Uvod u Veb i Internet tehnologije
Baze podataka i SQL

Milan Banković

Motivacija: kako baratati podacima u programima?

- Mnoge veb i desktop aplikacije u svom radu koriste veliku količinu podataka
- Podaci se obično nalaze u fajlovima na disku (u slučaju veb aplikacija na disku servera)
- Mogući pristupi:
  - ad-hoc pristup: svaka aplikacija sama definiše format i organizaciju podataka u fajlovima, kao i način na koji sa njima operiše
  - organizovani pristup: postoji specijalizovani softver koji barata podacima na organizovan i unapred definisan način, a sve aplikacije koriste usluge tog softvera za rad sa svojim podacima
- prvi pristup pogodan samo za veoma jednostavne programe
- drugi pristup omogućava rad sa mnogo većim skupovima podataka na efikasan i bezbedan način, te se češće koristi u praksi

Šta su baze podataka?

- Baza podataka je skup logički povezanih podataka kojima se pristupa na organizovan i konzistentan način
- Za rad sa bazama podataka koristi se specijalizovani softver koji se naziva sistem za upravljanje bazama podataka (SUBP) (engl. database menagement system (DBMS))
- SUBP implementira algoritme za efikasan i bezbedan pristup podacima u bazi
- Fizička organizacija podataka: način na koji su podaci fizički organizovani na disku
- Logička organizacija podataka: način na koji su podaci logički organizovani iz ugla korisnika (tj. kako ih korisnik vidi)
- Fizička i logička organizacija ne moraju imati veze jedna sa drugom
- Uloga SUBP je da sakrije od korisnika detalje fizičke organizacije i da mu omogući baratanje podacima na logičkom nivou

Vrste baza podataka

- Postoji više tipova baza podataka:
  - Relacione baze podataka
  - Objektno orijentisane baze podataka
  - XML baze podataka
  - ...
- Relacione baze podataka su ubedljivo najčešće korišćene, te se nadalje bavimo samo njima
- Sistemi za upravljanje relacionim bazama podataka (SURBP) (engl. relational database menagement system (RDBMS))

Relacione baze podataka

- Edgar Codd (1970): predložio relacioni model podataka
- Podaci su logički organizovani kao skup *n*-arnih relacija nad atributima
- Relacije možemo razumeti kao tabele, čije su vrste *n*-torke relacije
- Svaka vrsta tabele (*n*-torka relacije) predstavlja jedan entitet (tj. objekat iz stvarnog sveta o kome čuvamo podatke u bazi)
- Svaka kolona predstavlja jedan atribut (određenog tipa) pridružen entitetima
- Dodavanje novog entiteta određenog tipa u bazu se svodi na dodavanje nove vrste u odgovarajuću tabelu
- Primer: studentska baza podataka može imati entitet Student koji može imati atribute: ime, prezime, brIndeksa, smer, pol, itd.

Entiteti

- Entiteti mogu biti regularni (nezavisni), slabi (zavisni) i vezni (asocijativni).
- Regularni entiteti su entiteti koji mogu postojati nezavisno od drugih entiteta u bazi
- Primer: entitet Zaposleni u bazi neke firme predstavlja nezavisan entitet koji postoji sam za sebe u bazi
- Slabi entiteti su entiteti koji se odnose na neke druge entitete i ne mogu samostalno postojati u bazi
- Primer: entitet DeteZaposlenog u bazi firme predstavlja slabi entitet koji je uvek vezan za neki entitet tipa Zaposleni (tj. u pitanju je dete nekog od radnika), te ne može samostalno postojati u bazi
- Vezni entitet uspostavlja neku vrstu veze između dva ili više entiteta u bazi
- Primer: mozemo imati regularni entitet Projekat koji predstavlja projekat na kome firma radi i vezni entitet Angazovanje koji povezuje entitet tipa Zaposleni sa entitetom tipa Projekat (tj. zaposleni je angažovan na nekom projektu)

Još o veznim entitetima

- Vezni entiteti mogu povezivati više od dva entiteta iz baze
- Primer: možemo imati regularne entitete Student, Predmet i Nastavnik koje povezuje vezni entitet Polaganje (tj. veza koja definiše da je neki student polagao određeni predmet kod nekog nastavnika).
- Vezni entiteti mogu imati sopstvene atribute koji bliže opisuju vezu među entitetima
- Primer: entitet Polaganje može imati atribute Datum, Rok i Ocena

