**Handleiding database**

Inhoudsopgave

[Installatie postgreSQL(laptop) 3](#_Toc473801695)

[Installatie PostgreSQL (Raspberry Pi) 4](#_Toc473801696)

[Gegevens invoeren in pgAdmin4 6](#_Toc473801697)

[Gegevens toevoegen/verwijderen in pgAdmin4(python) 10](#_Toc473801698)

[Standaard templates voor database 10](#_Toc473801699)

[Gegevens woning toevoegen 10](#_Toc473801700)

[Gegevens noodmeldingen toevoegen (niet hoofdzakelijk) 11](#_Toc473801701)

[Gegevens noodmeldingen clearen 11](#_Toc473801702)

[Gegevens noodmeldingen verwijderen (niet noodzakelijk) 11](#_Toc473801703)

[Gegevens tabel selecteren 12](#_Toc473801704)

[Gegevens jaar/maand/dag van de datum van noodmeldingen selecteren 12](#_Toc473801705)

[Tabellen laatste rij lezen 12](#_Toc473801706)

[Gegevens noodmeldingen toevoegen (rpi) 14](#_Toc473801707)

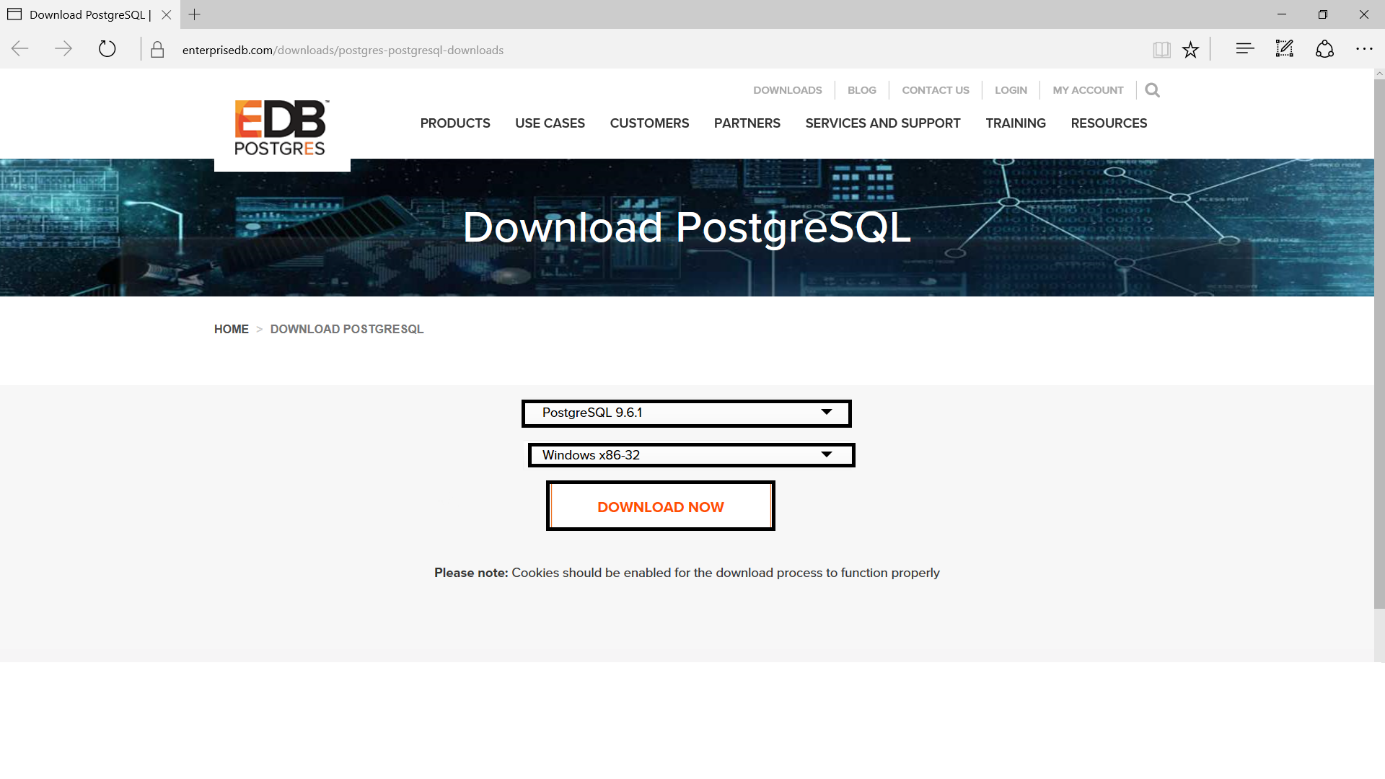
[Bronvermeldingen 15](#_Toc473801708)

