

XAMPP

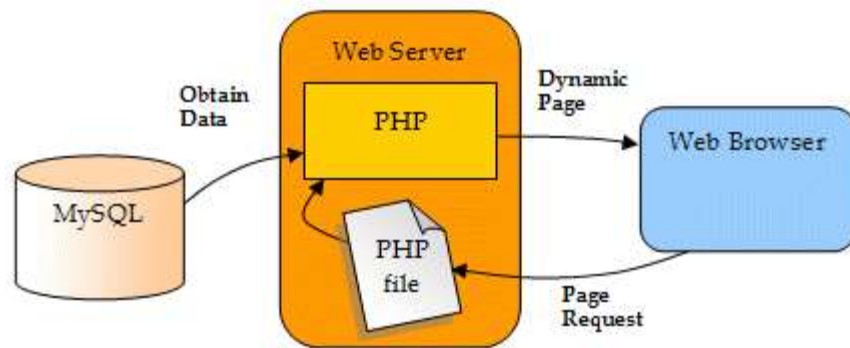


- Što je XAMPP?
- XAMPP – pednosti
- Instaliranje XAMPP-a
- XAMPP – Kontrolni panel
- XAMPP – direktorijumi
- Konfigurisanje XAMPP-a
- LAMP



ŠTO JE XAMPP?

- XAMPP je besplatna open-source platforma, koja sadrži:
 - Apache HTTP server,
 - MySQL bazu podataka,
 - PHP i
 - Perl programski jezik



Naziv **XAMPP** je skraćenica za:

- **X** (čita se „kros“ i znači kros-platforma)
- **A**pache HTTP server
- **M**ySQL
- **P**HP
- **P**erl



ZAŠTO KORISTITI XAMPP?

- Najpopularniji PHP razvojni paket
- Raspoloživ za Windows, Mac OS X i Linux
- Jednostavna instalacija i konfigurisanje
- Sasvim besplatan

INSTALIRANJE?

- Otići na: <https://www.apachefriends.org/download.html>
- Preuzeti i instalirati (može se zahtijevati pokretanja sa administratorskim pravima)
- Uobičajena windows instalacija Next->Next-> ... -> Finish

XAMPP Apache + MariaDB + PHP + Perl

Download

Click here for other versions



XAMPP for Windows

7.4.6 (PHP 7.4.6)



XAMPP for Linux

7.4.6 (PHP 7.4.6)



XAMPP for OS X

7.4.6 (PHP 7.4.6)

KONTROLNI PANEL

XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

XAMPP Control Panel v3.2.4

Service	Module	PID(s)	Port(s)	Actions
☐	Apache	17516 8964	80, 443	Stop Admin Config Logs
☐	MySQL	18256	3306	Stop Admin Config Logs
☐	FileZilla			Start Admin Config Logs
☐	Mercury			Start Admin Config Logs
☐	Tomcat			Start Admin Config Logs

Config Netstat Shell Explorer Services Help Quit

17:27:42 [main] Control Panel Ready
17:30:51 [mysql] Attempting to start MySQL app...
17:30:51 [mysql] Status change detected: running
17:31:05 [Apache] Attempting to stop Apache (PID: 17248)
17:31:05 [Apache] Attempting to stop Apache (PID: 9292)
17:31:05 [Apache] Status change detected: stopped
17:31:07 [Apache] Attempting to start Apache app...
17:31:07 [Apache] Status change detected: running

- ./htdocs – lokacija javnih html fajlova
- ./apache – lokacija konfiguracija
- ./mysql – lokacija MySQL baze podataka



- Apache konfiguracioni fajl (**httpd.conf**):
.\apache\conf\httpd.conf
- PHP konfiguracioni fajl (**php.ini**):
.\apache\bin\php.ini
- MySQL konfiguracioni fajl (**my.cnf**):
.\mysql\bin\mycnf

Više o instaliranju i konfigurisanju:

<https://pureinfotech.com/install-xampp-windows-10/>



LAMP



Linux



Apache



MySQL



Php

- **LAMP** je open-sorce Web razvojna platforma koja koristi **Linux** kao operativni sistem i **Apache** kao Web server.
- **MySQL** je upravljački sistem za relacionu bazu podataka
- **PHP** je objektno orijentisan skriptni jezik

Baza podataka je skup međusobno povezanih podataka, pohranjenih bez nepotrebne redundantnosti, s ciljem da na optimalan način posluže u raznim primjenama.

