

# Softver otvorenog koda

Žarko Zečević  
Elektrotehnički fakultet  
Univerzitet Crne Gore

# Predavanje 3

## LAMP STACK

### Ishodi učenja:

Nakon savladavanja gradiva sa ovog predavanja studenti će moći da:

- Instaliraju i konfigurišu LAMP stack
- Kreiraju i hostuju jednostavnu aplikaciju korišćenjem LAMP tehnologije
- Razumiju razliku između različitih vrsta načina kreiranja rezervnih kopija baza podataka i naprave naprave automatizovanu skriptu za kreiranje backup-a

# Lamp Stack

Skraćenica LAMP potiče od

- L = Linux
- A = Apache
- M = MySQL
- P = Perl/PHP/Python

LAMP stack je softver otvorenog koda koji omogućava brzi razvoj web aplikacija.

LAMP je moguće instalirati i na drugim platformama:

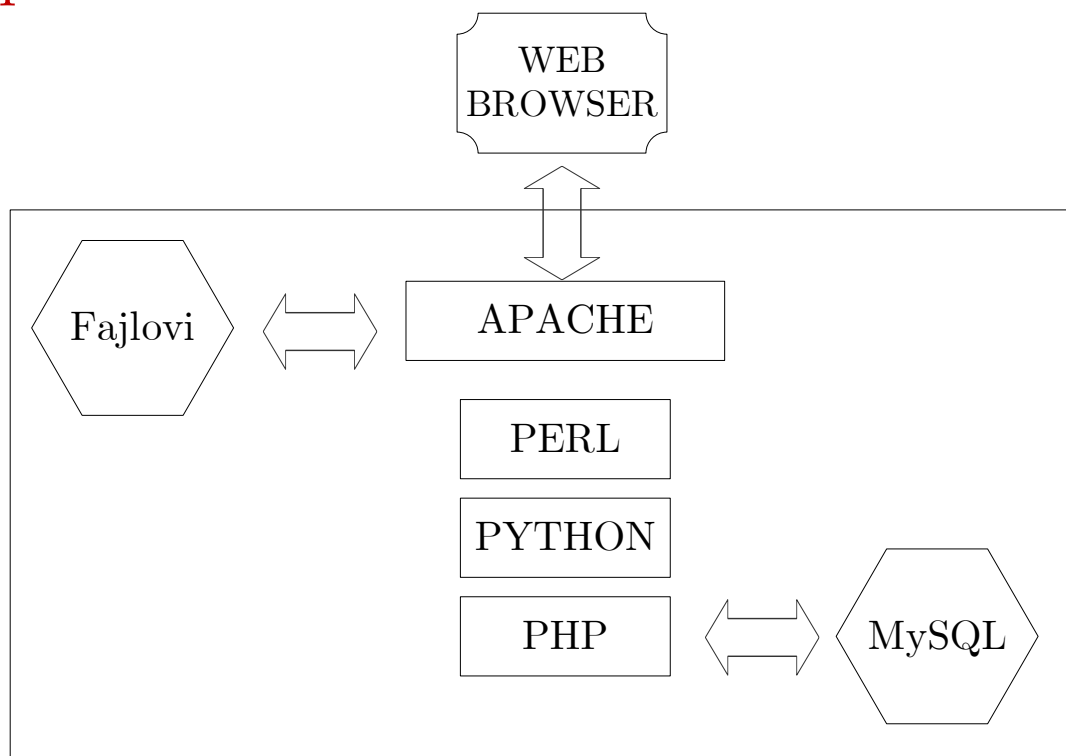
- Na Windowsu pod nazivom WAMP
- Na Mac-u pod nazivom MAMP

# Lamp Stack

Instalacija na Linuxu:

```
sudo apt-get install mysql-server
```

```
sudo apt-get install php libapache2-mod-php php-mcrypt  
php-mysql
```



PHP je široko-korišteni skriptni jezik generalne namjene koji je posebno pogodan za razvoj web aplikacija i koji može biti ugrađen u HTML kod.