# Installatie postgreSQL(laptop)

Versie: PostgreSQL: 9.6

Stappenplan:

1. Ga naar website: <http://www.enterprisedb.com/products/pgdownload.do> kies de juiste gegevens.

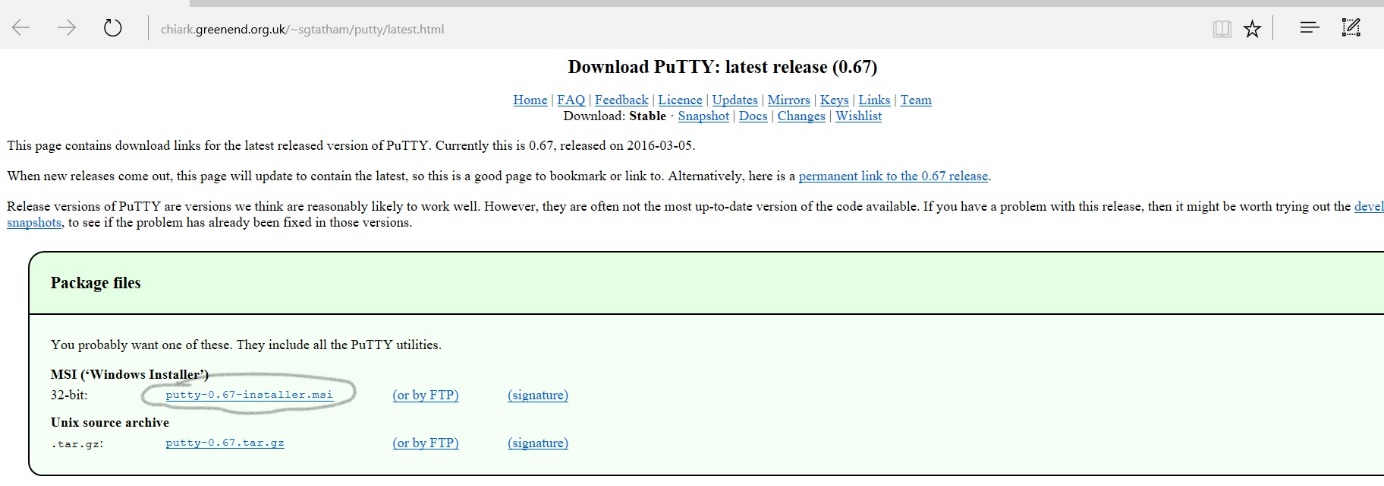


2 Voer het programma uit. Onthoudt tijdens de installatie ook het wachtwoord dat wordt ingevoerd.

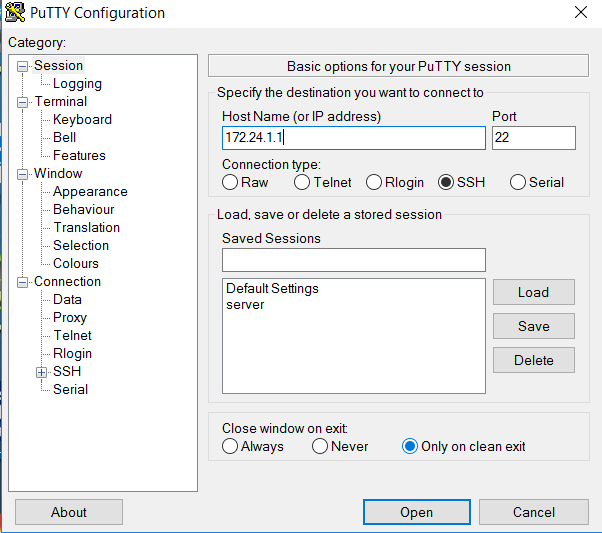
3 Stackbuilder hoef je niet te installeren.

