

Napredne računarske aplikacije

Predavanje broj: 02

Nastavne teme:

✓ Elementi sistema baza podataka:

Troslojna arhitektura apstrakcija; Rečnik podataka; DBMS jezici

✓ Modelovanje:

Konceptualni model; ER dijagram; Entitet; Instanca; Atribut; Veze

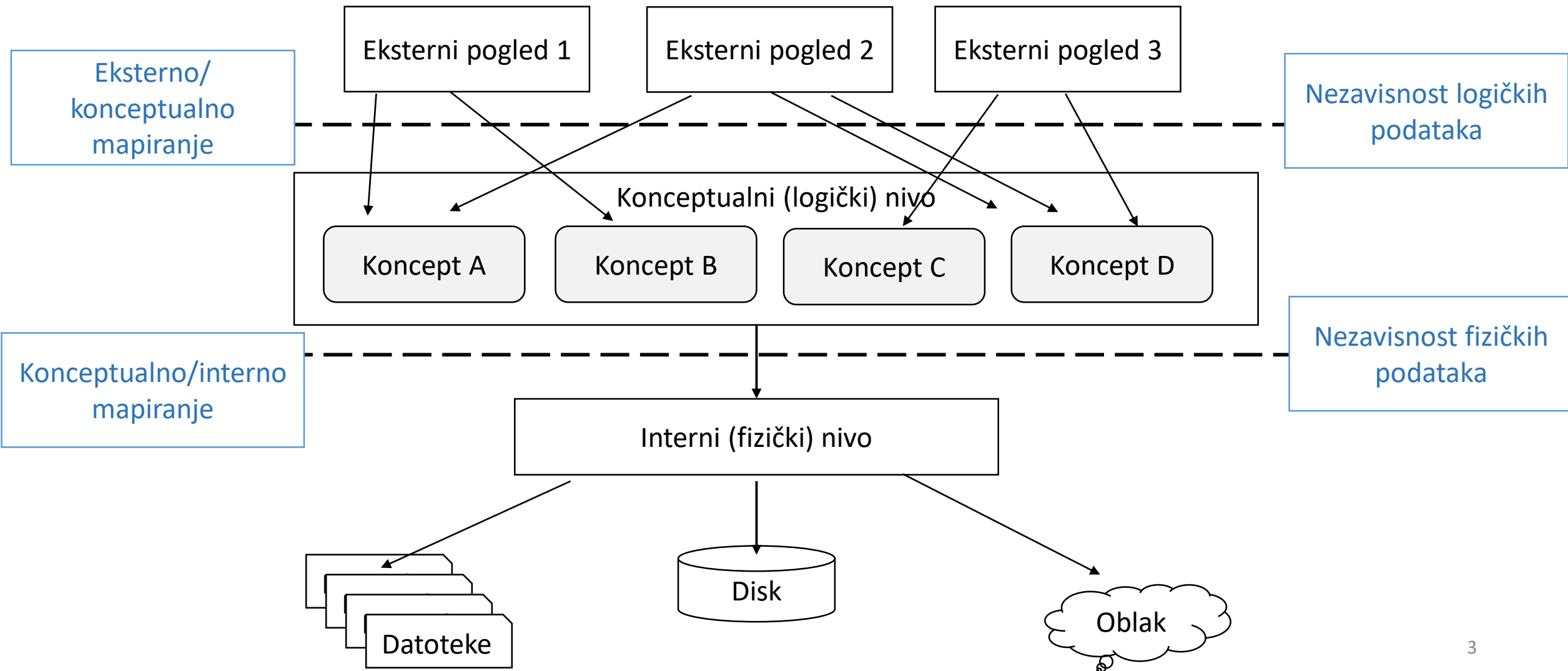
✓ Relacioni model:

Relacija; Ključevi; Referencijalni integritet; NULL vrednost;

Troslojna arhitektura za apstrakciju podataka

- Glavni cilj baza podataka je da se korisnicima pruži apstraktan pogled na podatke kako bi se sakrili detalji o tome kako se podaci skladište i održavaju.
- Tri sloja:
 - *eksterni (korisnik),*
 - *konceptualni (logički),*
 - *interni (fizički)*
- Različiti slojevi imaju različite zadatke.
- Promene na jednom sloju treba da imaju minimalan uticaj na ostale slojeve.
- Nezavisnost slojeva omogućava jednostavniji razvoj aplikacija i promenu platforme itd.

Troslojna arhitektura za apstrakciju podataka



Troslojna arhitektura za apstrakciju podataka

Eksterni sloj

- formira se na temelju konceptualnog nivoa
 - predstavljaja samo poglede (VIEW) prema potrebama pojedinih korisnika.
 - uobičajeno je da su to samo delovi ukupne baze podataka
 - obično ima više pogleda na jedinstvenu bazu podataka.
- Eksterni (korisnički) nivo se sastoji od skupa korisničkih modela. Korisnički model predstavlja pogled na jedan deo baze podataka koji je od interesa određenom korisniku. Definisanje korisničkih nivoa je neophodno radi određivanja prava pristupa delovima baze podataka. Eksterni nivo je predstavljen podšemom baze podataka.

Troslojna arhitektura za apstrakciju podataka

Konceptualni sloj

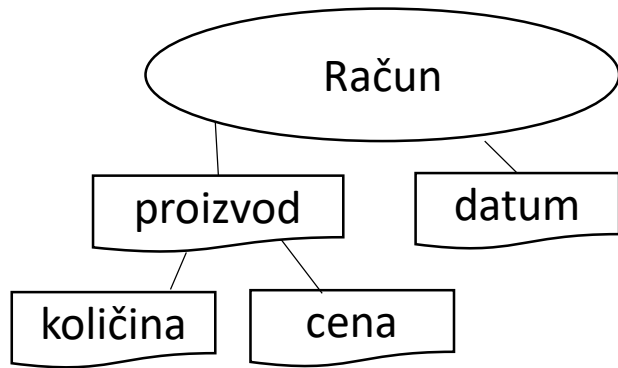
- specificira objekte, karakteristike objekata, veze, pravila integriteta, ponašanje objekata.
 - opisuje šta je smešteno u bazi i koji odnosi postoje između tih podataka.
 - najbliži stvarnosti
 - prikriva detalje fizičkog skladištenja struktura
 - fokusira se na opis entiteta, tipova podataka, veza i ograničenja
-
- Konceptualni nivo prikazuje logički model celokupne baze podataka kojim je definisan sadržaj baze podataka. Logički model predstavlja šemu baze podataka u okviru kojih su definisani atributi za svaku vrstu entiteta koji je obuhvaćen u modelu.