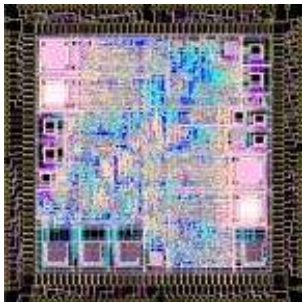
- Podaci se kreiraju nezavisno od programa koji ih koriste. Zajedničkim pristupom dodaju se novi podaci, te mijenjaju i premještaju postojeći.
- Podaci se pohranjuju u bazu podataka na organizovan način, koristeći odgovarajući model podataka.

- Baza podataka je kolekcija podataka.
- Sistem za upravljanje bazom podataka (DBMS – Database Management System) je softver koji kontroliše te podatke.

**Aplikacija
dolazi ovdje**

DBMS

DBMS interfejs
omogućuje da aplikacije
i sistem za upravljanje
podacima budu
izvedeni odvojeno



Sirovi podaci (goli metal)

- **Obezbjeđuje:**

- **Jezik za opis podataka** (DDL - Data definition language)

- **Jezik za rukovanje** podacima (DML - Data manipulation language)

- **Jezik za kontrolu** podataka (DCL - Data control language)

- **Često se ovi jezici** smatraju jednim jezikom – SQL.

- **DBMS obezbjeđuje**

- Trajnost

- Konkurentnost

- Integritet

- Bezbjednost

- Nezavisnost podataka

- **Rječnik podataka**

- Opisuje samu bazu podataka.

- **Fizička nezavisnost podataka.** Razdvaja se logička definicija baze od njene stvarne fizičke građe.

Na primjer, ako se fizička građa promijeni (na primjer, podaci se prepisu u druge datoteke na drugim diskovima), to neće zahtijevati promjene u postojećim aplikacijama.

- **Logička nezavisnost podataka.** Razdvaja se globalna logička definicija cijele baze podataka od lokalne logičke definicije za jednu aplikaciju.

Na primjer, ako se logička definicija promijeni (na primjer uvede se novi entitet ili veza), to neće zahtijevati promjene u postojećim aplikacijama.

- **Mogućnost oporavka nakon kvara.** Zaštita baze u slučaju kvara hardvera ili grešaka u radu sistemskog softvera.

- **Fleksibilnost pristupa podacima.** Korisnik može slobodno pretraživati podatke, i po želji uspostavljati veze među podacima.

U starijim mrežnim i hijerarhijskim bazama, staze pristupanja podacima bile su unaprijed definisane. Korisnik je mogao pretraživati podatke jedino onim redoslijedom koji je bio predviđen u vrijeme projektovanja i implementiranja baze.

- **Istovremeni pristup do podataka.** Mogućnost da veći broj korisnika istovremeno koristi iste podatke. Korisnici ne smiju ometati jedan drugoga.

ŠTO DONOSI DBMS?

- **Zaštita od neovlašćenog korišćenja.** Mogućnost da se korisnicima ograniče prava korišćenja baze.

Svakom korisniku se dodjeljuju ovlašćenja: što on smije, a što ne smije raditi s podacima.

- **Zadovoljavajuća brzina pristupa.** Operacije nad podacima moraju se odvijati dovoljno brzo, u skladu s potrebama određene aplikacije.

Na brzinu pristupa može se uticati izborom pogodnih fizičkih struktura podataka, te izborom pogodnih algoritama za pretraživanje.

- **Mogućnost podešavanja i kontrole.** Velika baza zahtijeva stalnu brigu: praćenje performansi, mijenjanje parametara u fizičkoj građi, rutinsko smještanje rezervnih kopija podataka.

Danas postoji više različitih DBMS-a:

mysql:

`www.mysql.org`

Open source, dosta moćan

MariaDB

PostgreSQL:

`www.postgresql.org`

Open source, moćan

Microsoft Access:

Jenostavan sistem sa puno korisnih grafičkih alata.