# Installatie PostgreSQL (Raspberry Pi)

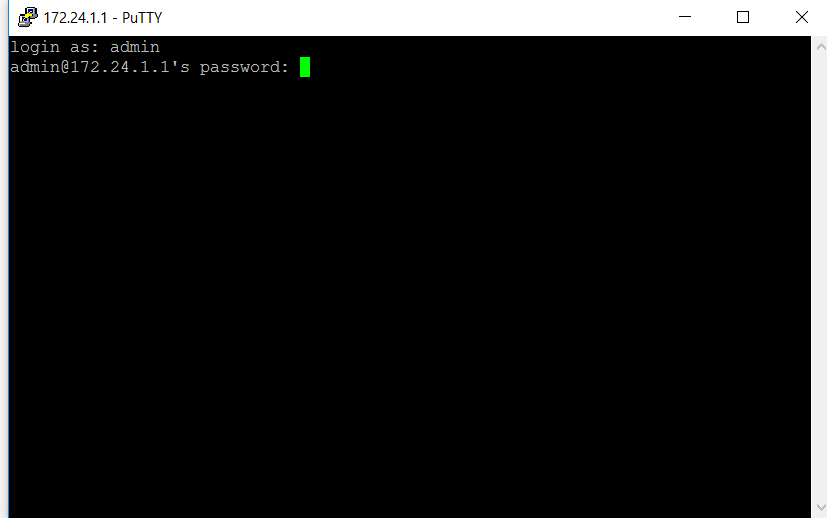
1. Download putty <http://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>



1. Start putty op en vul de host name IP-adres **(172.24.1.1)** in



1. Log in als admin met wachtwoord admin (wachtwoord is niet zichtbaar i.v.m. veiligheid



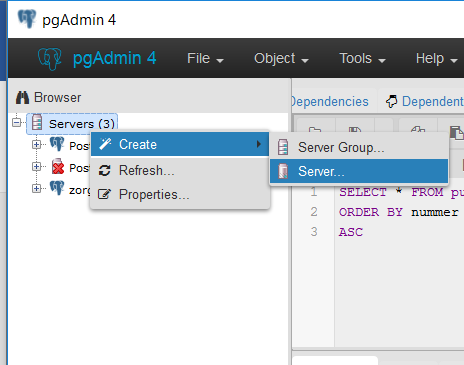
1. Voordat je begint ga je de meest recente updates installeren.Download dan postgreSQL met de naam **postgresql-9.4**.
2. **Installeer de client van PostgreSQL met de naam postgresql-client**
3. Verbind de server met command **psql -h server.domain.org database user**
4. **Sluit de client met command \q**
5. **Maak een pg\_hba.conf** file.
6. **Typ in de file host all all 172.24.0.0/16 md5** dit zorgt ervoor dat je toegang kan krijgen bij dat IP-adres (de IP-adres zal naar 192.168 moeten veranderen als de router bijv. 129144). Na het veranderen van het bestand, wordt het opgeslagen.
7. **Maak ook een back-up van het bestand ‘pg\_hba.conf’, de back-up is handig als er iets mis gaat.**
8. **Ga dan naar de bestand postgresql.conf. De volgende gegevens verander je van listen\_address = ‘localhost naar listen\_address = ‘\*’** nu accepteert postgreSQL verbindingen van elk IP-adres.
9. Download pgAdmin 3 in de rpi met de naam **pgadmin3**
10. **Met command sudo -u postgres psql postgres** kom je in het programma.
11. **Met command \password postgres** vul je een wachtwoord naar keuze in (bijv. admin)
12. Maak nu een database met de naam **zorggroep.**

Voor een volledigere uitleg ga naar <http://blog.bpwalters.com/setting-up-postgresql-on-raspberry-pi/>**[[1]](#footnote-1)** en <https://help.ubuntu.com/community/PostgreSQL>[[2]](#footnote-2)

