**Rukovanje MySQL-om**

**00:18:04**

Jedinica: 18 od 19

**Rezime**

* Projekat SQL Buddyja je akcentovan tako da se bazira na lakoj instalaciji, u okviru Wamp Servera, i jednostavnom korisničkom interfejsu;
* Odabirom „Insert” opcije imate mogućnost da unosite podatke u tabele koje ste kreirali;
* Ono što je potrebno prilikom promene passworda da uradite jeste da podesite password za root korisničko ime i uklonite anonimus korisnika, i kreirate novog korisnika, ukoliko je potrebno;
* Importom Baze u MySQL moguće je da uvedete vašu, već postojeću bazu;
* Baza je preporučljivo da ima .sql ekstenziju;
* Export baze podataka predstavlja izvoz baze iz SQL Buddy okvira;
* Biblioteka funkcija koju koristi php za uspostavljanje veze sa MySQL-om zove se **MySQLi**(slovo i je dodato od engleske reči improved, što znači poboljšan);
* MySQLi\_connect\_errno() vraća broj grešaka u slučaju neuspešnog izvršavanja; u slučaju da je uspešno izvršena, vraća nulu;
* Imena funkcija sa proceduralnim verzijama počinju sa ’mysqli\_’ i tim funkcijama se mora proslediti resurs koji ste prethodno dobili od funkcije mysqli\_connect();
* Potrebno je da bazu podataka sa kojom želite da se povežete navedete kao parametar funkcije mysqli\_connect();
* Izvršavanje upita ka bazi podataka obavlja se preko mysqli\_query() funkcije;
* Skup rezultata možete osloboditi pozivom metode: mysqli\_free\_result( $rezultati );
* Funkcija mysqli\_close(veza sa bazom) se koristi za raskidanje veza sa bazom.

Ova lekcija biće posvećena komunikaciji sa bazom podataka, kao i manipulaciji upitima i tabelama.

SQL Buddy je *open sorce*aplikacija bazirana na webu, pisana u PHP-u sa namerom da vrši administrativne poslove nad MySQL i SQLite, korišćenjem Web browsera. Projekat SQL Buddyja je akcentovan tako da se bazira na lakoj instalaciji, u okviru Wamp Servera i jednostavnom korisničkom interfejsu.

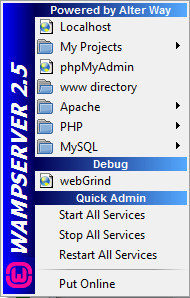
U okviru Wamp servera, o kome smo već govorili, nalazi se i podrška za MySQL bazu podataka. U okviru baze i opcije *sqlbuddy*možete jednostavno kreirati bazu podataka i tabele u njoj i manipulisati njima. S obzirom na to da je u okviru Wamp Servera, nećete imati dodatne komplikacije prilikom pozivanja vaše baze i pronalaženja njenog sadržaja.

Takođe, preporučljivo je i korišćenje *phpmyadmin* radnih okvira koji su ozbiljniji i specijalizovani za rad sa MySQL-om i koje ćemo koristiti u našem radu. Za objašnjenje korišćenja sqlbuddyja opredelili smo se da biste videli koje još alternative imate i opredelli se za najpogodniju u vašim aplikacijama. phpMyAdmin je najčešće korišćena web aplikacija za upravljanje MySQL-om.

Što se tiče SQL Buddyja, sada ćemo pokazati osovne mogućnosti koje vam pruža.

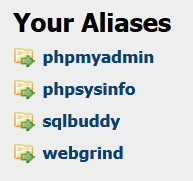
Objasnićemo kako da kreirate vašu željenu bazu u sledećim koracima:

1. Pokrenite Wamp server u okviru vašeg web browsera unosom localhost u prostor za adresu ili klikom direktno na Localhost u meniju Wamp servera.



