

TEHNIČKO VELEUČILIŠTE U ZAGREBU

**Politehnički specijalistički diplomske stručne studije
specijalizacija Informatika**

Napredne tehnike programiranja web servisa (.open-source)

Baze podataka

- **Baza podataka** je skup medusobno povezanih podataka, pohranjenih u vanjskoj memoriji računala
- Podaci su istovremeno dostupni raznim korisnicima i aplikacijskim programima
- U bazi podataka moguće je pohranjivati, mjenjati, brisati i čitati podatke

Baze podataka

- Korisnik (administrator baze podataka) može pristupiti preko softvera (programa) za rad sa bazama podataka
- **Sustav za upravljanje bazom podataka** (engl. Data Base Management System - DBMS) je poslužitelj (server) baze podataka
- Podaci u bazi su logički organizirani u skladu s nekim modelom podataka

Baze podataka

- Model podataka je skup pravila koja određuju kako može izgledati logička struktura baze
- DBMS podržava slijedeće modele:
 - **Relacijski model.** Zasnovan na matematičkom pojmu relacije.
 - **Mrežni model.** Baza je prikazana usmjerenim grafom
 - **Hijerarhijski model.** Baza je prikazana jednim stablom ili skupom stabala.
 - **Objektni model.** Baza je skup trajno pohranjenih objekata koji se sastoje od svojih internih podataka i “metoda” (operacija) za rukovanje s tim podacima.

Baze podataka

- Hijerarhijski i mrežni model bili su u upotrebi u 60-im i 70-im godinama 20. stoljeća
- Od 80-tih godina do danas koriste se relacijski modeli baza podataka, a sve više i objektni model baza podataka
- Baze podataka predstavljaju višu razinu rada s podacima u odnosu na klasične programske jezike

Ciljevi baza podataka

- Ciljevi koji se žele postići bazama podataka:
 - **Fizička nezavisnost podataka.** Razdvaja se logička definicija baze od njene stvarne fizičke građe. Ako se fizička građa promijeni, to neće zahtijevati promjene u postojećim aplikacijama (npr. podaci se prepišu u druge datoteke na drugim diskovima)
 - **Logička nezavisnost podataka.** Razdvaja se globalna logička definicija cijele baze podataka od lokalne logičke definicije za jednu aplikaciju. Ako se logička definicija promijeni, to neće zahtijevati promjene u postojećim aplikacijama (uveđe se novi zapis ili veza)

Ciljevi baza podataka

- Ciljevi koji se žele postići bazama podataka:
 - **Fleksibilnost pristupa podacima.** U starijim mrežnim i hijerarhijskim bazama, staze pristupanja podacima bile su unaprijed definirane. Korisnik je mogao pristupati podacima redoslijedom koji je bio predviđen kod projektiranja i implementiranja baze
 - **Istovremeni pristup do podataka.** Baza mora omogućiti da veći broj korisnika istovremeno koristi iste podatke. Pritom ti korisnici ne smiju ometati jedan drugoga.

Ciljevi baza podataka

- Ciljevi koji se žele postići bazama podataka:
 - **Čuvanje integriteta.** Nastoji se sačuvati korektnost i konzistencija podataka.
 - **Mogućnost oporavka nakon kvara.** Mora postojati pouzdana zaštita baze u slučaju kvara hardvera ili grešaka u radu sistemskog softvera.
 - **Zaštita od neovlaštenog korištenja.** Mogućnost da se korisnicima ograniče prava korištenja baze.

Ciljevi baza podataka

- Ciljevi koji se žele postići bazama podataka:
 - **Zadovoljavajuća brzina pristupa.** Operacije s podacima moraju se odvijati dovoljno brzo, u skladu s potrebama određene aplikacije.
 - **Mogućnost podešavanja i kontrole.** Velika baza zahtijeva stalnu brigu: praćenje performansi, mijenjanje parametara u fizičkoj građi, rutinsko pohranjivanje rezervnih kopija podataka, reguliranje ovlaštenja korisnika. Odgovorna osoba zove se *administrator baze podataka*.

Arhitektura baze podataka

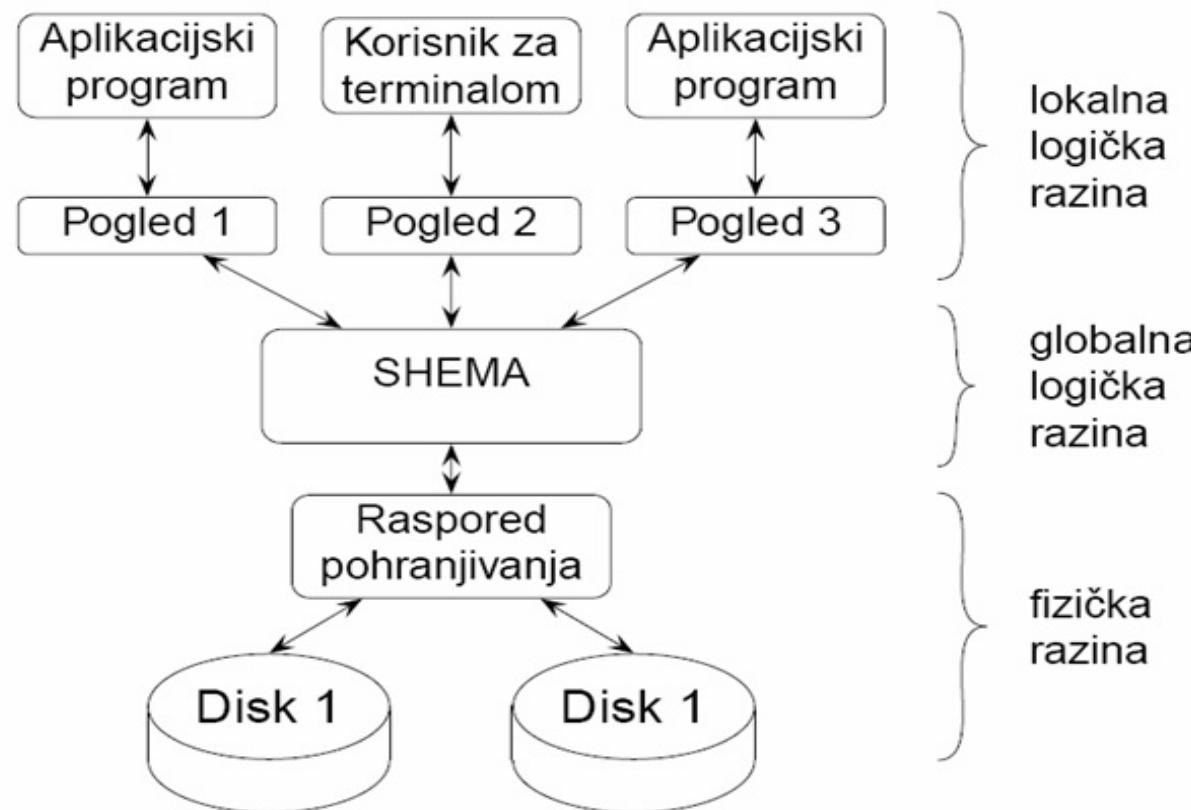
- Arhitektura baze podataka sastoji se od tri “sloja” i sučelja među slojevima:
 - **Fizička razina** odnosi se na fizički prikaz i raspored podataka na jedinicama vanjske memorije. To je aspekt kojeg vide samo sistemski programeri
 - **Globalna logička razina** odnosi se na logičku strukturu cijele baze. To je aspekt kojeg vidi projektant baze odnosno njen administrator.