Troslojna arhitektura za apstrakciju podataka

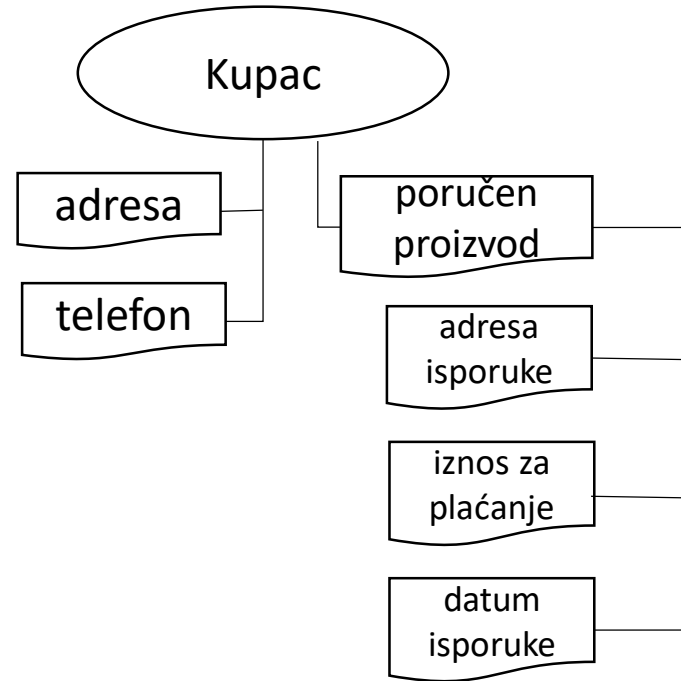
Interni sloj

- specificira kako se podaci fizički skladište i organizuju (npr. indeksi, putanja pristupa...)
- odnosi se na organizaciju datoteka i njihov fizički zapis na disku
- vremenom se menja:
 - uvode se novi memorijski uređaji za zapis podataka
 - koriste se novi indeksi za pristup podacima (npr. za često korišćene upite)
 - različita distribucija podataka na memorijskim modulima
 - baferi za često korišćene podatke
- Interni (fizički) nivo obuhvata sadržaj cele baze podataka i definiše način fizičke organizacije podataka na medijumima gde su podaci smešteni (formati zapisa u memoriji, organizacija i drugo.).

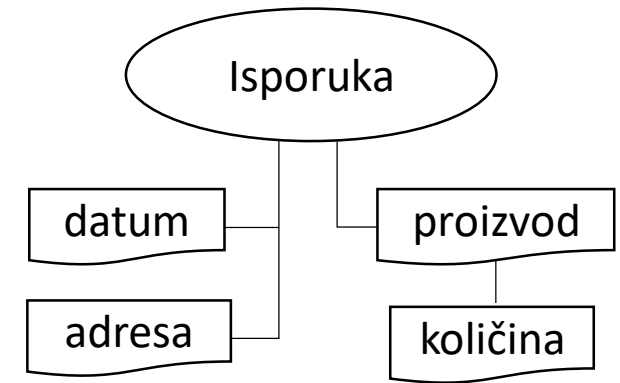
Odeljenje finansija



Odeljenje za odnose sa kupcima



Odeljenje logistike



Eksterni (korisnički) sloj

<i>Proizvod</i>	naziv, opis, cena,...
<i>Kupac</i>	ime, telefon, adresa, ...
<i>Račun</i>	kupac, datum, proizvodi(sa cenom i količinom)
<i>Isporuka</i>	račun, adresa, datum,

Konceptualni (logički) sloj



Interni (fizički) sloj

Rečnik podataka (katalog)

Rečnik podataka predstavlja skup podataka o podacima koji se koriste u analiziranom sistemu.

- Srce sistema baza podataka
- Sistemska baza podataka sa metapodacima

Rečnik podataka obuhvata tri celine:

- opis struktura podataka koje se koriste u tokovima podataka,
- opis polja definisanih nad podacima i
- opis domena.

DBMS jezici: DDL i DML

- ***Data Definition Language (DDL):***
 - jezik koji se koristi za definisanje šeme baze podataka.
 - jezik koji se koristi za definisanje elemenata konceptualnog, internog i eksternog sloja.
- ***Data Manipulation Language (DML):***
 - jezik koji se koristi za prikaz, modifikaciju, dodavanje i brisanje podataka.
 - jezik za pristup i manipulisanje podacima organizovanih odgovarajućim modelom podataka.

DBMS jezici: SQL

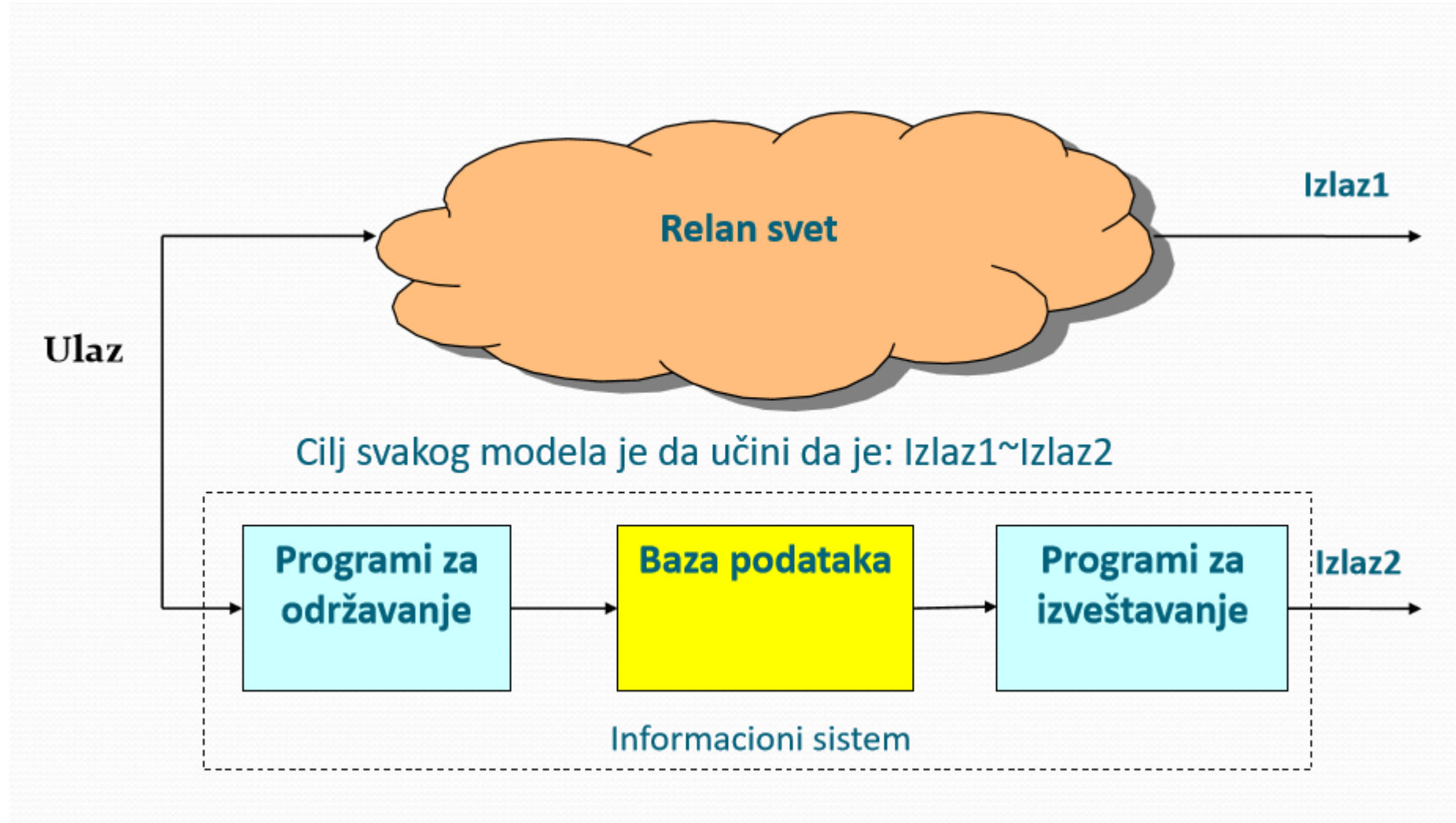
- SQL (*Structured Query Language*) je DDL i DML.
- SQL jeste jezik za upravljanje podacima posredstvom sistema za upravljanje bazom podataka
- Standardizovan je 1986. godine, ali i pored toga nije u potpunosti portabilan između različitih sistema za upravljanje bazama podataka
- Posедуje kompletnu sopstvenu sintaksu koju je potrebno savladati kako bi se na adekvatan način moglo rukovati podacima u bazi.
- Može se koristiti:
 - interaktivno (*intreactive SQL*)
 - ugrađen u odgovarajuće programsko okruženje (*embedded SQL*)

