Programska potpora komunikacijskim sustavima

11. predavanje, 31. svibnja 2023.

doc. dr. sc. Josip Vuković

SQL i baze podataka

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

- Predavanje izradio izv. prof. dr. sc. Marin Vuković

Creative Commons

- **slobodno smijete:**
  - **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
  - **remiksirati** — prerađivati djelo
- **pod sljedećim uvjetima:**
  - **imenovanje**. Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
  - **nekomercijalno**. Ovo djelo ne smijete koristiti u komercijalne svrhe.
  - **dijeli pod istim uvjetima**. Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.

U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.
Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.
Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licencije preuzet je s http://creativecommons.org/.

Sadržaj predavanja

- Dizajn baze podataka
- Kreiranje korisnika
- SQL upiti
- Primjeri iz koda

Kreiranje baze podataka

- Radimo web aplikaciju u kojoj želimo imati:
  - Studente
  - Predmete
  - Ocjene studenata na predmetima
- Dizajn baze i organizacija podataka
  - Važan korak koji će nam omogućiti jednostavno rukovanje podacima i eventualna **proširenja**
  - Također, razmisliti i o indeksiranju baze -> brža pretraga, no koje podatke indeksirati?

Kreiranje baze podataka - tablice

- Radimo web aplikaciju u kojoj želimo imati:
  - Studente -> tablica `studenti`
  - Predmete –> tablica `predmeti`
  - Ocjene studenata na predmetima -> tablica `ocjene`

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
id	INT	11	☑	☐	☐	☐	PRI		auto_increment		
ime	VARCHAR	30	☐	☐	☐	☑		NULL	None	UTF-8 Unicode	utf8_general_ci
prezime	VARCHAR	30	☐	☐	☐	☑		NULL	None	UTF-8 Unicode	utf8_general_ci

  - Ime i prezime?
    - hrvatski znakovi!
    - koje kodiranje koristiti (*Encoding / Collation*)?

Kreiranje baze podataka – tipovi podataka

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
id	INT	11	☑	☐	☐	☐	PRI		auto_increment		
ime	VARCHAR	30	☐	☐	☐	☑		NULL	None	UTF-8 Unicode	utf8_general_ci
prezime	VARCHAR	30	☐	☐	☐	☑		NULL	None	UTF-8 Unicode	utf8_general_ci

- Tipovi podataka
  - INT – integer definirane duljine (11)
    - Dobar za pretraživanje!
  - VARCHAR – promjenjivi niz znakova definirane duljine (30)
    - Relativno loš za pretraživanje!
- Još tipova podataka za različite svrhe…

Kreiranje baze podataka – tipovi podataka 2

- Još neki tipovi podataka…

Brojevi

```
TINYINT
SMALLINT
MEDIUMINT
INT
BIGINT
FLOAT
DOUBLE
DOUBLE PRECISION
REAL
DECIMAL
```

Znakovi, string, datoteke…

```
CHAR
VARCHAR
TINYTEXT
TEXT
MEDIUMTEXT
LONGTEXT
TINYBLOB
MEDIUMBLOB
BLOB
LONGBLOB
```

Vrijeme i datum

```
DATE
DATETIME
TIMESTAMP
TIME
YEAR
```

Kreiranje baze podataka - postavke

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation
id	INT	11	☑	☐	☐	☐	PRI		auto_increment		
ime	VARCHAR	30	☐	☐	☐	☑		NULL	None	UTF-8 Unicode	utf8_general_ci
prezime	VARCHAR	30	☐	☐	☐	☑		NULL	None	UTF-8 Unicode	utf8_general_ci

- `Allow Null`
  - Smije li podatak biti null – baca grešku!
- `Extra`
  - `Auto_increment`
    - kod dodavanja novog ”retka” automatski se povećava za jedan
    - Brisanje?
  - `On update CURRENT_TIMESTAMP`
  - `Serial default value`
    - Npr. 0 za integer
- `Key`
  - Važno – koji je primarni ključ?
  - Povezivanje više tablica s ključevima

Kreiranje baze podataka - konačno

studenti

Field	Type	Length
id	INT	11
ime	VARCHAR	30
prezime	VARCHAR	30

ocjene

Field	Type	Length
id	INT	11
predmetID	INT	11
studentID	INT	11
ocjena	INT	11

predmeti

Field	Type	Length
id	INT	11
naziv	VARCHAR	60

Dodavanje novog korisnika nad bazom

- Imamo bazu – moramo kreirati korisnika koji će joj pristupati
  - Zapravo, taj korisnik će predstavljati našu web aplikaciju
  - Točnije, web aplikacija će imati sve ovlasti kao i korisnik baze

```
grant all privileges on PPKS.* to ppksuser@localhost identified by
'ppksjakipassword';
```

