Αυτοματοποιημένη Διαχείριση Συστημάτων

Διάλεξη 6η

Θωμάς Καμαλάκης

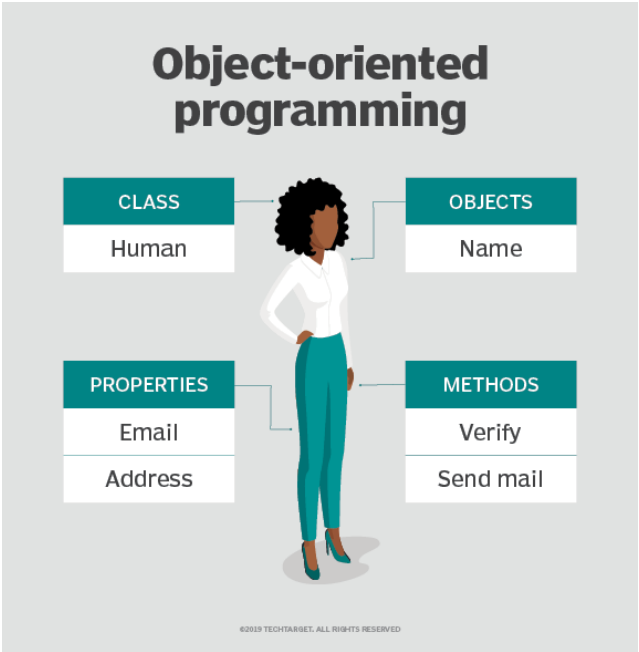
Χαροκόπειο Πανεπιστήμιο Αθηνών

Οκτώβριος 2022

Περιεχόμενα

Κλάσεις

- Η Python είναι μία *αντικειμενοστραφής* γλώσσα προγραμματισμού.
- Κάθε πράγμα που χρησιμοποιούμε άλλωστε στην Python είναι ένα *αντικείμενο*.
- Κάθε αντικείμενο έχει τα δικά του γνωρίσματα και μεθόδους.



Κλάσεις

- Κλάσεις: τύποι δεδομένων που ορίζονται από τον χρήστη.
- Αντικείμενα: Μεταβλητές που δημιουργούνται από τις κλάσεις.
- Γνωρίσματα: Εσωτερικές μεταβλητές που καθορίζουν την κατάσταση ενός αντικειμένου.
- Μέθοδοι: Συναρτήσεις που ορίζονται στην κλάση και καθορίζουν την συμπεριφορά του αντικειμένου.

Κλάσεις

- Ενθυλάκωση (encapsulation): Όλες οι απαραίτητες πληροφορίες για ένα αντικείμενο βρίσκονται μέσα του.
- Αφαίρεση (abstraction): Δεν πρέπει να μας ενδιαφέρει εσωτερικά πως είναι η κλάση όταν την χρησιμοποιούμε.
- Κληρονομικότητα (inheritance): Οι κλάσεις μπορεί να βασίζονται σε άλλες κλάσεις και να επαναχρησιμοποιούν στοιχεία τους.
- Πολυμορφισμός: Αν δύο κλάσεις έχουν παρόμοιες μεθόδους μπορούμε να τις διαχειριστούμε με τον ίδιο τρόπο.

Δημιουργία αντικειμένων

- Όταν δημιουργούμε ένα αντικείμενο που ανήκει σε μία κλάση, καλείται η __init__()

```
6  class user:
7
8      def __init__(self, name, surname, email, password):
9          self.name = name
10         self.surname = surname
11         self.email = email
12         self.password = password
13         self.display_name = name + " " + surname
```

Listing: user.py

- Η μέθοδος __str__ καθορίζει τι τυπώνεται όταν το αντικείμενο συνδυάζεται με την print()

```
16      def __str__(self):
17          return self.display_name + ' (' + self.email + ')'
```