- PHP je rekurzivni akronim (!) za *PHP: Hypertext Preprocessor*
- PHP je dostupan na <http://www.php.net>
- Zadnja verzija PHP-a je 8.2.6 iz 2023. godine

PHP se može koristiti na većini operativnih sistema:

- Linuxu i Unix-ovim varijantama (HP-UX, Solaris)
- Microsoft Windows-u
- Mac OS X (takođe Unix varijanta)
- RISC OS

PHP je podržan od strane mnogih web servera:

- Apache
- Microsoft IIS
- Nginx, itd.

Takođe, PHP može da komunicira sa većinom database management sistema:

- Adabas D, dBase, Empress, FilePro (read-only), Hyperwave, IBM DB2, Informix, Ingres, InterBase, FrontBase, mSQL, Direct MS-SQL, MySQL, ODBC, Oracle (OCI7 and OCI8), Ovrimos, PostgreSQL, SQLite, Solid, Sybase, Velocis, Unix dbm

# Za šta se koristi PHP?

PHP je uglavnom fokusiran na skriptiranje na strani web servera, pa se sa njim može odraditi bilo šta kao i sa ostalim CGI programima:

- Prikupljanje podataka iz HTML formi, generisanje dinamičkog sadržaja web stranice, slanje i primanje kolačića

Pomoću PHP-a se mogu odraditi i druge stvari:

- Pisanje skripti koje se izvršavaju sa komandne linije (za izvršavanje nije potreban web server ili browser)
- Idealan je pisanje skripti koje se izvršavaju pomoću crone-a (na Linux-u i Unix-u) ili Task Scheduler-a na Windows-u

# Za šta se koristi PHP?

- PHP se može koristiti i za pisanje desktop aplikacija, mada ovo nije njegova osnovna namjena
- Skriptiranje na strani web-servera je tradicionalna i osnovna namjena PHP-a

Da bi mogli da koristiti PHP potrebni su:

- PHP parser (CGI ili serverski modul), web server i web browser
- Web server treba da bude povezan sa instalacijom PHP-a
- Rezultat PHP programa se pregleda preko browsera (najčešće HTML, a može i pdf, slike, video ili neke druge vrste tekstualnih fajlova)



# Osnovna sintaksa PHP-a

```
<!DOCTYPE HTML PUBLIC "-
//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?php
      echo "Hi, I'm a PHP script!";
    ?>
  </body> </html>
```

PHP server intepretira kod unutar taga `<?php ?>` i vraća HTML kao rezultat.

# Osnovna sintaksa PHP-a

- PHP instrukcije se moraju završavati tačkom sa zarezom (sem zadnje, ispred taga `?>` kojim se zatvara PHP blok).
- PHP podržava više različitih stilova za komentare:
  - `//` Komentar u jednoj liniji (kao u C++)
  - `/*` Stavljanje komentara u više linija  
druga linija komentara `*/`
  - `#` Komentar u jednoj liniji (kao u Linux Shell-u)

Treba voditi računa o tome da server interpretira `?>` kao zatvaranje PHP bloka, čak iako se nalazi u jednolinijskom komentaru (ovo ne važi za višelinijske blokove).

# Konfiguracija PHP-a

Konfiguracija PHP-a se vrši pomoću **php.ini** fajla koji se nalazi na lokaciji `/etc/php/7.0/apache2/php.ini`. Kod serverskih verzija PHP-a ovaj fajl se čita jedanput, prilikom podizanja web servera.

Neka pravila koja se odnose na sintaksu ovog fajla:

- Linije koje počinju sa `;` se ignorišu
- Većina linija ima format *direktiva = vrijednost*
- Imena direktiva su case-sensitive (razlikuju se mala i velika slova)
- Vrijednosti direktiva mogu biti PHP konstante (pr. `E_ALL`, `M_PI`), INI konstante (`On`, `Off`, `True`, `False`, `Yes`, `No` ili `None`), regularni izrazi (`E_ALL & ~E_NOTICE`) ili stringovi.

