



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
ELEKTRONIKOS FAKULTETAS
ELEKTRONINIŲ SISTEMŲ KATEDRA

Atvirojo kodo įranga mokslui, verslui ir valdymui

Laboratorinis darbas Nr.6

Ataskaita

Atliko: AKSfm-16 gr. stud. Andžej Jakonis

Tikrino: Doc. dr. Raimond Laptik

Įvertinimas:.....

(pažymys, dėstytojo parašas)

Vilnius, 2017

Uždaviniai

- Pasirinkti ir įdiegti atvirojo kodo duomenų bazę. Pagrįsti pasirinkimą bei pateikti licenciją.
- Sukonfigūruoti bei sukurti pirmą duomenų bazę.
- Sukurti lentelę, užpildyti ją bei patikrinti turinį

1. Duomenų bazės pasirinkimas

Tarp keturių lyderiaujančių duomenų bazių dvi yra atvirojo kodo – MySQL bei PostgreSQL . Pagal populiarumą MySQL pirmauja, tačiau technologiniu atžvilgiu jos yra gana panašios. MySQL yra spartesnė duomenų bazė, bei pats duomenų bazių kūrimas yra paprastesnis. Tačiau šiame laboratoriniame darbe nutarta naudoti PostgreSQL. Priežastis – anksčiau jau dirbta su MySQL, todėl nutarta susipažinti su PostgreSQL.

PostgreSQL – atviro kodo , objektų reliacinė duomenų bazių valdymo sistema. Veikia daugelyje operacinių sistemų, įskaitant Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) ir Windows. Turi vidinį programavimo sąsają, pritaikytą dirbti su C, C++, Java, Perl, Python, Ruby, Tcl ir kt.

PostgreSQL išleidžiama pagal PostgreSQL licenciją, liberalią atvirojo kodo licenciją, panašią į BSD arba MIT licencijas. Licencija teigia, kad:

„Leidimas naudoti, kopijuoti, modifikuoti ir platinti šią programinę įrangą bei jos dokumentus bet kokiam tikslui, be mokesčio ir be raštiško sutikimo suteikiamas su sąlyga, kad aukščiau paminėtas autorių teisių pranešimas ir ši pastraipa bei šios dvi pastraipos bus visose kopijose.“

Taigi naudojant šią licenciją, pirmoje eilutėje rašome licencijos savininką (vartotojas ar organizacija) bei metus, o pagrindiniame licencijos tekste, įrašome savo organizacijos pavadinimą (vardą ir pavardę, jeigu tai pavienis vartotojas).

2. Duomenų bazės diegimas ir vartotojo kūrimas

2.1. PostgreSQL duomenų bazė Ubuntu operacinėje sistemoje diegiama terminale įrašant:

```
Sudo apt-get install postgresql postgresql-contrib
```

```
andzej@ubuntu:~$ sudo apt-get install postgresql postgresql-contrib
[sudo] password for andzej:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer re
linux-headers-4.10.0-28 linux-headers-4.10.0-28-generic
```

Pav. 1 Postgresql diegimas

- 2.2. Įrašius PostgreSQL, automatiškai sukuriamas vartotojas postgres (tiksliau tariant PostgreSQL vartotojai vadinami rolėmis (angl. *roles*). Taigi, naudojant rolę postgres sukuriamas vartotojas andzej bei duomenų bazė mydb. Rolė sukuriamą parašant komandų eilutėje:

```
Createuser -P -s -e andzej
```

```
postgres@ubuntu:~$ createuser -P -s -e andzej
Enter password for new role:
Enter it again:
CREATE ROLE andzej PASSWORD 'md55a3b448c187fb06816c1569294423d46' SUPERUSER CREA
TEDB CREATEROLE INHERIT LOGIN;
```

Pav. 2 Naujos rolės kūrimas

- 2.3. Naudojant tą patį super vartotoją postgres sukuriamą duomenų bazė mydb:

```
CREATE DATABASE mydb;
```

- 2.4. Toliau suteikiamos visos privilegijos vartotojui andzej:

```
GRANT ALL PRIVILEGES ON DATABASE mydb TO andzej;
```

```
andzej@ubuntu:~$ sudo -i -u postgres
[sudo] password for andzej:
postgres@ubuntu:~$ psql
psql (9.5.10)
Type "help" for help.

postgres=# create database mydb;
CREATE DATABASE
postgres=# \l
postgres=#
postgres=# grant all privileges on database mydb to andzej;
GRANT
postgres=#
```

Pav. 3 Naujos duomenų bazės kūrimas ir privilegijų suteikimas

3. Lentelės kūrimas ir pildymas.

- 3.1. Prisijungiama prie PostgreSQL duomenų bazės naudojant vartotoją andzej:

Sudo -i -u andzej

3.2. Prisijungiama prie duomenų bazės:

Psql mydb

3.3. Sukuriama lentelė test.

```
andzej@ubuntu:~$ sudo -i -u andzej
andzej@ubuntu:~$ psql
psql (9.5.10)
Type "help" for help.

andzej=# \c mydb;
You are now connected to database "mydb" as user "andzej".
mydb=# \d
No relations found.
mydb=# CREATE TABLE TEST(
mydb(# ID INT PRIMARY KEY NOT NULL,
mydb(# NAME TEXT NOT NULL,
mydb(# PHONE TEXT NOT NULL);
CREATE TABLE
mydb=# \d
          List of relations
Schema | Name | Type  | Owner
-----+-----+-----+-----
public | test | table | andzej
(1 row)
```

Pav. 4 Naujos lentelės kūrimas

3.4. Komanda \d pateikia sąrašą visų lentelių. Šiuo atveju tai lentelė Test.

3.5. Terminale įrašome komandą SELECT *FROM TABLE; patikriname, ar lentelė tuščia.

3.6. Užpildome sukurtą lentelę įrašius INSERT INTO TEST (ID, NAME, PHONE) VALUES (1, 'Andzej', '867122479');. Dar kartą įrašome SELECT *FROM TABLE; bei patikriname, ar sėkmingai užpildoma lentelė.

```
(1 row)

mydb=# SELECT * FROM TEST;
 id | name | phone
----+-----+-----
(0 rows)

mydb=# INSERT INTO TEST (ID, NAME, PHONE) VALUES (1, 'Andzej', '867122479');
INSERT 0 1
mydb=# SELECT * FROM TEST;
 id | name  | phone
----+-----+-----
  1 | Andzej | 867122479
(1 row)

mydb=#
```

Pav. 5 Lentelės pildymas