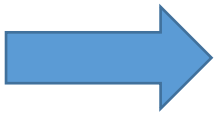
  - Je li uvijek potrebno navesti „all”?
    - Read, write…

SQL naredbe

- Naredbe za manipuliranje podacima u bazi podataka
- *Structured Query Language*
- Vrlo slične za različite baze podataka
- Najčešće:
  - **INSERT** – dodavanje
  - **UPDATE** – ažuriranje
  - **SELECT** – čitanje
  - **DELETE** – brisanje
- Ali i još puno puno drugih…
  - CREATE DATABASE
  - ALTER DATABASE
  - CREATE TABLE
  - ALTER TABLE
  - …

SQL naredbe - dodavanje

id	ime	prezime
1	Marin	Vuković
2	Krešimir	Pripužić



id	ime	prezime
1	Marin	Vuković
2	Krešimir	Pripužić
3	Test-ime	Test-prezime

```
INSERT INTO studenti (ime, prezime) VALUES ('Test-ime', 'Test-prezime');
```

(uočiti: id je “auto-increment”)

SQL naredbe - dodavanje

id	predmetID	studentID	ocjena
1	1	1	2

id	predmetID	studentID	ocjena
1	1	1	2
2	1	2	3

```
INSERT INTO ocjene (predmetID, studentID, ocjena) VALUES (1, 2, 3)
```

Što ako više puta pozovemo istu naredbu?

id	predmetID	studentID	ocjena
1	1	1	2
2	1	2	3
3	1	2	3
4	1	2	3

“Idempotentnost”

SQL naredbe - ažuriranje

Ne dodaje redak nego mijenja postojeći

id	predmetID	studentID	ocjena
5	1	2	3

id	predmetID	studentID	ocjena
5	1	2	5

Uočiti: id=5 - zašto?

```
UPDATE ocjene SET predmetID=1, ocjena=5 WHERE studentID=2
```

Što ako ne postoji takav “redak”?

```
UPDATE ocjene SET predmetID=1, ocjena=5 WHERE studentID=1
```

SQL naredbe – čitanje/odabir

Dohvaća podatke iz baze podataka

```
SELECT * FROM studenti
```

id	ime	prezime
1	Marin	Vuković
2	Krešimir	Pripužić
4	Test-ime	Test-prezime

```
SELECT * FROM studenti WHERE id=1
```

id	ime	prezime
1	Marin	Vuković

SQL naredbe – čitanje/odabir

Različite mogućnosti kod uvjeta (WHERE)

```
SELECT * FROM studenti WHERE ime="Marin"
```

id	ime	prezime
1	Marin	Vuković

```
SELECT * FROM studenti WHERE prezime LIKE '%ić%'
```

id	ime	prezime
1	Marin	Vuković
2	Krešimir	Pripužić

Poredavanje rezultata (sort)

```
SELECT * FROM studenti WHERE prezime LIKE '%ić%' ORDER BY prezime ASC;
```

id	ime	prezime
2	Krešimir	Pripužić
1	Marin	Vuković

SQL naredbe – čitanje/odabir

Sortiranje rezultata

```
SELECT * FROM studenti WHERE prezime LIKE '%ić%' ORDER BY prezime ASC;
```

id	ime	prezime
2	Krešimir	Pripužić
1	Marin	Vuković

Što ako ne postoje rezultati?

```
SELECT * FROM studenti WHERE prezime LIKE '%ne-postoji%' ORDER BY prezime ASC;
```

SQL naredbe – čitanje/odabir

Povezivanje podataka iz više tablica - UNION

```
SELECT prezime FROM studenti WHERE prezime LIKE '%ić%' UNION SELECT
predmetID FROM ocjene WHERE ocjena=5;
```

prezime
Vuković
Pripužić
2

Složeni(ji) upiti

```
SELECT * FROM studenti WHERE id IN (SELECT studentID FROM ocjene WHERE ocjena = 5;
```

id	ime	prezime
2	Krešimir	Pripužić

SQL naredbe – brisanje

Brišemo cijelu tablicu

```
DELETE * FROM ocjene;
```

Brisanje s uvjetom

```
DELETE FROM ocjene WHERE ocjena=5;
```

Primjeri koda - konfiguracija

- Primjer PHP ali više-manje je slično u svim programskim jezicima
- Konfiguracija – postavke za spajanje na bazu podataka

```
$db_host = "localhost"
$db_name = "PPKS"
$username = "ppksuser"
$password = "ppksjakipassword";
```