# Neka važnija podešavanja

`session.gc_maxlifetime = vrijeme_u_sec`

- Ovo podešavanje odnosno na dužinu trajanja sesije korisnika. Podrazumjevana vrijednost je 24 minuta

`short_open_tag = Off`

- Short open tag izgleda ovako: `<? ?>`. Ova opcija mora biti ugašena ako želite da koristite XML funkcije

`warn_plus_overloading = Off`

- Ova postavka javlja upozorenje ukoliko se `+` koristi sa stringom

`upload_max_filesize = 10M`

- Maksimalna dozvoljena veličina fajla koji se uploaduje

`max_file_uploads = integer`

- Maksimalan broj fajlova koji se može simultano uploadovati

# Neka važnija podešavanja

`memory_limit = 8M`

Maksimalna dozvoljena količina memorije koju skripte može da „konzumira“ (8MB)

`auto-prepend-file = [path/to/file]`

- Ukoliko se ovdje specificira neka putanja, PHP će je automatski dodati na početak svakog PHP fajla

`auto-append-file = [path/to/file]`

- Ukoliko se ovdje specificira neka putanja, PHP će je automatski dodati na kraj svakog PHP fajla

`disable_functions = [function1,function2...]`

- Ova direktiva omogućava blokiranje određenih PHP funkcija

`disable_classes = [class1,class2...]`

- Ova direktiva omogućava blokiranje određenih PHP klasa

# Neka važnija podešavanja

`precision = 12`

- Broj cifara kojim se prikazuju brojevi sa pomičnim zarezom

`file_uploads = [on/off]`

- Pomoću ove opcije se može omogućiti/onemogućiti upload fajlova preko PHP skripti

`mysql.default_host = hostname`

- Podrazumjevana IP adresa mysql servera koji se koristi ukoliko drugi server nije specificiran u PHP kodu

`mysql.default_user = username`

- Podrazumjevano korisničko ime koje se koristi za konekciju na mysql server ukoliko drugo nije specificirano

`mysql.default_password = password`

- Podrazumjevana korisnička šifra koja se koristi za konekciju na mysql server ukoliko druga nije specificirana

# MySQL DBMS

- MySQL je popularni, skalabilni, open-source sistem za upravljanje bazama podataka (eng. database management system, DBMS)
- Može se pokrenuti na Windows-u, Linux-u i mnogim drugim operativnim sistemima.
- MySQL je bio u vlasništvu švedske kompanije MySQL AB, koju je kasnije kupio Sun Microsystems. Danas je Oracle vlasnik MySQL-a.
- Poznati projekti koji koriste (ili su koristili) MySQL:
  - Joomla
  - Wordpress
  - phpBB
  - myBB
  - Drupal
  - Wikipedia
  - Google
  - Facebook
  - Twitter
  - Youtube

# Karakteristike MySQL-a

- Pisan u C i C++
- Postoje razvijeni API-ji za C, C++, Java, Perl, PHP, Python, Ruby, itd.
- Podrška za transakcione (InnoDB) i netransakcione (MyISAM) mašine za skladištenje. Kod transakcionih baza podataka promjene u bazi koje nastaju izvršavanjem seta logično povezanih sql komandi se prihvataju jedino u slučaju ako se svaki upit uspješno izvrši. Ukoliko se uspješno ne izvrši svaki upit, onda se promjene u bazi povlače i vraća se prethodno stanje. Na ovaj način se obezbjeđuje integritet podataka.
- Default-na mašina za skladištenje je InnoDB



# Instalacija na Linux-u

MySQL server se instalira pomoću komande (ukoliko već nije instaliran kao dio LAMP stacka):

```
$ sudo apt-get install mysql-server
```

```
$ mysql_secure_installation
```

MySQL programi koji se koriste za komandne linije:

**mysql** – sql klijent u kome se mogu kucati SQL upiti

**mysqladmin** – klijent za administratore

**mysqlcheck** – provjere integriteta baze podataka

**mysqldump** i **mysqlhotcopy** se koriste za backupovanje baza podataka

**mysqlimport** – importovanje baza

**mysqlshow** – prikazivanje informacija o bazi

# MySQL klijent

Na MySQL server se loguje pomoću sql klijenta:

```
$ mysql -uUSER -pPASS
```

Obično se klijent koristi na istoj mašini na kojoj se nalazi i server. Ne preporučuje se otvaranje porta 3036 spolja (port na kojem osluškuje mysql server).