Modelovanje

Modelovanje je postupak kojim se realni svet svodi na određeni broj objekata.

- Modelovanje se radi paralelno sa analizom potreba.
- Kako se informacije prikupljaju, objekti se identifikuju, dodeljuju im se imena koristeći termine bliske krajnjim korisnicima.
- Objekti se modeluju i analiziraju **korišćenjem dijagrama objekti-veze (ER dijagrami)**.
- Ako model nije tačan, modifikuje se, što ponekad zahteva da se prikupe dodatne informacije.
- Ciklus pregledanja i modifikovanja se nastavlja sve dok se ne dobije potvrda da je model korektan.

Modelovanje



Modelovanje

Modelovanje je postupak koji obuhvata tri koraka:

Izbor (selekcija) - izdvajanje bitnih objekata

- Mnoštvo objekata iz realnog sveta redukuje se na manji skup objekata, koji će činiti objekte modela.
- Složenost realnog sistema smanjuje.
- Selekcija se ne odnosi samo na objekte nego i na njihove osobine, kao i na međusobne veze (relacije) između objekata.

Dodeljivanje imena (za svaki objekat, vezu, atribut)

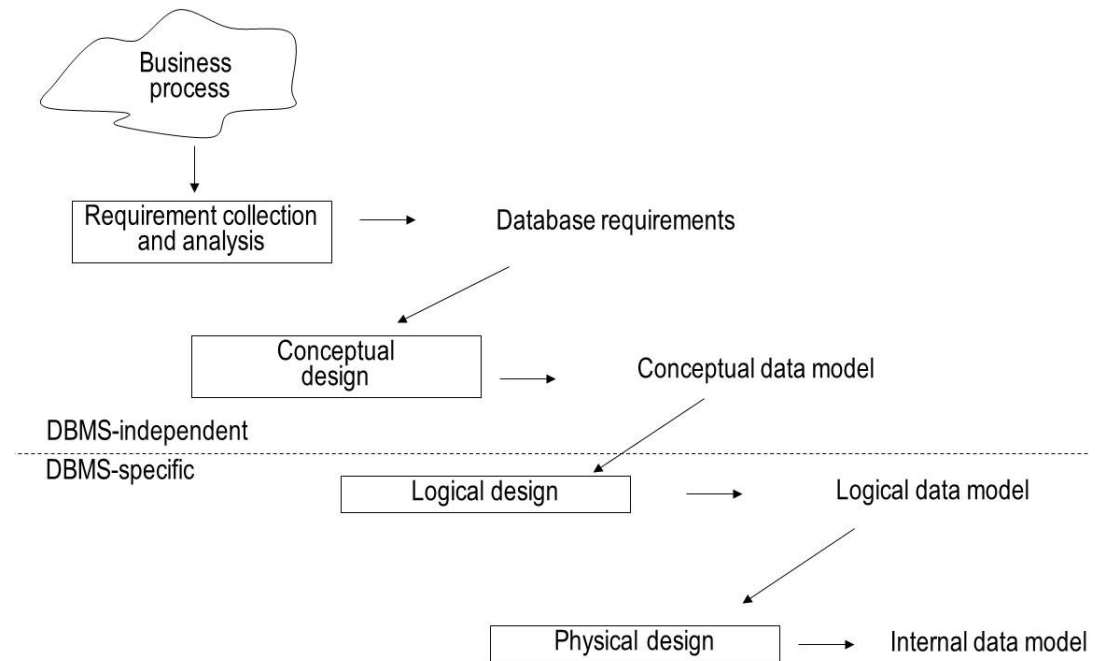
- Svakom objektu u realnom svetu, svakoj vezi između uočenih objekata, kao i svakom atributu (svojstvu) uočenog objekta ili veze dodeljuje se ime.

Klasifikacija

- Nehomogeni skup objekata i odnosa se svrstava u homogene klase i tipove objekata.
- Klasifikacija uvek zavisi od područja primene.

Faze razvoja baze podataka

- Prilikom implementiranja nove baze podataka, lako je upasti u zamku pokušavajući na brzinu odraditi posao, bez prethodnog utroška vremena na proces dizajna, odnosno projektovanja.
- Koliko je ova faza bitna, najbolje se može videti nakon završetka projekta, kada je zbog loše dizajnirane baze podataka neophodno često raditi skupe redizajne i reimplementacije.
- Projektovanje baze podataka je slično projektovanju kuće. Nerealno je početi sa zidanjem bez prethodno urađenog detaljnog planiranja. Takođe, dobar dizajn omogućava da kasnije originalnu građevinu proširite, odnosno dozidate, bez potrebe za rušenjem prethodno izgrađenog.



Faze razvoja baze podataka

- Analiza zahteva
 - Pre nego što sagledamo kako ćemo nešto uraditi, potrebno je prvo da vidimo šta je uopšte potrebno da uradimo. To ćemo najbolje učiniti sakupljanjem što više neophodnih informacija i njihovom analizom
- Konceptualni dizajn
 - Nakon što smo prikupili i analizirali zahteve, potrebno je ove zahteve pretočiti u neku vrstu formalnog, konceptualnog dizajna. U tu svrhu se najčešće koriste razni dijagrami za modelovanje entiteta i relacija među njima
- Logički dizajn
 - Na kraju, logičkim dizajnom se vrši mapiranje konceptualnog dizajna na konkretne tabele i relacije odabranog sistema baza podataka

Konceptualni model

Konceptualni model nastaje kao rezultat faze u razvoju baze podataka koja se odnosi na konceptualni dizajn.