Listing: user.py

```
1  # import the user class
2  from user import user
3
4  # create a user object
5  a = user('Thomas', 'Kamalakis', 'thkam@hua.gr', 'hua123')
6
7  # print the user object
8  print(a)
```

```
thomas@kirkhome2:~/Documents/mscpython/code/lecture6$ python3 -i userex.py
Thomas Kamalakis (thkam@hua.gr)
```

Ορισμός Μεθόδων

- Δεν είναι λίγο περίεργο να αποθηκεύουμε το password απευθείας;
- Ας το κάνουμε hash

```
21      # Update the user password hash
22      def set_password(self,plain_text):
23          h = sha256( plain_text.encode() )
24          self.password = h.hexdigest()
```

Listing: user.py

```
7   class user:
8
9       def __init__(self, name, surname, email, password):
10          self.name = name
11          self.surname = surname
12          self.email = email
13          self.display_name = name + " " + surname
14          self.set_password( password )
```

Listing: user.py

Ορισμός Μεθόδων

```
28      # verify password
29      def verify_password(self, plain_text):
30          h = sha256( plain_text.encode() )
31          return self.password == h.hexdigest()
32  #---checkpassword2
33
34  #---asdict1
35      def as_dict(self):
36          return {
37              'name' : self.name,
38              'surname' : self.surname,
39              'email' : self.email,
40              'display_name' : self.display_name,
41              'password' : self.password
42          }
43  #---asdict2
```

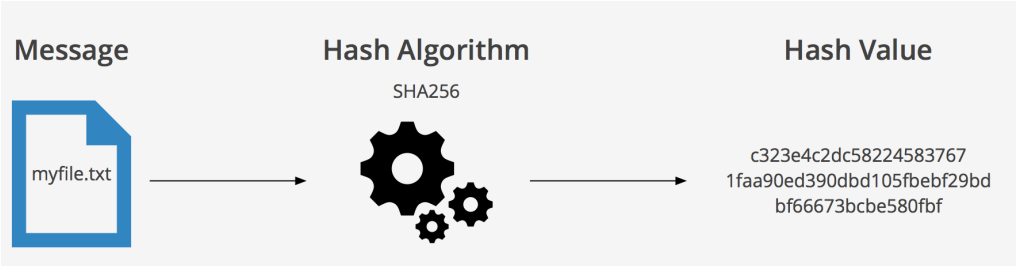
Listing: user.py

```
1  # import the user class
2  from user1 import user
3
4  # create a user object
5  a = user('Thomas', 'Kamalakis', 'thkam@hua.gr', 'hua123')
6
7  check = a.verify_password('hua')
8  print(check)
9
10 check = a.verify_password('hua123')
11 print(check)
```

Listing: userpass.py

SHA256

- Βασίζεται σε cryptographic hash functions.
- Πρόκειται για συναρτήσεις που πολύ δύσκολα αντιστρέφονται.
- Δημιουργήθηκε από την NSA (2001).
- Χρησιμοποιείται σε πολλές εφαρμογές (TLS, SSL, SSH, κτλ)



Ορισμός Μεθόδων

```
35      def as_dict(self):
36          return {
37              'name' : self.name,
38              'surname' : self.surname,
39              'email' : self.email,
40              'display_name' : self.display_name,
41              'password' : self.password
42          }
```

Listing: user.py

Λίγο διαφορετική __init__()

```
7  class user:
8
9      def __init__(self, *args):
10         # if there is one argument:
11
12         if len(args) == 1:
13             arg = args[0]
14             if isinstance(arg, dict):
15                 self.__set_user(arg['name'],
16                                 arg['surname'],
17                                 arg['email'],
18                                 arg['password'])
19
20             elif isinstance(arg, list):
21                 self.__set_user(arg[0],
22                                 arg[1],
23                                 arg[2],
24                                 arg[3])
25         elif len(args) == 4:
26             self.__set_user(args[0],
27                             args[1],
28                             args[2],
29                             args[3])
30         else:
31             raise TypeError('You need to supply a single argument that is either a list or a dict, or
       4 string arguments ')
```

Listing: user.py

Ιστορικά

- Πρόκειται για μία *σχεσιακή* βάση.
- Τα δεδομένα είναι οργανωμένα σε πίνακες
- Οι στήλες αποτελούν τα attributes, οι γραμμές τα αντίστοιχα δεδομένα.
- Ξεκίνησε από το UC Berkeley (1982)
- Είναι αρκετά διαδεδομένη βάση και χρησιμοποιείται πολύ συχνά σε συστήματα παραγωγής.