Unutar mysql klijenta se kucaju sql upiti. Ispod je dat primjer komande za kreiranje korisnika i baze podataka.

```
mysql> CREATE DATABASE proba;
```

```
mysql> CREATE USER 'stud'@'localhost' IDENTIFIED BY 'stud';
```

```
mysql> GRANT ALL PRIVILEGES ON proba.* TO 'stud'@'localhost';
```

```
mysql> FLUSH PRIVILEGES;
```

Iz mysql klijenta se izlazi pomoću komande **exit** ili **quit**.

# PHP API

```
<?php
$mysqli = new mysqli("localhost","user","password","my_db");
// provjera da li je uspostavljena konekcija
if ($mysqli -> connect_errno) {
    echo "Failed to connect to MySQL: " . $mysqli ->
connect_error;
    exit();
// Izvršavanje upita
if ($result = $mysqli -> query("SELECT * FROM Persons")) {
    echo "Returned rows are: " . $result -> num_rows;
    // Free result set
    $result -> free_result();}
$mysqli -> close();}
?>
```

# Backup baza podataka

Da bi se osiguralo da se podaci iz baza podataka mogu povratiti u bilo kom trenutku, potrebno je redovno raditi backup.

Kreiranje rezervnih kopija je potrebno iz više razloga:

- Pad operativnog sistema
- Hardverski problemi (problemi sa hard diskom, RAM memorijom, itd)
- Korisnici neke mogu greškom obrisati tabelu (DROP upit) ili podatke (DELETE FROM upit)
- Direktno editovanje fajlova u kojima su sačuvani podaci obično dovodi do gubitka integriteta tabela i grešaka

# Metode za backup baza podataka

- **Korišćenje backup programa**
  - mysqldump (offline backup, baze se moraju prethodno zaključati)
  - mysqlhotcopy (online backup, samo za MySIAM)
- **Fizičko kopiranje** fajlova sa podacima (\*.frm, \*.MYD, and \*.MYI fajlovi). Fajlovi se čuvaju u direktorijumu **/var/lib/mysql/naziv\_baze\_podataka**

Ova vrsta backup-a se koristi sa MySIAM mašine (mysqlhotcopy zapravo samo automatizuje ovaj proces).

- **Kreiranje Delimited-Text** fajlova
- **Inkrementalni backup** koristeći binarni log
- **Backup na replikacione slejv servere**

# Backup pomoću programa mysqldump

- **Mysqldump** je backup program koji se koristi za eksportovanje baza podataka i njihov transfer na drugi server (ne mora biti MySQL).
- Mysqldump na standardnom izlazu kreira skup CREATE sql upita čijim se izvršavanjem kreiraju objekti iz baze podataka (baze podataka, tabele, itd), i skup INSERT sql upita kojima se upisuju podaci u tabele.
- Izlaz iz ovog programa može biti sačuvan u fajl i kasnije se može učitati koristeći **mysqlimport**. Dostupne su opcije kojima se može kontrolisati koji objekti će biti eksportovani. Takođe se može izabrati format izlaznog fajla: CSV, delimited text, XML, itd.

# Backup pomocí programu mysqldump

## Jedna ili više tabela

```
$ mysqldump -u [uname] -p[pass] db_name table1 >  
table_backup.sql
```

```
$ mysqldump -u [uname] -p[pass] db_name table1 table2 >  
table_backup.sql
```

## Jedna ili više baza podataka

```
$ mysqldump -u [uname] -p[pass] db_name1 db_name2 >  
db_backup.sql
```

## Sve baze podataka

```
$ mysqldump -u [uname] -p[pass] --all-databases > all_db_backup.sql
```

## Čitavi MySQL

```
$ mysqldump -u [uname] -p[pass] db_name > db_backup.sql
```

## Kopiranje na drugi server

```
$ mysqldump --opt db_name | mysql -host = remote_host -C  
db_name
```

# Backup pomoću programa mysqldump

## Auto-kompresija koristeći gzip

```
$ mysqldump -u [uname] -p[pass] db_name | gzip >  
db_backup.sql.gz
```

## Backup na udaljeni server

```
$ mysqldump -P 3306 -h [ip_address] -u [uname] -p[pass] db_name >  
db_backup.sql
```

Eksportovana baza se može učitati na sljedeći način (pretpostavlja se da baza ne postoji, već se ona automatski kreirati pomoću komande ispod):

```
$ mysql -u [uname] -p[pass] [db_to_restore] < [backupfile.sql]
```

