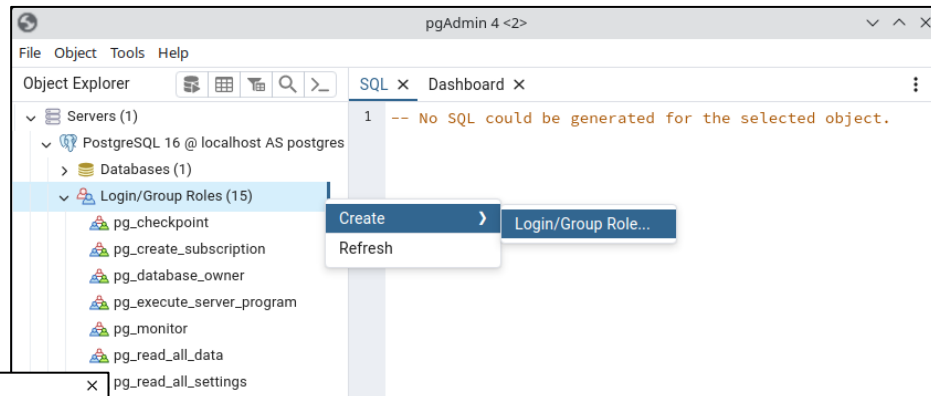
To add/create a **new database user**:

Step 1: Right click with the mouse in the Login/Group Roles and select

Create/Group Role

Step 2: Set the user name, password, etc.

Step 3: Save! The new user will be added to the list



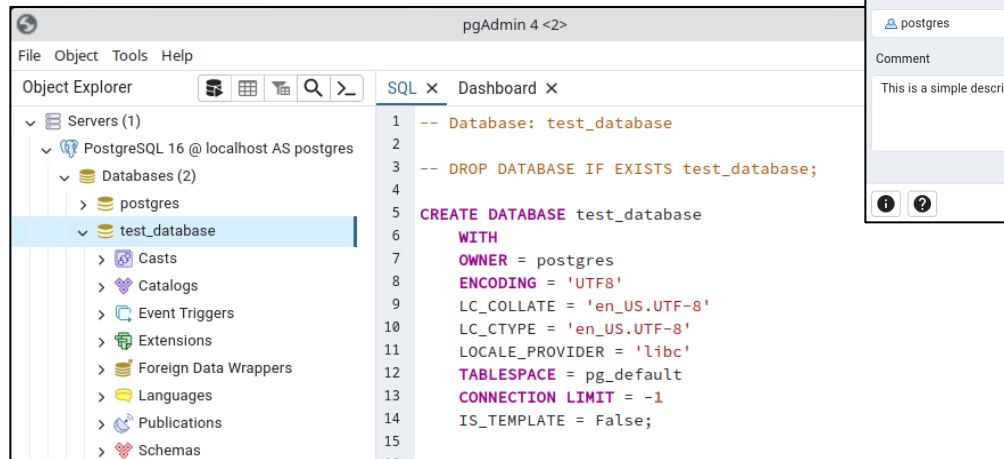
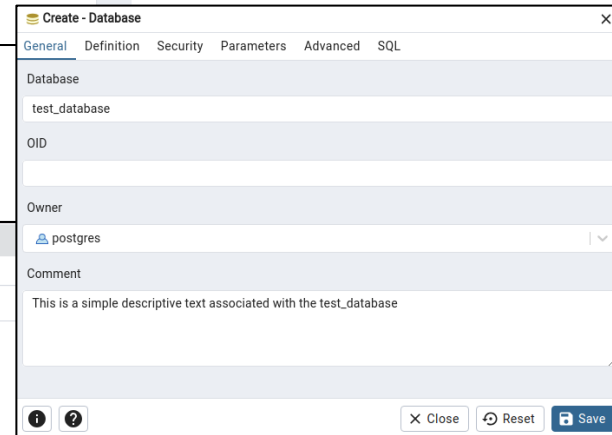
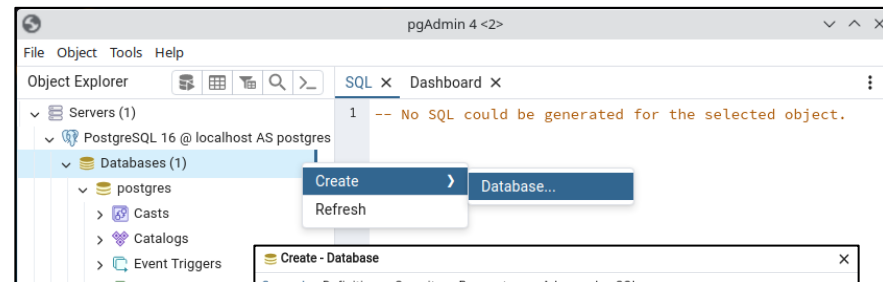
Set up PostgreSQL from pgAdmin

To add/create a **new database**:

Step 1: Right click with the mouse on Databases and select Create/Database...

Step 2: Fill out the form and set the database properties

Step 3: Save! The new database will be added to the list.



Software required

Software install

- PostgreSQL
- PostGIS
- **pgAdmin**
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- **Java**
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Java can be installed in Ubuntu using the prepackaged binaries via the usual apt commands. These slides show how to install the Open Java Development Kit (OpenJDK) v. 11.

The next steps are taken and adapted from: <https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/How-do-I-install-Java-on-Ubuntu>

Step 1: Refresh your server's local package index

```
sudo apt update
```

Step 2: Install the JDK

```
sudo apt install default-jdk
```

Step 3: Check that Java is installed and which version

```
java -version
```

You should see something like this in the console:

```
openjdk version "11.0.21" 2023-10-17
OpenJDK Runtime Environment (build 11.0.21+9-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.21+9-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
```

Software required

Software install

- PostgreSQL
 - PostGIS
 - pgAdmin
 - **Java**
 - 3DCityDB Suite
- Database setup
Imp/Exp connection
Additional schemas
ADE plug-ins
Further resources

Step 4: Find out where apt installed Java on Ubuntu

```
update-alternatives --config java
```

You should see something like this in the console:

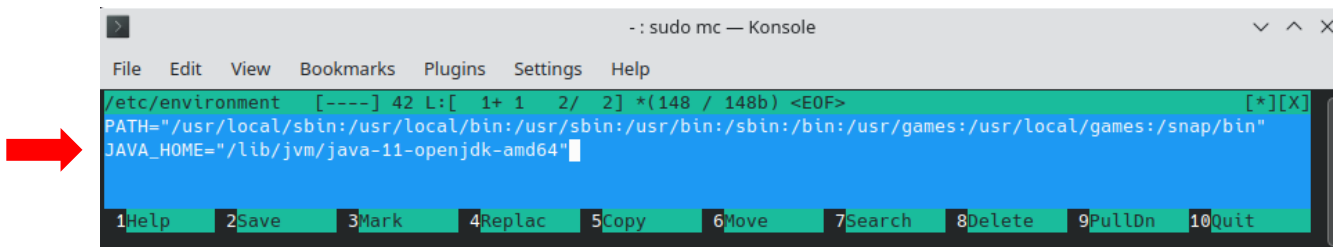
```
There is only one alternative in link group java (providing /usr/bin/java):
/usr/lib/jvm/java-11-openjdk-amd64/bin/java ←
Nothing to configure.
```

Step 5: Add JAVA_HOME to the environment. Open and edit `/etc/environment` file. You can use whatever editor you prefer. For example, the Midnight Commander editor (mc).

```
sudo mc -e /etc/environment
```

Paste the following line at the bottom of the file. Check that the string in red is the same as the one indicated in the previous point but without /bin/java.

JAVA_HOME="/lib/jvm/java-11-openjdk-amd64"



Software required

Software install

- PostgreSQL
- PostGIS
- pgAdmin
- Java
- 3DCityDB Suite

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Step 6: Force the Ubuntu terminal to reload the environment configuration file

```
source /etc/environment
```

Step 7: Check that JAVA_HOME has been set

```
echo $JAVA_HOME
```

You should see something like this in the console:

```
/lib/jvm/java-11-openjdk-amd64
```

Done! 😊

3DCityDB Suite

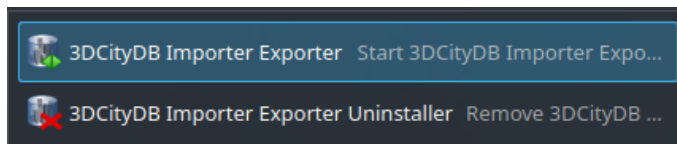


Once you have downloaded the **.jar** file of the 3DCityDB Suite you can launch the installer of the Importer/Exporter from the directory where the jar file is saved. For example:

```
java -jar 3DCityDB-Importer-Exporter-5.4.0-Setup.jar
```

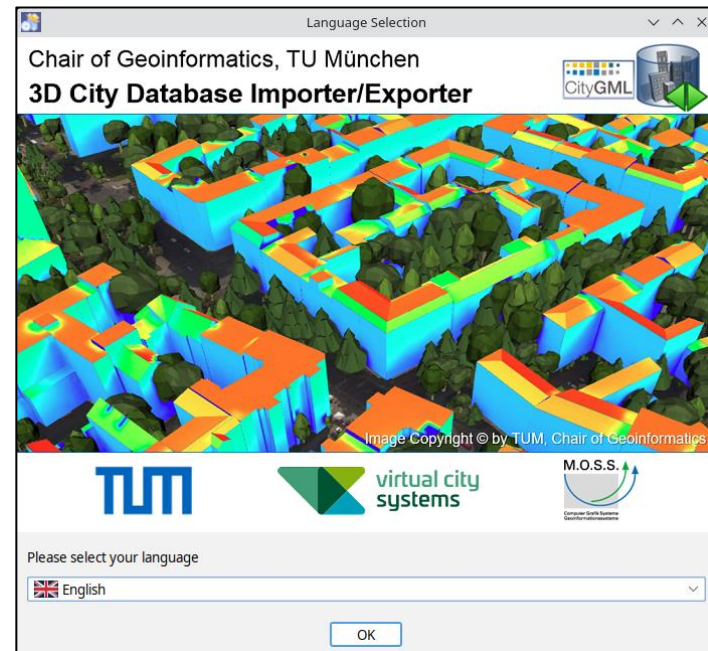
Follow the installation instructions. The software will be generally installed in your home directory e.g.
/home/giorgio/3DCityDB-Importer-Exporter

Upon installation, the Importer/Exporter can be launched directly from the desktop menu



Or, alternatively, via the command console from within the installation directory:

```
./3DCityDB-Importer-Exporter &
```



Overview

**Install required
software**

**Set up the
database**

**Connect to the
database via the
Importer/Exporter**

**Add additional
database schemas
(Optional)**

**Install ADE plug-ins
(Optional)**

Database setup

Procedure overview

Software required

Software install

Database setup

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

- 1) In PostgreSQL (e.g. via the pgAdmin GUI)
 - Connect to the PostgreSQL server
 - Create a new empty database that will contain your 3D city model data
 - Add the extensions for PostGIS, PostGIS-raster, etc. to the empty database
 - See the next slides for details
- 2) From the 3DCityDB installation folder
 - Authorise the execution of the shell files using the following command from the command prompt:
`chmod u+x CREATE_DB.sh`
 - Edit the **CONNECTION_DETAILS.sh** file and run the **CREATE_DB.sh** script to create the tables (and other objects) in the empty database you have created in the previous step
 - See the next slides for details
- 3) Connect to the database (e.g. via pgAdmin) just to check that you created the tables
- 4) Connect to the database from the 3DCityDB Importer/Exporter

Connecting to the database

- AFTER you have successfully installed PostgreSQL, you can access the database server via pgAdmin

ALTERNATIVELY

- You do not have PostgreSQL installed on your own computer, but you know the connection parameters to connect to a remote server
- **In both cases**, you will need information about:
 - **Server name** or **IP address** ("localhost" if it is on your own computer)
 - **Database name** (generally "postgres" if it is on your own computer)
 - **Port** (generally 5432 if it is on your own computer)
 - **Username, Password** (e.g. the ones created before, see previous slides)