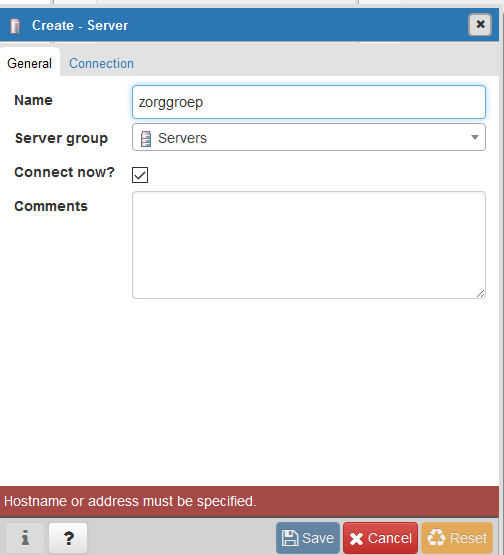
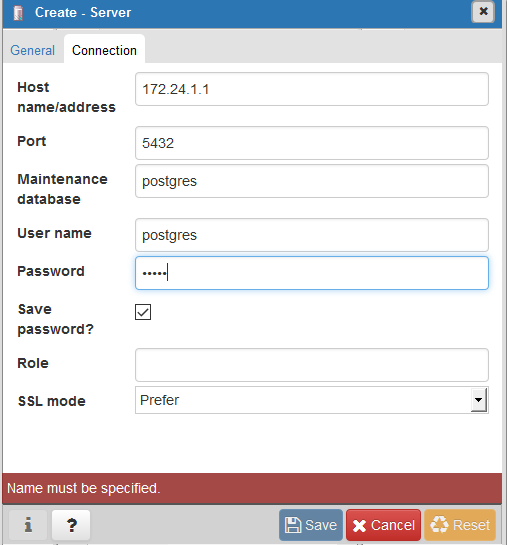
# 

# Gegevens invoeren in pgAdmin4

1. Zet de wifi op PI3-AP en vul het wachtwoord admin in om de Raspberry Pi (rpi) te verbinden (als het nog niet gedaan is)
2. Start pgadmin 4 op je laptop
3. Maak een nieuwe server

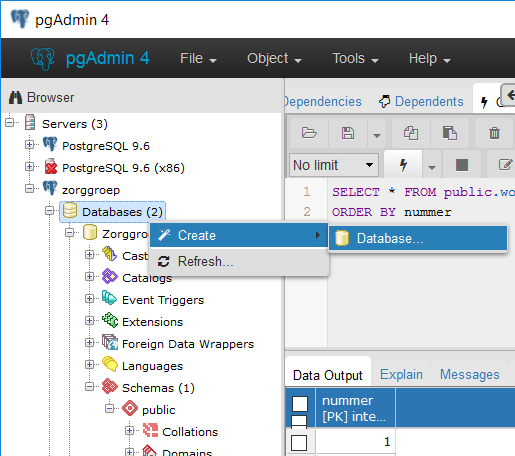
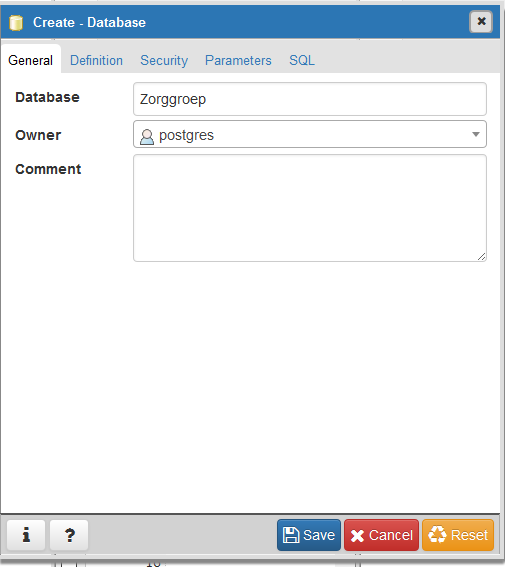