Arhitektura baze podataka

- Arhitektura baze podataka sastoji se od tri “sloja” i sučelja među slojevima:
 - **Lokalna logička** razina odnosi se na logičku predodžbu o dijelu baze kojeg koristi pojedina aplikacija. To je aspekt kojeg vidi korisnik ili aplikacijski programer.

Arhitektura baze podataka

Arhitektura BP



Jezici za rad s bazama podataka

- Komunikacija korisnika i DBMS-a odvija se pomoću:
 - **Jezik za opis podataka** (Data Description Language - DDL). Služi projektantu baze ili administratoru u svrhu zapisivanja sheme ili pogleda. Tim jezikom definiramo podatke i veze među podacima
 - **Jezik za manipuliranje podacima** (Data Manipulation Language - DML). Služi programeru za uspostavljanje veze između aplikacijskog programa i baze
 - **Jezik za postavljanje upita** (Query Language - QL). Služi korisniku za interaktivno pretraživanje baze

Modeliranje baze podataka

- **Entitet** je nešto o čemu želimo spremati podatke. Entitet može biti objekt ili biće (kuća, student, auto), odnosno događaj ili pojava (nogometna utakmica, praznik, servisiranje auta)
- Entitet je opisan **atributima** (atributi kuće su: adresa, broj katova, boja fasade).
- **Ključ** je atribut, ili skup atributa, čije vrijednosti jednoznačno odreduju primjerak entiteta zadano tipa.
 - Ne mogu postojati dva različita primjerka entiteta istog tipa s istim vrijednostima kandidata za ključ
 - Tip entiteta *AUTO*, kandidat za ključ je atribut *REG BROJ*
 - Tip entiteta *STUDENT*, kandidat za ključ je atribut *JMBAG*

Modeliranje baze podataka

- Entitet odnosno objekt "prema srodstvu" može biti:
 - **jaki** - postoji neovisno od drugih objekata
 - **slabi** - egzistencijalno i/ili identifikacijski zavisi o jakom objektu
- **Jaki entitet** je primjerice entitet ŠKOLA, jer može postojati sam za sebe.
- **Slabi entitet** je primjerice RAZRED jer ovisi o entitetu ŠKOLA i ako se iz baze podataka obrišu podaci o školi brišu se podaci i o razredu te škole.

Modeliranje baze podataka

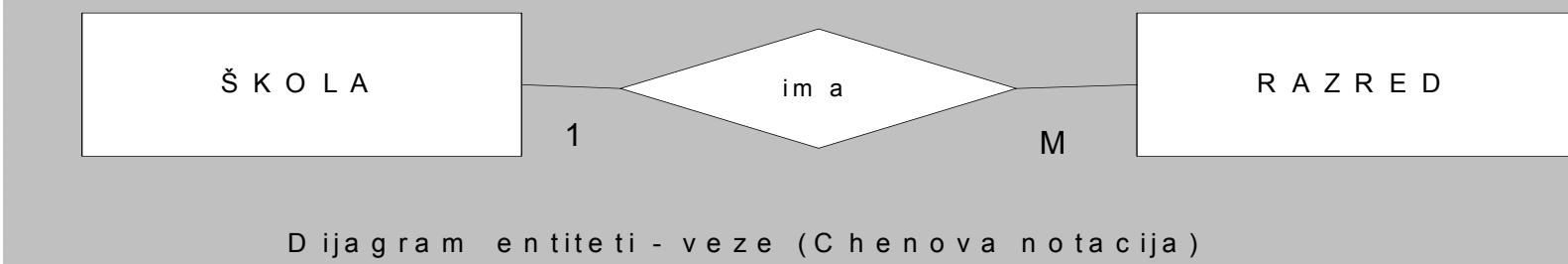
- Entiteti su opisani svojstvima (atributima). Atributi mogu biti:
- **identifikacijski** - jednoznačno i nedvosmisleno identificira jednu pojavu entiteta među svim ostalima, trajno je pridružen entitetu i često se zove ključni atribut ili šifra
- **opisni** - opisuje kvalitativna ili kvantitativna svojstva entiteta, mijenjaju se s vremenom jer se mijenjaju i stanja i svojstva entiteta
- **izvedeni** - izvode se logičkim ili aritmetičkim operacijama iz definiranih vrijednosti nekih drugih atributa, uključene su formule, algoritmi i logički izrazi

Modeliranje baze podataka

- Vezom se povezuju entiteti. Veza se imenuje a naziv veze opisuje ulogu entiteta u vezi.
- Naziv veze je najčešće glagol ili imenica.



D i a g r a m e n t i t e t i - v e z e (M a r t i n o v a n o t a c i j a)



D i a g r a m e n t i t e t i - v e z e (C h e n o v a n o t a c i j a)

Modeliranje baze podataka

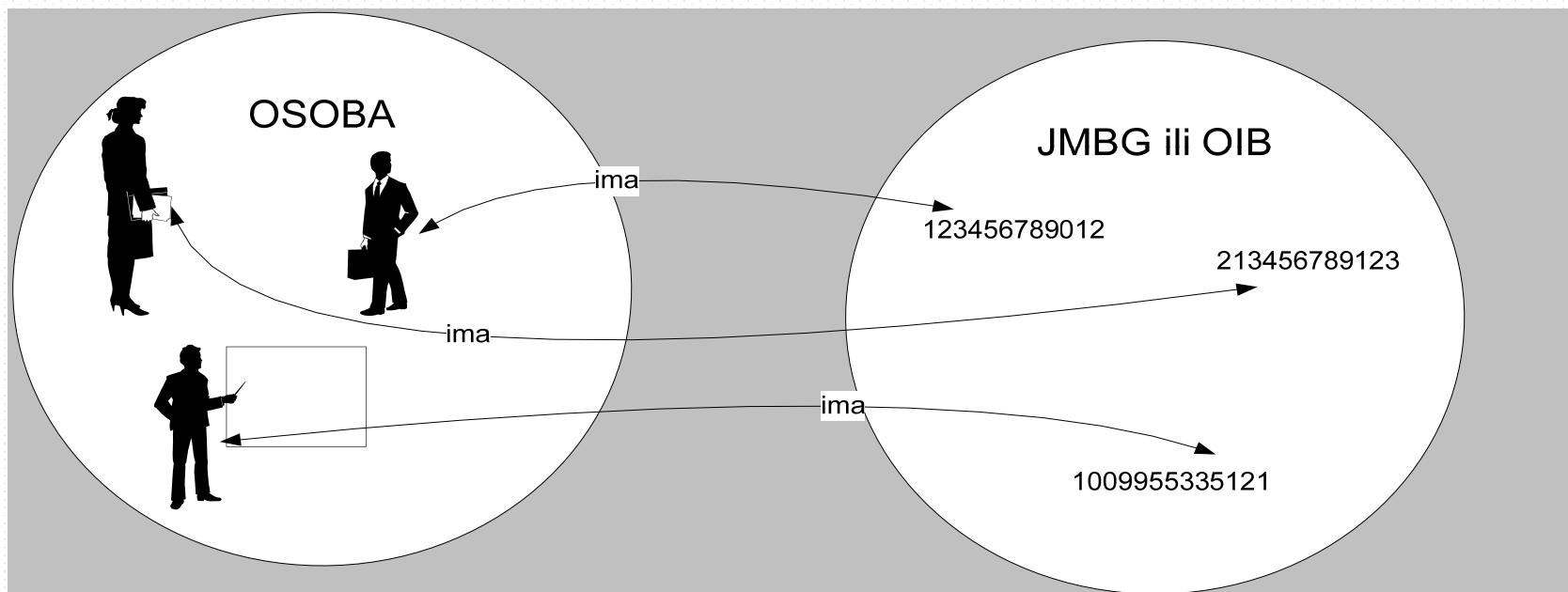
- Prema redu veze (ili stupnju veze pri čemu je stupanj veze jednak broju entiteta koji sudjeluju u vezi) razlikuju se:
 - **unarna**, gdje se radi o vezi između dvije pojave istog tipa entiteta (često se koristi naziv rekurzivna veza),
 - **binarna**, gdje se radi o vezi između dva entiteta, i
 - **ternarna**, gdje se radi o tri međusobno povezana entiteta.