Software required

Software install

Database setup

• Database connection

- Database creation
- Create tables etc.
- Check via pgAdmin

Imp/Exp connection

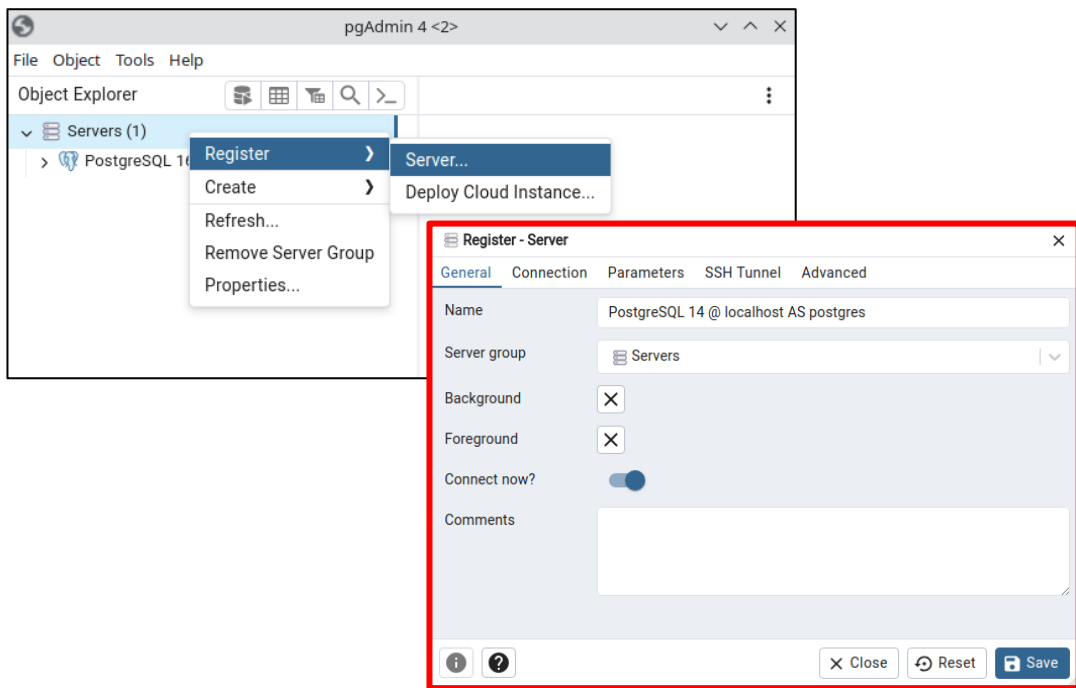
Additional schemas

ADE plug-ins

Further resources

Connecting to the database from pgAdmin

- Create a (link to the) database server
 - You are actually creating a connection to the database server from pgAdmin
 - Click on menu Object\Create\Server **OR** right mouse-click\Create\Server and fill out the fields
 - **Please note:** this step may not be required if you already have a server connection established

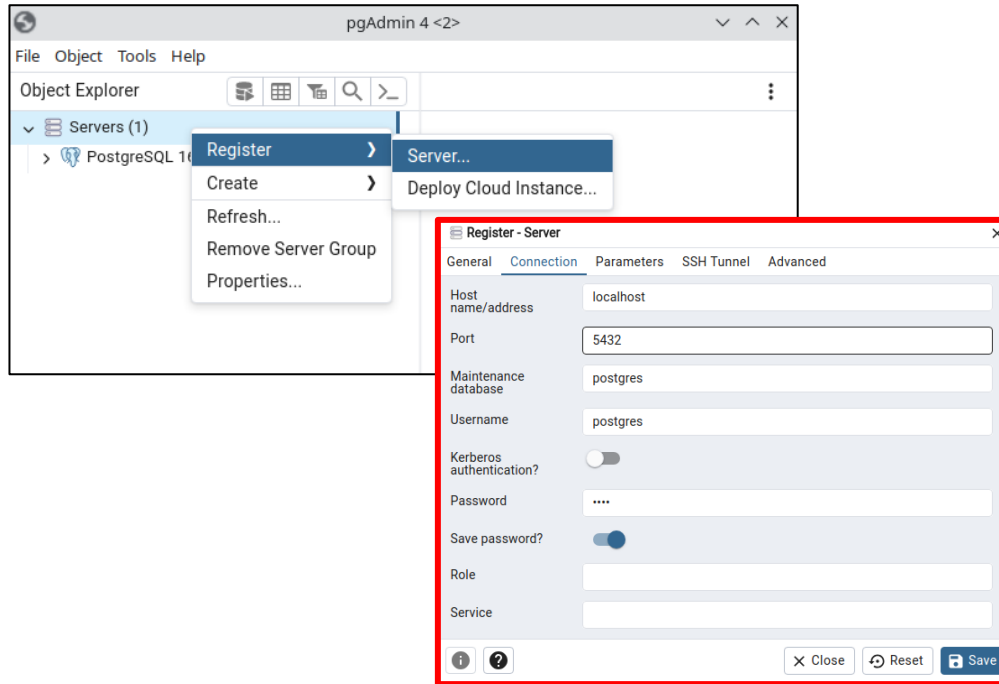


In the "General" tab, you simply add a name to identify your connection. Here, FOR EXAMPLE, the string is "PostgreSQL-12 on my PC".

Software required
Software install
Database setup
• **Database connection**
• Database creation
• Create tables etc.
• Check via pgAdmin
Imp/Exp connection
Additional schemas
ADE plug-ins
Further resources

Connecting to the database from pgAdmin

- Create a (link to the) database server
 - You are actually creating a connection to the database server from pgAdmin
 - Click on menu Object\Create\Server **OR** right mouse-click\Create\Server and fill out the fields
 - **Please note:** this step may not be required if you already have a server connection established



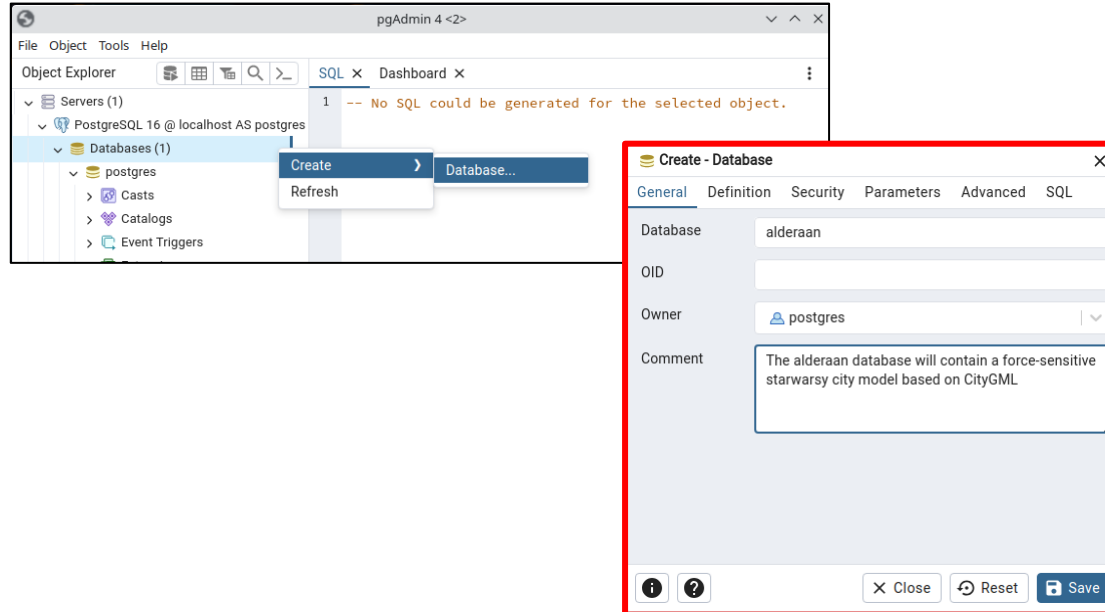
In the "Connection" tab,
you add the connection
parameters

- Host name / IP address
- Port
- Database name
- Username
- Password

Software required
Software install
Database setup
• **Database connection**
• Database creation
• Create tables etc.
• Check via pgAdmin
Imp/Exp connection
Additional schemas
ADE plug-ins
Further resources

Database creation

- Once you have created a connection, you must create the database that will contain your city model data
 - Click on menu Object\Create\Database **OR** right mouse-click\Create\Database and fill out the fields
 - Choose the name you want, ideally the name of the city



Here, **for example**, it is "alderaan", but you can choose any name you want

Best if you use only small letters!

Software required
Software install
Database setup
• Database connection
• **Database creation**
• Create tables etc.
• Check via pgAdmin
Imp/Exp connection
Additional schemas
ADE plug-ins
Further resources

Database creation

Software required

Software install

Database setup

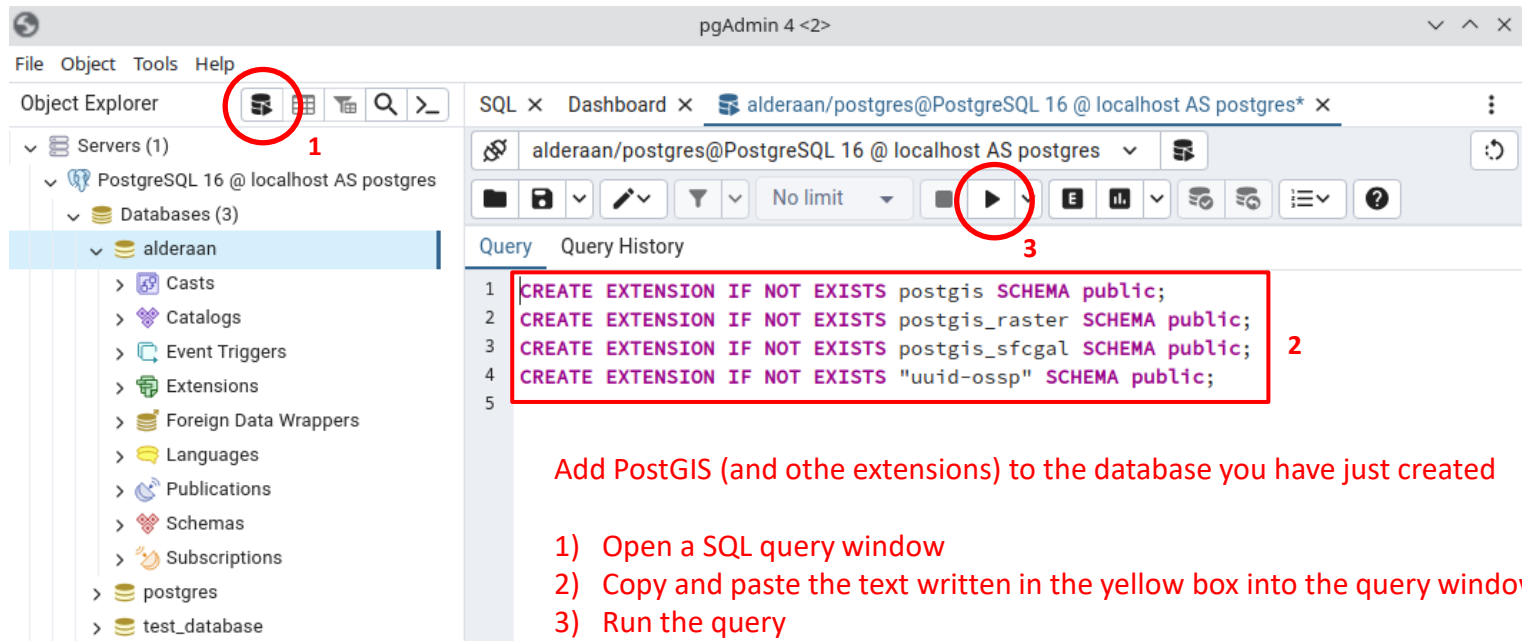
- Database connection
- **Database creation**
- Create tables etc.
- Check via pgAdmin

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources



pgAdmin 4 <2>

File Object Tools Help

Object Explorer

Servers (1)

PostgreSQL 16 @ localhost AS postgres

Databases (3)

alderaan

Casts

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas

Subscriptions

postgres

test_database

SQL x Dashboard x alderaan/postgres@PostgreSQL 16 @ localhost AS postgres*

alderaan/postgres@PostgreSQL 16 @ localhost AS postgres

No limit

Query Query History

```

1 CREATE EXTENSION IF NOT EXISTS postgis SCHEMA public;
2 CREATE EXTENSION IF NOT EXISTS postgis_raster SCHEMA public;
3 CREATE EXTENSION IF NOT EXISTS postgis_sfcgal SCHEMA public;
4 CREATE EXTENSION IF NOT EXISTS "uuid-oss" SCHEMA public;
5

```

Add PostGIS (and other extensions) to the database you have just created

- 1) Open a SQL query window
- 2) Copy and paste the text written in the yellow box into the query window
- 3) Run the query

```

CREATE EXTENSION IF NOT EXISTS postgis SCHEMA public;
CREATE EXTENSION IF NOT EXISTS postgis_raster SCHEMA public;
CREATE EXTENSION IF NOT EXISTS postgis_sfcgal SCHEMA public; -- optional, but good to have
CREATE EXTENSION IF NOT EXISTS "uuid-oss" SCHEMA public; -- optional, but good to have

```

Database creation

- Check that you have correctly installed the extensions in your database
 - Open "Extensions" item in your database (e.g. "alderaan")
 - Check that the extensions are listed there (the "plpgsql" one is installed by default)