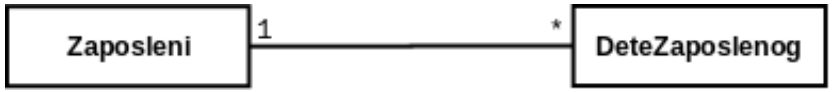
Odnosi među entitetima

Razlikujemo sledeće osnovne tipove odnosa među entitetima:

- jedan-prema-više (engl. one-to-many): jednom entitetu tipa X može odgovarati više entiteta tipa Y, ali svakom entitetu tipa Y odgovara tačno jedan entitet tipa X
  - Primer: odnos između regularnog i od njega zavisnog slabog entiteta; odnos između regularnih i veznih entiteta
- više-prema-više (engl. many-to-many): jednom entitetu tipa X može odgovarati više entiteta tipa Y i obratno
  - Primer: više zaposlenih mogu raditi na istom projektu, a svaki zaposleni može raditi na više projekata
- jedan-prema-jedan (engl.one-to-one): postoji $1-1$ pridruživanje između entiteta tipa X i tipa Y
  - retko se koristi, jer se u tom slučaju može izvršiti spajanje entiteta u jedan (npr. entiteti Zaposleni i BiografijaZaposlenog se mogu spojiti u jedan)

Odnosi među entitetima (primeri)

- Zaposleni može imati 0 ili više dece, a svako dete je dete tačno jednog zaposlenog

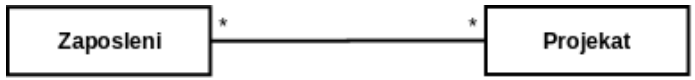


(u pitanju je odnos regularan-slab entitet)

- Zaposleni može imati više angažovanja, ali se svako angažovanje odnosi na tačno jednog zaposlenog. Slično, za svaki projekat može postojati više angažovanja različitih zaposlenih, ali se svako angažovanje opet odnosi na tačno jedan projekat

(u pitanju su odnosi između regularnih i veznog entiteta)

- Ako vezni entitet nema sopstvene atribute, već samo povezuje tačno dva entiteta, onda se u dijagramu može izostaviti



Svaki zaposleni može biti angažovan na više projekata, a svaki projekat može angažovati više zaposlenih (odnos više-prema-više)