*18.1. WAMP menu*

2. Odaberite iz menija (na dnu stranice) opciju *sqlbuddy*

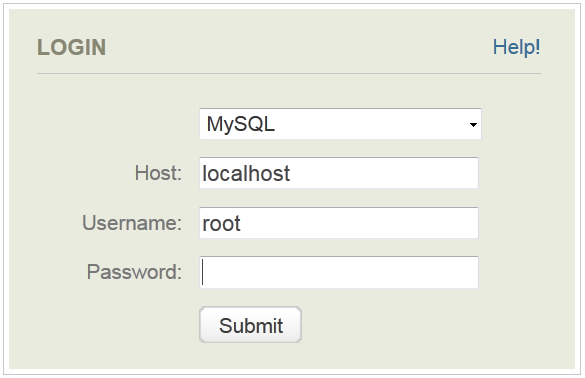
**

*18.2. sqlbuddy link*

Na istu lokaciju možete doći i jednostavnim pristupanjem sledećoj adresi:

http://localhost/sqlbuddy/login.php

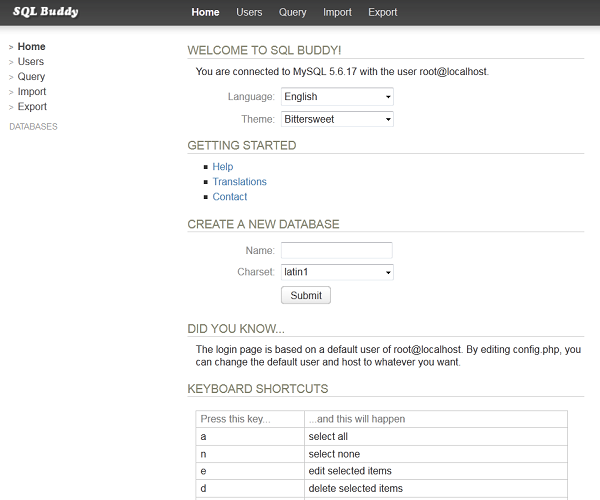
Napomena: Ukloliko se na strani pojavi forma za logovanje slična sledećoj:

**

*18.3. sqlbuddy login*

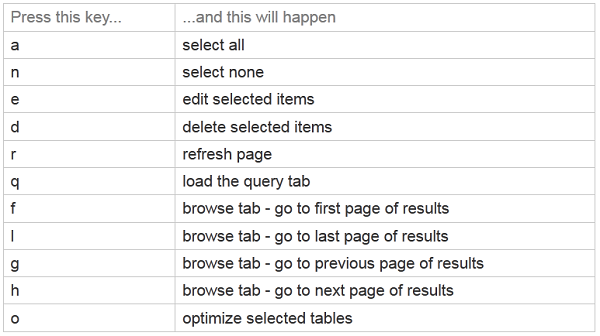
dovoljno je samo odabrati opciju Submit, a polje za Password ostaviti prazno. Naravno, ukoliko ste ranije vršli izmene parametara za logovanje, ovde je potrebno uneti validne vrednosti za te parametre.

3. Na ekranu će biti prikazana strana sledeće sadržine:

**

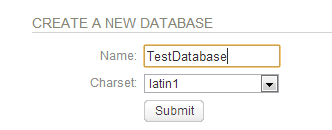
*18.4. sqlbuddy home*

4. U donjem delu prikazane stranice nalazi se i spisak prečica koje će vam olakšati kretanje i kreiranje vaših podataka.



*18.5. sqlbubby – keyboard shortcuts*

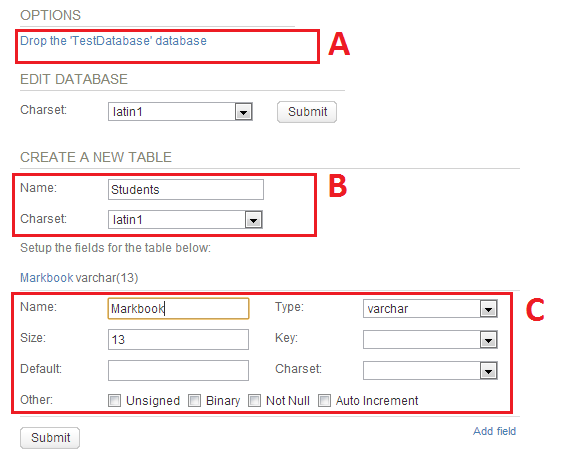
5. Kreiranje baze podataka se izvršava unosom željenog imena i odabirom pisma koje se koristi. Klikom na Submit dugme, vaša baza podataka je kreirana.