Software required
Software install

Database setup

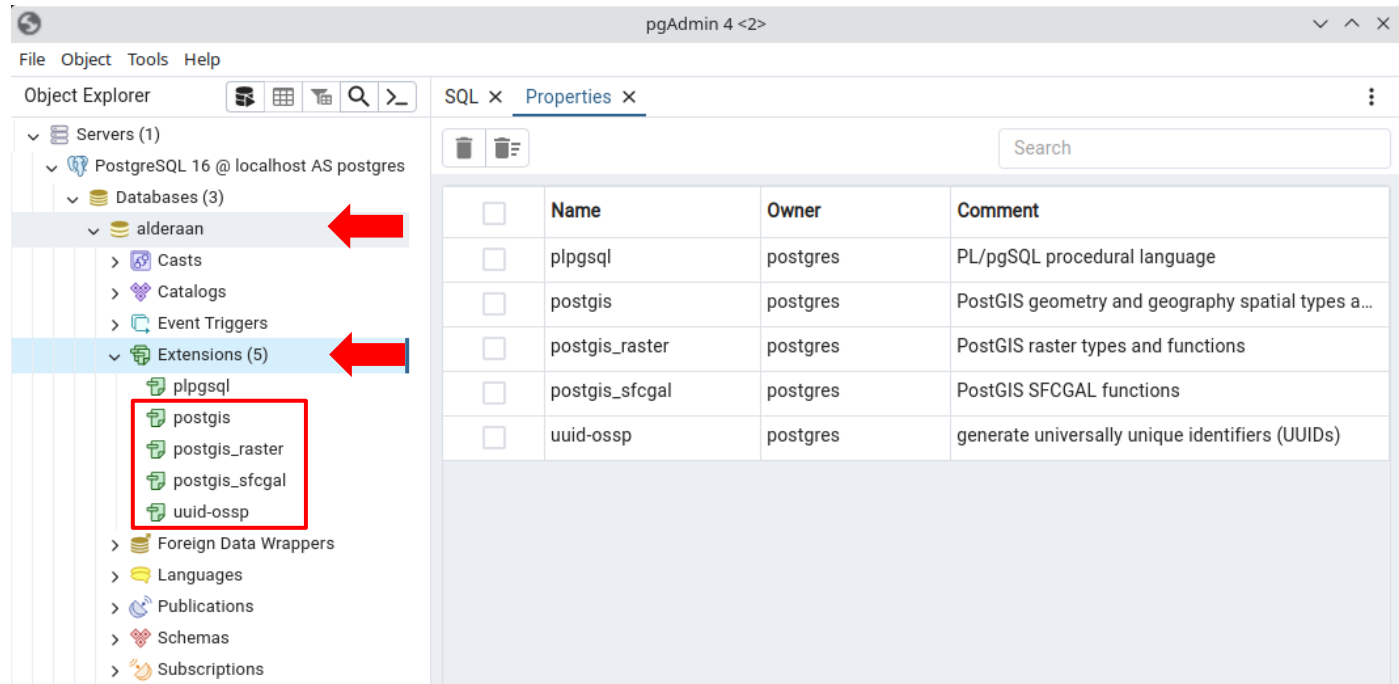
- Database connection
- **Database creation**
- Create tables etc.
- Check via pgAdmin

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources



Create tables and other database objects

NOTA BENE: The detailed installation guide can be found here:

<https://3dcitydb-docs.readthedocs.io/en/latest/first-steps/index.html>

In the following slides, only the main points are presented

Software required

Software install

Database setup

- Database connection
- Database creation
- **Create tables etc.**
- Check via pgAdmin

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

- Go to the **3DCityDB installation folder** and change to the 3dcitydb\postgresql\ShellScripts\Unix directory. It should look like in the next slides
 - Then, if not done previously:
`chmod u+x CREATE_DB.sh`
`chmod u+x DROP_DB.sh` (Only needed if you want to drop the database)
 - Open the **CONNECTION_DETAILS.sh** file in a text editor and insert your PostgreSQL connection details. For example, using Midnight Commander (or anything equivalent)
`mc -e CONNECTION_DETAILS.sh`
 - Run the **CREATE_DB.sh**. It will generate all needed database objects (schemas, tables, etc.)
`./CREATE_DB.sh`

Create tables and other database objects

Software required

Software install

Database setup

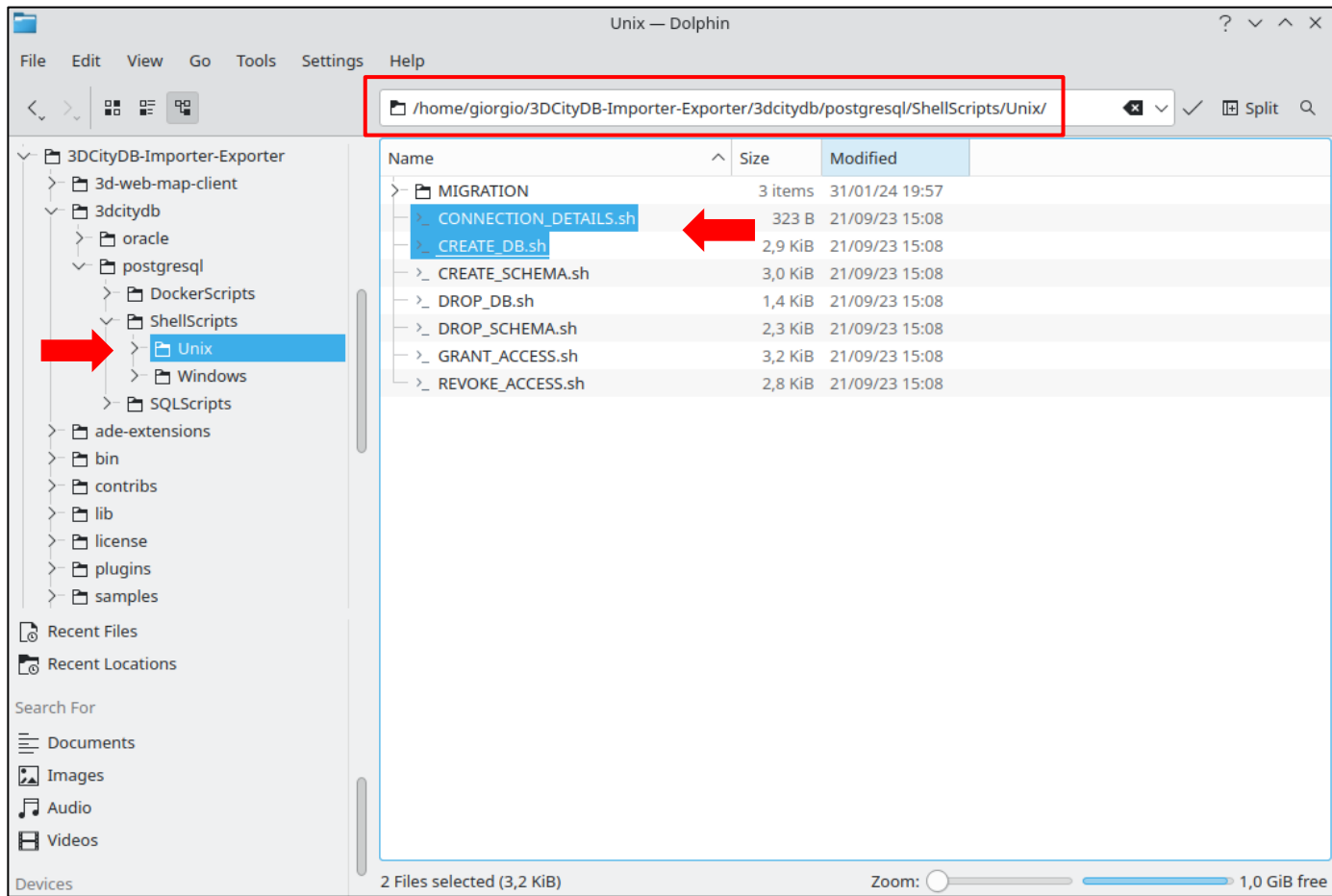
- Database connection
- Database creation
- **Create tables etc.**
- Check via pgAdmin

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources



Create tables and other database objects

Software required

Software install

Database setup

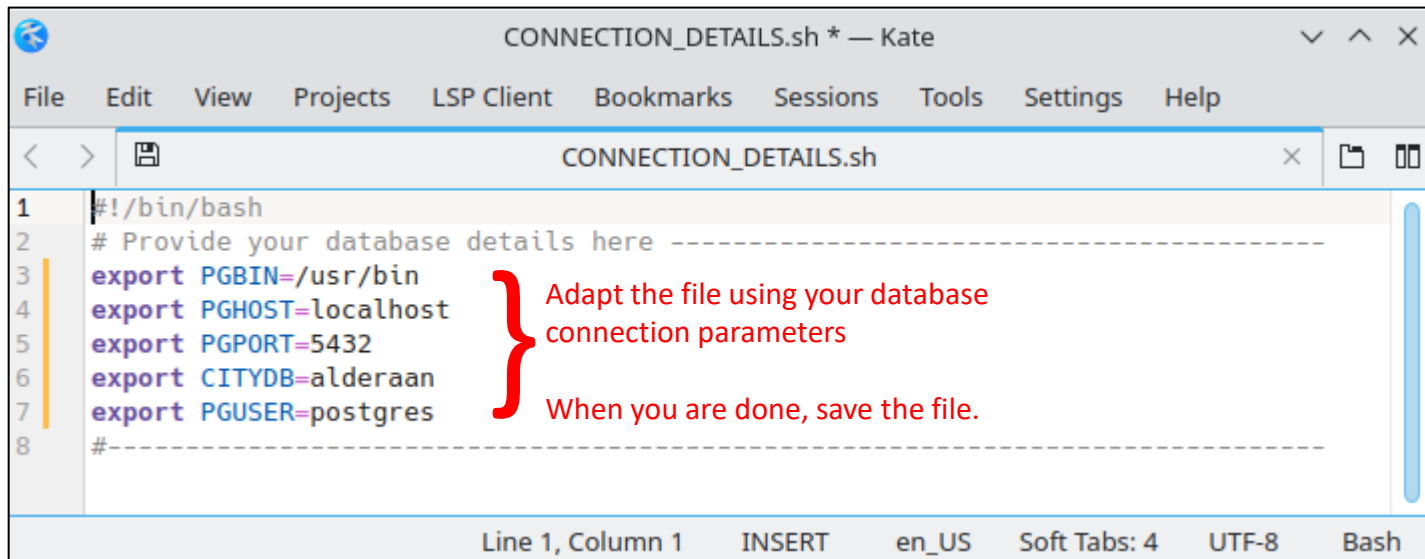
- Database connection
- Database creation
- **Create tables etc.**
- Check via pgAdmin

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources



```

1  #!/bin/bash
2  # Provide your database details here -----
3  export PGBIN=/usr/bin
4  export PGHOST=localhost
5  export PGPORT=5432
6  export CITYDB=alderaan
7  export PGUSER=postgres
8  #-----
  
```

Adapt the file using your database connection parameters

When you are done, save the file.

Line 1, Column 1 INSERT en_US Soft Tabs: 4 UTF-8 Bash

- The **PGBIN** variable contains the path to the psql executable. Depending on your PostgreSQL installation, it could be in other (but similar) locations.
- **PGHOST** is the IP address of the PostgreSQL server. If you are using the database on your own computer, you can write "localhost", otherwise you have to write the IP address of the server
- **PGPORT** is the port PostgreSQL is listening to. Generally, it is 5432 (this is the default value).
- **CITYDB** contains the name of the database that you have created before and that will contain the 3DCityDB tables, e.g. "alderaan"
- **PGUSER** contains the name of the user to connect to the database. On your local machine, you may use postgres. You will be asked the password during the installation process later.

Create tables and other database objects

Find out the EPSG codes that apply to your city or region.