- Sjetimo se:

```
grant all privileges on PPKS.* to ppksuser@localhost identified by 'ppksjakipassword';
```

  - Ovo smo unosili na MySQL konzoli ili u nekom alatu za pristup bazi
  - Npr. konzola: `mysql -u root -p`

Primjeri koda – spajanje na bazu

- Spajanje na bazu podataka – klasa `dbConnect`:

```php
<?php

class dbConnect{
    private $_mysqli;

    public function __construct(){
        include ('configuration.php');
        $this->mysqli = new mysqli($db_host, $username, $password, $db_name);
        $this->mysqli->select_db($db_name);
        if (mysqli_connect_errno()) {
            printf("Connect failed: %s\n", mysqli_connect_error());
            exit();
        }
    }
```

- I metode za različite manipulacije baze
  - Dodaj, ažuriraj, odaberi….

Primjeri koda - metode

```php
function getStudents(){
    $query = "SELECT * FROM studenti";
    echo $query;
    $result = $this->mysqli->query($query) or die($this->mysqli->error.__LINE__);
    $results;
    $cnt=0;
    while ($row = $result->fetch_row()) {
        $cnt++;
        $results[$cnt] = $row;
    }
    return $results;
}

function getStudents2(){
    $query = "SELECT id, ime, prezime FROM studenti";
    echo $query;
    $result = $this->mysqli->query($query) or die($this->mysqli->error.__LINE__);
    $results;
    $cnt=0;
    while ($row = $result->fetch_row()) {
        $cnt++;
        $results[$cnt]["id"] = $row[0];
        $results[$cnt]["ime"] = $row[1];
        $results[$cnt]["prezime"] = $row[2];
    }
    return $results;
}
```

Primjeri koda - metode

```php
function updateOcjene($studentID, $predmetID, $ocjena){
    $query = "UPDATE ocjene SET predmetID=".$predmetID.", ocjena=".$ocjena." WHERE studentID=".$studentID;
    echo $query;
    $this->mysqli->query($query) or die($this->mysqli->error.__LINE__);
}

function dodavanjeOcjene($studentID, $predmetID, $ocjena){
    $query = "INSERT INTO ocjene (predmetID, studentID, ocjena) VALUES (".$predmetID.", ".$studentID.", ".$ocjena.")";
    echo $query;
    $this->mysqli->query($query) or die($this->mysqli->error.__LINE__);
}
function brisanjeOcjene($studentID, $predmetID){
    $query = "DELETE FROM ocjene WHERE studentID=".$studentID." AND predmetID=".$predmetID;
    echo $query;
    $this->mysqli->query($query) or die($this->mysqli->error.__LINE__);
}
```

Primjeri koda – poziv metoda

- session_start()
  - dajemo uputu poslužitelju weba da stvori sjednicu
  - praćenje sjednice, stanja, varijabli…
- require_once(…)
  - Učitaj datoteku jednom
- $dbConnector
  - Instanca klase dbConnect
  - Pozivamo metode klase

```php
session_start();
require_once('dbconnect.php');
$dbConnector = new dbConnect();
$dbConnector->dodavanjeOcjene(1, 1, 2);
$dbConnector->updateOcjene(1, 1, 5);
$dbConnector->brisanjeOcjene(1, 1);
echo $dbConnector->getStudents();
```

Neki problemi ovakvog pristupa 1

- Vjerovanje korisniku?

```
function getStudents($ime){
    $query = "SELECT * FROM studenti WHERE ime='".$ime."'";
```

- SQL injection!

```
SELECT * FROM studenti WHERE ime='Marin' OR '1'='1';
```

  - Nikada ne smijemo vjerovati korisniku i njegov unos izravno slati na bazu
  - Koristiti barem:

```
private function sanitize($input) {
    if (get_magic_quotes_gpc()) {
        $input = stripslashes($input);
    }
    $output = mysqli_real_escape_string($this->mysqli, $input);
    return $output;
}
```

Neki problemi ovakvog pristupa 2

- Danas se SQL upiti iz koda ne izvode ovako!
  - Iako su same SQL naredbe iste
- Koriste se „pripremljene izjave” (*prepared statements*)

```
$stmt = $pdo->prepare("SELECT * FROM studenti WHERE id=:id");
$stmt->execute(['id' => $id]);
$student = $stmt->fetch();
```

- Još bolje:
  - Perzistencija objekata u bazu
  - Npr. objekt „student”
    - Parametri ime, prezime
  - Čitav objekt se pohranjuje u i dohvaća iz baze