Konceptualni model sadrži:

- Strukturu podataka – statički opis stanja sistema koji čine objekti, njihova svojstva, veze između objekata i njihovih svojstava.
- Operacije nad strukturom modela - prouzrokuju kretanje i promene tj. dinamiku realnog sistema.
- Ograničenja (constraints) - pravila koja razdvajaju dopuštena od nedopuštenih stanja realnog sistema i u svojoj prirodi deo su strukture modela podataka
Posledica ograničenja iz realnog sveta.

Konceptualni model ne određuje fizički oblik u kome se podaci čuvaju.

– NIJE VEZAN ZA ODREĐEN DBMS!

Konceptualni model

Izrada konceptualnog modela obuhvata sledeće korake:

1. Identifikaciju entiteta
 - ključne reči tokom intervjua korisnika
 - koji podaci bi trebalo da se skladište?
 - postojeći formulari, dokumenti i kartoteke
 - nezavisni i zavisni entiteti
2. Identifikaciju atributa
3. Identifikaciju ključeva
4. Definisavanje veza
5. Određivanje kardinalnosti veza
6. Normalizaciju
7. Definisavanje referencijalnog integriteta veza

Entitet

- ***Entitet*** predstavlja **objekat** posmatranja koji se može **jednoznačno odrediti, identifikovati i razlikovati**.
- Pojam od važnosti koji treba da bude zastupljen u bazi podataka.
- Entitet predstavlja koncept koji treba da omogući skladištenje i praćenje odgovarajućih podataka.
- Entiteti su ograničeni na objekte koji se u relacionoj bazi podataka, u narednim fazama, mogu predstaviti upotrebom tabela.

Entitet

- **Identifikovanjem entiteta se utvrđuje klasa entitet.**
- Entiteti jedne klase se međusobno razlikuju.
- Svaki entitet mora imati ime koje je jedinstveno.
- Ne mogu postojati dva ista entiteta u jednoj bazi podataka.
- Entitet po svojoj prirodi može biti:
 - deo okruženja: radnik, student, aparat, zgrada
 - apstraktan pojam: zvanje, boja, mera
 - događaj: upis, ispit, pregled
 - asocijacija: student – predmet, radnik-projekat, predmet-profesor itd...

Instanca

- Entiteti imaju **instance**.
- Instanca entiteta je jedinstvena **pojava entiteta**.

ENTITET	INSTANCA
OSOBA	Mahatma Gandhi, Žika Mikić...
BREND	Nike, Champion, Adidas...
TIP PROIZVODA	Cipele, Čizme, Patike
POSAD	Električar, Vozač, Bravar
ŽIVOTINJA	Pas, Mačka, Krava...
VOZILO	Kabriolet, Prikolica, Karavan...

Entiteti i instance

Da li je PAS instanca ili entitet?

Entiteti i instance

ZAVISI

- Šta ako smo zainteresovani za **različite vrste životinja**?
- Tada ima smisla razmišljati o **entitetu ŽIVOTINJA** sa **instancama PAS, MAČKA, KONJ** i tako dalje.
- Šta ako vodimo **uzgajivačnicu pasa**? Moramo da čuvamo **podatke o različitim rasama pasa**, ali ne o nekim drugim vrstama životinja. Za uzgajivačnicu pasa, mnogo je prirodnije misliti o **entitetu PAS** sa **instancama TERIJER, PUDLA, LABRADOR** i tako dalje.

Atribut

- **Atribut** je jedno od **svojstava entiteta** koji posmatramo i za koji sakupljamo podatke.
- **Atribut** ima samo **jednu vrednost za jednu instancu entiteta**.
- Atribut je poseban deo informacije koji:
 - Opisuje entitet
 - Kvantifikuje entitet
 - Kvalifikuje entitet
 - Klasifikuje entitet
 - Specifikuje entitet

Entitet - Atribut

ENTITETI	Atributi
MUŠTERIJA	Prezime, godine starosti, broj cipela, mesto stanovanja, e-mail
KOLA	Model, težina, katalogska cena
NARUDŽBINA	Datum naručivanja, datum isporuke
POSLO	Naziv, opis
TRANSAKCIJA	Iznos, datum transakcije
RADNI UGOVOR	Datum početka, plata

Atribut

- Atributi imaju **vrednosti**.
- **Vrednost atributa** može biti broj, niz znakova, datum, zvuk, slika itd...
- To su takozvani **tipovi podatka** ili formati.
- Vrednost atributa se definiše dodelom odgovarajućeg tipa podatka.

Entitet – Atribut -Instanca

ENTITETE	INSTANCE
opisuju	su opisane
Atributi	Vrednostima atributa
koji pripadaju određenom	koje se pamte kao
Tipu podataka	Podaci

Entitet – Atribut -Instanca

Struktura baze podataka - šema (konceptualni model)	Stanje baze podataka (fizička implementacija)
ENTITET	INSTANCA
Atributi	Vrednosti atributa

Relevantni atributi

- Identifikovanje atributa u postupku modelovanja podrazumeva **izbor relevantnih atributa za opis entiteta.**
- Broj relevantnih atributa nije fiksna i zavisi od konkretnog procesa i obrade podataka koju treba obaviti.
- Relevantne attribute definiše kompetentna osoba:
 - u skladu sa zahtevima (informacijama) jer od toga zavisi upotrebljivost dobijenih informacija

Relevantni atributi

Primer 1:

- Sa aspekta zarada, atribut “adresa” nije relevantan
- Sa aspekta organizovanja prevoza relevantan atribut

Primer 2:

- Sa aspekta odeljenja za plate atribut “vrsta bolesti” nije relevantan
- Sa aspekta zdravstvenog odeljenja istog preduzeća relevantan atribut

Primer 3:

Za sektor za urbanizam definišemo entitet ULICA sa sledećim relevantnim atributima koji ga opisuju:

- ULICA <naziv, dužina, širina, vrsta_kol, god_izg, kuca....>
- atribut promena temeprature?????

Mera relevantnih atributa

- **Premalo atributa:**
 - model će biti jednostavan za predstavljanje i analizu,
 - mala verodostojnost modela,
 - ograničen je broj upotrebljivih informacija
- **Previše atributa:**
 - velik verodostojnost modela,
 - kompleksnost velika,
 - manipulacija podacima teško izvodljiva,
 - dobijaju se konfuzne informacije.
- **Prepoznavanje prave mere relevantnih atributa** u modelovanju nekog procesa jedan od ključnih zadataka projektanta baze podataka.