Ako podatke želimo da učitamo u postojeću bazu, tada se to radi pomoću sljedeće komande:

```
$ mysqlimport -u [uname] -p[pass] [dbname] < [backup.sql]
```



# Kreiranje razgraničenih fajlova

Pomoću ovog metoda se čuvaju samo podaci iz tabele, ali ne i njena struktura.

Pomoću SQL upita se selektuju željene vrste i upisuju u fajl na serveru. Primjer sintakse ove komande je:

```
SELECT * INTO OUTFILE 'naziv fajla' FROM naziv_tabele;
```

Primjer kreiranja fajla u kome su kolone razdvojene zarezom (comma-separated values, (CSV)) je dat ispod:

```
SELECT a,b,a+b INTO OUTFILE '/tmp/result.txt'  
FIELDS TERMINATED BY ',' OPTIONALLY  
ENCLOSED BY '"' LINES TERMINATED BY '\n'  
FROM table;
```

# Kreiranje inkrementalnih backup-ova

Po default-u, server kreira binarne log fajlove u data direktorijumu. Binarni log fajlovi sadrže potrebne informacije, odnosno sql upite čijim se izvršavanjem stanje baze vraća na prethodno stanje.

Server kreira inkrementalne bin log fajlove, odnosno čuvaju se samo informacije o promjenama u odnosu na prethodni backup.

Spisak binarnih fajlova se može izlistati pomoću komande:

```
mysql> SHOW BINARY LOGS;
```

Da bi se koristila ova vrsta backup-a, server mora biti startovan sa opcijom --log-bin.

# Kreiranje inkrementalnih backup-ova

Da bi se omogućilo kreiranje binarnih fajlova potrebno je prvo otvoriti fajl `/etc/mysql/my.cnf` i u njemu dodati liniju `log-bin = mybinlog`. Nakon toga je potrebno specificirati za koje baze podataka treba uključiti binarno logovanje, ili za koje baze ne treba raditi binarno logovanje:

- 1) `binlog_do_db = mydb`
- 2) `binlog_ignore_db = mydb` Restauracija backup fajlova se vrši pomoću programa `mysqlbinlog`.

Binarni fajlovi između ostalog sadrže informacije i o tome kada su izvršeni upiti, pa je moguće specificirati i iz kog vremenskog perioda želimo da vratimo promjene.

# Linux crontab

**Cron** predstavlja daemon proces čija je osnovna uloga izvršavanje skripti/komandi u tačno određeno vrijeme.

Komande se čuvaju u tabelama, odnosno **crontab** fajlovima, čija je sintaksa sljedeća:

**MIN HOUR DOM MON DOW CMD**

MIN – minut (0-59), pr: 0, 1-12, \* (\* znači, svakog min.)

HOUR – sat (0-24), pr: 0, \*, 0-24/2 (svaka dva sata)

DOM – dan u mjesecu (0-31)

MON – mjesec (0-12)

DOW – dan u nedjelji (1-7)

CMD – komanda ili skripta

# Linux crontab

Svaki korisnik može da edituje svoj crontab fajl pomoću komande:

`crontab -e`

dok se postojeći crontab fajl briše na sljedeći način:

`crontab -r`

Administrator može da promijeni crontab korisnika na sljedeći način:

`crontab -u ime_korisnika -e`

Spisak svih crontab fajlova se nalazi u direktorijumu

`/var/spool/cron/crontabs`

# Linux crontab

Primjer crontab-a kojim se radi backup baze podataka dat je ispod:

```
# crontab -e
SHELL=/bin/bash
MAILTO=root@example.com
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local
/sbin
# backup baze podataka BAZA poda nazivom backup_datum.sql.gz
# backup se radi svakog dana u 00.14h
14 0 * * * mysqldump -uUSER -pPASS BAZA | gzip >
~/BACKUP/backup_`date '+%d-%m-%Y'`.sql.gz
# brisanje backup fajlova starijih od 7 dana
find ~/BACKUP -type f -mtime +7 -name '*.gz' -exec rm {} /
```