If you do not know them, you can search for them here: <https://epsg.org/>

Here some examples:



- **Netherlands**
 - **Horizontal datum EPSG: 28992**
 - **Vertical datum EPSG: 5109**
 - (Will automatically create the GMLsrsName: **urn:ogc:def:crs,crs:EPSG::28992,crs:EPSG:5109**)
- **Trento (Italy)**
 - **Horizontal datum EPSG: 25832**
 - **Vertical datum EPSG: 5214**
 - (Will automatically create the GMLsrsName: **urn:ocg:def:crs,crs:EPSG::25832,crs:EPSG::5214**)
- **Vienna (Austria)**
 - **Horizontal datum EPSG: 31256**
 - **Vertical datum EPSG: 1267**
 - (Will automatically create the GMLsrsName: **urn:ocg:def:crs,crs:EPSG::31256,crs:EPSG::1267**)

Software required
Software install
Database setup

- Database connection
- Database creation
- **Create tables etc.**
- Check via pgAdmin

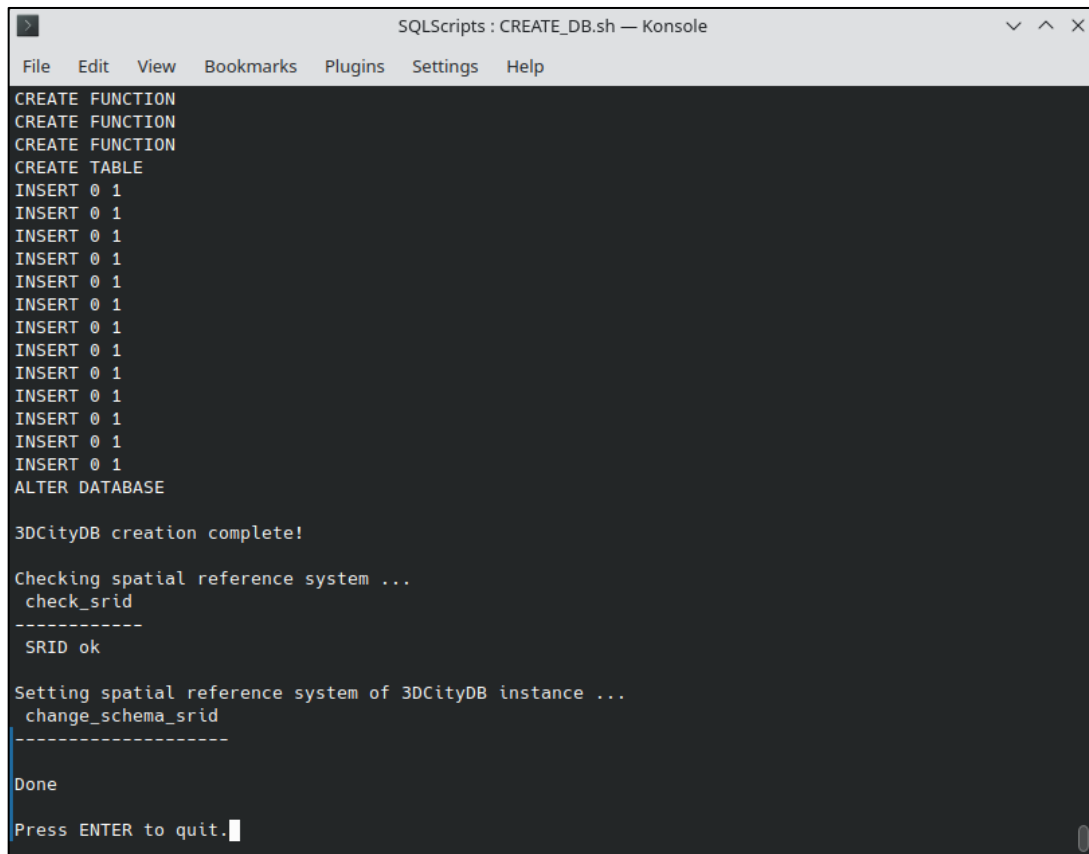
Imp/Exp connection
Additional schemas
ADE plug-ins
Further resources

- Set the EPSG codes for horizontal and vertical datum, as shown in the image here



Create tables and other database objects

- Upon successful installation, you should get something like this



```
SQLScripts : CREATE_DB.sh — Konsole
File Edit View Bookmarks Plugins Settings Help
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
CREATE TABLE
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
ALTER DATABASE

3DCityDB creation complete!

Checking spatial reference system ...
  check_srid
-----
SRID ok

Setting spatial reference system of 3DCityDB instance ...
  change_schema_srid
-----

Done
Press ENTER to quit.
```

Software required

Software install

Database setup

- Database connection
- Database creation
- **Create tables etc.**
- Check via pgAdmin

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources

Check via pgAdmin

- Open pgAdmin and check that the **citydb** and **citydb_pkg** schemas are there. The **citydb** schema should contain 66 tables

Software required

Software install

Database setup

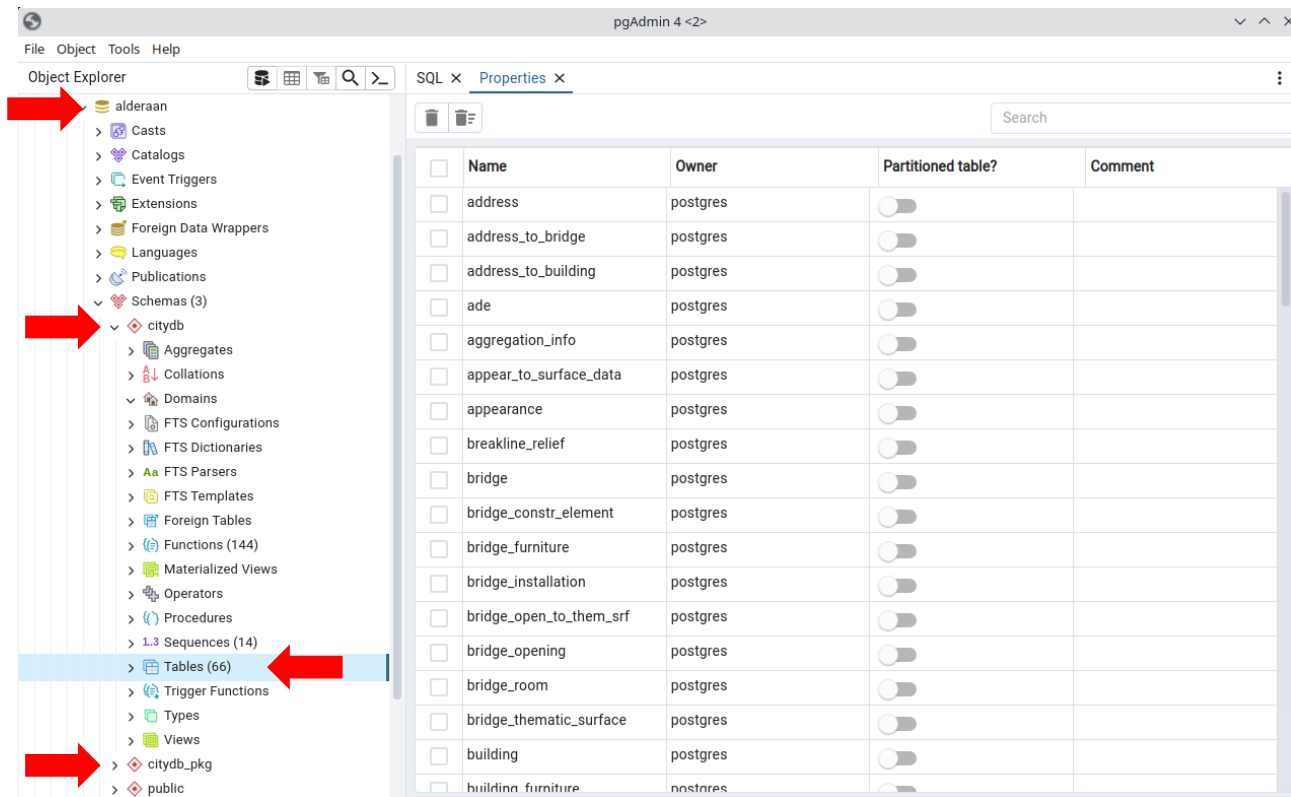
- Database connection
- Database creation
- Create tables etc.
- Check via pgAdmin

Imp/Exp connection

Additional schemas

ADE plug-ins

Further resources



The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays the database structure. A red arrow points to the 'citydb' schema under 'Schemas (3)'. Another red arrow points to the 'Tables (66)' entry under 'citydb'. A third red arrow points to the 'citydb_pkg' schema. On the right, the Properties window for the 'citydb' schema is open, showing a table with columns: Name, Owner, Partitioned table?, and Comment. The table lists 66 tables, all owned by 'postgres' and not partitioned.

| Name | Owner | Partitioned table? | Comment |
|-------------------------|----------|--------------------------|---------|
| address | postgres | <input type="checkbox"/> | |
| address_to_bridge | postgres | <input type="checkbox"/> | |
| address_to_building | postgres | <input type="checkbox"/> | |
| ade | postgres | <input type="checkbox"/> | |
| aggregation_info | postgres | <input type="checkbox"/> | |
| appear_to_surface_data | postgres | <input type="checkbox"/> | |
| appearance | postgres | <input type="checkbox"/> | |
| breakline_relief | postgres | <input type="checkbox"/> | |
| bridge | postgres | <input type="checkbox"/> | |
| bridge_constr_element | postgres | <input type="checkbox"/> | |
| bridge_furniture | postgres | <input type="checkbox"/> | |
| bridge_installation | postgres | <input type="checkbox"/> | |
| bridge_open_to_them_srf | postgres | <input type="checkbox"/> | |
| bridge_opening | postgres | <input type="checkbox"/> | |
| bridge_room | postgres | <input type="checkbox"/> | |
| bridge_thematic_surface | postgres | <input type="checkbox"/> | |
| building | postgres | <input type="checkbox"/> | |
| building_furniture | postgres | <input type="checkbox"/> | |

Overview

**Install required
software**

**Set up the
database**

**Connect to the
database via the
Importer/Exporter**

**Add additional
database schemas
(Optional)**

**Install ADE plug-ins
(Optional)**

Connecting to the database via Importer/Exporter

- AFTER you have successfully installed PostgreSQL, you can access the database server via pgAdmin

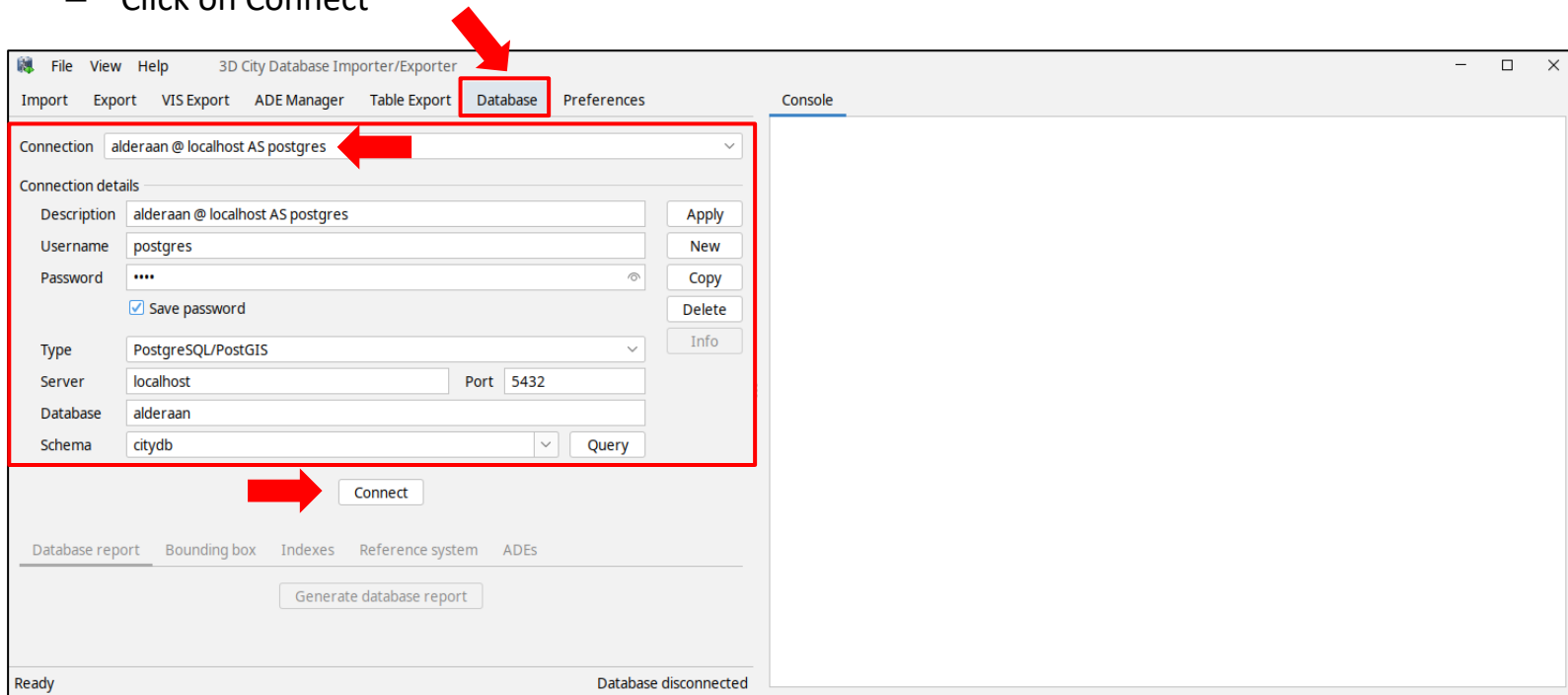
ALTERNATIVELY

- You do not have PostgreSQL installed on your own computer, but you know the connection parameters to connect to a remote server
- **In both cases**, you will need information about:
 - **Server name** or **IP address** ("localhost" if it is on your computer)
 - **Database name** (generally "postgres" if it is on your computer)
 - **Port** (generally 5432 if it is on your computer)
 - **Username, Password** (e.g. the ones created before if it is on your computer)

Software required
Software install
Database setup
**Imp/Exp
connection**
Additional schemas
ADE plug-ins
Further resources