Prost/Složen atribut

- **PROST ATRIBUT** se ne može rastavljati na delove bez gubitka svakog značenja – može se opisati samo jednim elementarnim (atomarnim) podatkom.
 - Visina (cm), Ocena, Smer itd.
- **SLOŽEN ATRIBUT** se sastoji od više prostih atributa. Može se rastaviti na jednostavnije.
 - Adresa (Ulica, Broj, Mesto, ...),
 - DatumRođenja (Dan, Mesec, Godina),
 - Ulica (Naziv, KucniBroj)

Mandatorni/Opcionalni atribut

- Mandatorni i opcionalni atributi
 - **MANDATORNI**: atributi koji moraju da imaju vrednost.
 - U većini poslova koji prate informacije o zaposlenima, ime zaposlenog je neophodno.
 - **OPCIONALNI**: atributi koji mogu imati vrednost ili ostati nepopunjeni.
 - Broj telefona često nije potreban, izuzev u mobilnim ili bežičnim aplikacijama.

E-mail adresa može biti mandatorni atribut za ZAPOSLENOG ako modelujete e-mail aplikaciju.

Svaka osoba **mora** da ima **datum rođenja**, a **može** da ima **datum venčanja**, **datum diplomiranja** i sl.

Promenljivi/Nepromenljivi Atribut

- Neki atributi (kao godine starosti) imaju vrednosti koje se stalno menjaju. Oni se nazivaju **promenljivi atributi**.
- Drugi atributi (kao datum naručivanja, datum rođenja) će se menjati retko ili nikad. To su **nepromenljivi atributi**
- Ako postoji izbor između atributa, **koristite nepromenljive**.
- Na primer, upotrebite datum rođenja umesto godina starosti.

JEDINSTVENI IDENTIFIKATOR (UID – unique identifier)

- **UID je atribut ili kombinacija atributa koji razlikuju jednoznačno identifikuje svaki primerak klase entitet.**
- Svaki student je opisan u nekoliko crta ili atributa.
- Koji atribut ili atributi vam dozvoljavaju da odaberete jednog studenta od ostatka grupe?
- To je jedinstveni identifikator studenta.



carmax.com

Domen atributa

- ***Domen atributa*** je skup svih mogućih vrednosti koje atribut može poprimiti.
- Domen je uži pojam od tipa podataka.
- Označava se sa D_i
 - $\text{Dom}(A_i) = \{\text{Vrednost}_1, \text{Vrednost}_2, \dots, \text{Vrednost}_n\}$
- Ulica i Prezime su tekstualni tip podataka,
 - ali su im domeni potpuno različiti
- Naučno zvanje je tipa text,
 - {docent, vanredni profesor, redovni profesor}
- Poštanski broj je petocifren, ali nisu svi petocifreni brojevi u igri

Domen atributa

- Entitet STUDENT < BrInd, ime, prezime, SP, adresa>
- Definisane domena atributa SP:
 - $\text{Dom}(\text{SP}) = \{\text{AVT}, \text{ASUV}, \text{EI}, \text{ELITE}, \text{IS}, \text{NET}, \text{NRT}, \text{RT}\}$
 - Skup svih mogućih gradova je domen atributa Grad.
- Može se definisati jedan domen za više atributa, obrnuto ne.
 - $D_i = \{\text{"crna"}, \text{"braon"}, \text{"zelena"}, \text{"plava"}, \text{"crvena"}, \text{"žuta"}\}$
- Najčešće korišćeni domeni:
 - Skup celih brojeva
 - Skup decimalnih brojeva
 - String – niz alfanumeričkih karaktera
 - Datum
 - Logički podatak
 - Novčani
 - Korisnički domeni

Entitet - Atribut

- Nekada je teško razlikovati entitete od atributa.
- Ako je **atribut dovoljno složen**, tako da ima svoje dodatne attribute, može se posmatrati kao **novi entitet**.
- Adresa može biti entitet i atribut.
 - Entitet: ako je potrebno voditi evidenciju poslovanju prodavnica po gradovima ili evidenciju o broju stanovništva po gradovima i sl...
ADRESA <ulica, broj, poštanski broj, grad>
 - Atribut: podaci o zaposlenima u nekom preduzeću
RADNIK <ime, prezime, adresa, kvalifikacija, datum_zaposlenja,...>

Entitet - Atribut

- Ulica može biti entite i atribut:
 - entitet: ako se vode podaci o npr. Sektoru za urbanizam
 - ULICA <naziv, dužina, širina, vrsta_kol, god_izg, kuca....>
 - atribut entiteta ZGRADA, vode se podaci u sektoru opštine o zgradama
 - ZGRADA <ulica, kucni_br, god_izg, br_spratova, br_stanova...>
- Prednost u slučaju da je atribut – višestruka upotreba istog koda.
- Mana u slučaju da je atribut – komplikovaniji kod za izdvajanje različitih informacija u zavisnosti od potreba.

VEZE IZMEĐU ENTITETA

- U realnom sistemu objekti nisu međusobno izolovani
 - **objekti** se nalaze u **međusobnoj interakciji**
- Npr. veza “studenti polažu predmet”
 - Između entiteta **Student** i **Predmet** postoji veza **Student_Polaže_Predmet** koja se može nazvati **Ispit**
- **Entiteti između kojih postoji veza zovu se učesnici veze** (*participants*)

VEZE IZMEĐU ENTITETA

- Učestvovanje nekog entiteta u vezi može biti **potpuno** i **delimično**
 - Potpuno – zapisi entiteta mogu da postoje samo ako postoje i u drugom (**slab entitet**)
 - Delimično – zapisi entiteta mogu da postoje bez obzira na drugi entitet (**jak entitet**)
- IS prodavnice:
 - Dobavljač – **jak entitet**
 - Artikal – **slab entitet**
- IS banke:
 - Klijent– **jak entitet**
 - Transakcija – **slab entitet**