**

*18.6. Kreiranje nove tabele*

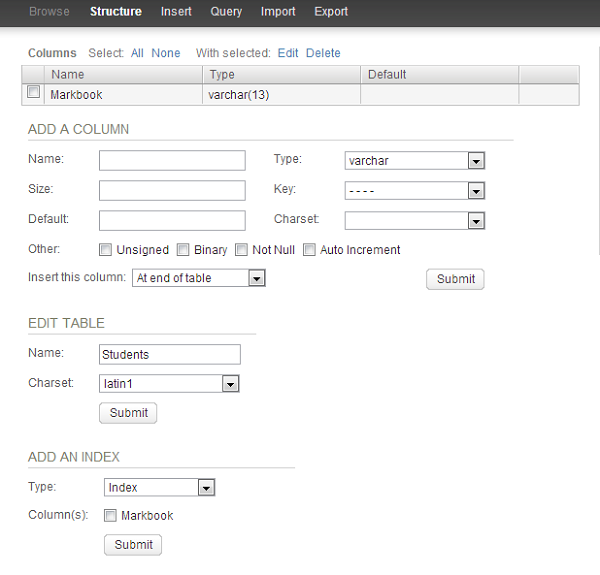
*6.*Kada ste kreirali vašu bazu, dobijate pregled kao na slici. U nastavku imate više mogućnosti za rad.

A. Opcija za uklanjanje baze  
B. Kreiranje tabele  
C. Popunjavanje polja radi unosa novih podataka u bazu



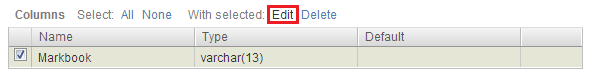
*18.7. sqlbuddy – kreiranje tabele 2*

7. Nakon potvrđivanja unosa tabele i prvog polja u njoj, prikazaće vam se stranica u nastavku sa prikazanim poljima u tabeli i otvara vam se mogućnost da dalje unosite polja i manipulišete primarnim i sekundarnim ključevima tabele.

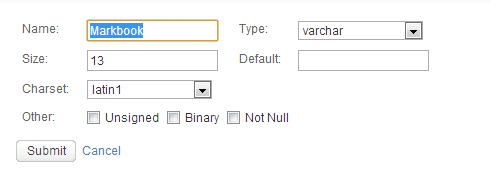
**

*18.8. Editovanje tabele*

8. Kreiranim poljima takođe možete manipulisati. Nakon kreiranja, sva vaša polja biće tabelarno prikazana. Na stranici iznad njih možete uočiti manji meni za rad. Odabirom na *Edit*otvoriće vam se polja za manipulaciju, tačnije, izmenu vašeg polja koje ste selektovali.

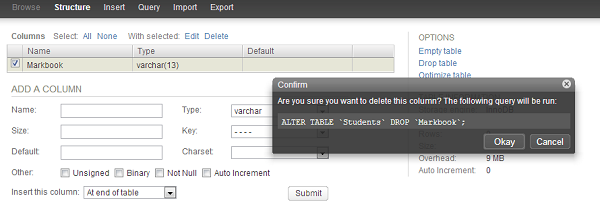


*18.9. sqlbuddy – opcija za editovanje polja*

**

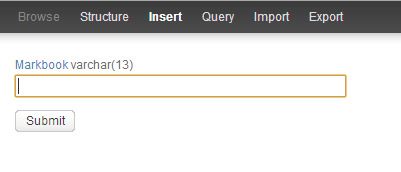
*18.10. sqlbuddy – forma za editovanje polja*

9. Takođe, selektovanjem *Delete*opcije zahtevaće se od vas da potvrdite brisanje označenog polja i izvršenja operacije brisanja.



*18.11. sqlbuddy - delete*

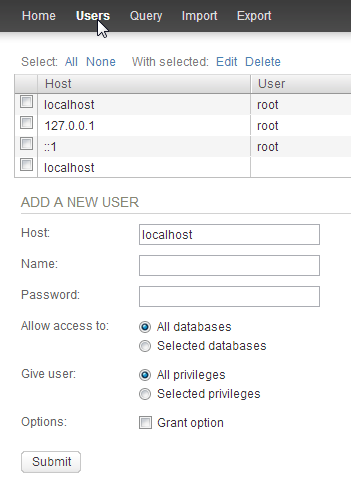
Odabirom „Insert” opcije imate mogućnost da unosite podatke u tabele koje ste kreirali.