Connecting to the database via Importer/Exporter

- Launch the 3DCityDB Importer/Exporter, select the "Database" tab
 - The Description field contains a simple string to identify the connection
 - Fill out the remaining fields with the connection parameters
 - Click on Connect



3D City Database Importer/Exporter

File View Help Import Export VIS Export ADE Manager Table Export **Database** Preferences Console

Connection: alderaan @ localhost AS postgres

Connection details

Description: alderaan @ localhost AS postgres [Apply]

Username: postgres [New]

Password: [Copy]

☒ Save password [Delete]

Type: PostgreSQL/PostGIS [Info]

Server: localhost Port: 5432

Database: alderaan

Schema: citydb [Query]

[Connect]

Database report Bounding box Indexes Reference system ADEs

[Generate database report]

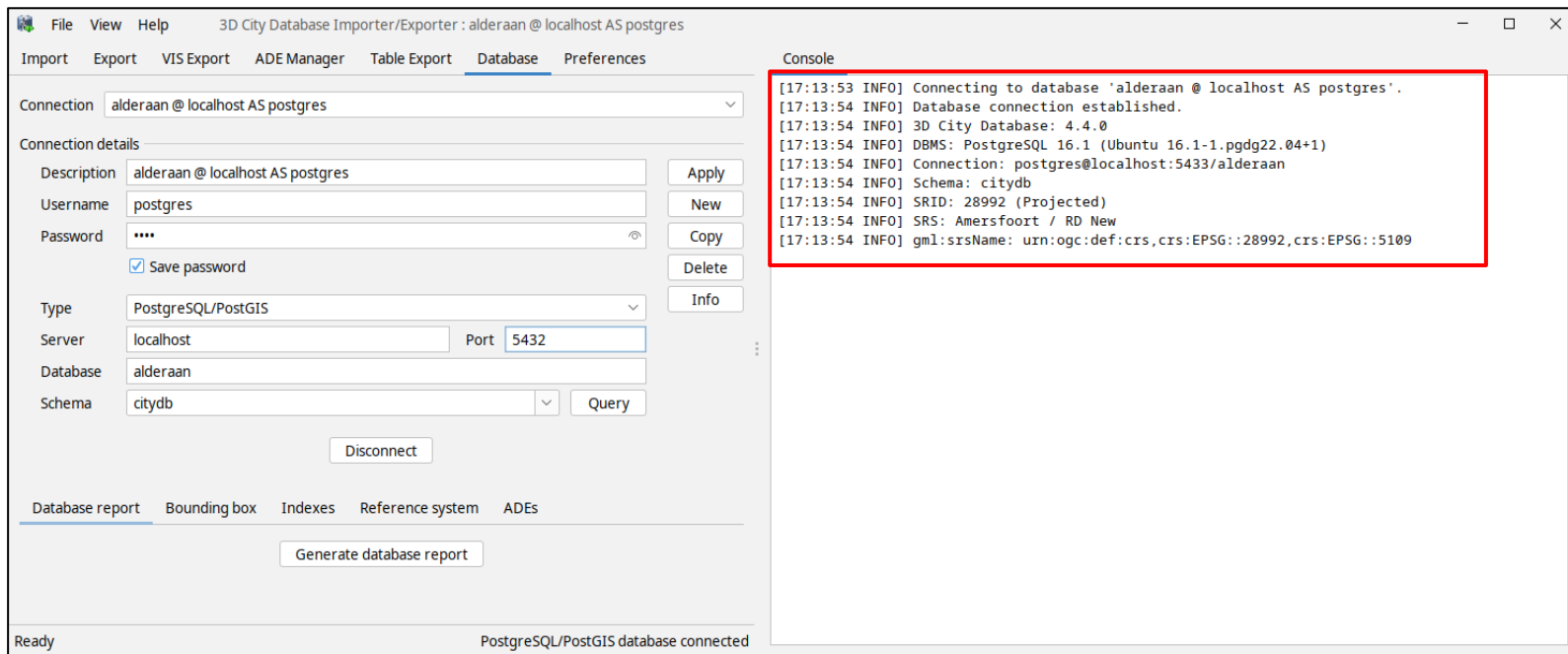
Ready Database disconnected

Software required
Software install
Database setup
Imp/Exp
connection
Additional schemas
ADE plug-ins
Further resources

Connecting to the database via Importer/Exporter

- Launch the 3DCityDB Importer/Exporter, select the "Database" tab
 - Upon successful connection, you will see the notification in the console

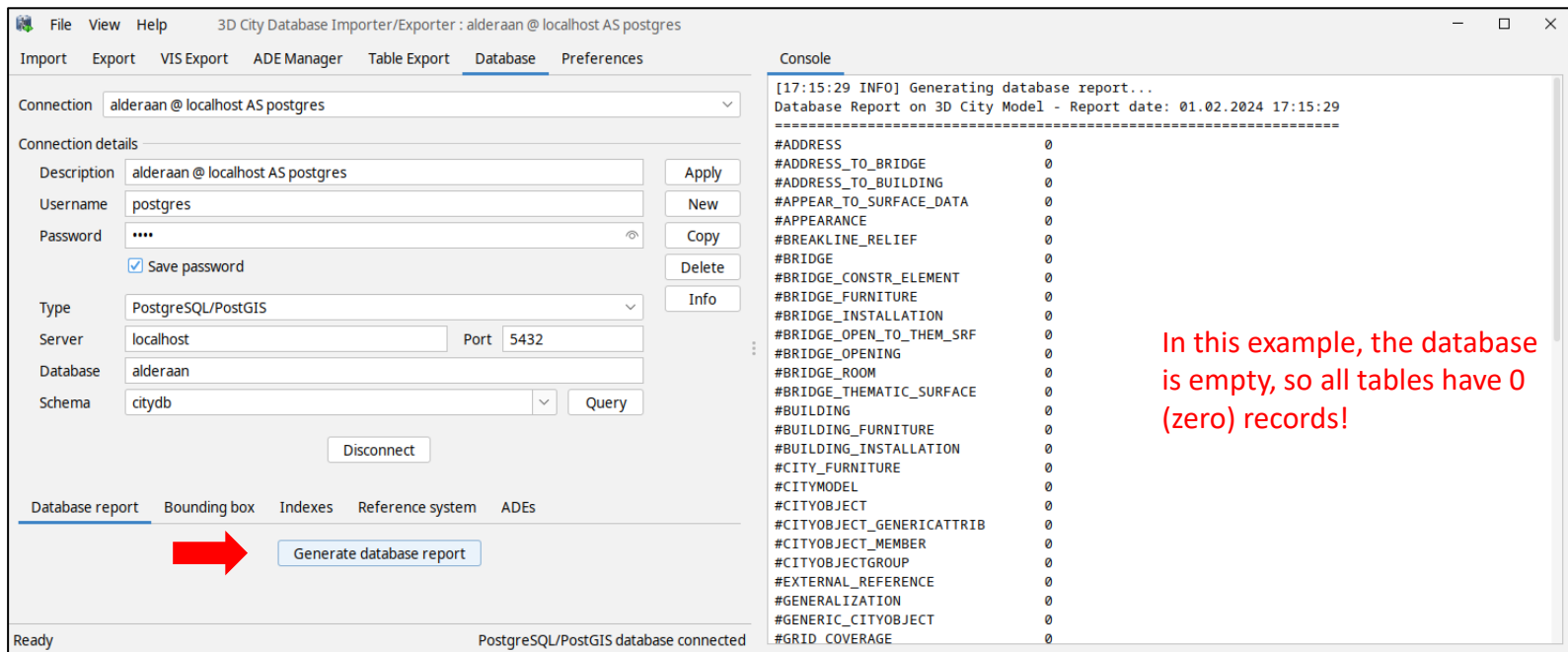
Software required
 Software install
 Database setup
Imp/Exp
connection
 Additional schemas
 ADE plug-ins
 Further resources



Connecting to the database via Importer/Exporter

- Launch the 3DCityDB Importer/Exporter, select the "Database" tab
 - You can optionally also **Generate a database report**
 - If starting from an empty database, it will simply show that all tables are empty! 😊

Software required
Software install
Database setup
Imp/Exp connection
Additional schemas
ADE plug-ins
Further resources



The screenshot shows the 3D City Database Importer/Exporter application. The 'Database' tab is selected, displaying connection details for 'alderaan @ localhost AS postgres'. A red arrow points to the 'Generate database report' button. The console window on the right shows the output of the report, listing various tables and their record counts, all of which are 0.

Connection: alderaan @ localhost AS postgres

Connection details:

- Description: alderaan @ localhost AS postgres
- Username: postgres
- Password: ****
- Save password: ☒
- Type: PostgreSQL/PostGIS
- Server: localhost
- Port: 5432
- Database: alderaan
- Schema: citydb

Buttons: Apply, New, Copy, Delete, Info, Disconnect, Query

Database report | Bounding box | Indexes | Reference system | ADEs

Generate database report

Console:

```
[17:15:29 INFO] Generating database report...
Database Report on 3D City Model - Report date: 01.02.2024 17:15:29
=====
#ADDRESS 0
#ADDRESS_TO_BRIDGE 0
#ADDRESS_TO_BUILDING 0
#APPEAR_TO_SURFACE_DATA 0
#APPEARANCE 0
#BREAKLINE_RELIEF 0
#BRIDGE 0
#BRIDGE_CONSTR_ELEMENT 0
#BRIDGE_FURNITURE 0
#BRIDGE_INSTALLATION 0
#BRIDGE_OPEN_TO_THEM_SRF 0
#BRIDGE_OPENING 0
#BRIDGE_ROOM 0
#BRIDGE_THEMATIC_SURFACE 0
#BUILDING 0
#BUILDING_FURNITURE 0
#BUILDING_INSTALLATION 0
#CITY_FURNITURE 0
#CITYMODEL 0
#CITYOBJECT 0
#CITYOBJECT_GENERICATTRIB 0
#CITYOBJECT_MEMBER 0
#CITYOBJECTGROUP 0
#EXTERNAL_REFERENCE 0
#GENERALIZATION 0
#GENERIC_CITYOBJECT 0
#GRID_COVERAGE 0
```

In this example, the database is empty, so all tables have 0 (zero) records!

Overview

**Install required
software**

**Set up the
database**

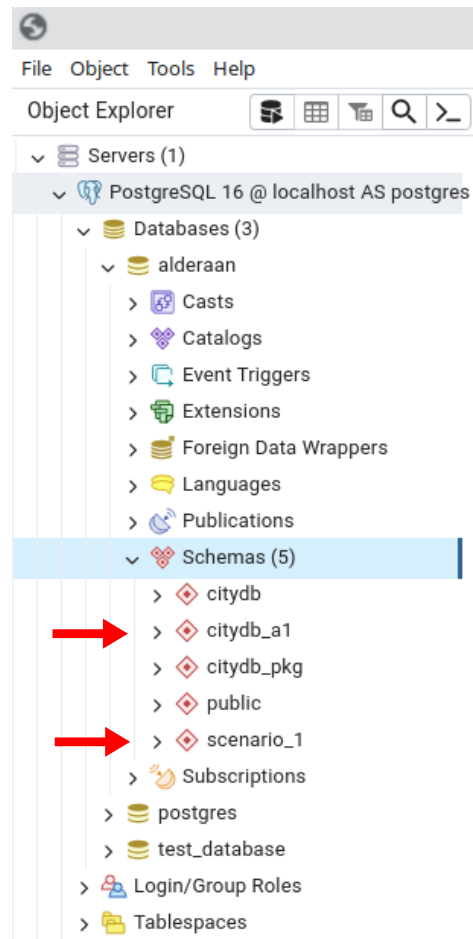
**Connect to the
database via the
Importer/Exporter**

**Add additional
database schemas
(Optional)**

**Install ADE plug-ins
(Optional)**

3D City Database: additional schemas

- **To add additional schemas** (OPTIONAL, besides the default citydb):
 - Go to the same directory where the citydb installation script are. In the command shell run
`chmod u+x CREATE_SCHEMA.sh`
`chmod u+x DROP_SCHEMA.sh`
 - Check the connection parameters in file **CONNECTION_DETAILS.sh** (should be the same as before)
 - Run the script
`./CREATE_SCHEMA.sh`
 You will be requested to enter the name of the additional schema (e.g. "citydb_a1", or "scenario_1", etc.)
 - You can choose any name you want, but try to use only small letters
 - The new schema and its contents will be added automatically. The new schema will have the same CRS of the citydb schema
 - You can repeat these steps and add more schemas to the same database. At the end you will have
 - The citydb schema and n additional schemas
 - Only one citydb_pkg schema