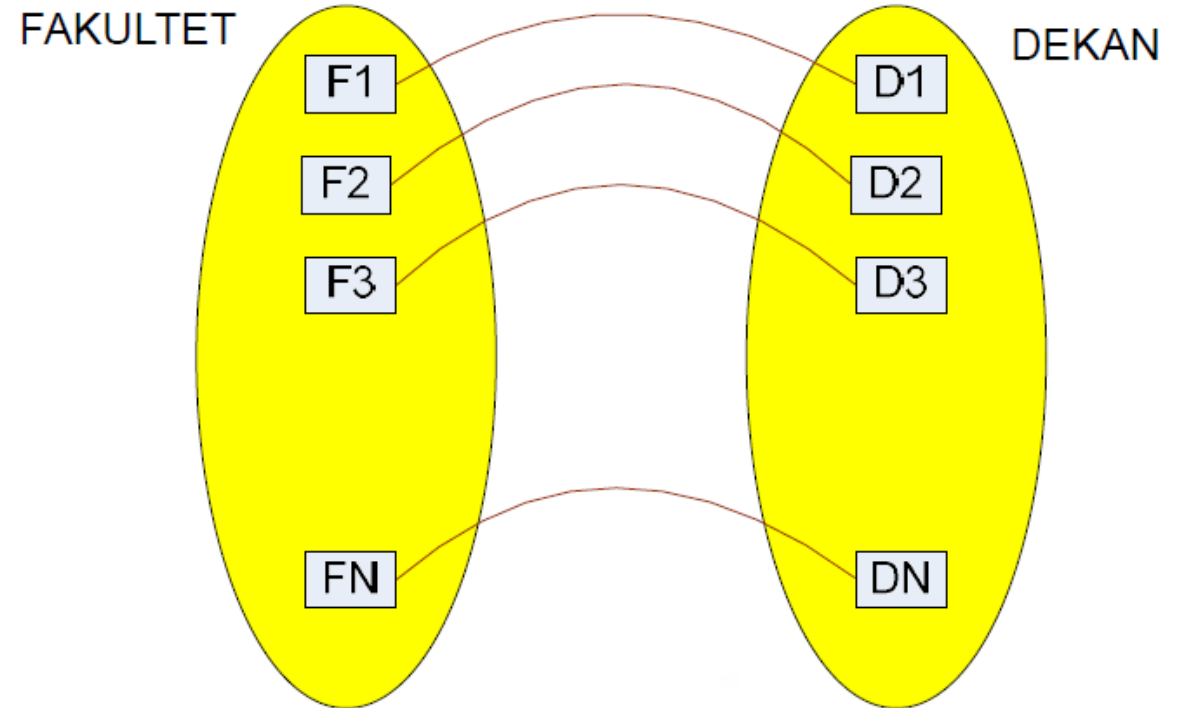
VEZE IZMEĐU ENTITETA

- Odnosi između objekta – deo modela podataka
- Analiziraju se tipovi odnosa koji se uspostavljaju.
- Odnosi između entiteta posmatranja prikazuju se najčešće primenom logike skupova i preslikavanja njihovih elemenata.

- Odnosi između entiteta:
 - 1:1; 1: N; N:M
 - Drugi način prikazivanja (1:1; 1:*; *:*)

VEZA 1:1

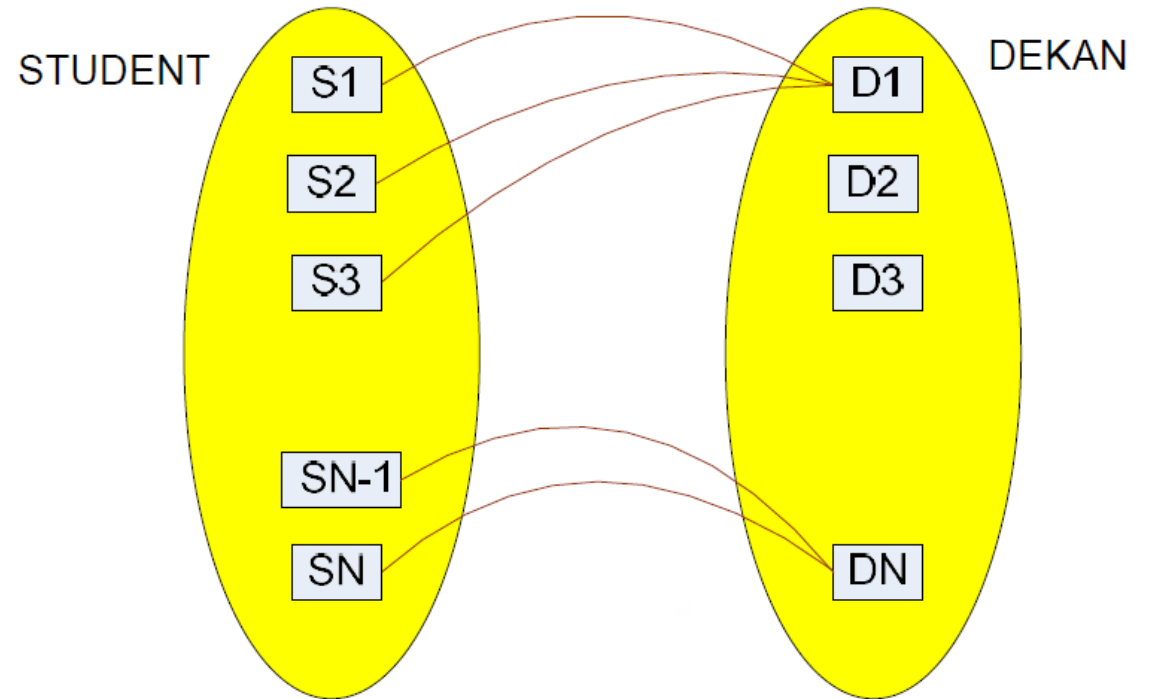
- Najjednostavniji način preslikavanja.
- Svaki element skupa X može se preslikati na najviše jedan element skupa Y.
- Svaki element skupa Y može biti preslikan na najviše jedan element skupa X.



Odnos između entiteta FAKULTET i DEKAN

Veza 1:N ili N:1

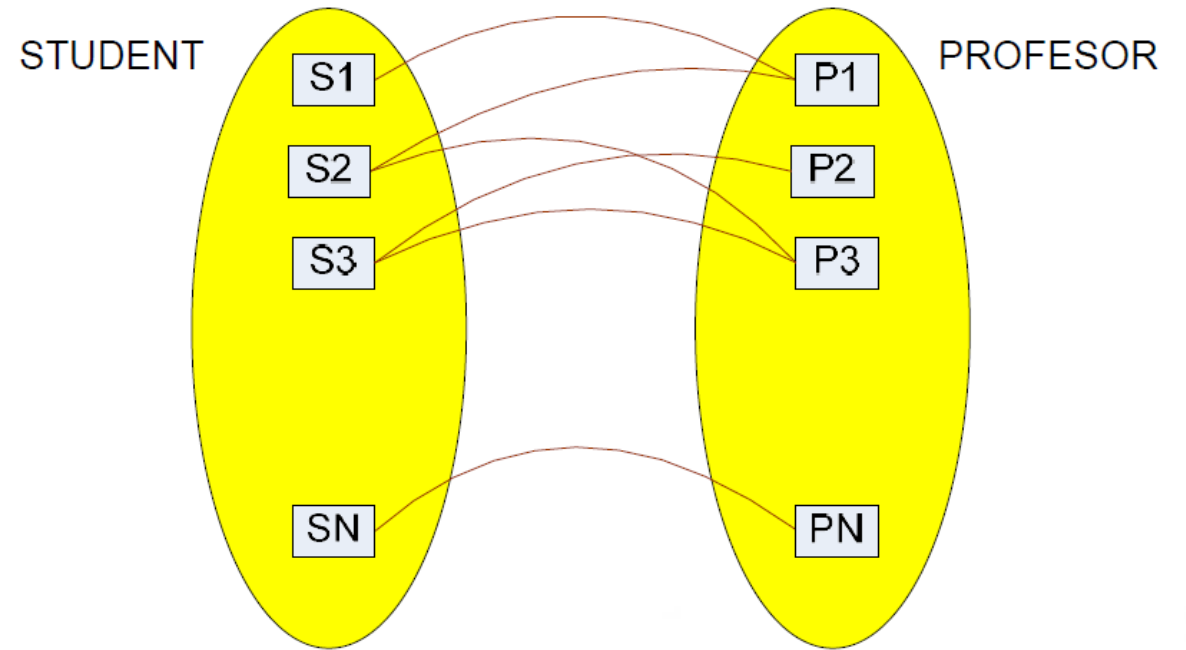
- Više elemenata skupa X može se preslikati na najviše jedan element skupa Y.
- Jedan element skupa Y može se preslikati na više elemenata skupa X.



Odnos između entiteta STUDENT i DEKAN

Veza M:N

- Svaki elemenata skupa X može se preslikati na više elemenata skupa Y.
- Svaki elemenata skupa Y može se preslikati na više elemenata skupa X.



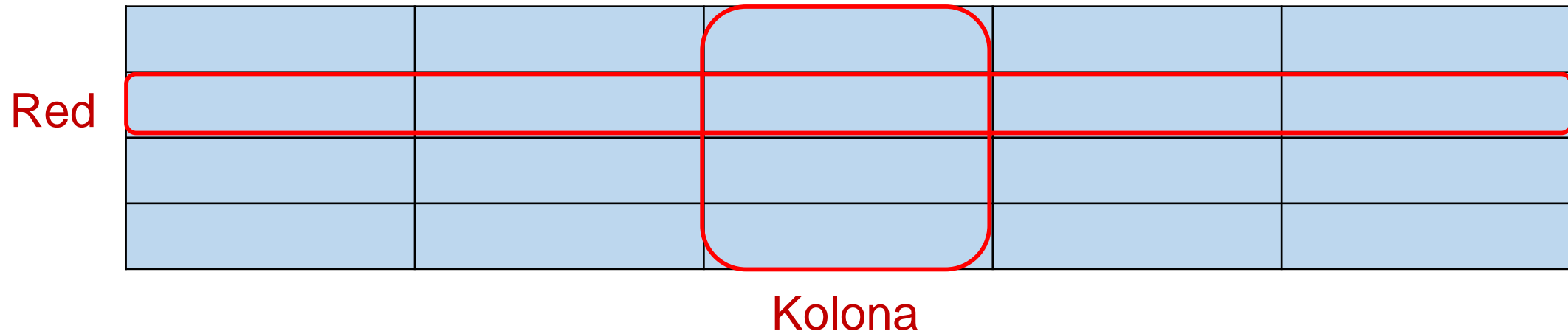
Odnos između entiteta STUDENT i PROFESOR

MAPIRANJE TERMINA

Dizajn KONCEPTUALNI MODEL	Fizička implementacija u BP RELACIONI MODEL
Entitet	Tabela
Instanca	Red
Atribut	Kolona
UID	Primarni ključ
Veza	Strani ključ

RELACIJA

- Relacioni model podataka na realni svet gleda putem tabela.
- Relacija predstavlja specifičan tip tabele.
- Relacija je dvodimenzionalna tabela koja poseduje specifične karakteristike.
- Tabela se sastoji od redova i kolona.



KARAKTERISTIKE RELACIJE

- Redovi sadrže podatke o instancama entiteta.
- Kolone sadrže podatke o atributima entiteta.
- Čelija (polje) tabele sadrži jednu vrednost.
- Sve vrednosti u jednoj koloni su istog tipa podataka.
- Svaka kolona ima jedinstveno ime.
- Ne mogu postojati dva identična reda.
- Redosled kolona nije od važnosti.
- Redosled redova nije od važnosti.

JEDNOSTAVNA RELACIJA

EmployeeNumber	FirstName	LastName
100	Yousef	Hassan
101	Jerry	Cadley
104	Yi	Chang
107	Shir	Anavi

PRIMER TABELE KOJI NIJE RELACIJA

EmployeeNumber	Phone	LastName
101	335-6421, 454-9744	Hassan
100	215-7789	Cadley
104	610-9850	Chang
107	299-9090	Anavi
100	215-7789	Cadley

Sve relacije su tabele, ali nisu sve tabele relacije!

TERMINOLOGIJA -SINONIMI

TABELA	RED	KOLONA
FAJL (RETKO)	ZAPIS	POLJE
RELACIJA	N-TORKA	ATRIBUT

RELACIONI MODEL

- Relacioni model podataka predstavlja teorijsku osnovu za baze podataka relacionog tipa.
- 1970. prvi put su dati principi i struktura relacionog modela podataka (E.F.Codd).
- Relacioni model podataka razmatra sledeće komponente:
 - strukturnu (objekti, entiteti) - predstava podataka
 - integritetsku (ograničenja) – zaštita podataka
 - manipulativnu (ažuriranje) – manipulisanje podacima

RELACIONI MODEL

- **Osnovni pojmovi relacionog modela podataka:**
 - atribut,
 - domen,
 - šema relacije i
 - relacija
- **Šema relacije R** u oznaci $R(A_1, A_2, \dots, A_N)$, je konačan skup atributa $\{A_1, A_2, \dots, A_N\}$ i konačan skup ograničenja nad vrednostima tih atributa.

RELACIONI MODEL

- Šema relacije kojom se opisuje klasa studenata, gde su relevantni atributi Broj indeksa i Ime studenta, može da bude:
 - STUDENT (BrInd, Ime)
- Relacija nad ovakvom šemom u jednom trenutku može da bude


(BrInd	Ime)
2017123	J.Jankovic
2018456	P.Petrovic
2016222	J.Jovanovic
2016555	M.Markovic

RELACIONI MODEL


- **Relaciona baza podataka je konačan skup relacija** koje su definisane odgovarajućim šemama relacija $\{R_i\}$ i konačan skup ograničenja koja važe između datih šema.
- U jednoj bazi podataka postoji više relacija (tabela) koje su međusobno povezane.
- Suština relacionih baza je vezama i posebnim ograničenjima koja se odnose na veze.

RELACIONA BAZA PODATAKA

- Konceptualni model predstavljamo grafički – ER dijagramom
Na osnovu ER-dijagrama kreiramo relacionu bazu podataka koja se sastoji iz tabela koje su međusobno povezane.