# Backup strategije

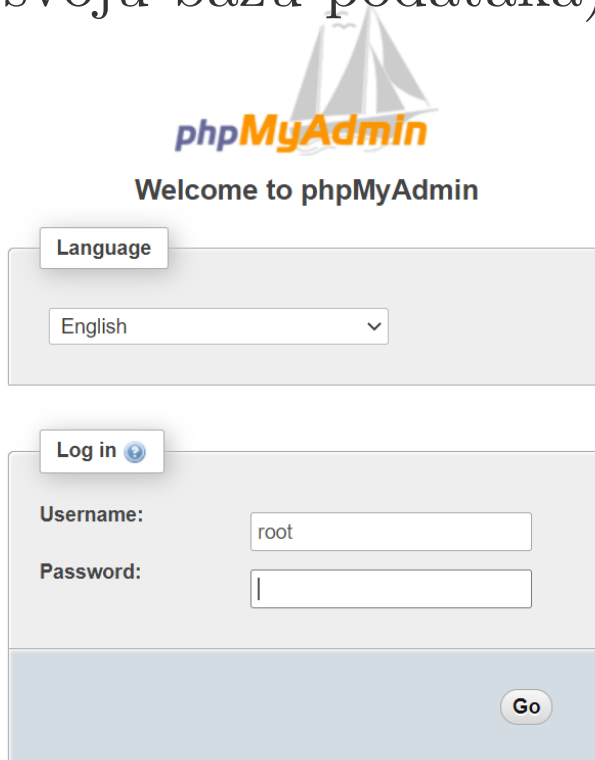
- Backup treba raditi često i redovno
- Backup treba raditi prije i nakon strukturnih promjena u bazi
- Treba uključiti kreiranje binarnih log fajlova
- Rezervne kopije ne treba čuvati na istom fajl sistemu (disku) na kome se nalazi i baza podataka
- Potrebno je raditi periodične backupove čitave baze podataka korišćenjem mysqldump komande
- Potrebno je raditi periodične inkrementalne backupove
- Automatizovati proces backupovanja baza
- Potrebno je odabrati odgovarajući storage za čuvanje backup fajlova

# PHPMyAdmin

PHPMyAdmin – web interfejs za rad sa bazama podataka

URL: <http://89.188.32.136/phpmyadmin>

Koristite iste korisničke podatke kao za logovanje na Linuxu (svak ima svoju bazu podataka).



phpMyAdmin

Welcome to phpMyAdmin

Language

English

Log in

Username: root

Password:

Go



# PHPMyAdmin

The screenshot displays the PHPMyAdmin web interface. At the top left, the 'phpMyAdmin' logo is visible, along with navigation icons for home, help, and settings. Below the logo are 'Recent' and 'Favorites' tabs. A sidebar on the left lists a tree view of databases, including 'New', 'anjapavicevic98', 'balsajanjusevic98', 'bobanbanjevic99', 'denisljuca98', 'denissuljevic97', 'egzonmilla99', 'filipandjelic98', 'ilijaradojicic', 'ilijavulanovic96', 'information\_schema', 'ivanacoric98', 'jovanaikovic99', 'ljubomirjankovic', 'lukam', and 'markokandic98'. The main content area is titled 'Server: localhost' and contains a navigation bar with buttons for 'Databases', 'SQL', 'Status', 'User accounts', 'Export', and 'Import'. Two settings panels are shown: 'General settings' and 'Appearance settings'. The 'General settings' panel includes a 'Change password' link and a 'Server connection collation' dropdown menu set to 'utf8mb4\_unicode\_ci'. The 'Appearance settings' panel includes a 'Language' dropdown menu set to 'English', a 'Theme' dropdown menu set to 'pmahomme', and a 'Font size' dropdown menu set to '82%'. A 'More settings' link is also present at the bottom of the appearance settings panel.