**

*18.12. sqlbuddy – insert*

**Promena passworda za pristup *sqlbuddy* radnoj provršini**

Default (osnovno) podešavanje za pristup MySQL-u predviđeno je sa usernameom: „root” i bez passworda, (odnosno, potvrdom na prazno polje predviđeno za unos passworda). Takođe, postoji korišćenje anonimus korisnika bez passworda. Ono što je potrebno da uradite jeste da podesite password za *root*korisničko ime i uklonite anonimus korisnika, i kreirate novog korisnika, ukoliko je potrebno.



*18.13. sqlbuddy – upravljanje korisnicima*

Podešavanje passworda za glavnog korisnika „root”:

* 1. Ulogujte se na SQL Buddy preko URL-a http://localhost/sqlbuddy, sa korisničkim imenom „root” i praznim passwordom,
  2. Odaberite opciju „Users”,
  3. Čekirajte red u kome su *User – root*i *Host – localhost*i odaberite „Edit”,
  4. Unesite novi password u predviđeno polje za unos (Change password),
  5. Ponovite postuke za root@127.0.0.1 (IP adresa za localhost) i root@: :1 (IPv6 localhost adresa).

Za uklanjanje anonimus korisnika potrebno je da uradite sledeće:

* 1. Ulogujte se na SQL Buddy sa nalogom glavnog korisnika,
  2. Odaberite opciju „Users“,
  3. Čekirajte red sa anonimus korisnikom (red u kome je user prazno polje) i odaberite opciju „Delete“.

Ne bi trebalo da koristite glavnog korisnika „root“ za osnovne operacije. Trebalo bi da „root“ bude upotrebljen samo za kreiranje novih korisnika. Takođe, možete kreirati novog korisnika za upotrebu i rad na standardnim operacijama. Za kreiranje novog korisnika zvanog „wampuser“, potrebno je da uradite sledeće:

* 1. Ulogujte se na SQL Buddy sa korisničkim imenom „root“ ,
  2. Odaberite opciju „Users“,
  3. U delu ADD NEW USER, unesite „localhost“ u polje za unos koje pripada *Hostu*, a „wampuser“ u polje za unos koje pripada *Name* i „xxxx“ u polje za unos koje pripada *Passwordu.*Za početak nemojte čekirati „Grant Option“, samo potvrdite klikom na dugme „Submit“.

**Export i Import baza**

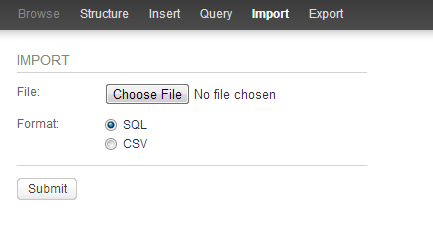
Kada radite sa MySQL-om, imate mogućnost Importa i Exporta baza iz ili u MySQL.

**Import**

Importom Baze u MySQL moguće je da uvedete vašu već postojeću bazu. Baza je preporučljivo da ima .sql ekstenziju.

Bazu možete uvesti na sledeći način:

* 1. Klikom na *Odaberi datoteku*birate fajl koji želite da uvezete,
  2. Potvrdom na *Submit* vaša baza biće importovana i nalaziće se na spisku baza za korišćenje sa leve strane.