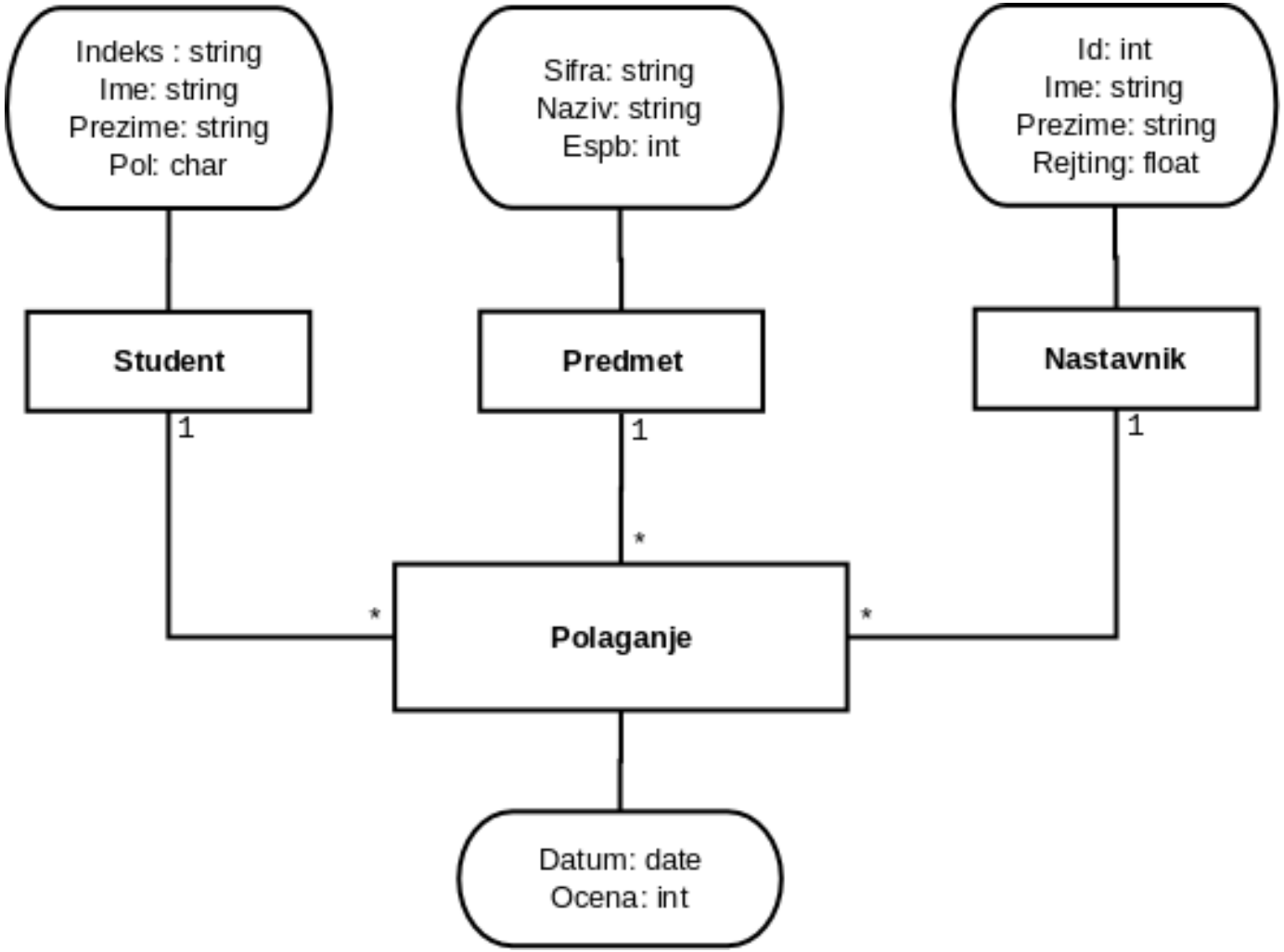
Integritet podataka

- Kao što je ranije rečeno, entiteti se preslikavaju u vrste odgovarajućih tabela
- Za svaki tip entiteta imamo posebnu tabelu, čije kolone odgovaraju atributima entiteta
- Integritet entiteta označava da svaka vrsta u tabeli predstavlja jedinstven entitet tog tipa
  - Ostvaruje se pomoću primarnog ključa (engl. primary key): primarni ključ je podskup skupa atributa takav da jednoznačno određuje entitet
  - Svaka tabela mora imati primarni ključ
  - SURBP ne dozvoljava da postoje dve vrste sa istom vrednošću primarnog ključa
- Kod slabih i veznih entiteta (odnos jedan-prema-više) dodatno postoji i referencijalni integritet koji označava da svaka vrsta u tabeli predstavlja (slabi ili vezni) entitet koji se odnosi na tačno jedan postojeći entitet u nekoj drugoj tabeli
  - Ostvaruje se pomoću stranog ključa (engl. foreign key): to je podskup skupa atributa koji odgovara primarnom ključu tabele sa kojom postoji odgovarajući odnos
  - atributi stranog ključa se obično namenski dodaju u tabelu da bi se uspostavila veza sa odnosnim tabelama
  - SURBP ne dozvoljava kreiranje entiteta za koje ne postoje odgovarajući entiteti u odnosnim tabelama

Postupak projektovanja baze podataka

- Najpre se napravi model entiteta i njihovih odnosa
- Zatim se za svaki regularni entitet kreira jedna tabela čije kolone odgovaraju atributima entiteta
  - Obavezno je identifikovati i definisati primarni ključ
  - Ukoliko nema prirodnih kandidata za primarni ključ, tada se dodaje jedan celobrojni atribut (Id) koji se proglašava za primarni ključ (tzv. surogat ključ)
- Za svaki slabi entitet formira se tabela koja pored kolona koje odgovaraju atributima slabog entiteta dodatno ima i kolone koje odgovaraju primarnom ključu regularnog entiteta na koji se slabi entitet odnosi
  - Dodatni atributi predstavljaju strani ključ u odnosu na regularni entitet
  - I kod slabih entiteta obavezno je definisati primarni ključ
- Za svaki odnos više-prema-više definiše se posebna tabela koja sadrži kolone koje odgovaraju primarnim ključevima dvaju tabela koje čine taj odnos
  - odgovarajuće kolone preuzete iz dvaju tabela koje čine taj odnos su upravo strani ključevi ka tim tabelama
  - svi atributi čine primarni ključ ove vezne tabele
- Za svaki vezni entitet definišemo tabelu koja sadrži kolone koje odgovaraju atributima tog entiteta, kao i kolone koje odgovaraju primarnim ključevima tabela na koje se ta veza odnosi
  - odgovarajuće kolone preuzete iz odnosnih tabela predstavljaju strane ključeve ka tim tabelama
  - primarni ključ se mora zasebno identifikovati (po potrebi, može se uvesti surogat ključ)