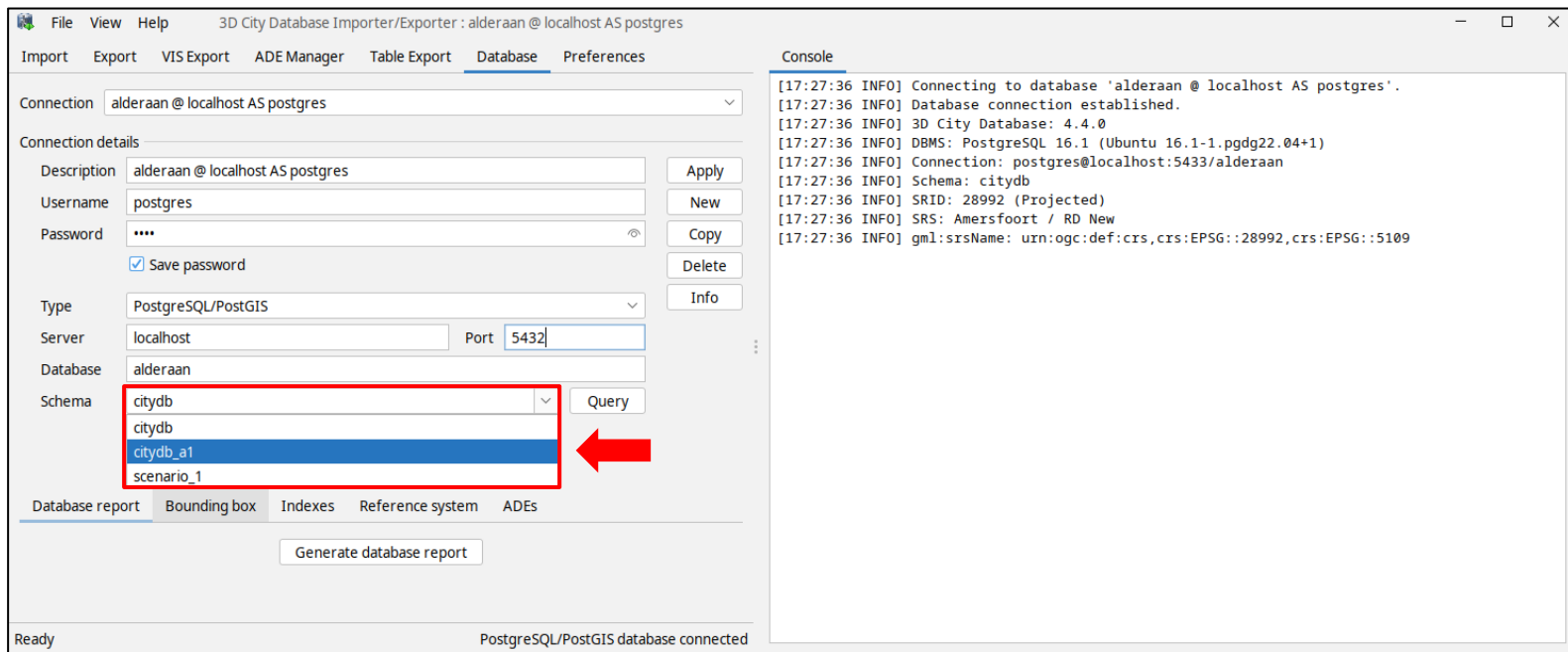


Software required
 Software install
 Database setup
 Imp/Exp connection
Additional schemas
 ADE plug-ins
 Further resources

3D City Database: additional schemas

- When using the Importer/Exporter, you can choose which schema to use to import/export data from the GUI.

Software required
Software install
Database setup
Imp/Exp connection
Additional schemas
ADE plug-ins
Further resources



Overview

**Install required
software**

**Set up the
database**

**Connect to the
database via the
Importer/Exporter**

**Add additional
database schemas
(Optional)**

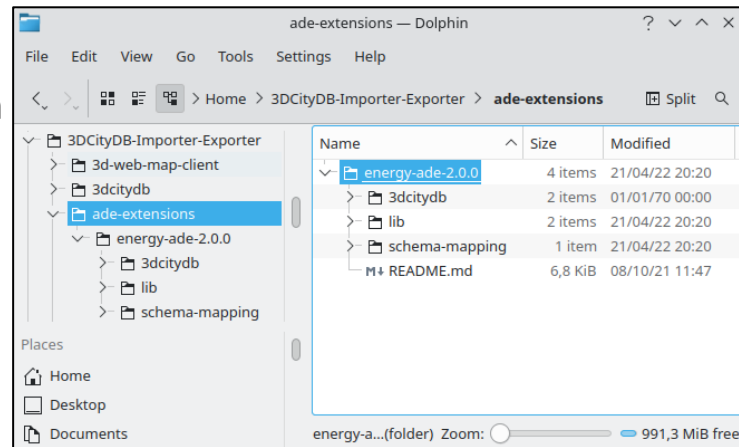
**Install ADE plug-ins
(Optional)**

ADE plugin installation

Please note: These slides refer to the Energy ADE plug-in for the 3DCityDB. However, a similar procedure can be followed for other ADEs.

Software required
Software install
Database setup
Imp/Exp connection
Additional schemas
ADE plug-ins
• Installation
• Data import
• Data export
Further resources

- 1) Download the **energy-ade-citydb** extension for the Importer/Exporter
 - <https://github.com/3dcitydb/energy-ade-citydb/releases/>
- 2) Unzip it in folder ade-extensions of your 3DCityDB install path



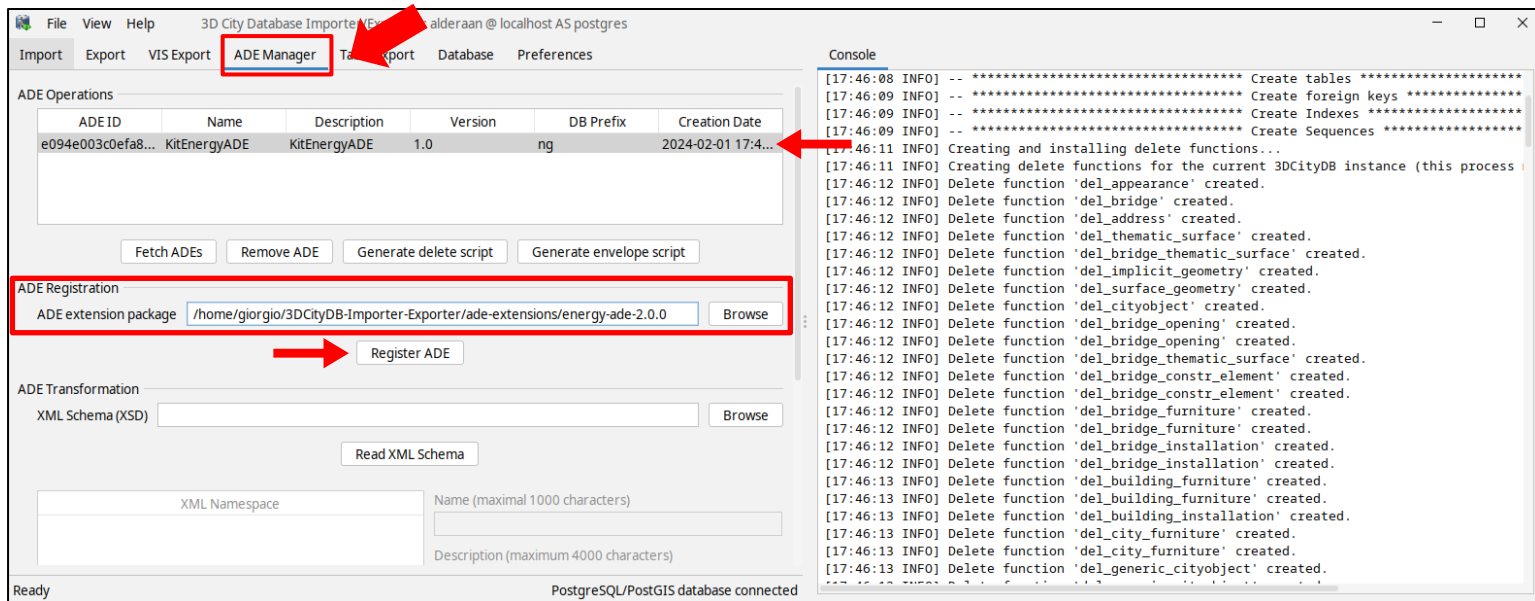
Detailed instructions available on-line

- <https://3dcitydb-docs.readthedocs.io/en/latest/plugins/ade-manager/index.html>

ADE plugin installation

- From the Importer/Exporter, connect to an existing 3DCityDB instance
 - In the "ADE extension package" add the path to the plug-in folder unzipped before
 - "Register" the ADE from the ADE Manager tab
 - The ADE will be added to the ADE list (and all tables, etc. will be added to the current schema)

Software required
 Software install
 Database setup
 Imp/Exp connection
 Additional schemas
ADE plug-ins
 • Installation
 • Data import
 • Data export
 Further resources



The screenshot shows the 3D City Database Importer/Exporter application window. The 'ADE Manager' tab is selected and highlighted with a red box. A red arrow points to the 'ADE Manager' tab. The 'ADE Operations' table lists the installed ADEs:

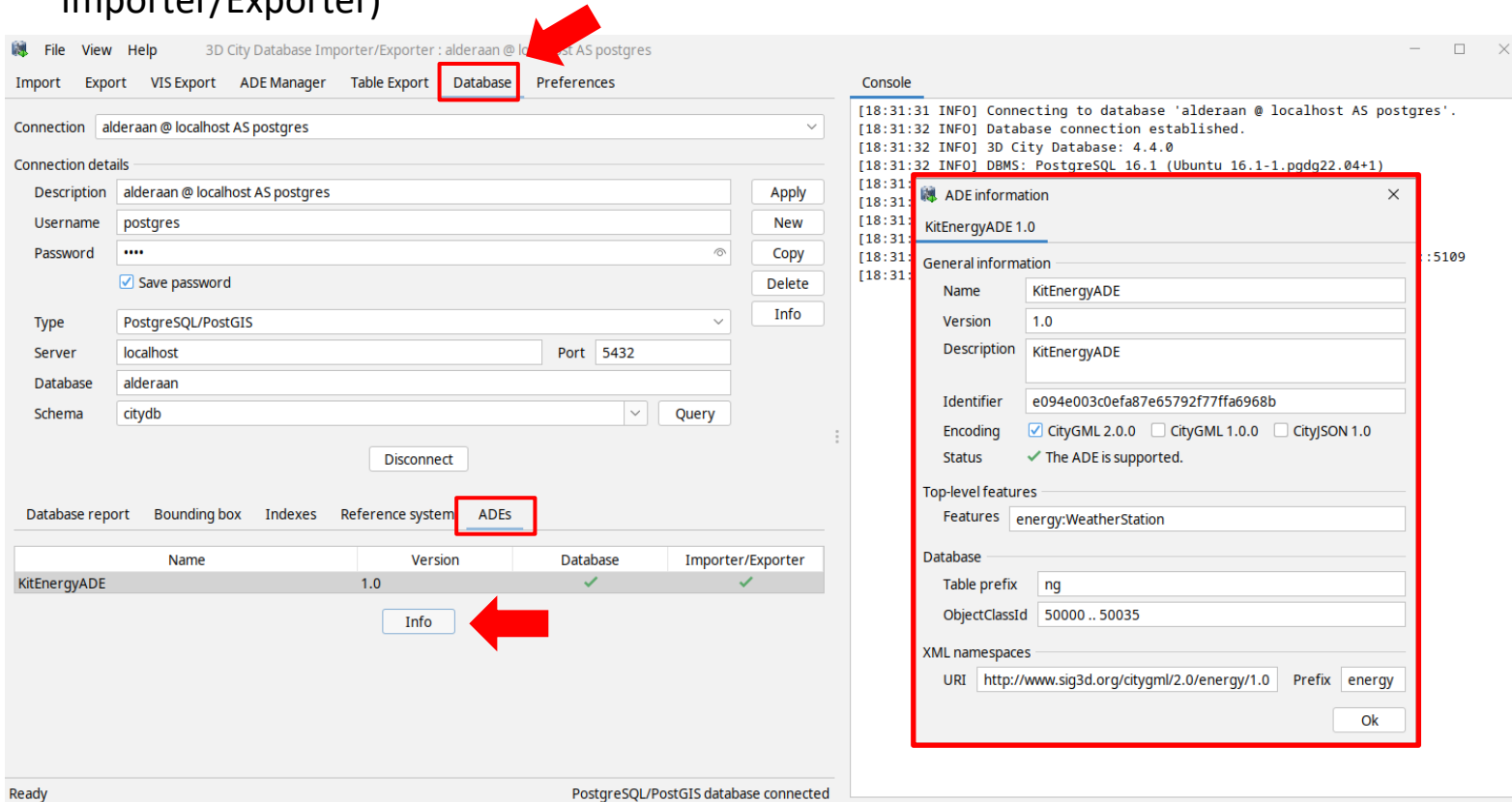
| ADE ID | Name | Description | Version | DB Prefix | Creation Date |
|-------------------|--------------|--------------|---------|-----------|--------------------|
| e094e003c0efa8... | KitEnergyADE | KitEnergyADE | 1.0 | ng | 2024-02-01 17:4... |

Below the table are buttons: 'Fetch ADEs', 'Remove ADE', 'Generate delete script', and 'Generate envelope script'. The 'ADE Registration' section is also visible, with the 'ADE extension package' field containing the path '/home/giorgio/3DCityDB-Importer-Exporter/ade-extensions/energy-ade-2.0.0'. A red box highlights this field and the 'Register ADE' button. A red arrow points to the 'Register ADE' button. The 'Console' window on the right shows the execution of SQL commands to create tables, foreign keys, indexes, and sequences, and then delete functions for the current 3DCityDB instance.