1. Vul de volgende gegevens in:

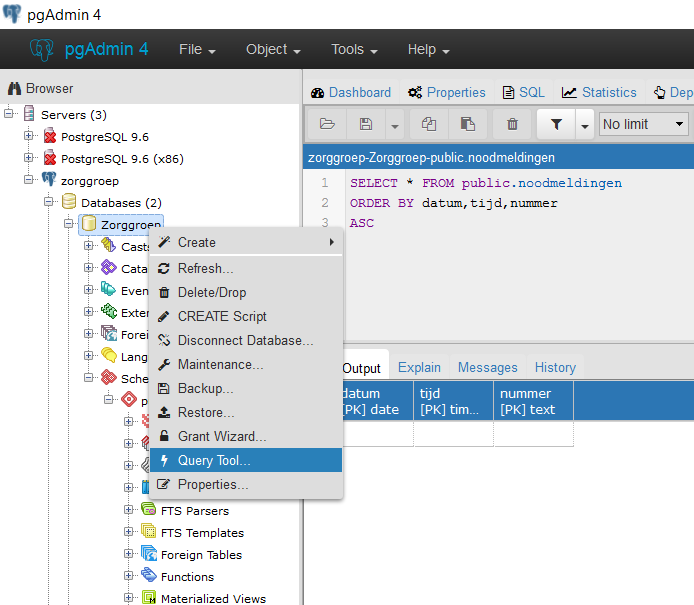
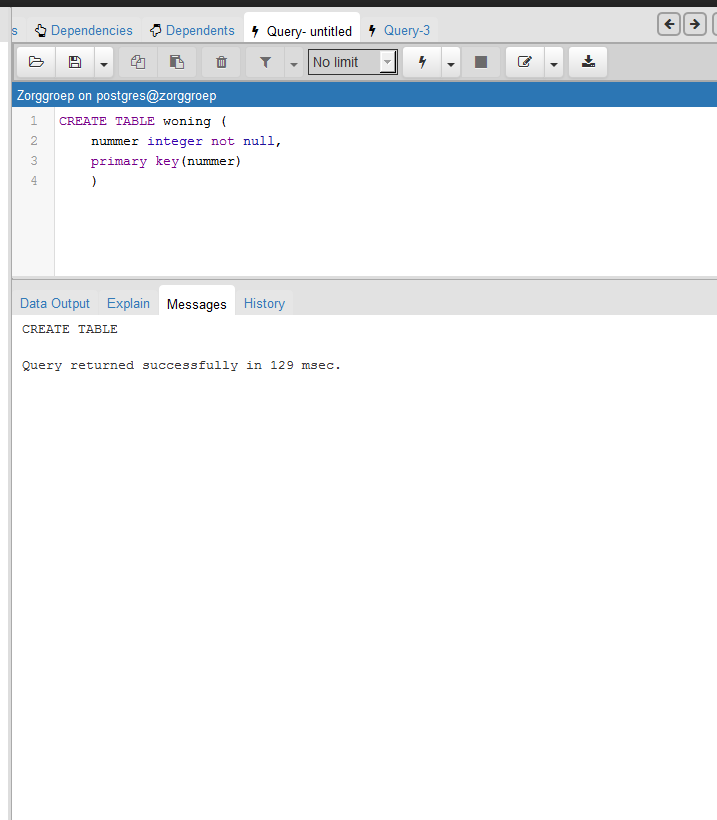
 