Primer celog modela



Tabele (relacije) izvedene iz modela

Student	
Indeks*	string
Ime	string
Prezime	string
Pol	char

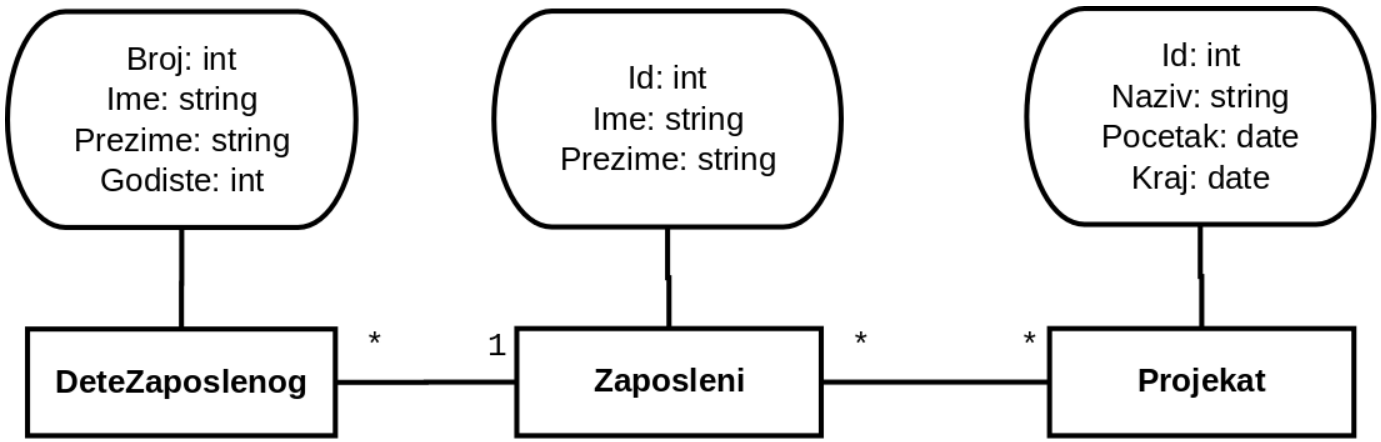
Predmet	
Sifra*	string
Naziv	string
Espb	int

Nastavnik	
Id*	int
Ime	string
Prezime	string
Rejting	float

Polaganje	
StudentIndeks*	string
PredmetSifra*	string
NastavnikId	int
Datum*	date
Ocena	int

- Atributi označeni zvezdicama čine primarne ključeve
- U tabeli Nastavnik uveden je surogat ključ (Id), jer ni jedan podskup prirodnih atributa nije garantovao jedinstvenost
- Strani ključevi u tabeli Polaganje: StudentIndeks (tabela Student), PredmetSifra (tabela Predmet) i NastavnikId (tabela Nastavnik)
- U tabeli Polaganje, pretpostavili smo da ni jedan student ne može polagati dva puta isti ispit istog dana (alternativa: surogat ključ)

Primer celog modela (2)



Tabele (relacije) izvedene iz modela

Zaposleni	
Id*	string
Ime	string
Prezime	string

DeteZaposlenog	
Broj*	int
Ime	string
Prezime	string
Godiste	int
ZaposleniId*	int

Projekat	
Id*	int
Naziv	string
Pocetak	date
Kraj	date

Angazovanje	
ZaposleniId*	int
ProjekatId*	int

- Atributi označeni zvezdicama čine primarne ključeve
- U tabeli Zaposleni uveden je surogat ključ (mogu postojati dva zaposlena koji se isto zovu i prezivaju)
- U tabeli DeteZaposlenog uvedena je kolona ZaposleniId koja je strani ključ u odnosu na tabelu Zaposleni
- Primarni ključ u tabeli DeteZaposlenog sastoji se iz identifikatora zaposlenog (ZaposleniId) i rednog broja deteta tog zaposlenog (Broj)
- U tabeli Projekat uveden je surogat ključ
- Za odnos više-prema-više izmedju tabela Zaposleni i Projekat uvodimo dodatnu tabelu Angazovanje koja sadrži primarne ključeve entiteta iz ove dve tabele na koje se odnosi. Svaki od atributa pojedinačno predstavlja strani ključ ka odgovarajućoj tabeli, a oba zajedno čine primarni ključ tabele Angazovanje