ADE plugin installation

- Check also in the database tab the ADEs properties (you may have to restart the Importer/Exporter)

Software required
Software install
Database setup
Imp/Exp connection
Additional schemas
ADE plug-ins
• Installation
• Data import
• Data export
Further resources



The screenshot shows the '3D City Database Importer/Exporter' application. The 'Database' tab is selected, and the 'ADEs' sub-tab is active. A red arrow points to the 'Database' tab, and another red arrow points to the 'Info' button for the 'KitEnergyADE' entry in the table below.

| Name | Version | Database | Importer/Exporter |
|--------------|---------|----------|-------------------|
| KitEnergyADE | 1.0 | ✓ | ✓ |

The 'ADE information' dialog box for 'KitEnergyADE 1.0' is open, showing the following details:

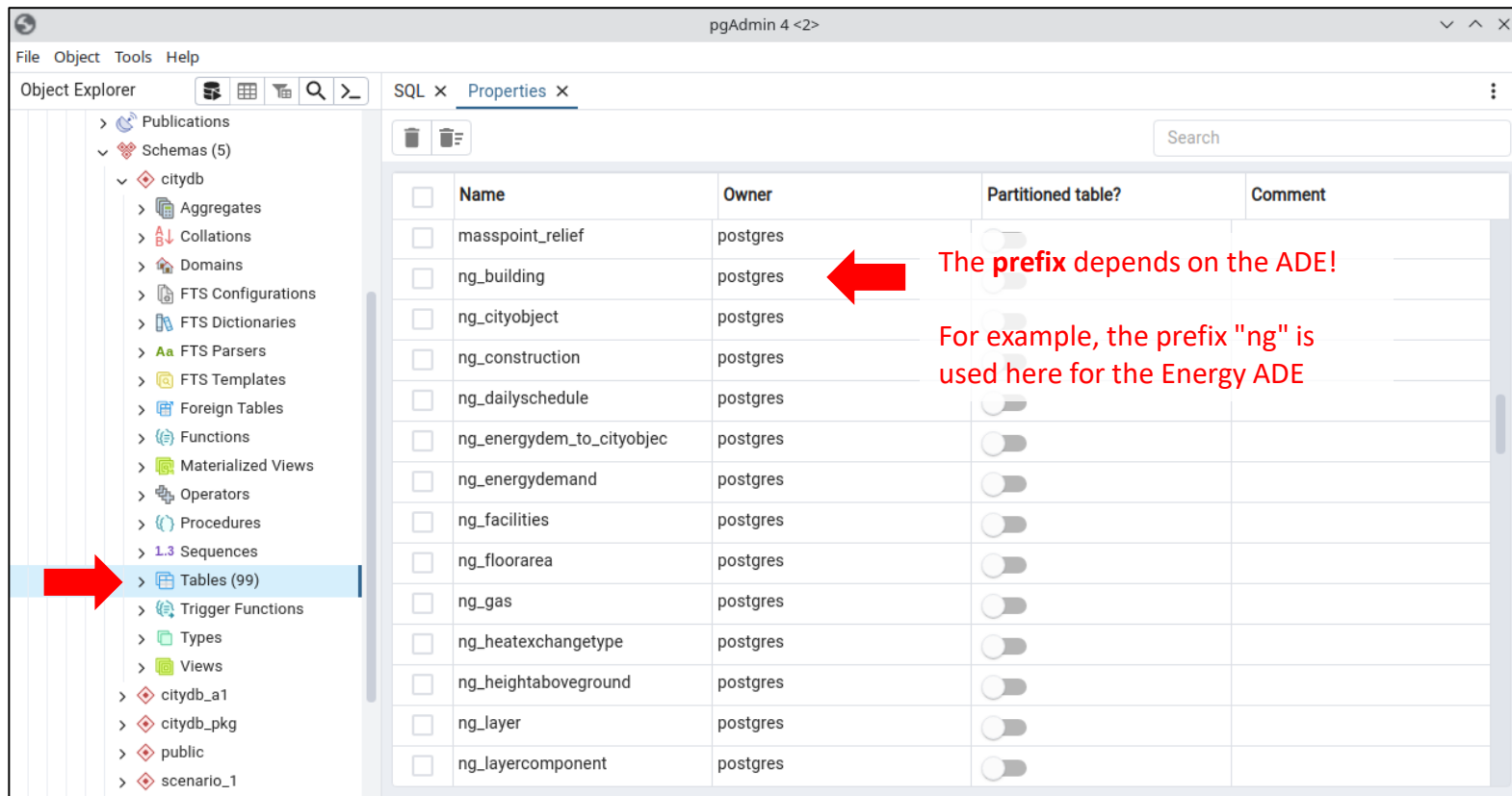
- General information:**
 - Name: KitEnergyADE
 - Version: 1.0
 - Description: KitEnergyADE
 - Identifier: e094e003c0efa87e65792f77ffa6968b
 - Encoding: ☒ CityGML 2.0.0 ☐ CityGML 1.0.0 ☐ CityJSON 1.0
 - Status: ✓ The ADE is supported.
- Top-level features:**
 - Features: energy:WeatherStation
- Database:**
 - Table prefix: ng
 - ObjectClassId: 50000 .. 50035
- XML namespaces:**
 - URI: http://www.sig3d.org/citygml/2.0/energy/1.0
 - Prefix: energy

The 'Ready' status bar at the bottom indicates 'PostgreSQL/PostGIS database connected'.

ADE plugin installation

- Check in pgAdmin: new tables (and functions) with prefix "ng" have been added

Software required
Software install
Database setup
Imp/Exp connection
Additional schemas
ADE plug-ins
• Installation
• Data import
• Data export
Further resources



pgAdmin 4 <2>

File Object Tools Help

Object Explorer

- Publications
- Schemas (5)
 - citydb
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (99)**
 - Trigger Functions
 - Types
 - Views
 - citydb_a1
 - citydb_pkg
 - public
 - scenario_1

SQL x Properties x

| <input type="checkbox"/> | Name | Owner | Partitioned table? | Comment |
|--------------------------|---------------------------|----------|--------------------------|---------|
| <input type="checkbox"/> | masspoint_relief | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_building | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_cityobject | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_construction | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_dailyschedule | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_energydem_to_cityobjec | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_energydemand | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_facilities | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_floorarea | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_gas | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_heatexchangetype | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_heightaboveground | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_layer | postgres | <input type="checkbox"/> | |
| <input type="checkbox"/> | ng_layercomponent | postgres | <input type="checkbox"/> | |

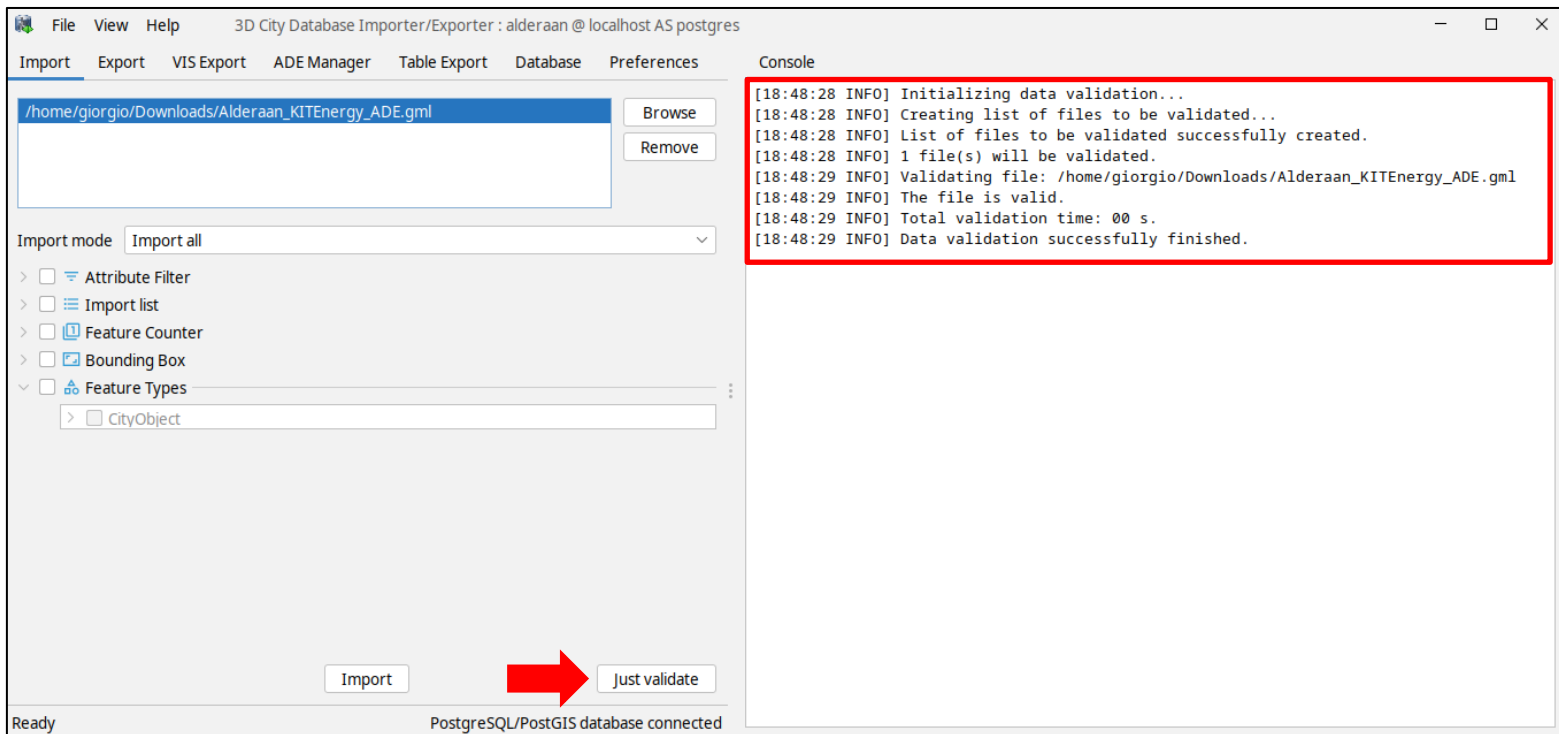
The prefix depends on the ADE!

For example, the prefix "ng" is used here for the Energy ADE

ADE data import

- To import ADE data into the extended 3DCityDB, the procedure is the same as with non-ADE data via the Import tab

Software required
Software install
Database setup
Imp/Exp connection
Additional schemas
ADE plug-ins
• Installation
• **Data import**
• Data export
Further resources



ADE data import

- To import ADE data into the extended 3DCityDB, the procedure is the same as with non-ADE data via the Import tab

