

Лабораториска вежба 4 – Креирање на бази на податоци со MySQL

Име и презиме	Број на индекс	Група

Вовед во MySQL

Еден од најраспространетите и најкористените датотечни сервери е MySQL кој пред се е од отворен тип кој овозможува управување и со релациони бази на податоци кој овозможува повеќе-кориснички пристап. Целиот изворен код на развојниот проект на MySQL е на располагање под GNU лиценца кој е поседуван и спонзориран од страна на Oracle Corporation. Во основа е систем за управување со релациони бази на податоци кој во општата дистрибуција не доаѓа со графичка околина т.е. е базиран на терминал. Во последно време најчесто се среќава како дел од пакет за развој на веб апликации т.е. со дистрибуции како XAMPP или WAMP Кои со себе имаат и PHP и Apache сервер. Заедно со овие дистрибуции доаѓа и phpMyAdmin кој овозможува web базиран графички интерфејс кон MySQL серверот.

Работа со MySQL

Најавување на серверот:

```
feit:~$> mysql -u user -p
```

опцијата `-u` овозможува внесување на корисничкото име, додека пак опцијата `-p` значи дека по извршувањето на командата, ќе може да се вметне лозинката.

Преглед на бази на податоци:

```
mysql> show databases;
```

Со оваа команда ќе ги видите сите бази кои моментално се креирани на серверот.

НАПОМЕНА: секоја команда завршува со точка-запирка „;“.

```
+-----+
| Database      |
+-----+
| information_schema |
| feitdb        |
| mysql         |
+-----+
```

Креирање на нова база на податоци:

```
mysql> create database ime_na_baza;
```

Пример:

```
mysql> create database feitdb;
```

Query OK, 1 row affected (0.00 sec)

Користење на одредена база

Пред да се почне да се користи некоја база истата мора да се означи како тековна база за работа. Користејќи ја наредбата `use database_ime`: може да се означи некоја од базите како тековна за работа.

```
mysql> use feitdb;
Database changed
```

Листање на сите табели еднаш чим ќе се селектира дадена база може да се прави со помош на наредбата `show tables`; Доколку самошто е креирана базата би требало да се добие следниот излез:

```
mysql> show tables;
Empty set (0.01 sec)
```

Креирање на табела

Креирањето на дадена табела во рамки на некоја база се прави со помош на наредбата `CREATE TABLE` чија синтакса е дефинирана како:

```
create table ime_tabela (atrib1 data_tip1, atrib2 data_tip2, atrib3 data_tip3);
```

каде што најпрво се наведува името на табелата, а потоа во рамки на мали загради се наведуваат атрибутите кои се членови на табелата заедно со податочниот тип.

Пример:

```
mysql> create table ime_tabela (atrib1 varchar(20), atrib2 varchar(10), atrib3 int(1));
Query OK, 0 rows affected (0.07 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_feitdb |
+-----+
| ime_tabela      |
+-----+
1 row in set (0.01 sec)
```

Листа од типови на податоци во MySQL:

Текстуални типови:

CHAR()	A fixed section from 0 to 255 characters long.
VARCHAR()	A variable section from 0 to 255 characters long.
TINYTEXT	A string with a maximum length of 255 characters.
TEXT	A string with a maximum length of 65535 characters.
BLOB	A string with a maximum length of 65535 characters.
MEDIUMTEXT	A string with a maximum length of 16777215 characters.
MEDIUMBLOB	A string with a maximum length of 16777215 characters.
LONGTEXT	A string with a maximum length of 4294967295 characters.
LOB	A string with a maximum length of 4294967295 characters.

Броеви:

TINYINT()	-128 to 127 normal 0 to 255 UNSIGNED.
SMALLINT()	-32768 to 32767 normal 0 to 65535 UNSIGNED.
MEDIUMINT()	-8388608 to 8388607 normal 0 to 16777215 UNSIGNED.
INT()	-2147483648 to 2147483647 normal 0 to 4294967295 UNSIGNED.
BIGINT()	-9223372036854775808 to 9223372036854775807 normal 0 to 18446744073709551615 UNSIGNED.
FLOAT	A small number with a floating decimal point.
DOUBLE(,)	A large number with a floating decimal point.
DECIMAL(,)	A DOUBLE stored as a string , allowing for a fixed decimal point.

Датуми:

DATE	YYYY-MM-DD.
DATETIME	YYYY-MM-DD HH:MM:SS.
TIMESTAMP	YYYYMMDDHHMMSS.
TIME	HH:MM:SS.

Одржување на интегритет

Одржување на интегритет на ентитет т.е. дефинирање на примарни клучеви се прави со помош на PRIMARY KEY(atrib_name) директивата. Откако ќе се дефинираат сите атрибути на една табела, се дефинира примарниот клуч. Дополнително се дефинира интегритет на атрибут преку тоа дали може или неможе да има нулева вредност.

Одржување на референтен интегритет т.е. дефинирање на надворешни клучеви се прави со помош на FOREIGN KEY со синтакса:

FOREIGN KEY (ime_atrib) REFERENCES ime_tabela (ime_kluc_atrib)

Пример:

```
CREATE TABLE sorabotnik
(
  id INT NOT NULL,
  ime VARCHAR(30),
  PRIMARY KEY (id)
);
```

```
CREATE TABLE fakturi
(
  faktura_id INT NOT NULL,
  suma FLOAT,
  kupuvac_id INT NOT NULL,
  PRIMARY KEY(faktura_id),
  FOREIGN KEY (kupuvac_id) REFERENCES kupuvac (id)
);
```