Modeliranje baze podataka

- Prema tipu povezanosti (pridruživanja) određuje se kardinalnost veza koja može biti:
 - jednostavno ili potpuno pridruživanje ($1 : 1$),
 - uvjetno pridruživanje ($1 : M$),
 - složeno ili višeznačno pridruživanje ($M : N$).

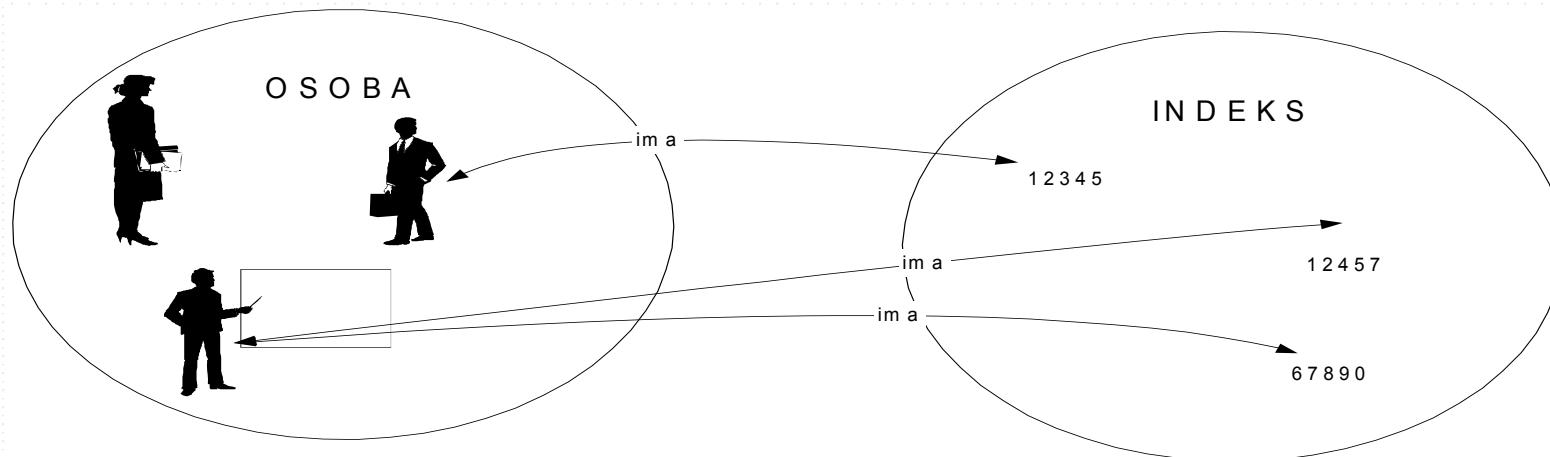
Modeliranje baze podataka

- **Jednostavno ili potpuno pridruživanje (1 : 1)**
- Svaki član iz skupa pojava jednog entiteta povezan je s jednim i samo jednim članom iz skupa pojava drugog entiteta.



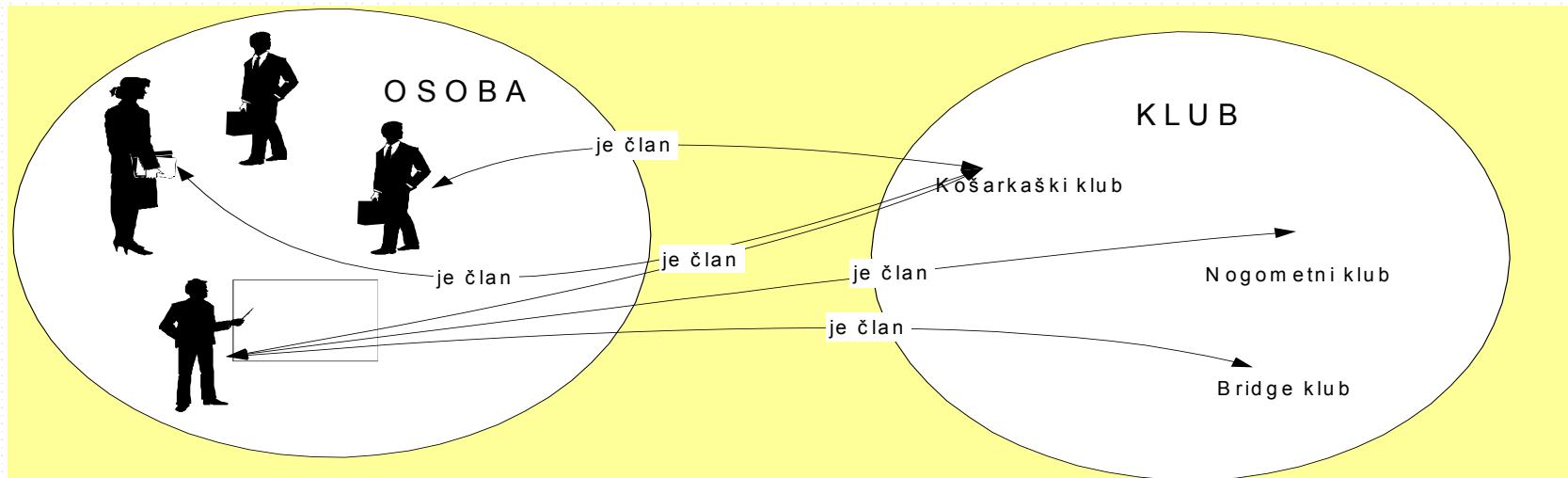
Modeliranje baze podataka

- **Uvjetno pridruživanje (1 : M)**
- Svaki član iz skupa pojava jednog entiteta povezan s jednim ili niti jednim ili s više članova iz skupa pojava drugog entiteta, pri čemu je svaki član iz skupa pojava drugog entiteta povezan samo s jednim članom iz skupa pojava prvog entiteta.



Modeliranje baze podataka

- **Složeno ili višeznačno pridruživanje (M : N)**
- Svaki član iz skupa pojava jednog entiteta povezan je s jednim, niti jednim ili s više članova iz skupa pojava drugog entiteta (ne postoje ograničenja u povezanosti članova skupa pojava oba entiteta).

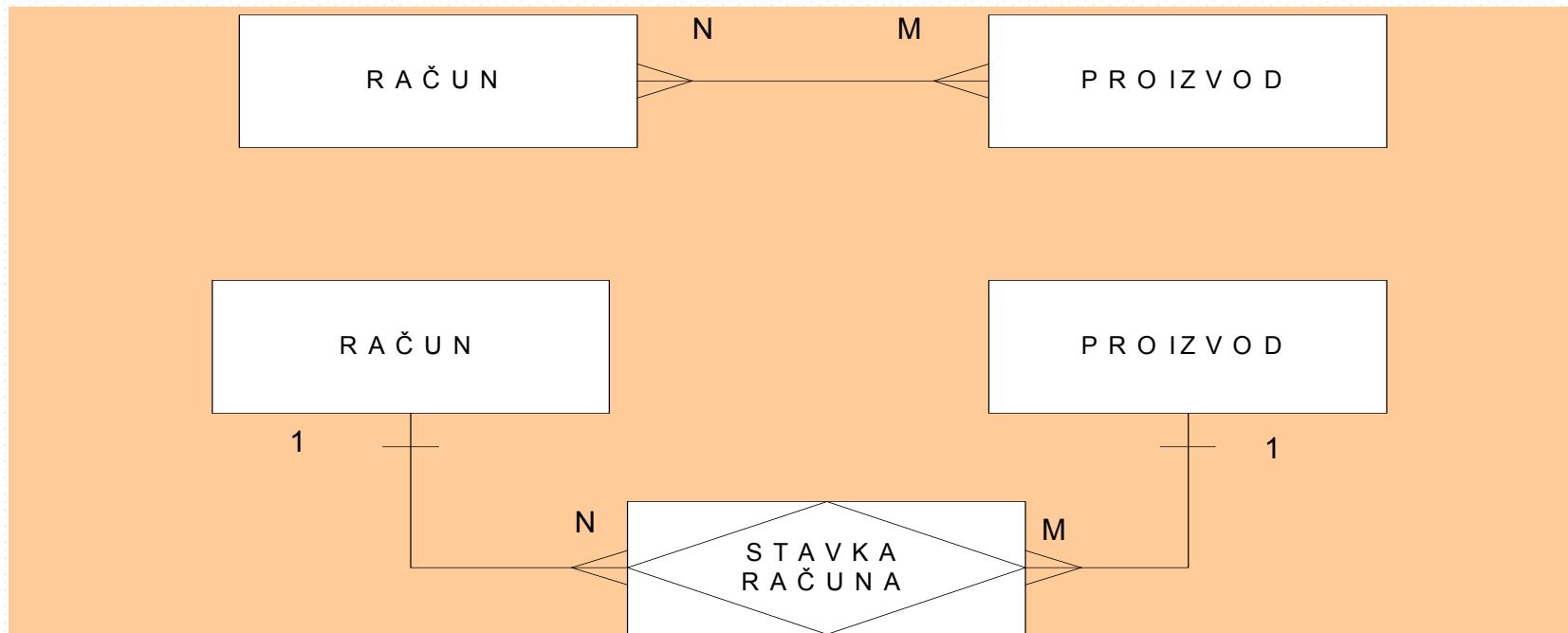


Modeliranje baze podataka

- Kardinalnost veze može mijenjati ovisno o tomu koji se entitet promatra. Tako je moguće da:
 - 1 osoba posjeduje 0, 1 ili više automobila
 - 1 automobil pripada 1 osobi
- Trgovac jednom dobavljaču može poslati više narudžbi, ali se jedna narudžba odnosi samo na jednog dobavljača. Tip pridruživanja je 1:M.

Modeliranje baze podataka

- Samo se veze tipa pridruživanja 1:1 i 1:M mogu implementirati u relacijskoj bazi podataka. Stoga se svaka veza tipa M:N treba pretvoriti u dvije veze tipa 1:M i N:1



Uvod u relacijski model

- Svaka baza podataka u sebi ima jednu ili više tablica sa podacima npr. tablica u kojoj su pohranjeni podaci o zaposlenicima TVZ-a.
- U svaku tablicu se spremaju podaci o jedinki npr. zaposleniku i to se zove zapis (eng. record)
- Tablica se sastoji od zapisa koji pohranjuje podatke zaposlenika
- Tablica ima stupce koji se nazivaju atributi

Uvod u relacijski model

- Normalizacija je pristup izradi baze podataka čiji će dizajn biti optimiziran za izvršavanje upita nad bazom podataka.
- Smanjiti mogućnost dobivanja anomalija u podacima baze.
- Višestruko zapisivanje istog podatka otežava mijenjanje sadržaja baze podataka.
- Normalizacija je pristup izradi baze podataka čiji će dizajn biti optimiziran za izvršavanje generalnih upita na bazu.

Uvod u relacijski model

- Tablica sa podacima o zaposlenicima

localhost > work26293 > tvz_employee

	Polje	Vrsta	Uspoređivanje	Atributi	Null	Zadano	Dodatno	Aktivnost
<input type="checkbox"/>	<u>employee_id</u>	int(11)			Ne	None	AUTO_INCREMENT	       
<input type="checkbox"/>	<u>employee_firstname</u>	varchar(255)	utf8_general_ci		Ne			       
<input type="checkbox"/>	<u>employee_lastname</u>	varchar(255)	utf8_general_ci		Ne			       
<input type="checkbox"/>	<u>employee_department</u>	int(11)			Ne			       

Označi sve / Ukloni sve označe S odabirom:       

 Prikaz ispisa  Prikaz relacija  Predloži strukturu tablice  Track table
 Dodaj 1 polja Pri završetku tablice Pri početku tablice Poslije employee_id 

Indeksi: 

Aktivnost	Naziv ključa	Vrsta	Jedinstveno	Pakirano	Polje	Najvažnije	Uspoređivanje	Null	Komentar
 	PRIMARY	BTREE	Da	Ne	employee_id	0	A		

Uvod u relacijski model

- Podaci u tablici tvz_employee

← T →	employee_id	employee_firstname	employee_lastname	employee_department
<input type="checkbox"/>  	1	Pero	Perić	smjer informatika
<input type="checkbox"/>  	2	Mirko	Mirić	smjer informatika
<input type="checkbox"/>  	3	Stipe	Stipić	smjer računarstvo
<input type="checkbox"/>  	4	Matko	Matić	smjer elektrotehnika

Podaci u stupcu „employee_department” moramo staviti u odnos sa drugom tablicom. To se naziva **relacija podataka**

- Da bi mogli staviti zapise u odnos moram uvesti pojам ključa

Uvod u relacijski model

- Podaci u tablici tvz_employee

← T →	employee_id	employee_firstname	employee_lastname	employee_department
<input type="checkbox"/>  	1	Pero	Perić	smjer informatika
<input type="checkbox"/>  	2	Mirko	Mirić	smjer informatika
<input type="checkbox"/>  	3	Stipe	Stipić	smjer računarstvo
<input type="checkbox"/>  	4	Matko	Matić	smjer elektrotehnika

Podaci u stupcu „employee_department” moramo staviti u odnos sa drugom tablicom. To se naziva **relacija podataka**

- Da bi mogli staviti zapise u odnos moram uvesti pojам ključa

Uvod u relacijski model

- Ključ koji se može koristiti za jedinstveno identificiranje reda u tablici naziva se jedinstvenim ključem.
- Kada se ključ sastoji od podataka koji imaju vanjsko, realno značenje (OIB osobe, JMBAG studenta, ISBN knjige) tada se taj ključ naziva "**prirodni**" ključ.
- Ako prirodan ključ nije prikidan, može se dodijeliti **arbitraran ili generiran** ključ.