Software required

Software install

Database setup

Imp/Exp connection

Additional schemas

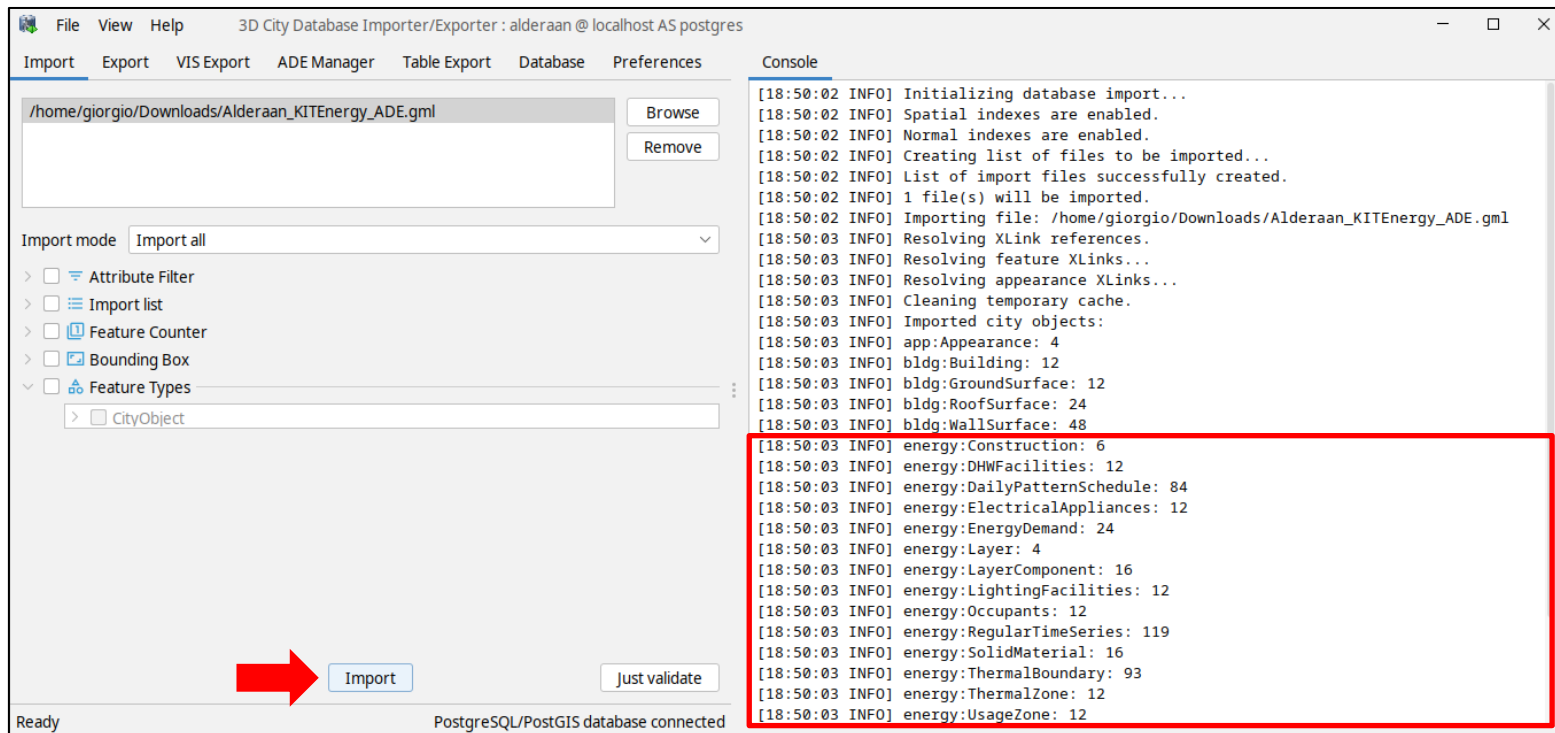
ADE plug-ins

• Installation

• **Data import**

• Data export

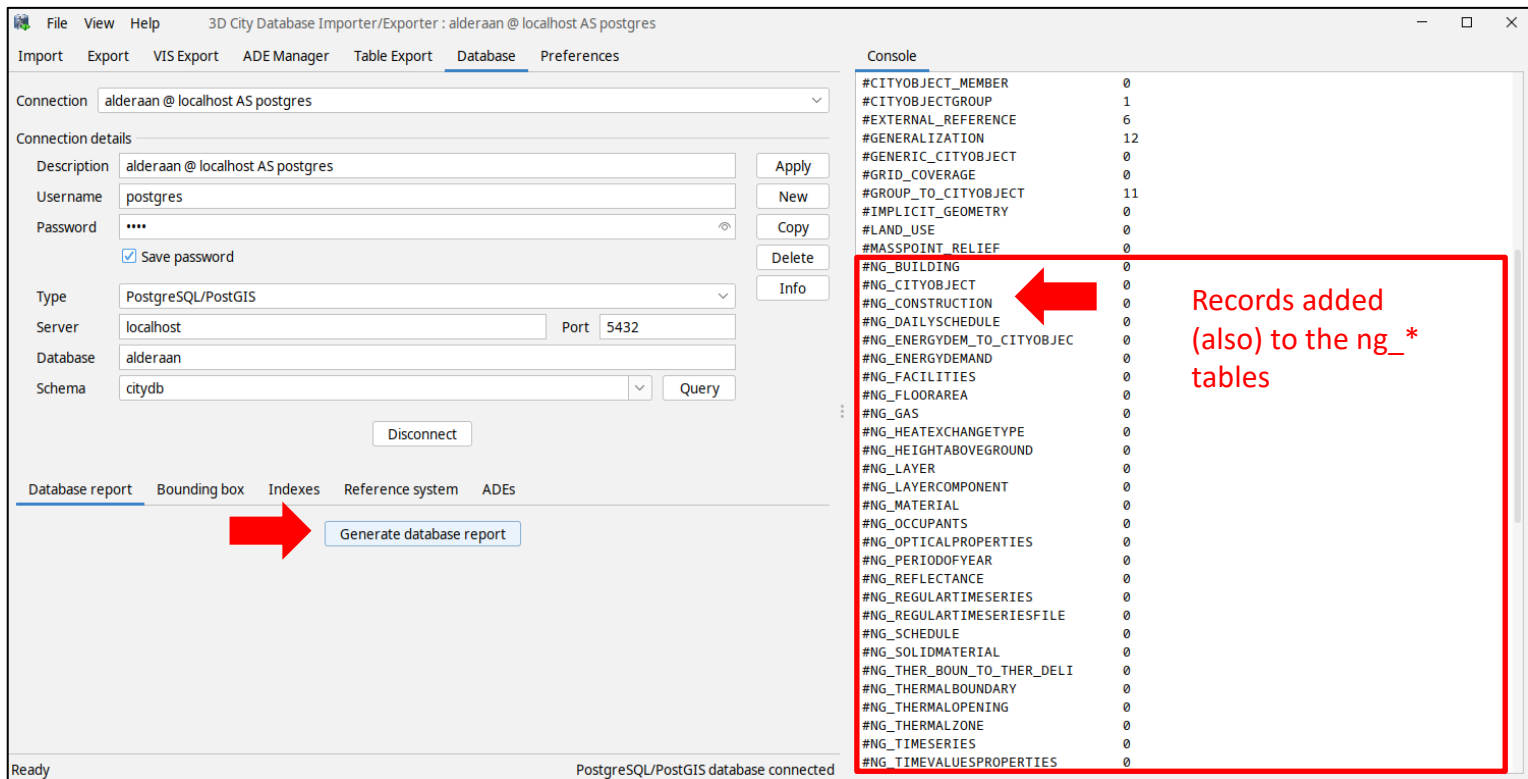
Further resources



ADE data import

- Check also the database report in the database tab!

Software required
Software install
Database setup
Imp/Exp connection
Additional schemas
ADE plug-ins
• Installation
• **Data import**
• Data export
Further resources



The screenshot shows the '3D City Database Importer/Exporter' application window. The 'Database' tab is selected, displaying connection details for 'alderaan @ localhost AS postgres'. A red arrow points to the 'Generate database report' button in the 'Database report' sub-tab. On the right, the 'Console' window shows a list of tables and their record counts. A red box highlights the bottom portion of this list, and a red arrow points to the 'ng_*' tables.

Console Output:

| | |
|----------------------------|----|
| #CITYOBJECT_MEMBER | 0 |
| #CITYOBJECTGROUP | 1 |
| #EXTERNAL_REFERENCE | 6 |
| #GENERALIZATION | 12 |
| #GENERIC_CITYOBJECT | 0 |
| #GRID_COVERAGE | 0 |
| #GROUP_TO_CITYOBJECT | 11 |
| #IMPLICIT_GEOMETRY | 0 |
| #LAND_USE | 0 |
| #MASSPOINT_RELIEF | 0 |
| #NG_BUILDING | 0 |
| #NG_CITYOBJECT | 0 |
| #NG_CONSTRUCTION | 0 |
| #NG_DAILYSCHEDULE | 0 |
| #NG_ENERGYDEM_TO_CITYOBJEC | 0 |
| #NG_ENERGYDEMAND | 0 |
| #NG_FACILITIES | 0 |
| #NG_FLOORAREA | 0 |
| #NG_GAS | 0 |
| #NG_HEATEXCHANGETYPE | 0 |
| #NG_HEIGHTABOVEGROUND | 0 |
| #NG_LAYER | 0 |
| #NG_LAYERCOMPONENT | 0 |
| #NG_MATERIAL | 0 |
| #NG_OCCUPANTS | 0 |
| #NG_OPTICALPROPERTIES | 0 |
| #NG_PERIODOFYEAR | 0 |
| #NG_REFLECTANCE | 0 |
| #NG_REGULARTIMESERIES | 0 |
| #NG_REGULARTIMESERIESFILE | 0 |
| #NG_SCHEDULE | 0 |
| #NG_SOLIDMATERIAL | 0 |
| #NG_THER_BOUN_TO_THER_DELI | 0 |
| #NG_THERMALBOUNDARY | 0 |
| #NG_THERMALOPENING | 0 |
| #NG_THERMALZONE | 0 |
| #NG_TIMESERIES | 0 |
| #NG_TIMEVALUESPROPERTIES | 0 |

Records added (also) to the ng_* tables

ADE data export

- Conceptually analogous to the procedure without ADE content. Simply choose what to export, and run the exporter!

Software required

Software install

Database setup

Imp/Exp connection

Additional schemas

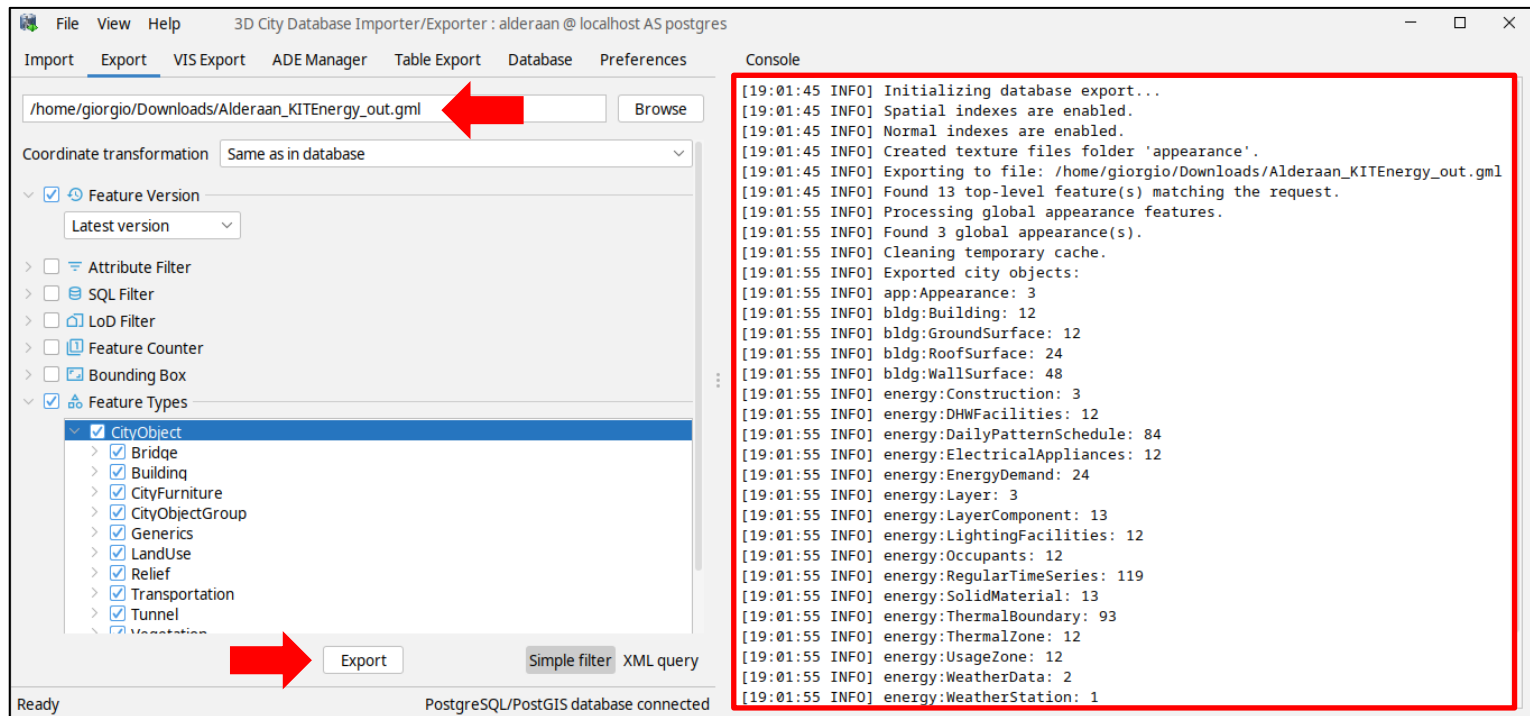
ADE plug-ins

- Installation

- Data import

- Data export**

Further resources



Further resources

- For further information, check the official 3DCityDB documentation regarding the installation procedure details
- **Online documentation**
 - <https://3dcitydb-docs.readthedocs.io/en/latest/>
- **Online tutorial by TU Munich**
 - <https://github.com/3dcitydb/tutorials>

Software required
Software install
Database setup
Imp/Exp connection
Additional schemas
ADE plug-ins
Further resources



Dr. Giorgio Agugiaro

g.agugiaro@tudelft.nl

3D Geoinformation Group

TU Delft

The Netherlands

<https://3d.bk.tudelft.nl/gagugiaro>