

Serverske tehnologije

PHP, MySQL

Ak. god. 2011/2012

PHP postavke

- PHP interpreter ima cijeli niz parametara koji utječu na ponašanje interpretera (i vaše aplikacije pisane u php-u).
- Ti parametri se postavljaju prilikom instalacije/konfiguracije php interpretera.
- Postavke php-a možete dobiti pozivom funkcije `phpinfo()` u svojoj php skripti, ali ponekada web host pružatelji usluga onemogućavaju izvođenje te funkcije zbog sigurnosti.
- http://www.fesb.hr/~kiki/primjeri_internet_2/svojstva_php.php

<http://pzi.fesb.hr/proba.php>

PHP postavke

- PHP interpreter može biti instaliran ili kao CGI interpreter ili kao modul Web servera.
- Kada se PHP koristi kao modul Web servera npr. Apache modul, nasljeđuje prava korisnika Web servera. To utječe na sigurnost izvođenja PHP programa. Npr. ako se preko PHP skripte pristupa bazi, ukoliko baza nema ugrađenu vlastitu kontrolu pristupa, baza treba biti dostupna korisniku koji pristupa PHP stranici (obično je to na Apache web serveru "nobody" korisnik). Danas većina Web hosting pružatelja usluga nudi PHP instaliran kao CGI interpreter zbog veće sigurnosti, ali ovakav način rada je sporiji.

Gotove funkcije

- PHP jezik ima cijeli niz biblioteka gotovih funkcija.

<http://hr.php.net/manual/en/funcref.php>

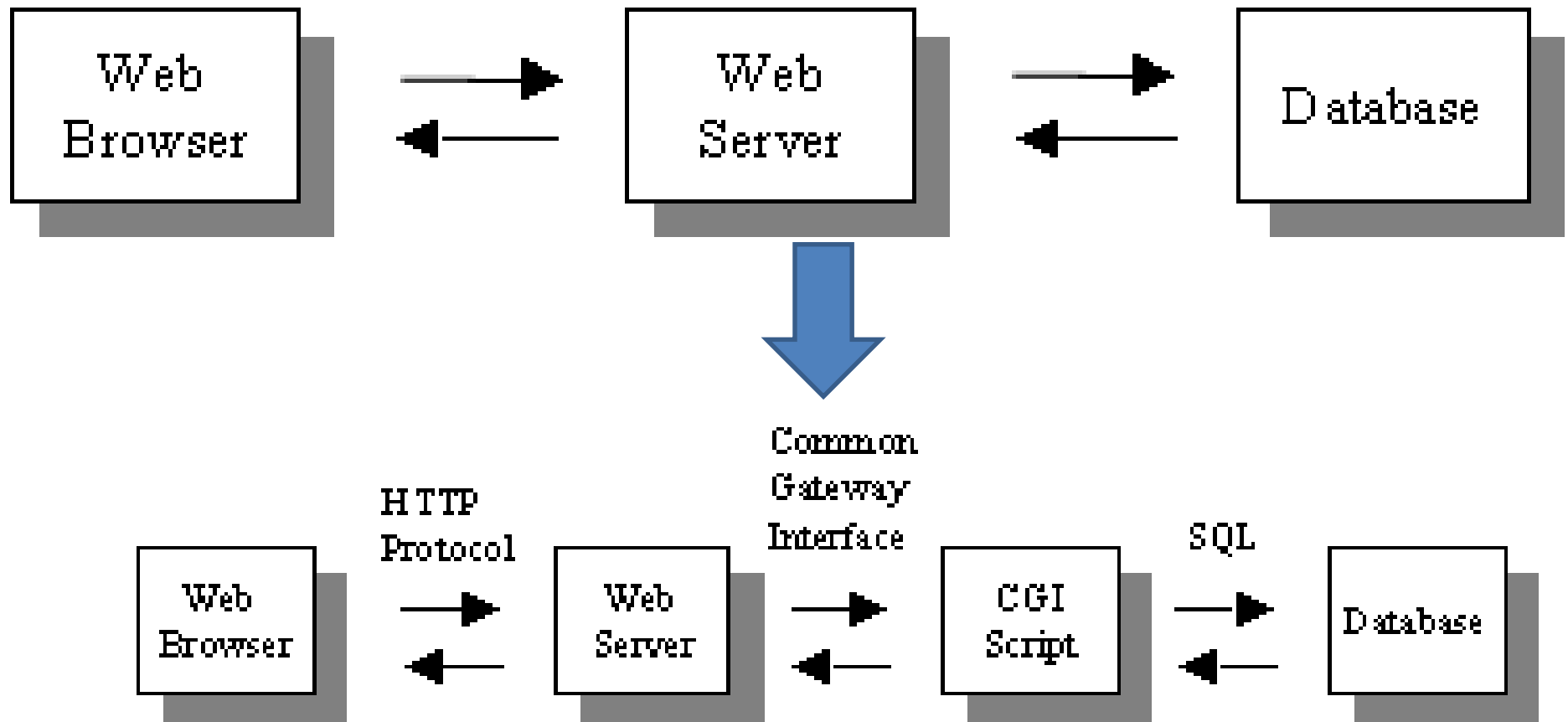
Podrška za baze podataka

- Jedna od velikih prednosti PHP-a je i široka podrška za pristup različitim bazama podataka.
- dBase
- mSQL
- Sybase
- MySQL
- Oracle
- SQLite
- PostgreSQL

BAZE PODATAKA NA WEBU

- Baze na webu uglavnom koriste klijent-server komunikacijsku paradigmu. Ako koristimo takvu bazu podataka, tada se web aplikacija može nalaziti na jednom serveru, a sama baza na nekom drugom serveru.
- Web server treba biti u stanju postaviti upit prema bazi podataka kao njezin klijent te odgovor baze vratiti nazad svom HTTP klijentu koji je i postavio upit.
- Komunikacija između Web servera i servera baze (tj. servisa baze) ostvaruje se opet preko posrednika (*MiddleWare*).
- Jedan posrednik može biti CGI skripta koja zahtjeve web servera prosljeđuje bazi te odgovore baze prosljeđuje nazad web serveru ili php modul integriran u web server koji izvršava php skriptu ili

BAZE PODATAKA NA WEBU



BAZE PODATAKA NA WEBU

- Obično se između baze podataka i aplikacije koja pristupa bazi razvijaju API-ji koji programeru pružaju sučelje od aplikacije prema bazi. API definira koje i kako formatirane naredbe program može poslati bazi te kakav odgovor baze program može dobiti nazad.
- ODBC (*Open Database Connectivity*) je jedan API za pristup bazi podataka koji je razvio Microsoft.
- Implementacije ODBC postoje na različitim OS-ovima (Windows, Linux, Mac OS).

BAZE PODATAKA NA WEBU

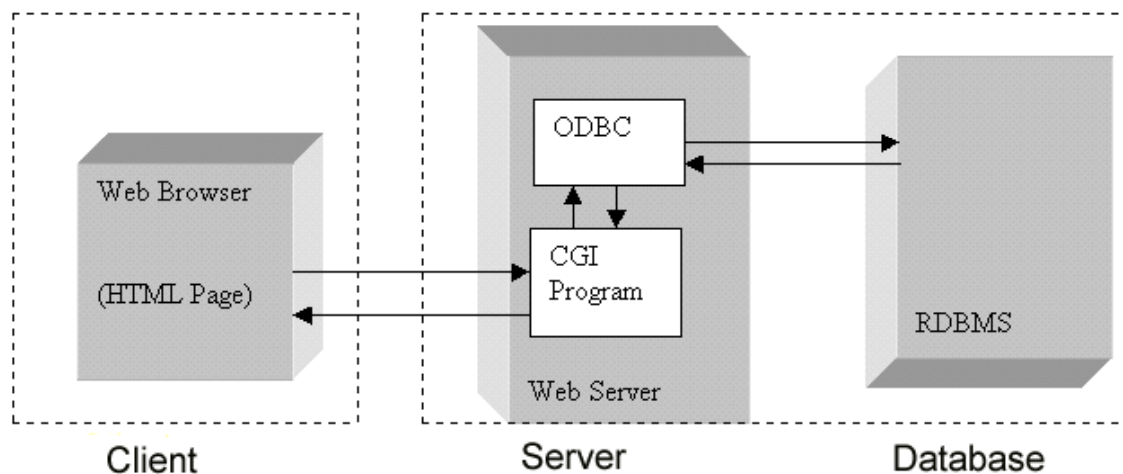
- Uz API treba biti implementiran i softverski driver koji omogućava aplikaciji koja koristi API (npr. ODBC API) da se spoji na bazu.
- ODBC driveri su razvijeni za SQL, Oracle, PostgreSQL, MySQL, itd. te za baze podataka koje nemaju pristup preko mreže tzv. “desktop baze” poput MS Accessa.

BAZE PODATAKA NA WEBU

- DBI (*DataBase Interface*) je također API za pristup bazi podataka razvijen za programski jezik Perl. DBI moduli su razvijeni za Oracle, MySQL, MiniSQL, Sybase, itd.
- MySQL baza ima razvijen API za C, C++, Eiffel, Java, Perl, PHP, Python i TCL. Za PHP, API za pristup MySQL bazi je sastavni dio PHP-a.
- MySQL je višekorisnička, višenićna (multi-thread) SQL baza podataka. SQL (*Structured Query Language*) je standardizirani jezik baze podataka. MySQL je klijent/server aplikacija koja se sastoji od server *dimona* (*daemon* - program koji se vrti u pozadini bez direktne interakcije s korisnikom) mysqld i razlićitih klijentskih programa i biblioteka.

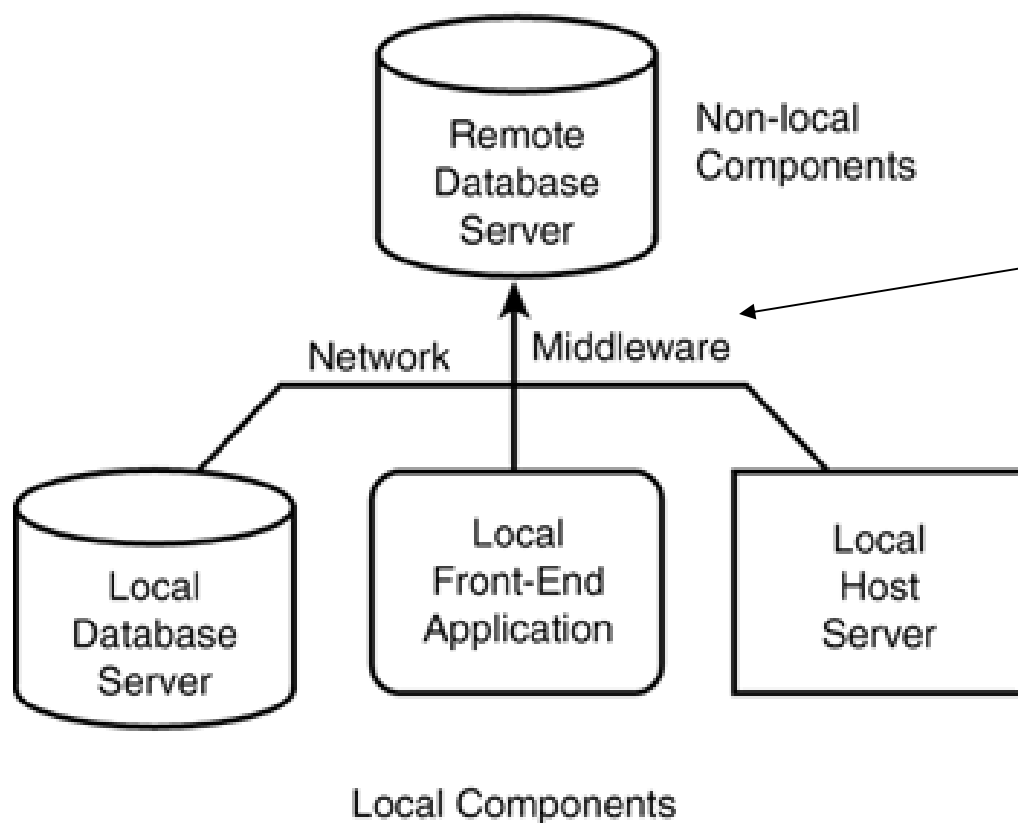
BAZE PODATAKA NA WEBU

Baza se može nalaziti na istom serveru kao i web aplikacija.



Relational Database Management System

BAZE PODATAKA NA WEBU



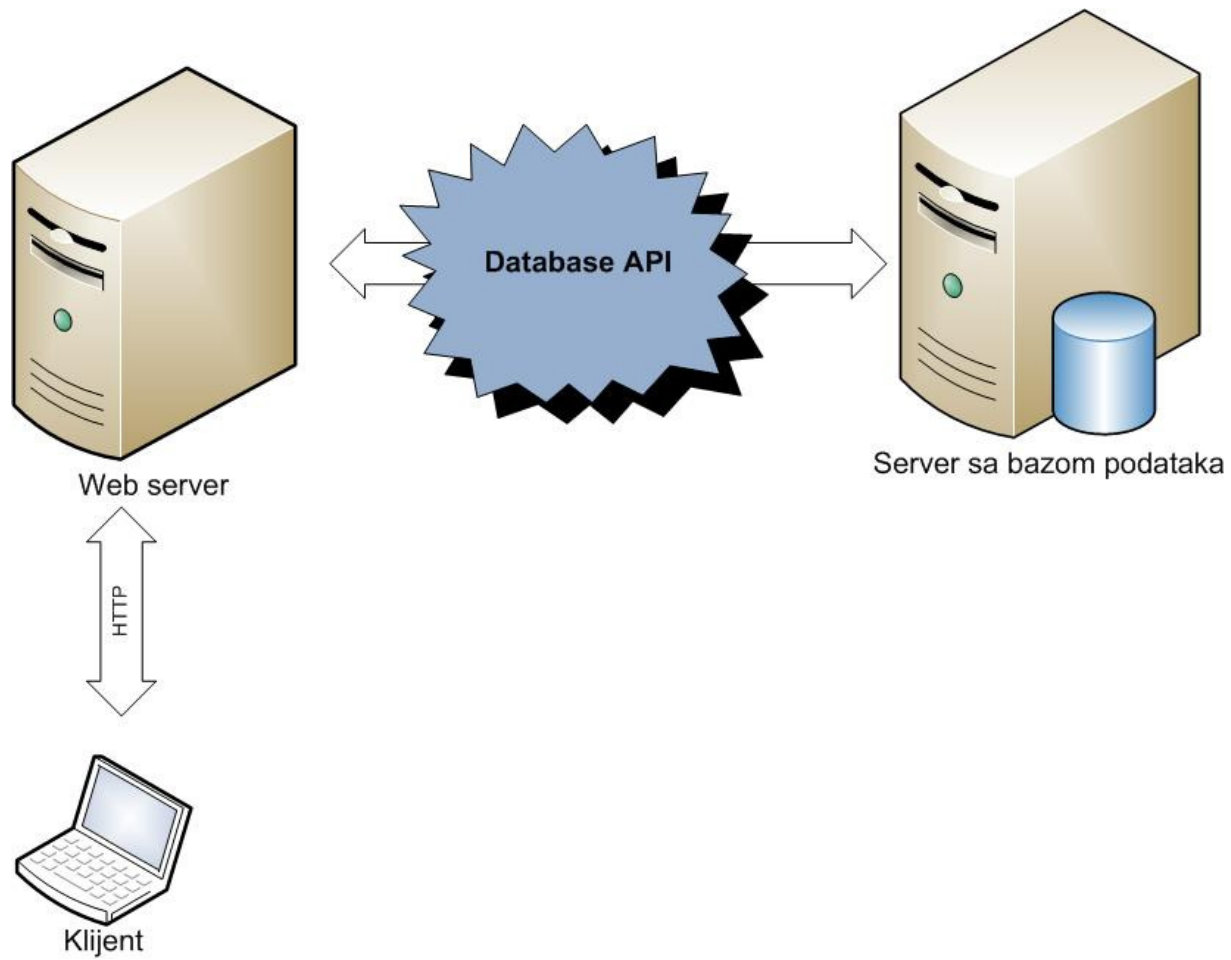
ODBC, JDBC,...

Baza se može nalaziti na dislociranom serveru ako se radi o bazi koja ima podršku za mrežni pristup. Kao middleware za pristup bazi se opet koriste ODBC, JDBC i druga sučelja.

MySQL konekcija

- Klijenti se na MySQL server mogu spojiti na dva načina. Jedan način je korištenje Unix *domain socketa* (*file socketa*) tj. lokalnog soketa (ili korištenje *named pipes* na Windowsima), a drugi način je korištenje TCP/IP konekcije tj. soketa.
- Unix *domain socket* i *named pipes* se koriste samo za lokalne konekcije. Web server i MySQL server moraju biti na istom računalu.
- Ako se koristi opcija sa spajanjem preko TCP/IP konekcije server baze osluškuje na portu (obično 3306) dolazne zahtjeve klijenata (kao i drugi servisi) i na njega se bilo koji klijent može spojiti sa bilo kojeg računala na mreži.

MySQL



MySQL klijenti

- Kada imate instaliranu MySQL bazu po defaultu jedini MySQL klijent koji dolazi preinstaliran sa bazom je konzolni klijent koji se poziva naredbom (u konzoli):
- `mysql -u korisnicko_ime -p;`
- Nakon toga se mogu pozivati SQL naredbe.
Npr:
- `SHOW databases;`

PHP MySQL klijent

- phpMyAdmin je gotova klijentska aplikacija u php-u koji omogućava pristup i administriranje MySQL baze kroz grafičko web sučelje.
- Aplikacija je besplatna, relativno se jednostavno instalira i koristi.
- Primjer:
- <http://pzi.fesb.hr/phpMyAdmin-3.4.8-english>
- <http://www.morsko-prase.hr/phpMyAdmin>

MySQL i PHP

- Funkcije koje omogućuju pristup MySQL serveru (servisu): <http://hr.php.net/manual/en/ref.mysql.php>
- Postupak spajanja php klijenta na MySQL bazu se izvodi u nekoliko koraka:
 1. Klijent treba otvoriti konekciju (mrežnu ili lokalnu) prema serveru sa bazom
 2. Klijent treba odabrati bazu (možemo imati cijeli niz različitih baza na istom MySQL serveru)
 3. Nakon toga klijent može generirati SQL upite na bazu
 4. Obrada dohvaćenih podataka

1. spajanje na MySQL bazu

- `mysql_connect` – funkcija za otvaranje konekcije prema MySQL serveru

```
resource mysql_connect ( [string $server [, string  
    $username [, string $password [, bool $new_link  
    [, int $client_flags]]]] )
```

- Parametri funkcije koji se u deklaraciji funkcije navode u zagradama `[]` su opcionalni prilikom navođenja funkcije.
- Povratna vrijednost je tipa *resource* i to je identifikator konekcije ili `FALSE` u slučaju neuspjeha.

1. spajanje na MySQL bazu

- *server* – ime MySQL servera, može sadržavati i broj porta na kojem server sluša "hostname:port" ili put do file socketa ":/path/to/socket" (npr. pzi.fesb.hr:3306 – ako je dozvoljeno spajanje preko mreže na bazu, /var/run/mysqld/mysqld.sock – preko lokalnog soketa)
- Ako se za ime MySQL servera koristi ime "localhost" ili "localhost:port" ignorirati će se pokušaj spajanja preko TCP soketa i konekcija će se pokušati uspostaviti preko lokalnog soketa (ili named pipe na Windowsima). Ako se želi konekcija preko TCP/IP treba umjesto "localhost" koristiti "127.0.0.1".

1. spajanje na MySQL bazu

- *username* – korisničko ime za spajanje na MySQL server
- *password* - lozinka
- *new_link* – ako se funkcija pozove po drugi put sa istim argumentima neće se otvarati nova konekcija nego će se dohvatiti identifikator već otvorene konekcije.
- *client_flags* – može biti kombinacija konstanti MYSQL_CLIENT_SSL, MYSQL_CLIENT_COMPRESS, MYSQL_CLIENT_IGNORE_SPACE ili MYSQL_CLIENT_INTERACTIVE.

MySQL i PHP

- Konekcija prema MySQL serveru se zatvara kada se izvrši skripta u kojoj je konekcija otvorena ukoliko se u skripti već negdje nije eksplicitno zatvorila konekcija pozivom funkcije `mysql_close()`.
- Izvršavanje skripte završava sa zadnjom linijom u pozvanoj skripti.

2. odabir baze

- `mysql_select_db` – funkcija za biranje baze koja će koristiti nakon spajanja na MySQL server

```
bool mysql_select_db ( string  
    $database_name [, resource  
    $link_identifier] )
```

- Povratna vrijednost je `TRUE` ako se funkcija uspješno izvršila, a `FALSE` u slučaju greške.

2. odabir baze

- Obavezni parametar funkcije `mysql_select_db` je string *database_name* koji sadrži ime baze koja se želi koristiti.
- Parametar *link_identifier* je opcionalan. To je MySQL konekcija tj. identifikator konekcije koji je vratila funkcija `mysql_connect()`. Ako nije naveden podrazumijeva se korištenje identifikator od zadnjeg poziva funkcije `mysql_connect()` prije poziva funkcije `mysql_select_db`. Ako ne postoji identifikator (npr. funkcija `mysql_connect` uopće nije pozvana) funkcija će pokušati sama kreirati link.

3. postavljanje SQL upita

- `mysql_query` – funkcija za slanje upita trenutno izabranoj MySQL bazi

`resource mysql_query (string $query [,
resource $link_identifier])`

- Povratna vrijednost je tipa *resource*. Sadržaj povratne vrijednosti za `SELECT`, `SHOW`, `DESCRIBE`, `EXPLAIN` i druge upite koji vraćaju neke podatke je resurs sa tim podacima. Za SQL upite sa `UPDATE`, `DELETE`, `DROP`, i sličnim naredbama je `TRUE` u slučaju uspješnog upita. U slučaju greške povratna vrijednost je uvijek `FALSE`.

3. postavljanje SQL upita

- Obavezni parametar funkcije je string *query* koji sadrži SQL formatirani upit, s time da na kraju ne bi trebao ići znak ;
- *link_identifier* je MySQL konekcija tj. identifikator konekcije koji je vratila funkcija `mysql_connect()`. Ako nije naveden podrazumijeva se korištenje identifikator od zadnjeg poziva funkcije `mysql_connect()` prije poziva funkcije `mysql_query` . Ako ne postoji identifikator (npr. funkcija `mysql_connect` uopće nije pozvana) funkcija će pokušati sama kreirati link.

4. obrada dohvaćenih podataka

- Za obradu dohvaćenih podataka mogu se koristiti različite funkcije poput `mysql_fetch_row`, `mysql_fetch_array`,...

`array mysql_fetch_row (resource
$result)`

-
- `mysql_fetch_row` – funkcija dohvaća jedan redak iz resursa vraćenog pozivom funkcije `mysql_query`.

4. obrada dohvaćenih podataka

- `mysql_fetch_array` – funkcija pristupa slogovima dohvaćenim iz baze u obliku asocijativnog niza (elementima se pristupa preko imena kolone), numeričkog niza (elementima se pristupa preko imena) ili oboje:

`array mysql_fetch_array (resource $result [, int $result_type])`

- *result* – resurs iz kojeg se dobavljaju dohvaćeni slogovi pozivom funkcije `mysql_query`.
- *result_type* – oblik niza u kojem će se vratiti podaci je definiran jednom o sljedećih konstanti `MYSQL_ASSOC`, `MYSQL_NUM`, defaultna vrijednost je `MYSQL_BOTH`.

MySQL i PHP

// 1. konekcija na server

```
$db = mysql_connect("mysqlserver","username","password");
```

// 2. selekcija baze

```
mysql_select_db("imebaze", $db);
```

// 3. postavljanje SQL upita

```
$result = mysql_query("SELECT * FROM nekatablica", $db) ;
```

// 4. dohvaćanje jednog po jedno retka

```
while ($myrow=mysql_fetch_row($result))
```

```
{
```

```
    foreach ($myrow as $value) echo $value;
```

```
}
```

MySQL i PHP

```
<body>
<? if($_REQUEST[submit])
{ // 1. konekcija na server
  $db = mysql_connect("mysqlserver","username","password");
  // 2. selekcija baze
  mysql_select_db("imebaze",$db) or die("Došlo je do problema, pokušajte ponovo ili kontaktirajte
  svog administratora");
  // 3. postavljanje SQL upita
  $sql="INSERT INTO imetablice(imekolone1, imekolone2) VALUES('$_REQUEST[var1]',
  '$_REQUEST[var2]')";
  $result=mysql_query($sql) or die("Došlo je do problema, pokušajte ponovo ili kontaktirajte svog
  administratora");
  printf ("Podaci su ubačeni u bazu!"); }
else
{?>
<FORM ACTION="<?=$PHP_SELF?>" METHOD=POST>
<input type="text" name="var1"/><input type="text" name="var2"/>
<INPUT TYPE="submit" VALUE="Pošalji u bazu" name=submit></FORM>
<?}?>
</body></html>
```