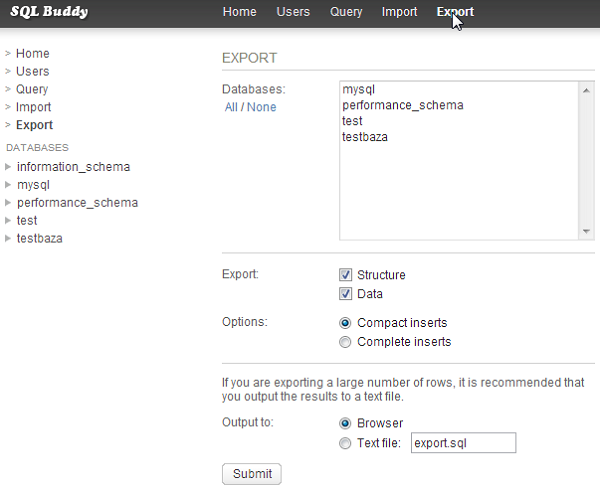
**

*18.14. sqlbuddy - import*

**Export**

Export baze podataka predstavlja izvoz baze iz SQL Buddy okvira. Baza će biti „spakovana“ u .sql fajl i kao takva moći će da se uveze u neke druge okvire baza podataka. Export se vrši na sledeći način:

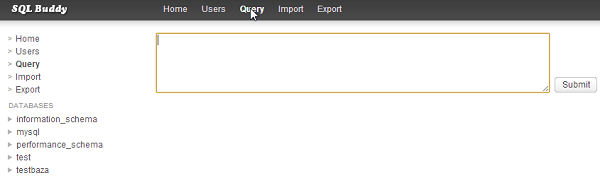
* 1. Iz ponuđene liste odaberete željenu bazu koju želite da izvezete,
  2. U odeljku *Export:* možete odabrati da li izvozite strukturu baze i bazu, najbolje je da čekirate obe opcije i upotpunite bazu,
  3. U *Options:* delu birate da li želite kompaktan, smanjeni insert ili kompletan odnosno ceo insert iz baze podataka,
  4. I u poslednjem delu *Output to:* *Browser* ili *Text file* su opcije koje možete odabrati u zavisnosti od broja redova u odabranoj bazi podataka; kada imate veći broj redova, preporučuje se odabir *Text file,*
  5. Klikom na *Submit*potvrđujete i završavate vaše exportovanje baze.

  
*18.15. sqlbuddy – export*

**Query**

U svakom trenutku, ukoliko želite da komunicirate sa bazom podataka, možete pozvati *Query* opciju i u okviru polja za unos pisati svoje upite. Klikom na Submit potvrđujete izvršenje ispisanog upita.

U ovom trenutku sve što je nama potrebno za rad sa bazom nalazi se u olakšanom delu SQL Buddyja, bez potrebe da znate sintaksu kreiranja upita nad vašom bazom.



*18.16. sqlbuddy – query*

**Uspostavljanje veze sa bazom podataka**

Biblioteka funkcija koju koristi PHP za uspostavljanje veze sa MySQL-om zove se ***MySQLi***(slovo *i*je dodato od engleske reči *improved,* što znači poboljšan). Sa korišćenjem ove biblioteke imate mogućnost da koristite i objektno orijentisanu i proceduralnu sintaksu.

Sada ćemo koristiti proceduralnu sintaksu za pristup bazi podataka. Evo i primera:

|  |  |
| --- | --- |
| 1 | @ $db = mysqli\_connect('localhost', 'root', '','testDB'); |

Kao rezultat ove [funkcije](https://www.link-elearning.com/linkdl/opisPojma.php?id=139107) dobićemo vezu sa bazom podataka tipa *resurs.*Ako koristimo proceduralnu sintaksu kao u primeru i dobijeni rezultat kao resurs, potrebno je da dobijeni resurs prosledite svim daljim funkcijama iz biblioteke MySQLi koje upotrebite. Imena funkcija sa proceduralnim verzijama počinju sa ’*mysqli\_’* i tim funkcijama se mora proslediti resurs koji ste prethodno dobili od funkcije *mysqli\_connect().*

Potrebno je da uvek proverite rezultat pokušaja uspostavljanja veze. Razlog tome je što preostali deo skripta neće raditi bez funkcionalne povezanosti sa bazom podataka. Ovakvu proveru možete izvršiti sledećim kodom:

|  |  |
| --- | --- |
| 1  2  3  4  5 | if(mysqli\_connect\_errno())  {  echo 'Connection failed! ';  exit;  } |

Funkcija koju smo upotrebili, mysqli\_connect\_errno(), vraća broj grešaka u slučaju neuspešnog izvršavanja. U slučaju da je konekcija uspešno izvršena, vraća nulu.