Τι θα κάνουμε

- Θα στήσουμε μία Postgres στο debian VM
- Θα χρησιμοποιήσουμε την βιβλιοθήκη psycopg2 για να αυτοματοποιήσουμε διάφορες διεργασίες
- Θα δούμε παραδείγματα Create Read Update Delete (CRUD)
- πρόκειται για μία απλή περίπτωση απομακρυσμένου ελέγχου συστήματος.
- Φυσικά πολλά από τα παραδείγματα εύκολα μεταφέρονται και σε άλλες σχεσιακές βάσεις, π.χ. MySQL

Εγκατάσταση

```
Linux debian 5.10.0-19-amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# apt install postgresql postgresql-contrib
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libllvm11 libpq5 libsensors-config libsensors5 libxslt1.1 libz3-4 postgresql-13 postgresql-client-13 postgresql-client-common postgresql-common ssl-cert sysstat
Suggested packages:
  lm-sensors postgresql-doc postgresql-doc-13 libjson-perl isag
The following NEW packages will be installed:
  libllvm11 libpq5 libsensors-config libsensors5 libxslt1.1 libz3-4 postgresql postgresql-13 postgresql-client-13 postgresql-client-common postgresql-common postgresql-contrib
  ssl-cert sysstat
0 upgraded, 14 newly installed, 0 to remove and 0 not upgraded.
Need to get 43.1 MB of archives.
After this operation, 168 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Ο χρήστης postgres

```
root@debian:~# passwd postgres
New password:
Retype new password:
passwd: password updated successfully
```

```
root@debian:~# su - postgres
postgres@debian:~$
```

Δημιουργία Βάσης

```
root@debian:~# passwd postgres
New password:
Retype new password:
passwd: password updated successfully
root@debian:~# su - postgres
postgres@debian:~$ createdb userdb
postgres@debian:~$ psql userdb
psql (13.8 (Debian 13.8-0+deb11u1))
Type "help" for help.

userdb=# CREATE TABLE users (
userdb(#   email VARCHAR(100) PRIMARY KEY,
userdb(#   givenName VARCHAR(100),
userdb(#   surname VARCHAR(150),
userdb(#   password VARCHAR(200) );
CREATE TABLE
userdb=# \dt
         List of relations
 Schema | Name  | Type  |  Owner
--------+-------+-------+----------
 public | users | table | postgres
(1 row)

userdb=#
```

Δημιουργία Βάσης

```
userdb=# \d+ users
                                        Table "public.users"
  Column   |          Type          | Collation | Nullable | Default | Storage  | Stats target | Description
-----------+------------------------+-----------+----------+---------+----------+--------------+-------------
 email     | character varying(100) |           | not null |         | extended |              |
 givenname | character varying(100) |           |          |         | extended |              |
 surname   | character varying(150) |           |          |         | extended |              |
 password  | character varying(200) |           |          |         | extended |              |
Indexes:
    "users_pkey" PRIMARY KEY, btree (email)
Access method: heap
```

Απομακρυσμένη Σύνδεση

```
postgres=# ALTER USER postgres WITH PASSWORD 'hua123';
ALTER ROLE
```

```
  GNU nano 5.4          /etc/postgresql/13/main/postgresql.conf *
# 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 = '/var/lib/postgresql/13/main'          # use data in another d>
                                        # (change requires restart)
hba_file = '/etc/postgresql/13/main/pg_hba.conf'        # host-based authentica>
                                        # (change requires restart)
ident_file = '/etc/postgresql/13/main/pg_ident.conf'    # ident configuration f>
                                        # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is written.
external_pid_file = '/var/run/postgresql/13-main.pid'                   # write>
                                        # (change requires restart)


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

# - Connection Settings -

listen_addresses = '*'                  # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for >
                                        # (change requires restart)
port = 5432                             # (change requires restart)
max_connections = 100                   # (change requires restart)
```

Απομακρυσμένη Σύνδεση

```
  GNU nano 5.4          /etc/postgresql/13/main/pg_hba.conf *
# available for which authentication methods.
#
# Database and user names containing spaces, commas, quotes and other
# special characters must be quoted.  Quoting one of the keywords
# "all", "sameuser", "samerole" or "replication" makes the name lose
# its special character, and just match a database or username with
# that name.
#
# This file is read on server startup and when the server receives a
# SIGHUP signal.  If you edit the file on a running system, you have to
# SIGHUP the server for the changes to take effect, run "pg_ctl reload",
# or execute "SELECT pg_reload_conf()".
#
# Put your actual configuration here
# ----------------------------------
#
# If you want to allow non-local connections, you need to add more
# "host" records.  In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.




# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local   all             postgres                                peer

# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     peer
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
# IPv6 local connections:
host    all             all             ::1/128                 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     peer
host    replication     all             127.0.0.1/32            md5
host    replication     all             ::1/128                 md5
host    all             all             0.0.0.0/0               md5
host    all             all             ::/0                    md5
```

Ακούει κανείς; netstat

- Πολλές φορές θέλουμε να δούμε αν τρέχει ένα service όπως η postgres
- Μπορούμε να χρησιμοποιήσουμε διάφορα εργαλεία για αυτό
- Ένα από αυτά είναι το netstat
- netstat -tupln
  - -t : TCP connections
  - -u : UDP connections
  - -l : listening sockets (ποιός ακούει)
  - -p : να μας δείξει πιο πρόγραμμα (program) ακούει
  - -n : δείξε μας τα IP

Εγκατάσταση

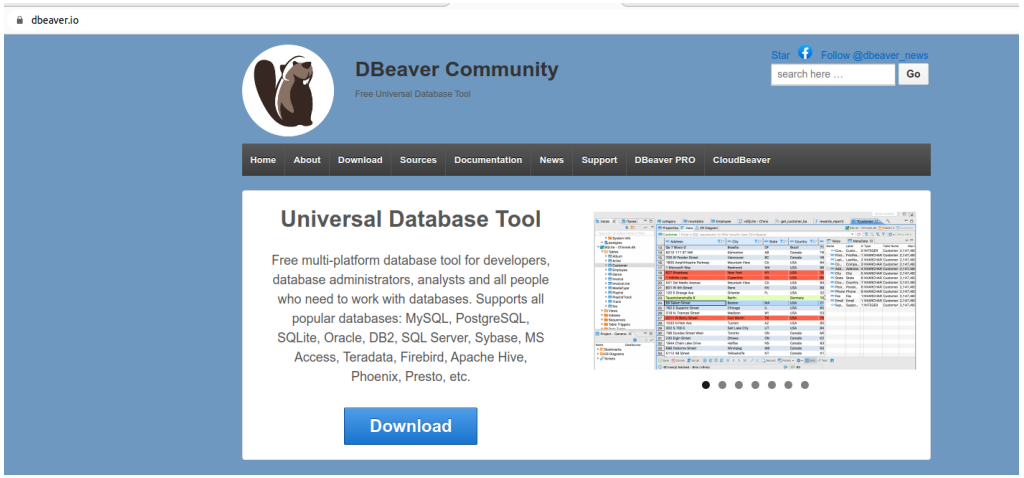
```
root@debian:~# apt-get install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 250 kB of archives.
After this operation, 1015 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian bullseye/main amd64 net-tools amd64 1.60+git2
0181103.0eebece-1 [250 kB]
Fetched 250 kB in 0s (976 kB/s)
apt-listchanges: Can't set locale; make sure $LC_* and $LANG are correct!
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
        LANGUAGE = "en_US:en",
        LC_ALL = (unset),
        LC_TIME = "el_GR.UTF-8",
        LC_MONETARY = "el_GR.UTF-8",
        LC_ADDRESS = "el_GR.UTF-8",
        LC_TELEPHONE = "el_GR.UTF-8",
        LC_NAME = "el_GR.UTF-8",
        LC_MEASUREMENT = "el_GR.UTF-8",
        LC_IDENTIFICATION = "el_GR.UTF-8",
        LC_NUMERIC = "el_GR.UTF-8",
        LC_PAPER = "el_GR.UTF-8",
        LANG = "en_US.UTF-8"
    are supported and installed on your system.
perl: warning: Falling back to a fallback locale ("en_US.UTF-8").
locale: Cannot set LC_ALL to default locale: No such file or directory
Selecting previously unselected package net-tools.
(Reading database ... 35467 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1) ...