Komercijalni sistemi:

Oracle (`www.oracle.com`)

SQL Server (`www.microsoft.com/sql`)

DB2 (`www.ibm.com/db2`)

Više detalja na:

https://www.ucg.ac.me/objave_spisak/blog/5742

- SQL - Structured Query Language

- ANSI Standardi

- ☐ SQL-1989
- ☐ SQL-1992 (SQL2)
- ☐ SQL-1999 (SQL3)
- ☐ SQL-2003
- ☐ SQL-2006
- ☐ SQL-2008
- ☐ SQL-2011
- ☐ SQL-2016
- ☐ SQL-2019

- Različiti DBMS koriste različite SQL

■ SQL obezbjeđuje

- Jezik za opis podataka (DDL - data definition language)
- Jezik za rukovanje podacima (DML - data manipulation language)
- Jezik za kontrolu podataka (DCL - data control language)

■ Osim toga SQL

- se može koristiti iz drugih programskih jezika.
- Može se proširiti u cilju obezbjeđenja uobičajenih programskih konstrukcija (kao što su: if-then, petlje, promjenljive, itd.)

- SQL je deklarativan (neproceduralni) jezik
 - Proceduralni – navodi što kompjuter tačno treba da uradi.
 - Neproceduralni – opisuje zahtijavani rezultat (ne način kako to izračunati).
- Primjer: Neka je data baza podataka sa sljedećim tabelama:
 - Student sa atributima ID, Ime, Adresa.
 - Predmeti sa atributima Šifra, Naziv.
 - Upis sa atributima ID, Šifra, datum, ocjena.
- Dobiti listu studenata koji su odabrali predmet 'Baze podataka'.

Proceduralno programiranje:

```
Set P to be the first Predmet Record
Code = ''
While (P is not null) and (Code = '')
    If (P.Naziv = 'Baze podataka') Then
        Code = P.Code
Set P to be the next Predmet Record
Set IMENA to be empty
Set S to be the first Student Record
While S is not null
    Set U to be the first Upis Record
    While U is not null
        If (U.ID = S.ID) And
            (U.Code = Code) Then
            IMENA = IMENA + S.IME
        Set U to be the next Upis Record
Set S to be the next Student Record
Return IMENA
```

```
/* Nalazenje sifre predmeta */
/* 'Baze podataka' */

/* Lista imena studenata */

/* Za svakog studenta... */

/* Za svaku instacu Upisa... */
/* Ako je student */
/* upisao Baze podataka */
/* dodati ga u listu */
```

Neproceduralno programiranje:

```
SELECT Ime FROM Student, Upis
WHERE (Student.ID = Upis.ID)
AND (Upis.Code =
      (SELECT Code FROM Predmeti
       WHERE Naziv = 'Baze podataka'))
```

Više detalja na:

https://www.ucg.ac.me/objave_spisak/blog/5742

CREATE TABLE

```
<name> (  
    <col-def-1>,  
    <col-def-2>,  
        :  
    <col-def-n>,  
    <constraint-1>,  
        :  
    <constraint-k>)
```

■ Neohodno je navesti:

- ☐ ime tabele
- ☐ listu definicija kolona
- ☐ listu ograničenja (npr. ključevi)


```
<col-name> <type>  
[NULL|NOT NULL]  
[DEFAULT <val>]  
[constraint-1 [,  
constraint-2[,  
...]]]
```

- Svakoj koloni se zadaje ime i tip podatka koji će sadržavati
- Najčešći tipovi:
 - ☐ INT
 - ☐ FLOAT
 - ☐ CHAR (n)
 - ☐ VARCHAR (n)
 - ☐ DATE

- Kolone se mogu navesti kao **NULL** ili **NOT NULL**.
- **NOT NULL** kolone ne mogu imati **NULL** vrijednost.
- Ako naredbom ništa nije navedeno za kolone, podrazumijeva se **NULL**.
- Kolonama se može dodijeliti podrazumijevana (default) vrijednost.
- Samo se navede ključna riječ **DEFAULT** i zatim vrijednost, primjer:

`broj INT DEFAULT 0`

```
CREATE TABLE Student (  
    studID INT NOT NULL,  
    studIme VARCHAR(50) NOT NULL,  
    studAdresa VARCHAR(50) ,  
    studGodina INT DEFAULT 1)  
ENGINE=INNODB;
```

CONSTRAINT

<name>

<type>

<details>

■ Najčešći <type>:

☐ PRIMARY KEY

☐ UNIQUE

☐ FOREIGN KEY

☐ INDEX

- Ograničenja imaju ime – ograničenja pristupa zahtijevaju ime, ali neka druga ne.
- Ograničenja koja se odnose na jednu kolonu, mogu se uključiti u definiciju te kolone.