Uvod u relacijski model

- Ako uvedemo još jednu tablicu sa podacima
- Tablica sa podacima o odjelu

localhost ► work26293 ► tvz_department

Pretraživanje	Struktura	SQL	Traži	Tracking	Umetni	Izvoz	Uvoz	Operacije	Isprazni	Ispusti
	Polje	Vrsta	Uspoređivanje	Atributi	Null	Zadano	Dodatno		Aktivnost	
<input type="checkbox"/>	<u>department_id</u>	int(11)			Ne	None	AUTO_INCREMENT	      	     	
<input type="checkbox"/>	<u>depatment_name</u>	varchar(255)	utf8_general_ci		Ne			     	     	

Označi sve / Ukloni sve označe S odabirom:      

Prikaz ispisa Prikaz relacija Predloži strukturu tablice Track table

Dodaj 1 polja Pri završetku tablice Pri početku tablice Poslije department_id Kreni

Indeksi: 

Aktivnost	Naziv ključa	Vrsta	Jedinstveno	Pakirano	Polje	Najvažnije	Uspoređivanje	Null	Komentar
 	PRIMARY	BTREE	Da	Ne	department_id	0	A		

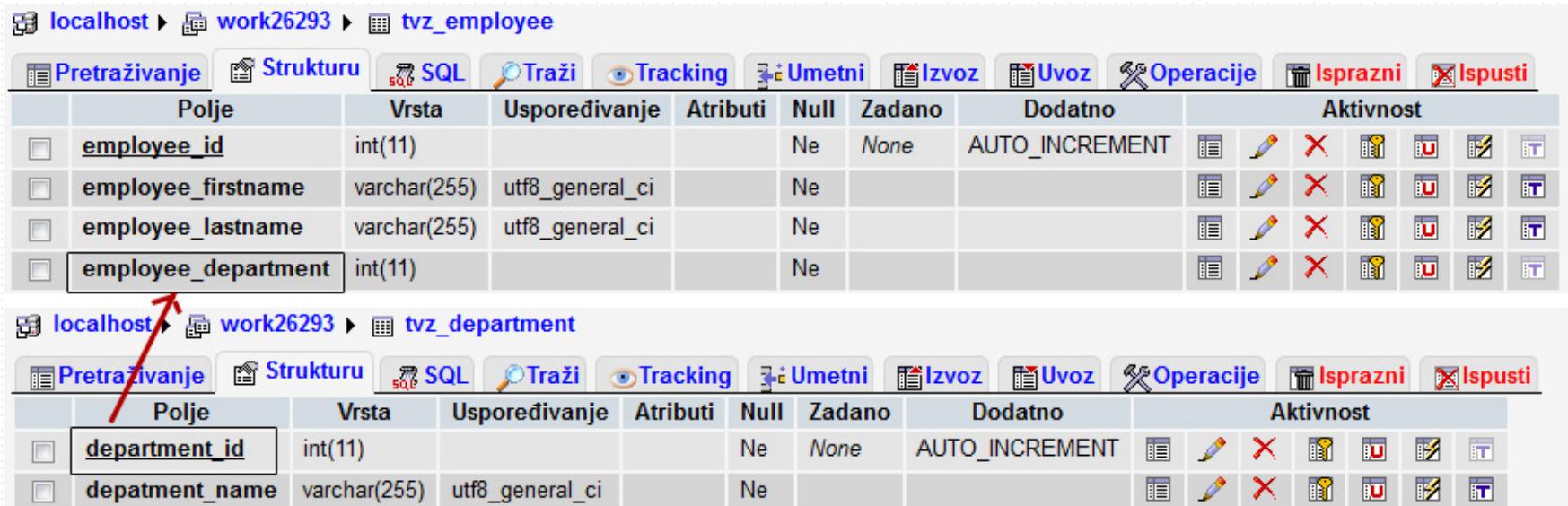
Uvod u relacijski model

- Prikaz sadržaja tablice tvz_department

←→	department_id	depatment_name
<input type="checkbox"/>  	1	smjer informatika
<input type="checkbox"/>  	2	smjer računarstvo
<input type="checkbox"/>  	3	smjer elektrotehnika

Uvod u relacijski model

- Primjer relacije tablica tvz_department i tvz_employee



The screenshot shows two database structures in MySQL Workbench:

- tvz_employee:** This table has four columns: employee_id (int(11), primary key, auto-increment), employee_firstname (varchar(255), utf8_general_ci), employee_lastname (varchar(255), utf8_general_ci), and employee_department (int(11)).
- tvz_department:** This table has two columns: department_id (int(11), primary key, auto-increment) and department_name (varchar(255), utf8_general_ci).

A red arrow points from the 'employee_department' column in the tvz_employee table to the 'department_id' column in the tvz_department table, indicating a foreign key relationship.

Uvod u relacijski model

- U primjeru 1 bilo bi gotovo nemoguće proširiti tablicu sa podacima o adresi odjela ili nekom drugom podatku bez da dođe do pogreške kod unosa podataka
- Za svakog djelatnika adresa odjela morala bi se unositi ručno i pojedinačno

localhost > work26293 > tvz_employee



The screenshot shows the 'tvz_employee' table structure in MySQL Workbench. The table has five columns: employee_id, employee_firstname, employee_lastname, employee_department, and employee_address. The 'employee_id' column is defined as int(11) with a primary key constraint (indicated by a red 'P' icon). The other four columns are defined as varchar(255) with utf8_general_ci collation. The 'employee_firstname' and 'employee_lastname' columns have a red 'X' icon in the 'Atributi' column, indicating they are not nullable. The 'employee_department' and 'employee_address' columns have a red 'X' icon in the 'Null' column, indicating they cannot be null. The 'employee_id' column has a red 'X' icon in the 'Zadano' column, indicating it is not defined as a default value. The 'employee_id' column also has a red 'X' icon in the 'Operacije' column, indicating it is not part of any foreign key relationships. The 'employee_id' column has a red 'X' icon in the 'Isprazni' column, indicating it cannot be deleted. The 'employee_id' column has a red 'X' icon in the 'Ispusti' column, indicating it cannot be truncated.