Takođe, preporuka je da kada uspostavljate vezu sa bazom podataka, red koda u kome se nalazi konekcija na bazu podataka započnete sa ’@’ kako biste imali mogućnost da potencijalne greške obradite kako smatrate da treba.

Kada radite na konekcijama sa bazom, potrebno je da znate i da postoji ograničen broj istovremenih veza koje se mogu uspostaviti sa MySQL serverom. U okviru MySQL-a postoji i parametar *max\_connections,* kojim se određuje maksimalan broj veza. Ovaj parametar svojim radom sprečava preopterećenje servera iz razloga prevelikog broja zahteva ili lošeg rada softvera.

Izvršavanjem SQL komande:

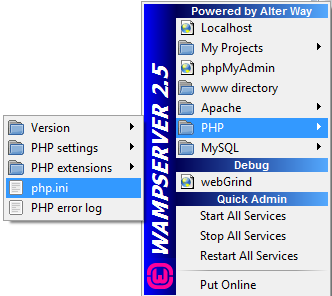
show variables like "max\_connections";

možete proveriti trenutna podešavanja za maksimalan broj konekcija.

Standardnu vrednost ovog parametra možete izmeniti sledećom SQL naredbom:

set global max\_connections = 200;

ili izmenama u datoteci php.ini.

**

*18.17. WampServer – php.ini*

**Selektovanje baze podataka**

Kada se sa bazom povezujete sa weba, potrebno je da bazu podataka sa kojom želite da se povežete navedete kao parametar funkcije mysqli\_connect(). Ukoliko iz tekuće baze želite da pređete u neku drugu, potrebno je da ispišete sledeći kod:

|  |  |
| --- | --- |
| 1 | mysqli\_select\_db( $conn, $dbName); |

u ovom primeru možemo uočiti poziv resursa kao parametra funkcije mysqli\_select\_db, koji smo spominjali.

**Upiti ka bazi podataka**

Izvršavanje upita ka bazi podataka obavlja se preko *mysqli\_query()* funkcije. Pre upotrebe same funkcije, potrebno je najpre definisati sam upit (query). Upit možete definisati na sledeći način:

|  |  |
| --- | --- |
| 1 | $query = "SELECT \* FROM users WHERE first\_name = '". $name . "'"; |

U ovom upitu $name predstavlja promenljivu koju je korisnik definisao (dodelio joj vrednost) u okviru aplikacije. Promenljiva se dalje poredi sa poljem u bazi i na taj način filtrira rezultat upita.

Nakon definisanja upita, možemo ga izvršiti na sledeći način:

|  |  |
| --- | --- |
| 1 | $result = mysqli\_query( $conn, $query ); |

Parametri funkcije mysqli\_query() su označeni promenljivama $conn i $query. Prvi parametar ($conn) predstavlja vezu ka bazi podataka (link) koju smo kreirali upotrebom funkcije mysqli\_connect(). Drugi parametar je sam upit koji smo takođe kreirali i smestili u promenljivu $query.  
  
Rezultati pozivanja željenog upita smeštaju se u promenljivu *$result* da bismo kasnije mogli da manipulišemo njima. Ukoliko dođe do greške prilikom izvršenja i funkcija se ne izvrši kako je planirano, rezultat će biti logička vrednost *false*.

**Učitavanje rezultata upita**

Upotrebom funkcije *mysqli\_num\_rows()* prebrojavamo redove koje je upit vratio. Kao parameter potrebno joj je proslediti identifikator rezultata, na sledeći način:

|  |  |
| --- | --- |
| 1 | $num\_results = mysqli\_num\_rows( $result ); |

Izvršenjem funkcije, broj redova koji ste dobili kao rezultat dalje možete korisno upotrebiti ukoliko želite, ili obraditi rezultat upita. Kada znate koliko redova imate, možete upotrebiti petlju for na sledeći način:

for ( $i = 0; $i < $num\_results; $i++ )  
{  
...OBRADA REZULTATA…  
}

**Raskidanje veze sa bazom podataka**

Skup rezultata možete osloboditi pozivom metode:

|  |  |
| --- | --- |
| 1 | mysqli\_free\_result( $results ); |

i nakon toga možete upotrebiti metod:

|  |  |
| --- | --- |
| 1 | mysqli\_close($conn); |

Da biste prekinuli vezu sa bazom podataka, nije uvek potrebno da koristite ovu metodu, jer će se veza svakako prekinuti po izvršavanju skripta.

**Vežba 1**

***Problem:***

Potrebno je napraviti kraću bazu podataka koristeći SQL Buddy razvojno okruženje i koristeći se proceduralnom sintaksom (funkcije iz biblioteke mysqli() ... ) napisati otvaranje konekcije, ispisivanje upita, pozivanje upita i zatvaranje konekcije.

***Rešenje:***

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | <?php  $servername = "localhost";  $username = "root";  $password = "";  $db = "testdb";    // Create connection  $conn = mysqli\_connect($servername, $username, $password, $db);    // Check connection  if (!$conn) {      die("Connection failed: " . mysqli\_connect\_error());  }  echo "Connected successfully! <br>";    $name = 'Peter';    $query = "SELECT \* FROM users WHERE firstname = '$name'";  $result = mysqli\_query($conn, $query);    if (mysqli\_num\_rows($result) > 0) {      // output data of each row      while($row = mysqli\_fetch\_assoc($result)) {          echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";      }  } else {      echo "0 results";  }    mysqli\_close($conn);  ?> |

Da bismo pristupili bazi podataka, potrebno je da znamo gde se ta baza podataka nalazi, potrebno je da znamo naziv baze kojoj želimo da pristupimo i potrebno je da imamo pristupne podatke. Upravo ćemo zato sve te podatke smestiti u promenljive:

|  |  |
| --- | --- |
| 1  2  3  4 | $servername = "localhost";  $username = "root";  $password = "";  $db = "testdb"; |

Sada možemo pristupiti kreiranju konekcije sa bazom:

|  |  |
| --- | --- |
| 1  2 | // Create connection  $conn = mysqli\_connect($servername, $username, $password, $db); |

Funkciji mysqli\_connect prosleđujemo parametre navedenim redosledom. Nakon konekcije možemo tu konekciju i proveriti:

|  |  |
| --- | --- |
| 1  2  3  4  5 | // Check connection  if (!$conn) {      die("Connection failed: " . mysqli\_connect\_error());  }  echo "Connected successfully! <br>"; |

Funkcija mysqli\_connect\_error će vratiti generisanu poruku koja bliže ukazuje na razlog neuspešne konekcije sa bazom.  
  
Kreiramo promenljivu $name i dodeljujemo joj vrednost koja treba biti pretražena u bazi.

|  |  |
| --- | --- |
| 1 | $name = 'Peter'; |

Kreiramo upit ka bazi:

|  |  |
| --- | --- |
| 1 | $query = "SELECT \* FROM users WHERE firstname = '$name'"; |

Nakon toga možemo izvršiti prethodno generisani upit:

|  |  |
| --- | --- |
| 1 | $result = mysqli\_query($conn, $query); |

Kao prvi parametar funkcije mysqli\_query postavljamo link ka bazi, dok je drugi sam upit.  
  
Ukoliko postoje selektovani redovi, odnosno ukoliko ima rezultata pretrage, potrebno je te rezultate i prikazati. Zato najpre proveravamo da li je broj vraćenih redova veći od 0 i ukoliko jeste, koristimo while petlju da bismo prošli kroz svaki red:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | if (mysqli\_num\_rows($result) > 0) {      // output data of each row      while($row = mysqli\_fetch\_assoc($result)) {          echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";      }  } else {      echo "0 results";  } |

Sa druge strane, ukoliko je broj selektovanih redova 0, znači da nema rezultata pretrage i korisnika treba obavestiti o tome.  
  
Na samom kraju ostaje da zatvorimo konekciju sa bazom:

|  |  |
| --- | --- |
| 1 | mysqli\_close($conn); |

Ovde treba napomenuti da bi konekcija i automatski bila završena nakon završetka skripte.

**Šta je potrebno proslediti kao argumente funkciji mysqli\_query()?**

 $db, $query

 $db

 $query

 NULL