Wachtwoord is admin

1. Maak een database Zorggroep en vul de naam van de database in:

1. Ga naar de query van de database en vul in:

**CREATE TABLE woning (**

**Nummer text not null,**

**Primary key(nummer)**

**)**

**CREATE TABLE noodmeldingen (**

**Datum date not null,**

**Tijd time not null,**

**Primary key (datum, tijd, nummer),**

**Nummer integer references woning(nummer)**

**)**

**CREATE TABLE users (**

**Id integer not null,**

**Username varchar (255) not null,**

**Password varchar (255) not null, primary key(id)**

**)**

**INSERT INTO woning (nummer) VALUES ('woning-1');**

**INSERT INTO woning (nummer) VALUES ('woning-2');**

**INSERT INTO woning (nummer) VALUES ('woning-3');**

**INSERT INTO woning (nummer) VALUES ('woning-4');**

**INSERT INTO woning (nummer) VALUES ('woning-5');**

**INSERT INTO woning (nummer) VALUES ('woning-6');**

**INSERT INTO woning (nummer) VALUES ('woning-7');**

**INSERT INTO woning (nummer) VALUES ('woning-8');**

**INSERT INTO woning (nummer) VALUES ('woning-9');**

**INSERT INTO woning (nummer) VALUES ('woning-10');**

**INSERT INTO woning (nummer) VALUES ('woning-11');**

**INSERT INTO woning (nummer) VALUES ('woning-12');**

**INSERT INTO woning (nummer) VALUES ('woning-13');**

**INSERT INTO woning (nummer) VALUES ('woning-14');**

**INSERT INTO woning (nummer) VALUES ('woning-15');**

**INSERT INTO woning (nummer) VALUES ('woning-16');**

**INSERT INTO woning (nummer) VALUES ('woning-17');**

**INSERT INTO woning (nummer) VALUES ('woning-18');**

**INSERT INTO woning (nummer) VALUES ('woning-19');**

**INSERT INTO woning (nummer) VALUES ('woning-20');**

**INSERT INTO woning (nummer) VALUES ('woning-21');**

**INSERT INTO woning (nummer) VALUES ('woning-22');**

**INSERT INTO woning (nummer) VALUES ('woning-23');**

**INSERT INTO woning (nummer) VALUES ('woning-24');**

**INSERT INTO woning (nummer) VALUES ('woning-25');**

**INSERT INTO woning (nummer) VALUES ('woning-26');**

**INSERT INTO woning (nummer) VALUES ('woning-27');**

**INSERT INTO woning (nummer) VALUES ('woning-28');**

**INSERT INTO woning (nummer) VALUES ('woning-29');**

**INSERT INTO woning (nummer) VALUES ('woning-30');**

**INSERT INTO woning (nummer) VALUES ('woning-31');**

**INSERT INTO woning (nummer) VALUES ('woning-32');**

**INSERT INTO woning (nummer) VALUES ('woning-33');**

**INSERT INTO woning (nummer) VALUES ('woning-34');**

**INSERT INTO woning (nummer) VALUES ('woning-35');**

**INSERT INTO woning (nummer) VALUES ('woning-36');**

**INSERT INTO woning (nummer) VALUES ('woning-37');**

**INSERT INTO woning (nummer) VALUES ('woning-38');**

**INSERT INTO woning (nummer) VALUES ('woning-39');**

**INSERT INTO woning (nummer) VALUES ('woning-40');**

**INSERT INTO woning (nummer) VALUES ('woning-41');**

**INSERT INTO woning (nummer) VALUES ('woning-42');**

**INSERT INTO woning (nummer) VALUES ('woning-43');**

**INSERT INTO woning (nummer) VALUES ('woning-44');**

**INSERT INTO woning (nummer) VALUES ('woning-45');**

**INSERT INTO woning (nummer) VALUES ('woning-46');**

**INSERT INTO woning (nummer) VALUES ('woning-47');**

**INSERT INTO woning (nummer) VALUES ('woning-48');**

**INSERT INTO woning (nummer) VALUES ('woning-49');**

**INSERT INTO woning (nummer) VALUES ('woning-50');**

**INSERT INTO users (id, username, password) VALUES (1, 'zorggroep', 'domotica')[[3]](#footnote-3)**

# Gegevens toevoegen/verwijderen in pgAdmin4(python)

Voordat er wordt gedaan aan python wordt Python 3 en pycharm gebruikt om de codes in te voeren. Na het installeren moeten de packages psycopg2 en datetime worden geïnstalleerd.

## Standaard templates voor database

Dit is de standaard code van elk database commando code.

**import** psycopg2 **as** psy2  
con = psy2.connect(**"dbname=Zorggroep user=postgres host=172.24.1.1 password=admin"**)  
dbname = **"Zorggroep"**cur = con.cursor()

#cur.execute(**""**)con.commit()  
cur.close()  
con.close()

Voor meer uitleg wat elk functie doet ga naar <https://www.tutorialspoint.com/postgresql/postgresql_python.htm>[[4]](#footnote-4) of <http://initd.org/psycopg/docs/usage.html> [[5]](#footnote-5)

## Gegevens woning toevoegen

De code is nodig wanneer er meer dan 50 woningen zijn.