Setting up net-tools (1.60+git20181103.0eebece-1) ...
Processing triggers for man-db (2.9.4-2) ...
root@debian:~#
```

netstat -tupln

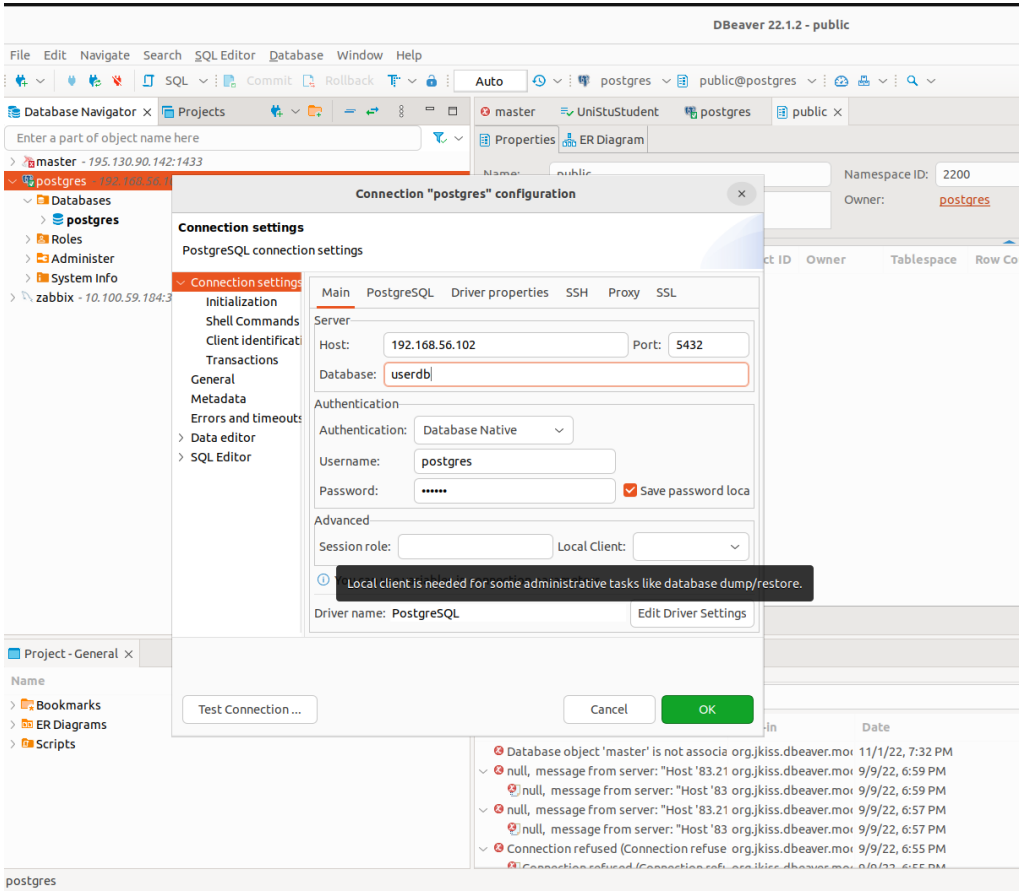
```
root@debian:~# netstat -tupln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
758/sshd: /usr/sbin
tcp        0      0 0.0.0.0:5432            0.0.0.0:*               LISTEN
4419/postgres
tcp6       0      0 :::22                   :::*                    LISTEN
758/sshd: /usr/sbin
tcp6       0      0 :::5432                 :::*                    LISTEN
4419/postgres
udp        0      0 0.0.0.0:68              0.0.0.0:*
689/dhclient
udp        0      0 0.0.0.0:68              0.0.0.0:*
647/dhclient
root@debian:~#
```

Dbeaver

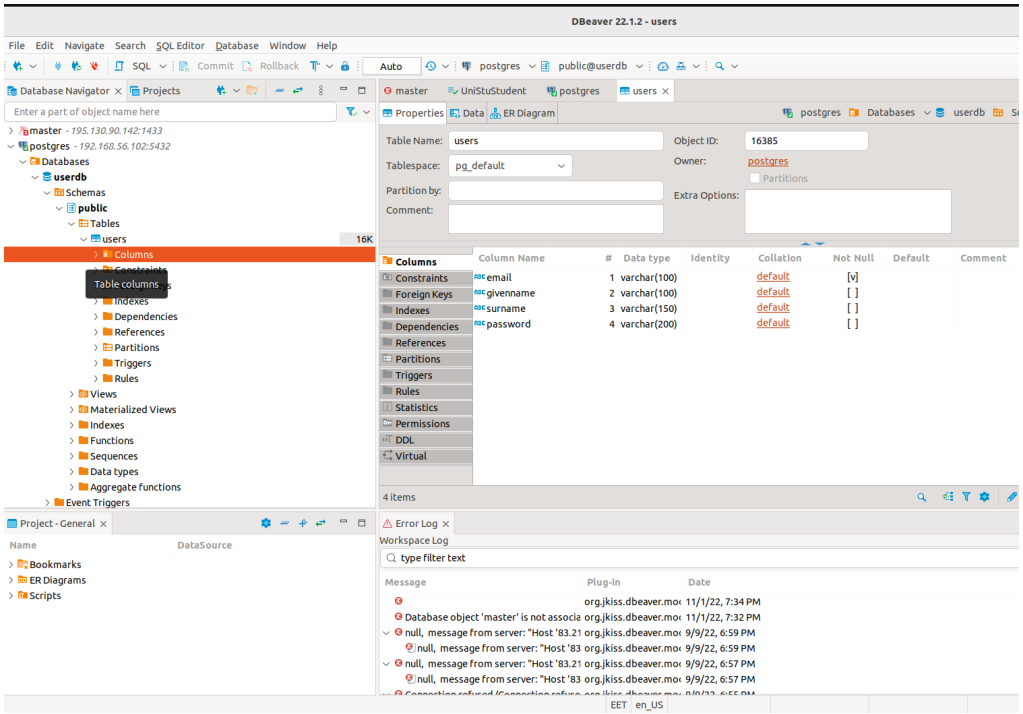
- Ένας πολύ καλός client για να διαχειριζόμαστε τις βάσεις
- ΕΛ/ΛΑΚ



Dbeaver


DBeaver 22.1.2 - public
File Edit Navigate Search SQL Editor Database Window Help
Database Navigator
Projects
Enter a part of object name here
master - 195.130.90.142:1433
postgres
Databases
postgres
Roles
Administer
System Info
zabbix - 10.100.59.184:3
Connection "postgres" configuration
Connection settings
PostgreSQL connection settings
Connection settings
Initialization
Shell Commands
Client identificati
Transactions
General
Metadata
Errors and timeouts
Data editor
SQL Editor
Main PostgreSQL Driver properties SSH Proxy SSL
Server
Host: 192.168.56.102
Port: 5432
Database: userdb
Authentication
Authentication: Database Native
Username: postgres
Password:
Save password loca
Advanced
Session role:
Local Client:
Local client is needed for some administrative tasks like database dump/restore.
Driver name: PostgreSQL
Edit Driver Settings
Test Connection ...
Cancel
OK
Project - General
Name
Bookmarks
ER Diagrams
Scripts