	Polje	Vrsta	Uspoređivanje	Atributi	Null	Zadano	Dodatno	Aktivnost
<input type="checkbox"/>	<u>employee_id</u>	int(11)			Ne	None	AUTO_INCREMENT	
<input type="checkbox"/>	<u>employee_firstname</u>	varchar(255)	utf8_general_ci		Ne			
<input type="checkbox"/>	<u>employee_lastname</u>	varchar(255)	utf8_general_ci		Ne			
<input type="checkbox"/>	<u>employee_department</u>	varchar(255)	utf8_general_ci		Ne			
<input type="checkbox"/>	<u>employee_address</u>	varchar(255)	utf8_general_ci		Ne			

Uvod u relacijski model

- Prikaz ručnog unosa podataka za tablicu tvz_employee

	← T →	employee_id	employee_firstname	employee_lastname	employee_department	employee_address
<input type="checkbox"/>	 	1	Pero	Perić	smjer informatika	Vrbik 8a
<input type="checkbox"/>	 	2	Mirko	Mirić	smjer informatika	Vrbik 8a
<input type="checkbox"/>	 	3	Stipe	Stipić	smjer računarstvo	Vrbik 8a
<input type="checkbox"/>	 	4	Matko	Matić	smjer elektrotehnika	Konavoska 2

Uvod u relacijski model

- Ako smo napravili relaciju, proširenje sa podacima nije problem.
- Svi podaci vežu se i nema ponavljanja unosa podataka.
- Podatak se unosi u **samo jednoj** tablici
`tvz_department`



The screenshot shows the 'Struktura' (Structure) tab of the phpMyAdmin interface for the 'tvz_department' table. The table has three columns: 'department_id' (int(11)), 'depatment_name' (varchar(255)), and 'department_address' (varchar(255)). The 'department_id' column is defined as AUTO_INCREMENT. The table is currently empty, indicated by the red 'X' icons in the 'Operacije' (Operations) column.

	Polje	Vrsta	Uspoređivanje	Atributi	Null	Zadano	Dodatno		Aktivnost
	<code>department_id</code>	int(11)			Ne	None	AUTO_INCREMENT	 	   
	<code>depatment_name</code>	varchar(255)	utf8_general_ci		Ne			 	   
	<code>department_address</code>	varchar(255)	utf8_general_ci		Ne			 	   

Uvod u relacijski model

- Prikaz ručnog unosa podataka za tablicu tvz_department

← T →	department_id	depatment_name	department_address
<input type="checkbox"/>  	1	smjer informatika	Vrbik 8a
<input type="checkbox"/>  	2	smjer računarstvo	Vrbik 8a
<input type="checkbox"/>  	3	smjer elektrotehnika	Konavoska 2

Uvod u relacijski model

- Prikazana baza podatka koja funkcioniра na opisani način je **relacijska baza podataka**
- Imena tablice, same baze podataka i atributa se pišu malim slovima engleske abecede bez znakova razmaka
- Svakom **atributu** tj. **polju** treba odrediti tip podatka koji će se u njega spremati
- MySQL tipovi podatka se dijele na:
 - a. Numeričke tipove
 - b. Tekstualni tipovi ili string
 - c. vremenske

Uvod u relacijski model

Numerički tipovi

- Mogu biti sa ili bez predznaka. Ako su sa predznakom mogu imati negativne vrijednosti.
 - **BIT(x)** – može biti između 1 i 64, ako se ne navede onda je 1.
 - **TINYINT** – od -128 do 127 ili bez predznaka od 0 do 255
 - **BOOL, BOOLEAN**
 - **SMALLINT** – od -32.768 do 32.767 ili bez predznaka od 0 do 65.535
 - **MEDIUMINT** – od -8.388.608 do 8.388.607 ili bez predznaka od 0 do 16.777.215
 - **INT** – od -2.147.483.648 do 2.147.483.647 ili bez predznaka od 0 do 4.294.967.295
 - **BIGINT** – od -9.223.372.036.854.775.808 do 9.223.372.036.854.775.807 ili bez predznaka od 0 od 18.446.744.073.709.551.615

Uvod u relacijski model

Numerički tipovi

- Mogu biti sa ili bez predznaka. Ako su sa predznakom mogu imati negativne vrijednosti.
 - **FLOAT(x, y)** – se koristi za pohranu realnih brojeva sa manjom preciznošću. x je ukupan broj znakova, a y broj decimalnih znakova
 - **DOUBLE(x, y)** – isto kao i FLOAT ali sa duplom preciznošću
 - **FLOAT(x)** – gdje je x preciznost u broj bitovima
 - **DECIMAL(x, y)** – slično double-u

Uvod u relacijski model

Tekstualni tipovi

- Tekstualna polja se dijele na fiksne i varijabilne. Fiksna polja će zauzimati uvijek istu količinu memorije za svoju pohranu. Varijabilna polja će zauzimati onoliku količinu memorije koliko je u njima pohranjeno podataka
 - **CHAR(x)** – Tekst od x znakova. Gdje x može biti između 1 i 255. **Char je fiksan tip polja.**
 - **VARCHAR(x)** - Tekst od maksimalno x znakova. Gdje x može biti između 1 i 255, ako se ne navede iznosi 1. **Varchar je varijabilan tip polja.**
 - **BINARY(x)** – Za pohranu binarnih vrijednosti. X je broj duljina polja u bytovima, maksimalno 255. **Fiksan tip polja.**
 - **VARBINARY(x)** – Ekvivalent varchar tipu ali za pohranu binarnih vrijednosti

Uvod u relacijski model

Tekstualni tipovi

- Tekstualna polja se dijele na fiksne i varijabilne. Fiksna polja će zauzimati uvijek istu količinu memorije za svoju pohranu.
Varijabilna polja će zauzimati onoliku količinu memorije koliko je u njima pohranjeno podataka
 - **TINYBLOB** – polje sa maksimalno 255 bytova. **Varijabilni tip.**
 - **TINYTEXT** – polje sa maksimalno 255 znakova. **Varijabilni tip.**
 - **BLOB(x)** – polje od maksimalno 65535 bajtova. **Varijabilni tip.**
 - **TEXT(x)** – polje sa maksimalno 65535 znakova. **Varijabilni tip.**
 - **MEDIUMBLOB** – maksimalno 16777215 bytova. **Varijabilni tip.**
 - **MEDIUMTEXT** - maksimalno 16777215 znakova. **Varijabilni tip.**
 - **LONGBLOB** – maksimalno 4G bytova. **Varijabilni tip.**

Uvod u relacijski model

Tekstualni tipovi

- Tekstualna polja se dijele na fiksne i varijabilne. Fiksna polja će zauzimati uvijek istu količinu memorije za svoju pohranu.
Varijabilna polja će zauzimati onoliku količinu memorije koliko je u njima pohranjeno podataka
 - **LONGTEXT** - maksimalno 4G znakova. Varijabilni tip.
 - **ENUM('1', '2', '3',..)** - polje koje može sadržavati jednu od predefiniranih vrijednosti ('vrijednost 1', 'vrijednost 2', 'vrijednost 3' itd.)
 - **SET ('1', '2', '3',..)** - polje koje može imati jednu ili više predefiniranih vrijednosti.