phpMyAdmin

Server: localhost

Databases SQL Status User accounts Export Import

### General settings

Change password

Server connection collation: utf8mb4\_unicode\_ci

### Appearance settings

Language: English

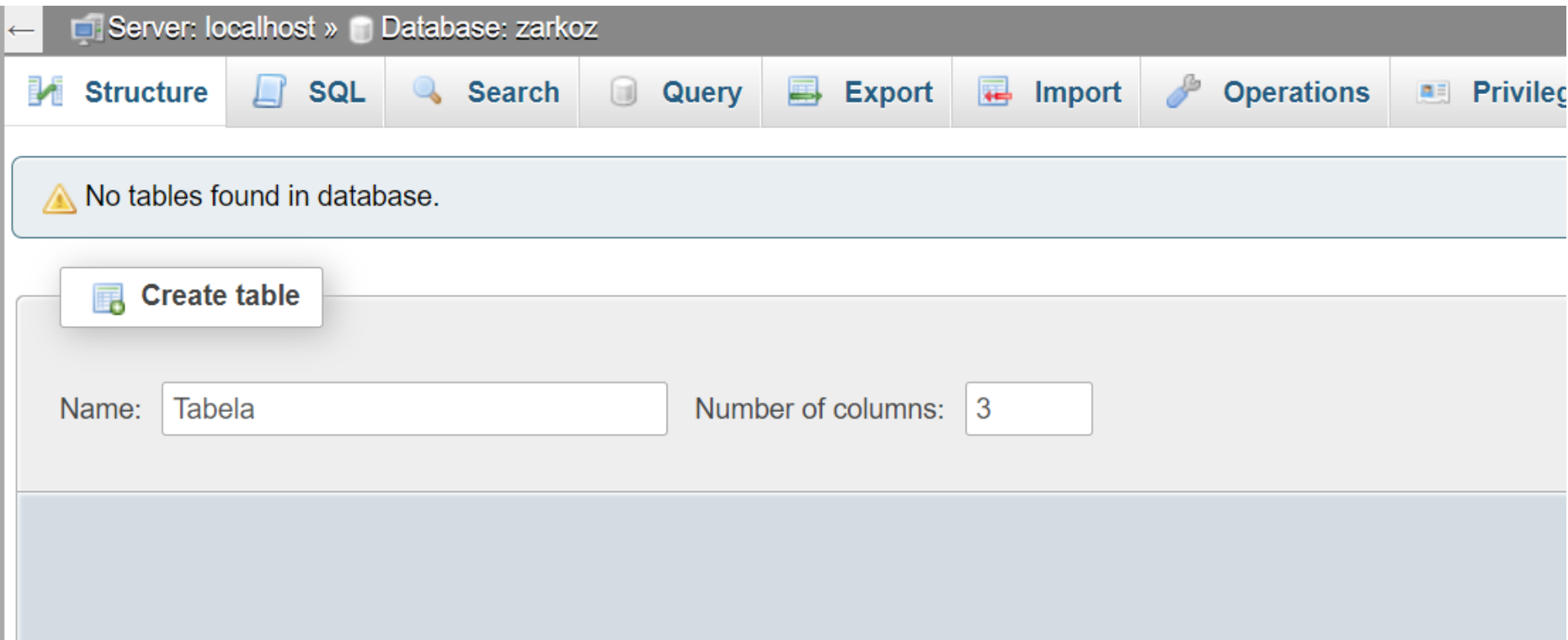
Theme: pmahomme

Font size: 82%

More settings

# PHPMyAdmin

U PHPMyAdmin-u moguće je kreirati, modifikovati i brisati tabele, importovati podatke, itd. PHPMyAdmin takođe sadrži interfejs za unošenje SQL upita.



The screenshot displays the PHPMyAdmin interface for a MySQL server. The top navigation bar shows the server as 'localhost' and the database as 'zarkoz'. The main menu includes options like Structure, SQL, Search, Query, Export, Import, Operations, and Privileges. A message indicates that no tables were found in the database. Below this, the 'Create table' form is visible, with the table name set to 'Tabela' and the number of columns set to 3.

Server: localhost » Database: zarkoz

Structure SQL Search Query Export Import Operations Privileges

⚠ No tables found in database.

Create table

Name:  Number of columns: