



# Principi softverskog inženjerstva

## Vežbe - VIII nedelja - PHP & MySQL

Dražen Drašković, asistent  
Elektrotehnički fakultet  
Univerziteta u Beogradu

# Šta je SQL?



- SQL (*Structured Query Language*) - strukturirani jezik za upite
- SQL je standardizovani jezik za pristupanje sistemima za upravljanje relacionim bazama podataka (RDBMS).
- SQL omogućava unošenje podataka u bazu i čitanje podataka iz nje.
- SQL se koristi u sistemima za upravljanje bazama podataka: MySQL, Oracle, PostgreSQL, Sybase, Microsoft SQL Server i drugi.



## *Funkcije za rad sa MySQL*

# Funkcije za rad sa MySQL-om

- Ove funkcije omogućavaju pristup MySQL database serveru.
- Podela:
  - mysql funkcije (MySQL functions)
    - proceduralni pristup
  - mysql\_i funkcije (MySQL Improved Extensions)
    - proceduralni pristup
    - objektno-orjentisani pristup

# MySQL funkcije



- Starija biblioteka funkcija za rad sa MySQL-om
- Ne podržava neka proširenja MySQL-a uvedena u verzijama 4.1 i novijim
- Proceduralni pristup
- MySQL funkcije nisu ugrađene u PHP.  
=> Treba dodati odgovarajuće dinamičko proširenje (dynamic extension) u php.ini fajlu:

```
extension=php_mysql.dll
```

# MySQL Improved Extensions

- Novija biblioteka funkcija za rad sa MySQL-om
- Podržava proširenja MySQL-a uvedena u verzijama 4.1 i novijim
- Proceduralni ili objektno-orijentisani pristup
- MySQL Improved f-je nisu ugrađene u PHP.  
=> Treba dodati odgovarajuće dinamičko proširenje (dynamic extension) u php.ini fajlu:

```
extension=php_mysql.dll
```



## *Čítanje podataka iz MySQL baze*

# Čitanje podataka iz baze



1. Uspostavljanje veze sa MySQL-om i označavanje trenutno aktivne baze podataka **mysqli\_connect**
2. Izvršavanje upita i dobijanje skupa rezultata **mysqli\_query**
3. Učitavanje reda rezultata iz skupa rezultata **mysqli\_fetch\_row**
4. Obrada dobijenih vrednosti za učitani red
5. Zatvaranje konekcije sa bazom podataka **mysqli\_close**



# Uspostavljanje veze sa bazom

```
mysqli mysqli_connect(  
    string host ,  
    string username ,  
    string password ,  
    string database )
```

- Uspostavlja vezu sa RDBMS-om, i označava aktivnu bazu podataka.
- Ova funkcija umesto objekta vraća rezultat tipa resurs (objekat tipa mysqli, odnosno konekciju na MySQL server) ili *false*, ako konekcija ne može da se uspostavi



**bool mysqli\_select\_db**  
**(mysqli connection , string database)**

Označava BP zadatu parametrom *database* kao trenutno aktivnu (BP nad kojom će se izvršavati upit koji se prosledi po ostvarenoj MySQL konekciji), u vezi sa MySQL-om zadatoj parametrom *connection*.

Funkcija vraća:

TRUE, u slučaju uspeha

FALSE, u slučaju neuspeha.

# Zadavanje upita bazi podataka

**mysqli\_result mysqli\_query**  
**(mysqli connection , string SQL\_command)**

Izvršava SQL iskaz čiji je tekst zadat parametrom *SQL\_command*.

Parametar *connection* je objekat tipa *mysqli* (konekcija sa MySQL-om) dobijen kao povratna vrednost funkcije *mysqli\_connect()*.

Ova funkcija vraća:

objekat tipa *mysqli\_result* (skup rezultat upita), za komande koje vraćaju rezultat SELECT, SHOW, DESCRIBE ili EXPLAIN

TRUE, za ostale komande u slučaju uspeha

FALSE, u slučaju neuspeha

# mysqli\_fetch\_row()



## array mysqli\_fetch\_row (mysqli\_result result\_set)

- Učitava red po red rezultata iz objekta tipa `mysqli_result`, dobijenog kao povratna vrednost funkcije `mysqli_query`
- Funkcija vraća:
  - numerički niz koji predstavlja red rezultata (prva kolona - element sa indeksom 0 itd.)
  - NULL, kada nema više redova u skupu rezultata

# mysqli\_close()



## **bool mysqli\_close (mysqli connection)**

- Zatvara konekciju sa MySQL-om koja je prethodno uspostavljena pozivanjem funkcije `mysqli_connect()`.
- Funkcija vraća:
  - TRUE, u slučaju uspeha
  - FALSE, u slučaju neuspeha

# Ostale mysql funkcije

- `mysql_free_result`
- `mysql_num_rows`
- `mysql_set_charset`
- *Napomena:*  
Potpun spisak i detaljna objašnjenja svih mysql funkcija u PHP dokumentaciji

# mysqli\_free\_result



```
void mysqli_free_result  
( mysqli_result result )
```

Oslobađa memoriju dodeljenu parametru *result*,  
dobijenim kao povratna vrednost f-je `mysqli_query`.

Bez poziva `mysqli_free_result` f-je memorija dodeljena  
skupu rezultata bi se oslobodila tek pri završetku PHP  
skripta.

# Obrada grešaka (#1)



- Pri upotrebi MySQL-ovih funkcija za rad s bazama podataka postoji verovatnoća da se pozivanje neke od navedenih funkcija završi neuspehom
- Uzroci neuspeha: RDBMS nije dostupan, zadat je pogrešan parametar, korisnik je uneo pogrešnu lozinku, upit je pogrešno sastavljen, ...
- U tom slučaju PHP će u većini slučajeva prikazati neku svoju grešku (najčešće na nivou Warning-a) i nastaviti sa izvršavanjem skripta



# Obrada grešaka (#2)



- Cilj je izbeći ovakvo ponašanje, i obezbediti sledeće:
  - omogućiti korisniku da ispisuje greške kako on želi (umesto PHP-ovih grešaka) i
  - omogućiti prekid izvršavanja skripta u slučajevima kada je to potrebno (ako ne uspe konekcija sa BP nema smisla nastaviti dalje,...)

# Funkcije za obradu MySQLi grešaka (#1)

- Grupa funkcija MySQL Improved sadrži i funkcije za obradu grešaka koje mogu nastati pri radu sa mysql\_i f-jama, koje omogućavaju otkrivanje i detaljno obaveštavanje o nastalim greškama
- Ove funkcije mogu omogućavaju programerima lakše otkrivanje i ispravljanje grešaka, one nisu od interesa krajnjim korisnicima

# Funkcije za obradu MySQLi grešaka (#2)

## **int mysqli\_connect\_errno ( )**

F-ja vraća:

broj (kod) greške nastale pri poslednjem pozivu f-je  
mysqli\_connect()  
0, ako nije došlo do greške

## **string mysqli\_connect\_error ( )**

F-ja vraća:

tekst opisa greške nastale pri poslednjem pozivu f-je  
mysqli\_connect()  
prazan string, ako nije došlo do greške

# Funkcije za obradu MySQLi grešaka (#3)

## **int mysql\_errno ( mysqli connection )**

F-ja vraća:

broj (kod) poslednje greške nastale pri pozivu neke mysqli f-je u  
konekciji *connection*

0, ako nije došlo do greške

## **string mysqli\_error ( mysqli connection)**

F-ja vraća:

tekst opisa poslednje greške nastale pri pozivu neke mysqli f-je u  
konekciji *connection*

prazan string, ako nije došlo do greške

# Funkcije za obradu MySQLi grešaka (#4)

- Korisno bi bilo napisati sledeće korisnički definisane funkcije:

```
function showerror_connection() {  
    die ("Greška ".mysql_connect_errno(). " :".  
        mysql_connect_error());  
}
```

```
function showerror($connection) {  
    die ("Greška ".mysql_errno($connection). " :".  
        mysql_error($connection));  
}
```

- Ove funkcije će programeri pozivati u slučaju otkrivanja greške pri pozivu neke mysql funkcije, kako bi dobili detaljne informacije o nastaloj grešci.

# Operator @



- Operator @ predstavlja **operator za kontrolu prikaza standardnih PHP-ovih poruka o greškama** u veb pretraživaču
- Ako se operator @ stavi ispred nekog izraza, bilo koja PHP-ova standardna greška koja je eventualno nastala u toku izračunavanja izraza neće biti prikazana u veb pretraživaču (što bi se desilo ako bi izostavili @)
- Podsetnik: `display_errors=On` (php.ini)

# Upotreba datoteka za umetanje (#1)

- Pogodno je parametre konekcije sa BP (host, username, password, database) čuvati u posebnoj datoteci
- Fleksibilnost podešavanja tih parametara na jednom, centralnom mestu olakšava:
  - testiranje sistema sa rezervnom ili udaljenom kopijom podataka, jednostavnim menjanjem imena baze podataka i/ili servera u datoteci.
  - testiranje sistema sa različitim kombinacijama korisničkih imena i lozinki, sa različitim ovlašćenjima

# Upotreba datoteka za umetanje (#2)

- Pogodno je i korisnički definisane funkcije za obradu grešaka čuvati u posebnoj datoteci.
- Datoteke za umetanje najčešće imaju ekstenziju .inc ili .php



# Upotreba datoteka za umetanje - bezbedonosni problem -

- Ako korisnik aplikacije zahteva datoteku za umetanje sa ekstenzijom .inc, njen sadržaj će se prikazati u veb pretraživaču, tako da se mogu videti parametri i lozinke za pristup RDBMS-u odnosno izvorni kodovi funkcija koji bi trebalo da ostanu skriveni

# Upotreba datoteka za umetanje - rešenje problema: prvi način -

- **Umesto ekstenzije .inc koristiti ekstenziju .php**  
(najbolje rešenje)
- U tom slučaju PHP parser obrađuje sadržaj datoteke za umetanje, ali se u veb pretraživaču ništa ne prikazuje jer datoteka ne sadrži telo skripta
- Jedini nedostatak je što se takve datoteke ne mogu lako razlikovati od drugih datoteka  
=> rešenje: sve datoteke za umetanje staviti u poseban folder (npr. inc)

# Upotreba datoteka za umetanje - rešenje problema: drugi način -

- **Datoteke za umetanje smestiti negde izvan stabla dokumenta u vašoj instalaciji Apache veb servera**
- U tom slučaju se u direktivi include ili require navodi kompletna putanja do datoteke

# Upotreba datoteka za umetanje - rešenje problema: treći način -

- **Podesiti Apache tako da se korisnicima ne dozvoljava učitavanje sadržaja datoteka sa ekstenzijom .inc**



## *Rad sa drugim bazama podataka*

# Rad sa drugim RDBMS



- Rad sa drugim RDBMS (*Relational Database Management System*) se odvija na sličan način kao sa MySQL-om (isti niz koraka).
- Za svaki od podržanih RDBMS-a postoji skup funkcija jako sličan onim za MySQL (PostgreSQL, Oracle, Microsoft SQL Server,...).

# ODBC



- Za RDBMS koje PHP ne podržava direktno (kao što je npr. *Microsoft Access*) na raspolaganju su ODBC funkcije (*Open DataBase Conectivity* - otvoreni standard za povezivanje sa bazama podataka)
- ODBC funkcije se mogu koristiti i za RDBMS koje PHP podržava.
- ODBC omogućava zamenu RDBMS-a u sloju baze podataka veb aplikacije, bez potrebe za promenom PHP skripta.