KREIRANJE TABELE - OGRANIČENJA

- Primarni ključ se definiše kroz ograničenja.
- **PRIMARY KEY** ograničenje uključuje **UNIQUE** i **NOT NULL** ograničenja.

- Za primarni ključ **<details>** je lista kolona koje sačinjavaju ključ.

```
CONSTRAINT <name>  
PRIMARY KEY  
(col1, col2, ...)
```

- Isto kao **PRIMARY KEY** , grupi kolona se može zadati **UNIQUE** ograničenje
- Ovime se definiše kandidat za ključ tabele.

<details> za **UNIQUE** ograničenje je lista kolona koja predstavlja kandidat za ključ.

```
CONSTRAINT <name>  
    UNIQUE  
    (col1, col2, ...)
```

```
CREATE TABLE Student (  
    studID INT NOT NULL,  
    studIme VARCHAR(50) NOT NULL,  
    studAdresa VARCHAR(50) ,  
    studGodina INT DEFAULT 1,  
    CONSTRAINT pkStudent  
        PRIMARY KEY (studID)  
) ENGINE=INNODB;
```

- Web aplikacija
- Olakšava upotrebu MySQL baze podataka (grafički interfejs)
- Pokreće se na:

<http://localhost/phpmyadmin>

- Preuzima se na:

http://www.phpmyadmin.net/home_page/downloads.php

Kako koristiti PHPMyAdmin (u više detalja):

https://www2.slideshare.net/karwanmst/mysql-database-with-phpmyadmin?from_action=save

Naziv baze: **idsys**

Tabele:

- **orgjedinice** – organizacione jedinice korisnika
- **korisnici** – korisnici čiji identifikatori ostvaruju pravo pristupa
- **terminali** – uređaji na kojima se očitavaju identifikatori
- **evidencije** – evidentiranje očitavanja identifikatora

MySQL – SQL – TABELA KORISNICI

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	IDKorisnika 	int(11)			No	None		AUTO_INCREMENT
☐	2	Prezime	varchar(25)	utf8mb4_general_ci		Yes	NULL		
☐	3	Ime	varchar(25)	utf8mb4_general_ci		Yes	NULL		
☐	4	TagID 	varchar(16)	utf8mb4_general_ci		No	None		
☐	5	IDOrgJed	int(11)			No	None		
☐	6	Aktivan	tinyint(1)			No	None		

MySQL – SQL – TABELA ORGJEDINICE

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	IDOrgJed 	int(11)			No	None		AUTO_INCREMENT
☐	2	NazivOrgJed	varchar(100)	utf8mb4_general_ci		Yes	NULL		
☐	3	IDRod	int(11)			Yes	NULL		

MySQL – SQL – TABELA EVIDENCIJE

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	ID 	int(11)			No	None		AUTO_INCREMENT
☐	2	IDKorisnika	int(11)			No	None		
☐	3	Vrijeme	datetime			No	None		
☐	4	TerminalID	int(11)			No	None		
☐	5	tagID	varchar(16)	utf8mb4_general_ci		No	None		

MySQL – SQL – TABELA TERMINALI

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
☐	1	TerminalID 	int(11)			No	None		AUTO_INCREMENT
☐	2	Adresa 	int(11)			No	None		
☐	3	Naziv	varchar(100)	utf8mb4_general_ci		Yes	NULL		
☐	4	Tip	int(11)			Yes	NULL		

- **INSERT** – dodavanje reda (zapisa) u tabelu.
- **UPDATE** – izmjena podataka u zapisu (zapisima) tabele
- **DELETE** – brisanje zapisa iz tabele
- **UPDATE i DELETE** koriste **WHERE** klauzulu kojom se specificira koje zapise izmijeniti ili ukloniti
- **BUDITE PAŽLJIVI**, netačnom **WHERE** klauzulom može se izgubiti puno podataka.

INSERT INTO

<table>

(col1, col2, ...)

VALUES

(val1, val2, ...)

- Broj kolona i vrijednosti mora biti isti.
- Ako se dodaju vrijednosti u svaku kolonu, ne mora se navoditi lista sa imenima kolona
- SQL ne zahtijeva da svaki zapis bude različit (osim ako neko ograničenje to ne nameće).

MySQL – SQL – INSERT - PRIMJER

Neka je polazna tabela sljedeća:

Student

ID	studIme	studAdresa	studGodina
1	Patar Marić	Slobode 23	1

```
INSERT INTO Student
```

```
(ID, studIme, studAdresa, studGodina)
```

```
VALUES (2, 'Marko Matić', 'Pobjede 12', 3)
```

Student

ID	studIme	studAdresa	studGodina
1	Patar Marić	Slobode 23	1
2	Marko Matić	Pobjede 12	3


```
UPDATE <table>
SET col1 = val1
    [,col2 =
    val2...]
[WHERE
    <condition>]
```

- U svim vrstama kod kojih je uslov zadovoljen postavljaju se zadate vrijednosti kolonama.
- BUDITE PAŽLJIVI - ako nije zadat uslov svi zapisi će biti promijenjeni.
- Vrijednosti su konstante ili algebarski izrazi.

MySQL – SQL – UPDATE - PRIMJER

UPDATE Student

SET studGodina = 2, studIme = 'Marina Šoć'

WHERE ID = 4

Student

ID	studIme	studAdresa	studGodina
1	Patar Marić	Slobode 23	1
2	Marko Matić	Pobjede 12	3
3	Ana Tot		1
4	Marina Šoć	Cetinjski put bb	2

UPDATE Student

SET studGodina = studGodina + 1

Student

ID	studIme	studAdresa	studGodina
1	Patar Marić	Slobode 23	2
2	Marko Matić	Pobjede 12	4
3	Ana Tot		2
4	Marina Šoć	Cetinjski put bb	3

- Ukljanja sve zapise koji zadovoljavaju uslov

DELETE FROM

<table>

[WHERE

<condition>]

- Ako neme uslova, onda će SVI zapisi biti obrisani – **BUDITE PAŽLJIVI!**
- Neke verzije SQL-a imaju i naredbu **TRUNCATE TABLE <T>** koja je kao i **DELETE FROM <T>** ali je u nekim situacijama brža.

MySQL – SQL – DELETE - PRIMJER

```
DELETE FROM  
Student  
WHERE studGodina = 2
```

Student

ID	studIme	studAdresa	studGodina
2	Marko Matić	Pobjede 12	4
4	Marina Šoć	Cetinjski put bb	3

Student

ID	studIme	studAdresa	studGodina
----	---------	------------	------------

- SQL komanda koja se najčešće koristi.
 - Upit prema grupi tabela. Rezultat je takođe tabela.
 - Puno opcija.
 - Obično postoji više načina za sastaviti bilo koji upit.

SELECT

[DISTINCT | ALL] <column-list>

FROM <table-names>

[WHERE <condition>]

[ORDER BY <column-list>]

[GROUP BY <column-list>]

[HAVING <condition>]

- (*[]* - optional, *|* - or)

Više detalja na:

https://www.ucg.ac.me/objave_spisak/blog/5742



<https://www.w3schools.com/php/>

Doraditi arduino skeč i PHP fajl tako da se omogući sljedeće:

- Nakon očitavanja MASTER kartice ulazi se u mod upisivanja i brisanja korisnika.
- Nakon primicanja nepoznatog tag-a treba napraviti da se upiše novi red u tabeli korisnici u kojem će se u koloni TagID upisati ID upravo očitano taga, u koloni TerminalID adresa uređaja, a u koloni Aktivan broj 1. U kolonama Ime, Prezime i IDOrgJed upisati vrijednost NULL. Kolona IDKorisnika je AUTO_INCREMENT i u njoj ne treba upisivati vrijednost kroz PHP program. (2 -1 bod)
- Nakon primicanja poznatog tag-a treba napraviti da se obriše red iz tabele korisnici u kojem je u koloni TagID upisan ID upravo očitano taga. (2-1 bod)
- Ponovnim primicanjem MASTER kartice vraća se u mod prepoznavanja.
- Kada je u modu prepoznavanja, treba da radi na sljedeći način:
- Nakon primicanja tag-a treba napraviti da se najprije u tabeli korisnici provjeri da li postoji red u kojem je u koloni TagID upisan ID upravo očitano taga. Ukoliko postoji, uzeti IDKorisnika čiji je to TagID. U tabeli evidencije upisati novi red, u kojem će se u koloni TagID upisati ID upravo očitano taga, u koloni TerminalID adresa uređaja, u koloni IDKorisnika broj preuzet iz tabele korisnici, dok u koloni vrijeme treba upisati trenutni datum i vrijeme. (4-3-2 poena)