Properties
ER Diagram
Namespace ID: 2200
Owner: postgres
Tablespace
Row Co
Date
Database object 'master' is not associa org.jkiss.dbeaver.mo 11/1/22, 7:32 PM
null, message from server: "Host '83.21 org.jkiss.dbeaver.mo 9/9/22, 6:59 PM
null, message from server: "Host '83 org.jkiss.dbeaver.mo 9/9/22, 6:59 PM
null, message from server: "Host '83.21 org.jkiss.dbeaver.mo 9/9/22, 6:57 PM
null, message from server: "Host '83 org.jkiss.dbeaver.mo 9/9/22, 6:57 PM
Connection refused (Connection refuse org.jkiss.dbeaver.mo 9/9/22, 6:55 PM
postgres

Dbeaver


DBeaver 22.1.2 - users
File Edit Navigate Search SQL Editor Database Window Help
Database Navigator
Projects
Enter a part of object name here
master - 195.130.90.142:1433
postgres - 192.168.56.102:5432
Databases
userdb
Schemas
public
Tables
users
16K
Columns
Table columns
Indexes
Dependencies
References
Partitions
Triggers
Rules
Views
Materialized Views
Indexes
Functions
Sequences
Data types
Aggregate functions
Event Triggers
Project - General
Name
DataSource
Bookmarks
ER Diagrams
Scripts
master
UniStuStudent
postgres
users
Properties
Data
ER Diagram
Table Name: users
Object ID: 16385
Tablespace: pg_default
Owner: postgres
Partition by:
Comment:
Extra Options:
Columns
Constraints
Foreign Keys
Indexes
Dependencies
References
Partitions
Triggers
Rules
Statistics
Permissions
DDL
Virtual
Column Name
Data type
Identity
Collation
Not Null
Default
Comment
email varchar(100) default [v]
givenname varchar(100) default [ ]
surname varchar(150) default [ ]
password varchar(200) default [ ]
4 items
Error Log
Workspace Log
type filter text
Message
Plug-in
Date
Database object 'master' is not associa
null, message from server: "Host '83.21
null, message from server: "Host '83
EET
en_US

Εγκατάσταση του psycopq2