Uvod u relacijski model

Vremenski tipovi

- Postoje sljedeći tipovi vremenskih polja
 - **DATE** – datum u obliku 'YYYY-MM-DD'. Opseg vrijednosti ide od '1000-01-01' do '9999-12-31'.
 - **DATETIME** – datum i vrijeme u obliku 'YYYY-MM-DD HH:MM:SS'. Opseg vrijednosti ide od '1000-01-01 00:00:00' do '9999-12-31 23:59:59'.
 - **TIMESTAMP** – datum i vrijeme u obliku 'YYYY-MM-DD HH:MM:SS'. Interno se pohranjuje kao broj sekundi protekao od Unix epohe (1.1.1970.). Opseg vrijednosti je od '1970-01-01 00:00:01' UTC (Coordinated Universal Time:
http://en.wikipedia.org/wiki/Coordinated_Universal_Time) do '2038-01-19 03:14:07' UTC

Uvod u relacijski model

Vremenski tipovi

- Postoje sljedeći tipovi vremenskih polja
 - **TIME** – Vrijeme u formatu 'HH:MM:SS'. Opseg vrijednosti je od '-838:59:59' do '838:59:59'.
 - **YEAR(2|4)** – Godina u obliku dvije ili 4 znamenke. Ako se ne navede pretpostavlja se 4.

Uvod u relacijski model

Naredbe SQL se dijele po svrsi postojanja na:

- Definicija podataka:
 1. CREATE
 2. DROP
 3. ALTER
- Manipulacija i pretraga nad podacima:
 1. SELECT
 2. INSERT
 3. UPDATE
 4. DELETE
 5. TRUNCATE

Uvod u relacijski model

Naredbe SQL se dijele po svrsi postojanja na:

- Kontrola prava korisnika baze podataka:
 1. GRANT
 2. REVOKE
- Kontrola transakcija:
 1. START TRANSACTION
 2. COMMIT
 3. ROLLBACK

Uvod u relacijski model

Naredbe SQL se dijele po svrsi postojanja na:

- Pomoćne naredbe:

1. USE
2. SET
3. DESCRIBE
4. EXPLAIN
5. SHOW

Uvod u relacijski model

CREATE stvara tablicu sa poljima i pripadajućim atributima

```
CREATE TABLE users (
    id int(11) UNSIGNED NOT NULL AUTO_INCREMENT,
    name varchar(50),
    lastname varchar(50),
    username varchar(100) NOT NULL,
    password varchar(100) NOT NULL,
    about text,
    created tinyint(1) UNSIGNED NOT NULL,
    modified tinyint(1) UNSIGNED NOT NULL,
    PRIMARY KEY (id),
    UNIQUE KEY username_key (username)
) ENGINE=InnoDB DEFAULT CHARSET=latin2 COLLATE=latin2_croatian_ci;
```

Uvod u relacijski model

Izlaz

localhost ► work26293 ► tvz_users "InnoDB free: 55296 kB"

Pretraživanje		Strukturu		SQL		Traži		Tracking		Umetni		Izvoz		Uvoz		Operacije		Isprazni		Ispusti		
Polje	Vrsta	Uspoređivanje		Atributi	Null	Zadano			Dodatno								Aktivnost					
<input type="checkbox"/> id	int(11)			UNSIGNED	Ne	None			AUTO_INCREMENT				<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
<input type="checkbox"/> name	varchar(50)	latin2_croatian_ci			Da	NULL							<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
<input type="checkbox"/> lastname	varchar(50)	latin2_croatian_ci			Da	NULL							<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
<input type="checkbox"/> username	varchar(100)	latin2_croatian_ci				Ne							<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
<input type="checkbox"/> password	varchar(100)	latin2_croatian_ci				Ne							<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
<input type="checkbox"/> about	text	latin2_croatian_ci				Da	NULL					<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/> created	tinyint(1)			UNSIGNED		Ne						<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
<input type="checkbox"/> modified	tinyint(1)			UNSIGNED	Ne							<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Označi sve / Ukloni sve oznake S odabirom:

Prikaz ispisa Prikaz relacija Predloži strukturu tablice Track table

Dodaj 1 polja Pri završetku tablice Pri početku tablice Poslije id Kreni

Indeksi:

Aktivnost	Naziv ključa	Vrsta	Jedinstveno	Pakirano	Polje	Najvažnije	Uspoređivanje	Null	Komentar
<input type="checkbox"/>	PRIMARY	BTREE	Da	Ne	id	0	A		
<input type="checkbox"/>	username_key	BTREE	Da	Ne	username	0	A		

Uvod u relacijski model

Drop je naredba obavlja suprotnu zadaću od naredbe create, odnosno briše bazu, tablicu, korisnika sa svim poljima i sadržajem

- **DROP DATABASE** – briše bazu podataka i sve tablice u njoj.
- **DROP TABLE** – briše tablicu
- **DROP USER** – briše korisnika baze podataka

Uvod u relacijski model

Alter SQL naredba služi za izmjenu već stvorenih objekata kao što su baza podataka ili tablica unutar baze.

ALTER DATABASE

- može promijeniti postavljeni character set ili poredak abecede baze podataka
- može dodati, oduzeti ili promijeniti bilo koji dio tablice. Npr. dodati polje, maknuti polje, dodati ključ itd.

Uvod u relacijski model

Select je sql naredba za dohvat tj. pretraživanje podataka iz baze podataka

Primjer:

```
SELECT name FROM tvz_users WHERE name='Pero'  
ORDER BY lastname ASC
```

Izlaz: Pero

Uvod u relacijski model

FROM sadrži popis tablica iz kojih se podaci uzimaju

Npr. "SELECT firstname, lastname FROM tvz_users;" će dohvatiti imena i prezimena svih osoba iz tablice po imenu "tvz_users".

WHERE sadrži uvjete po kojima se podaci pretražuju.

Npr. "SELECT firstname, lastname FROM tvz_users WHERE city='Zagreb';" dohvatiti će imena i prezimena svih osoba koje žive u Zagrebu, a nalaze se u tablici tvz_osobe.

Uvod u relacijski model

PHP:

```
<?php
$con=mysqli_connect("localhost", "root", "123", "my_db");
$query = "SELECT * FROM tvz_users";
$result = mysqli_query($con,$query);
while($row = mysqli_fetch_array($result)) {
    echo $row['firstname'] . " " . $row['lastname'];
    echo "<br />";
}
mysqli_close($con);
?>
```

IZLAZ: Svi podaci iz baze koji se nalaze u tablici *tvz_users*

Uvod u relacijski model

PHP:

```
<?php
$con=mysqli_connect("localhost", "root", "123", "my_db");
$query = "SELECT firstname, lastname FROM tablica_users
WHERE city='Zagreb'";
$result = mysqli_query($con,$query);
while($row = mysqli_fetch_array($result)) {
    echo $row['firstname'] . " " . $row['lastname'];
    echo "<br />"; }
mysqli_close($con);
?>
```

IZLAZ: Svi podaci iz baze koji se nalaze u tablici *tvz_users* i
sadrže podatak Zagreb u polju *city*

Uvod u relacijski model

ORDER BY određuje kako će se dohvaćeni podaci posložiti. Ključna riječ na kraju određuje da li je poredak rastući (ASC) ili padajući (DESC)

"SELECT firstname, lastname FROM tvz_users ORDER BY lastname ASC, firstname ASC;" dohvatiće sva imena i prezimena ljudi iz tablice tvz_users poredanih po prezimenu, a zatim po imenu (dvije osobe sa istim prezimenom će biti poredane po imenu).