**import** psycopg2 **as** psy2  
con = psy2.connect(**"dbname=Zorggroep user=postgres host=172.24.1.1 password=admin"**)  
dbname = **"Zorggroep"**cur = con.cursor()

cur.execute(**"INSERT INTO woning (nummer) VALUES (50)"**)[[6]](#footnote-6)con.commit()  
cur.close()  
con.close()

## Gegevens noodmeldingen toevoegen (niet hoofdzakelijk)

De code kan worden gebruikt in python op de laptop maar is niet noodzakelijk. De code in de Raspberry Pi gaat daar automatisch.

**import** psycopg2 **as** psy2  
**import** datetime[[7]](#footnote-7)  
con = psy2.connect(**"dbname=Zorggroep user=postgres host=172.24.1.1 password=admin"**)  
dbname = **"Zorggroep"**cur = con.cursor()  
tijd = datetime.datetime.today()  
nu\_datum = tijd.strftime(**"%d %B %Y"**)  
nu\_tijd = tijd.strftime(**"%H:%M:%S"**)  
print(nu\_datum)  
print(nu\_tijd)

nummer = 1

cur.execute(**"INSERT INTO noodmeldingen (datum, tijd, nummer) VALUES (%s, %s, %s, %s)"**(nu\_datum, nu\_tijd, nummer))  
con.commit()cur.close()  
con.close()

## Gegevens noodmeldingen clearen

Met die code verwijder je alles in de tabel.

**import** psycopg2 **as** psy2  
**import** datetime  
con = psy2.connect(**"dbname=Zorggroep user=postgres host=172.24.1.1 password=admin"**)  
dbname = **"Zorggroep"**cur = con.cursor()  
tijd = datetime.datetime.today()  
nu\_datum = tijd.strftime(**"%d %B %Y"**)  
psy2.extensions.register\_type(psy2.extensions.UNICODE, cur)  
cur.execute(**"truncate noodmeldingen"**)  
cur.execute(**"SELECT \* FROM noodmeldingen"**)  
**for** i **in** cur:  
 print(i)  
con.commit()  
cur.close()  
con.close()

## Gegevens noodmeldingen verwijderen (niet noodzakelijk)

De code hieronder is niet noodzakelijk als het hele bestand moet worden verwijderd

**import** psycopg2 **as** psy2  
**import** datetime  
con = psy2.connect(**"dbname=Zorggroep user=postgres host=172.24.1.1 password=admin"**)  
dbname = **"Zorggroep"**cur = con.cursor()  
tijd = datetime.datetime.today()  
nu\_datum = tijd.strftime(**"%d %B %Y"**)  
psy2.extensions.register\_type(psy2.extensions.UNICODE, cur)  
cur.execute(**"delete from noodmeldingen where datum != 2017-02-1"**)  
cur.execute(**"SELECT \* FROM noodmeldingen"**)  
**for** i **in** cur:  
 print(i)  
con.commit()  
cur.close()  
con.close()

## Gegevens tabel selecteren

Met die code wordt bepaalde gegevens van alle kolommen van een tabel weergegeven

**import** psycopg2 **as** psy2  
con = psy2.connect(**"dbname=Zorggroep user=postgres host=172.24.1.1 password=admin"**)  
dbname = **"Zorggroep"**cur = con.cursor()  
psy2.extensions.register\_type(psy2.extensions.UNICODE, cur)  
cur.execute("**SELECT \* FROM woning**") ***# noodmeldingen, users***  
for i in cur:  
 print(i)con.commit()  
cur.close()  
con.close()

## Gegevens jaar/maand/dag van de datum van noodmeldingen selecteren

Hierin kunnen bepaalde gegevens binnen een kolom geselecteerd worden.

**import** psycopg2 **as** psy2  
con = psy2.connect(**"dbname=Zorggroep user=postgres host=172.24.1.1 password=admin"**)  
dbname = **"Zorggroep"**cur = con.cursor()  
psy2.extensions.register\_type(psy2.extensions.UNICODE, cur)

cur.execute(**"select extract(year from datum), nummer from noodmeldingen"**)

***# extract(month from datum), extract(day from datum)***  
**for** i **in** cur:  
 print(i)  
con.commit()  
cur.close()  
con.close()

## Tabellen laatste rij lezen

Met die code wordt er in een file alle gegevens van de tabel ‘noodmeldingen’ geschreven. Daarna selecteert Python welk noodmelding het laatst is toegevoegd.