# ODBC klijent



- Kada se RDBMS-u pristupa pomoću ODBC funkcija neophodno je instalirati i ODBC klijent-a za dati RDBMS.
- ODBC klijenta treba da obezbedi proizvođač RDBMS-a, i on je za svaki RDBMS specifičan.





*Čítanje podataka na osnovu  
korisničkih unetih parametara*



- Napraviti HTML formu za pretraživanje prodavnica po nazivu:
  - Korisnik unese parametre pretraživanja u HTML form element i po završetku pritisne dugme Submit.
  - Na osnovu unetih parametara pretraživanja vrši se čitanje zahtevanih podataka iz baze i isti se prikazuju.
  - Koraci 1 i 2 se mogu ponavljati veći broj puta.

# Struktura skriptova



- Dva pristupa
  - Razdvojeni skriptovi:  
Sastoje se iz dve povezane komponente (dva odvojena fajla)  
*Podvarijanta: sa korišćenjem frejmova.*
  - Kombinovani skriptovi:  
Sastoje se jedne komponente (jednog fajla)

# Razdvojeni skriptovi



- Sastoje se iz dve povezane komponente (dva odvojena fajla)
  - html (ili php) fajl koji sadrži HTML formu za unos parametara koje prosleđuje ovom drugom php fajlu
  - php fajl koji prihvata prosleđene parametre, obrađuje ih, čita podatke iz baze i prikazuje rezultate; po pravilu, ovaj fajl sadrži link na prvi html (ili php) fajl za ponovno zadavanje parametara pretraživanja

# Kombinovani skriptovi #1



- Dve komponente razdvojenih skriptova, html (ili php) fajl koji sadrži HTML formu za unos parametara i PHP fajl koji vrši prijem, obradu i prikaz podataka, se nalaze u istom fajlu.
- Ove komponente su potpuno iste kao odgovarajuće komponente odvojenih skriptova.
- Treba dodati upravljački deo koji određuje kada se izvršava jedna, a kada druga komponenta.

# Kombinovani skriptovi #2



- Kada je kombinovani (jedinostveni) skript pozvan bez parametara, treba da se izvrši komponenta koja predstavlja HTML formu za unos parametara.
- Kada je kombinovani (jedinostveni) skript pozvan sa parametrima, prosleđenim od samog sebe, tačnije od komponente koja predstavlja HTML formu, potrebno je izvršiti PHP komponentu koja prima, obrađuje i prikazuje podatke.

# Struktura kombinovanih skriptova

```
<?php
    if (empty($_POST["poslato_iz_forme"])) {
?>
    ... HTML (PHP) form komponenta sa hidden
    elementom poslato_iz_forme=1
```

```
<?php
    }/* if */ else{
?>
    ... PHP komponenta za obradu i prikaz
```

```
<?php
    }//else
?>
```



## *Ugradnja parametara u hiperlinkove*



# Karakteristični use case



- Slučaj čitanja podataka iz BP na osnovu korisnički prosleđenih podataka ugrađenih u hiperlinkove (karakteristični use case)
  - Korisniku se prikazuje spisak stavki, koje ujedno predstavljaju hiperlinkove
  - Kada korisnik odabere određenu stavku tj. prati odgovarajući link, vrši se čitanje iz baze i prikaz podataka vezanih za odabranu stavku
  - Koraci 1 i 2 se mogu ponavljati veći broj puta

# Struktura skriptova



- Kod prosleđivanja parametara putem hiperlinkova, postoji obično dva povezana PHP fajla.
- Prvi fajl prikazuje sve stavke. Svaka stavka predstavlja hiperlink koji referencira drugi fajl, samo se za svaku stavku prosleđuje drugačija vrednost parametra.
- Drugi fajl prihvata parametar prosleđen putem hiperlinka od prvog fajla, čita iz baze i prikazuje podatke o izabranoj stavki u okviru prvog fajla; po pravilu ovaj fajl sadrži povratni link na prvi fajl.