Uvod u relacijski model

LIMIT određuje koliko će se redova maksimalno dohvatiti.

"SELECT firstname, lastname FROM tvz_users ORDER BY lastname ASC LIMIT 10;"

dohvatiti će prvih 10 imena i prezimena iz tablice tvz_users poredanih po prezimenu.

"SELECT firstname, lastname FROM tvz_users ORDER BY lastname ASC LIMIT 10, 5;" dohvatiti će 5 imena i prezimena poredanih po prezimenu odmaknutih od početka rezultata za 10 redaka, redove od 10 do 15.

Uvod u relacijski model

PHP:

```
<?php
$con=mysqli_connect("localhost", "root", "123", "my_db");
$query = "SELECT firstname, lastname FROM tvz_users ORDER
BY lastname ASC LIMIT 10";
$result = mysqli_query($con,$query);
while($row = mysqli_fetch_array($result)) {
    echo $row['firstname'] . " " . $row['lastname'];
    echo "<br />"; }
mysqli_close($con);
?>
```

IZLAZ: Dohvat prvih 10 podataka iz baze koji se nalaze u tablici *tvz_users* poredanih po prezimenu uzlazno

Uvod u relacijski model

Insert je naredba za upis novih sadržaja u tablicu

```
INSERT INTO tvz_users (firstname, lastname, city) VALUES  
('Iva', 'Ivić', 'Zagreb'), ('Joža', 'Horvat', 'Osijek');
```

```
INSERT INTO tvz_users (firstname, lastname, city) VALUES  
($firstname, $lastname, $city);
```

Uvod u relacijski model

PHP:

```
<?php
$con=mysqli_connect("localhost", "root", "123", "my_db");
$query = "INSERT INTO tvz_users (firstname, lastname, city)
VALUES ($firstname, $lastname, $city)";
$result = mysqli_query($con,$query);
mysqli_close($con);
?>
```

IZLAZ: Unos podataka u bazu

Uvod u relacijski model

Update je naredba za mijenjanje već postojećih podataka u tablicama

Naredba će promijeniti ime osobe u tablici gdje je vrijednost atributa id=3

```
UPDATE tvz_users SET firstname='Vedran' WHERE id=3;
```

Uvod u relacijski model

PHP:

```
<?php
$con=mysqli_connect("localhost", "root","123","my_db");
$query = "UPDATE tvz_users SET firstname='Vedran' WHERE
id=3";
$result = mysqli_query($con,$query);
mysqli_close($con);
?>
```

IZLAZ: Izmjena podataka u bazi gdje je id=3. Promjena će biti na polju **firstname** sa podatkom *Vedran*

Uvod u relacijski model

Delete je naredba za brisanje podataka iz tablica

Delete naredba će izbrisati sve zapise (retke) koji zadovoljavaju WHERE uvjet. Ako WHERE nije prisutan izbrisat će se svi zapisi

Naredba će obrisati sve atribute u redu gdje je vrijednost atributa id=3

```
DELETE FROM tvz_users WHERE id=3;
```

Uvod u relacijski model

PHP:

```
<?php
$con=mysqli_connect("localhost", "root", "123", "my_db");
$query = "DELETE FROM tvz_users WHERE id=3";
$result = mysqli_query($con,$query);
mysqli_close($con);
?>
```

IZLAZ: Brisanje podataka u bazi gdje je id=3.

Uvod u relacijski model

Truncate je naredba za brisanje svih podataka u tablici

```
TRUNCATE tvz_users;
```

Grant je naredba za kreiranje i davanje dozvola korisnicima baze podataka

Prava koje se mogu dodijeliti na:

- bazama podataka,
- tablicama,
- pojedinim kolonama u pojedinoj tablici

Uvod u relacijski model

root korisnik je glavni administrator MySQL servera i on ima sva prava.

Root korisnika se koristi da bi se stvorilo nove korisnike ili da bi se nešto popravilo ako dođe do greške u radu servera

Sa novom instalacijom MySQL servera dolazi root korisnik koji nema postavljenu zaporku. Ako se radi o serveru koji će biti javno dostupan na web-u, obavezno postaviti zaporku

Uvod u relacijski model

Grant naredba kreira korisnika „proba_korisnik” koji ima prava izvršavati naredbe *select*, *insert*, *update* i *delete* nad svim tablicama baze podataka „test”, te mu je zaporka za ulaz na MySQL server „zaporka123”.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON test.* TO
'proba_korisnik'@'localhost' IDENTIFIED
BY 'zaporka123';
```

Uvod u relacijski model

Revoke naredba je suprotna naredba od naredbe grant.
Briše prava korisnika.

Revoke naredba će maknuti pravo na izvršavanje
naredbe *insert* nad tablicama „test” baze korisniku
„proba_korisnik”

```
REVOKE INSERT ON test.* FROM 'proba_korisnik'@'localhost';
```

Uvod u relacijski model

Use je naredba koja definira sa kojom bazom podataka ćemo raditi.

MySQL server sadrži jednu ili više baza podataka. Da bi odredili sa kojom ćemo bazom raditi prvo se sa njom moramo povezati

```
USE db_name;
```

Uvod u relacijski model

Set je naredba za mijenjanje serverskih i klijentskih postavki.

Koristi se za npr. postavljanje character set veze klijenta iz PHP aplikacije sa MySQL serverom

```
SET NAMES 'utf8';
```

Uvod u relacijski model

Describe je naredba za opisivanje tablice. Ovom naredbom možemo dobiti prikaz tablice.

```
DESCRIBE products;
```

Field	Type	Null	Key	Default	Extra
product_id	int(10) unsigned	NO	PRI	NULL	auto_increment
url	varchar(100)	NO	UNI	NULL	
name	varchar(50)	NO		NULL	
description	varchar(255)	NO		NULL	
price	decimal(10,2)	NO		NULL	
visible	tinyint(1) unsigned	NO	MUL	1	

Uvod u relacijski model

PHP:

```
$res = mysql_query('DESCRIBE products');
while($row = mysql_fetch_array($res)) {
    echo "{$row['Field']} - {$row['Type']}\n";
}
```

IZLAZ:

product_id - int(10) unsigned
url - varchar(100)
name - varchar(50)
description - varchar(255)
price - decimal(10,2)
visible - tinyint(1) unsigned

Uvod u relacijski model

Explain je naredba za optimizaciju izvršavanja upita na bazu. Omogućava uvid u interno izvršavanje pojedinog upita nad bazom.

Show je generalna naredba za prikaz. Show može prikazati koje sve baze postoje na serveru, koje sve tablice postoje u pojedinoj bazi podataka, korisnike itd.

Pitanja