```
root@kirklaptop:/home/thomas# apt-get install libpq-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bridge-utils libgsoap-2.8.117 liblzf1 python3-docker python3-dockerpty
  python3-docopt python3-dotenv python3-texttable python3-websocket ubuntu-fan
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libssl-dev
Suggested packages:
  postgresql-doc-14 libssl-doc
The following NEW packages will be installed:
  libpq-dev libssl-dev
0 upgraded, 2 newly installed, 0 to remove and 39 not upgraded.
Need to get 2517 kB of archives.
After this operation, 12,9 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

```
root@kirklaptop:/home/thomas# pip3 install psycopg2
Collecting psycopg2
  Downloading psycopg2-2.9.5.tar.gz (384 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 384.3/384.3 KB 2.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: psycopg2
  Building wheel for psycopg2 (setup.py) ... -
```

Δημιουργία κλάσης userdb

```
74 class userdb:
75     def __init__(self, server_ip = SERVER_IP,
76                          db_name = DB_NAME,
77                          db_user = DB_USER,
78                          db_password = DB_PASSWORD ):
79 
80         self.server_ip = server_ip
81         self.db_name = db_name
82         self.db_user = db_user
83         self.db_password = db_password
```

Listing: user.py

Σύνδεση

```
86      def connect(self):
87          self.conn = psycopg2.connect(
88                          host = self.server_ip,
89                          database = self.db_name,
90                          user = self.db_user,
91                          password = self.db_password)
92
93      def close(self):
94          self.conn.close()
```

Listing: user.py

Εισαγωγή

```
97      def insert_user(self, user):
98          cursor = self.conn.cursor()
99          name = user.name
100         surname = user.surname
101         email = user.email
102         password = user.password
103         query = """INSERT INTO users (email, givenName, surname, password) VALUES ('%s', '%s', '%s',
            '%s') """ %(email, name, surname, password)
104         cursor.execute(query)
105         self.conn.commit()
```

Listing: user.py

Εισαγωγή

```
1 from user import user, userdb
2
3 a = userdb()
4 a.connect()
5 a.insert_user(
6     user('Thomas', 'Kamalakis', 'thkam@hua.gr', 'hua123')
7 )
8 a.close()
```

Listing: pqsl.py

Ανάγνωση

```
108     def get_user(self, email):
109         cursor = self.conn.cursor()
110         query = """SELECT * FROM users WHERE (email='%s') """ %email
111         cursor.execute(query)
112         if cursor.rowcount == 1:
113             u = cursor.fetchall()[0]
114             return user( u[1], u[2], u[0], u[3] )
```

Listing: user.py

Ανάγνωση

```
1 from user import user, userdb
2
3 a = userdb()
4 a.connect()
5 u = a.get_user('thkam@hua.gr')
6 print(u)
```

Listing: pqsl.py

Ενημέρωση

```
117     def update_name(self, name, email):
118         cursor = self.conn.cursor()
119         query = """UPDATE users SET givenName = '%s' WHERE email = '%s' """ %(name, email)
120         cursor.execute(query)
121         self.conn.commit()
122 
123     def update_surname(self, surname, email):
124         cursor = self.conn.cursor()
125         query = """UPDATE users SET surname = '%s' WHERE email = '%s' """ %(surname, email)
126         cursor.execute(query)
127         self.conn.commit()
```

Listing: user.py

Ενημέρωση

```
130     def update_password_hash(self, hash, email):
131         cursor = self.conn.cursor()
132         query = """UPDATE users SET password = '%s' WHERE email = '%s' """ %(hash, email)
133         cursor.execute(query)
134         self.conn.commit()
135 
136     def update_password(self, password, email):
137         cursor = self.conn.cursor()
138         hash = sha256( password.encode() ).hexdigest()
139         query = """UPDATE users SET password = '%s' WHERE email = '%s' """ %(hash, email)
140         cursor.execute(query)
141         self.conn.commit()
```

Listing: user.py

Εισαγωγή

```
1 from user import user, userdb
2
3 a = userdb()
4 a.connect()
5 a.update_name('Tom', 'thkam@hua.gr')
6 a.update_surname('Hanks', 'thkam@hua.gr')
7 a.update_password('hua', 'thkam@hua.gr')
```

Listing: pqsl.py

Εξαγωγή

```
144     def all_users(self):
145         cursor = self.conn.cursor()
146         query = """SELECT * FROM users """
147         cursor.execute(query)
148         rows = cursor.fetchall()
149         users = { u[0] : user( u[1], u[2], u[0], u[3] ) for u in rows }
150         return users
```

Listing: user.py

Εξαγωγή

```
153     def export_to_csv(self, filename):
154         with open(filename, 'w') as f:
155             writer = csv.writer(f, delimiter = ';')
156             writer.writerow(CSV_HEADERS)
157 
158             users = self.all_users()
159             for email, user in users.items():
160                 writer.writerow( [ user.email,
161                                    user.name,
162                                    user.surname,
163                                    user.password ] )
```

Listing: user.py

Εξαγωγή

```
1  from user import user, userdb
2
3  a = userdb()
4  a.connect()
5
6  users = a.all_users()
7  print(users)
8
9  a.export_to_csv('users.csv')
10 a.close()
```

Listing: pqslexport1.py