SURBP sistemi i upitni jezici

- SURBP ima zadatak da korisniku omogući da kreira i održava tabele koje predstavljaju model entiteta i odnosa kao i da manipuliše podacima u njima
- SURBP apstrahuje detalje fizičke organizacije podataka na disku i korisniku predstavlja podatke koristeći relacioni model kao vid logičke organizacije podataka
- Komunikacija između korisnika i SURBP se odvija putem posebnog jezika koji nazivamo upitni jezik (engl. query language)
- Upitni jezici su deklarativne prirode (opisuju šta želimo da uradimo, a ne kako to da uradimo)
- Postupak izvršavanja upita (naredbe) prepušta se SURBP sistemu
- Svi najpopularniji SURBP sistemi današnjice (Oracle, IBM DB2, MySQL, PostgreSQL, Microsoft SQL Server) koriste isti upitni jezik: SQL (engl. Structured Query Language)
- Ovaj jezik je standardizovan međunarodnim standardom ISO/IEC 9075
- Svi SURBP sistemi podržavaju najveći deo standarda, a imaju i neke svoje specifičnosti

Struktura SQL jezika

SQL jezik se sastoji iz više celina:

- jezik za definisanje podataka (engl. data definition language (DDL)):
  - Kreiranje i brisanje baza podataka (CREATE DATABASE, DROP DATABASE)
  - Kreiranje i brisanje tabela u bazi (CREATE TABLE, DROP TABLE)
  - Promena definicije tabele (ALTER TABLE)
  - Kreiranje i brisanje korisnika sistema (CREATE USER, DROP USER)
  - ...
- jezik za manipulaciju podacima (engl. data manipulation language (DML))
  - Dodavanje podataka u tabelu baze (INSERT INTO)
  - Brisanje podataka iz tabele (DELETE FROM)
  - Ažuriranje podataka u tabeli (UPDATE)
  - Pretraga (postavljanje upita) (SELECT)
- jezik za kontrolu pristupa podacima (engl. data control language (DCL))
  - Davanje prava korisniku za određenu operaciju nad bazom (GRANT)
  - Oduzimanje prava korisniku za određenu operaciju nad bazom (REVOKE)
  - ...

Tipovi podataka u SQL-u

- Logički tip: BOOLEAN (vrednosti TRUE i FALSE)
- Celobrojni tipovi: TINYINT (8bit), SMALLINT (16bit), INTEGER (32bit), BIGINT (64bit)
- Realni tipovi: FLOAT (jednostruka), DOUBLE (dvostruka)
- Realni brojevi u fiksnom zarezu: DECIMAL
- Tekstualni podaci: CHAR(n), VARCHAR(n), NCHAR(n), NVARCHAR(n) (konstante se zapisuju pod jednostrukim navodnicima)
- Datum i vreme: DATE, DATETIME (konstante se zapisuju u obliku 'yyyy-mm-dd' i 'yyyy-mm-dd hh:mm:ss')
- ...

Izrazi i naredbe u SQL-u

- SQL podržava standardnu sintaksu izraza koja uključuje aritmetičke operatore ($+$, $-$, $*$, $/$), relacione operatore ($<$, $>$, $<=$, $>=$, $=$, $<>$), logičke operatore (*AND*, *OR*, *NOT*)
- SQL podržava veliki broj skalarnih funkcija koje takođe mogu učestvovati u formiranju izraza (*LENGTH*(), *SUBSTR*(), *CONCAT*(), *SQRT*(), *LOG*(), ...)
- Pored skalarnih, postoje i agregatne funkcije (*AVG*(), *MIN*(), *MAX*(), *COUNT*(), *SUM*()) koje se mogu primenjivati na skup vrednosti (npr. na celu kolonu tabele)
- Naredbe u SQL-u se koriste za definisanje tabela koje predstavljaju model podataka (DDL jezik), za manipulaciju podacima u tabeli (DML jezik), kao i za kontrolu prava pristupa podacima (DCL jezik)
- Sintaksa naredbi u SQL-u je kompleksna i najbolje je ilustrovati je kroz primere (videti fajlove `studentska_baza_reset.sql` i `upiti.sql`)
- Najzanimljivija naredba je SELECT naredba kojom se postavljaju upiti nad bazom. Njena sintaksa je i najkomplikovanija

MySQL

- U nastavku ovog predmeta radićemo sa SURBP sistemom MySQL
- U pitanju je softver sa dvojnom licencom: slobodan (GPL) i vlasnički
- Razvoj: MySQL AB (1995)
- Kupio ih Sun Microsystems (2008)
- Oracle kupio Sun 2010. godine
- MariaDB (2010): projekat izveden iz MySQL, slobodan softver, kompatibilan sa MySQL-om
- Mogući načini rada:
  - iz konzole (komanda mysql pod linux-om)
  - korišćenjem veb interfejsa (phpmyadmin)
  - korišćenjem GUI aplikacije (MySQL Workbench)
  - iz programa pisanih u većini popularnih jezika (poput C-a, PHP-a, Python-a, Jave,...) koristeći odgovarajuće biblioteke