**import** psycopg2 **as** psy2  
con = psy2.connect(**"dbname=Zorggroep user=postgres host=172.24.1.1 password=admin"**)  
dbname = **"Zorggroep"**cur = con.cursor()  
cur.copy\_to(open(**database.txt'**, **'w'**), **'noodmeldingen'**, sep=**'|'**)[[8]](#footnote-8)  
con.commit()  
cur.close()  
con.close()  
**def** reading\_last\_line():  
 **with** open(**database.txt'**, **'r'**) **as** read:  
 print(read.readlines()[-1])  
  
reading\_last\_line()

# Gegevens noodmeldingen toevoegen (rpi)

Zie document: .docx

# Bronvermeldingen

Tutorialspoint, 2017, PostgreSQL – Python Interface, geraadpleegd op 23-01-2017 <https://www.tutorialspoint.com/postgresql/postgresql_python.htm>

Federico Di Gregorio en Daniele Varazzo, 2016, basic module usage, geraadpleegd op 22-01-2017 <http://initd.org/psycopg/docs/usage.html>

Mikini, 28-08-2015, PostgreSQL, geraadpleegd op 27-01-2017 <https://help.ubuntu.com/community/PostgreSQL>

D. Obbink, 14-12-2016, reader modelleren SQL, geraadpleegd op 23-01-2017 <https://hubl.hu.nl/archive/file/476243>

Python software foundation, 01-01-2017, 8.1 datetime – basic date and time types, geraadpleegd op 30-01-2017 <https://docs.python.org/2/library/datetime.html>

M.Jansen en D.A. Pronk, jaar van uitgave:?, Reader modelleren Installeren Postgres, geraadpleegd op 30-01-2017

<https://hubl.hu.nl/session/les-9-16?overview=530659> in 07 SQL Installatie handleiding postgres.docx

Ben Walters, 28-04-2016, setting up PostgreSQL on Raspberry Pi, geraadpleegd op 29-01-2017

<http://blog.bpwalters.com/setting-up-postgresql-on-raspberry-pi/>

Federico Di Gregorio en Daniele Varazzo, 2016, The cursor class, geraadpleegd op 31-01-2017 <http://initd.org/psycopg/docs/cursor.html#cursor.copy_to>

1. Ben Walters, 28-04-2016, setting up PostgreSQL on Raspberry Pi, geraadpleegd op 29-01-2017

   <http://blog.bpwalters.com/setting-up-postgresql-on-raspberry-pi/> [↑](#footnote-ref-1)
2. Mikini, 28-08-2015, PostgreSQL, geraadpleegd op 27-01-2017 <https://help.ubuntu.com/community/PostgreSQL> [↑](#footnote-ref-2)
3. M.Jansen en D.A. Pronk, jaar van uitgave:?, Reader modelleren Installeren Postgres, geraadpleegd op 30-01-2017 <https://hubl.hu.nl/session/les-9-16?overview=530659> in 07 SQL Installatie handleiding postgres.docx [↑](#footnote-ref-3)
4. Tutorialspoint, 2017, PostgreSQL – Python Interface, geraadpleegd op 23-01-2017 <https://www.tutorialspoint.com/postgresql/postgresql_python.htm> [↑](#footnote-ref-4)
5. Federico Di Gregorio en Daniele Varazzo, 2016, basic module usage, geraadpleegd op 22-01-2017 <http://initd.org/psycopg/docs/usage.html> [↑](#footnote-ref-5)
6. D. Obbink, 14-12-2016, reader modelleren SQL, geraadpleegd op 23-01-2017 <https://hubl.hu.nl/archive/file/476243>  [↑](#footnote-ref-6)
7. Python software foundation, 01-01-2017, 8.1 datetime – basic date and time types, geraadpleegd op 30-01-2017 <https://docs.python.org/2/library/datetime.html> [↑](#footnote-ref-7)
8. Federico Di Gregorio en Daniele Varazzo, 2016, The cursor class, geraadpleegd op 31-01-2017 <http://initd.org/psycopg/docs/cursor.html#cursor.copy_to> [↑](#footnote-ref-8)