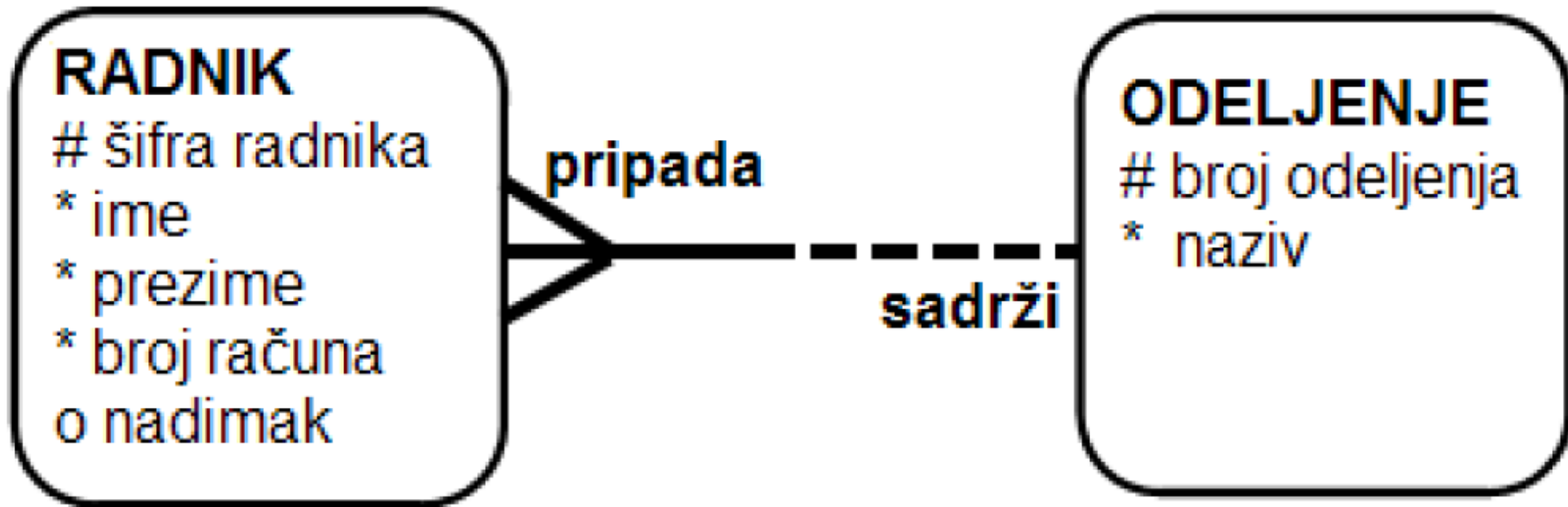
Konceptualni model  ER-dijagram

ER-dijagram  Relacioni model

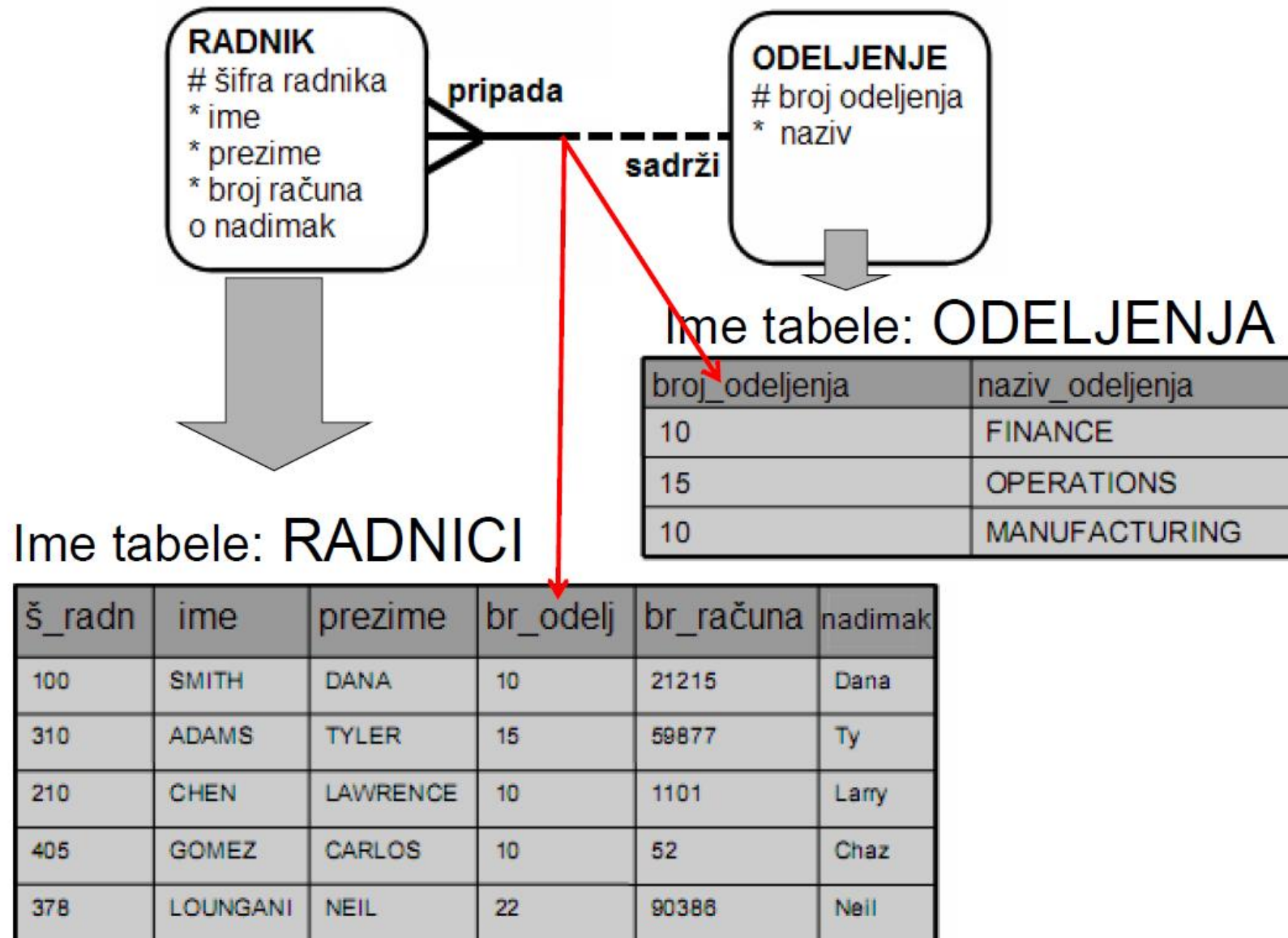
Relacioni model  Relaciona baza podata
fizička implementacija

Konceptualni model prikazan ER – dijagramom

Potrebno je napraviti model koji vodi evidenciju radnika po odeljenjima. Za svakog **radnika** potrebno je čuvati podatke o imenu, prezimenu, broju računa i ukoliko ima nadimak. Svako **odeljenje** ima svoj naziv. Radnici su raspoređeni po odeljenjima tako da jedan radnik radi samo u jednom odeljenju. U odeljenju može da radi jedan ili više radnika. Postoje i odeljenja koja su otvorena i kojima nisu dodeljeni radnici.



ER – dijagram u relacioni model

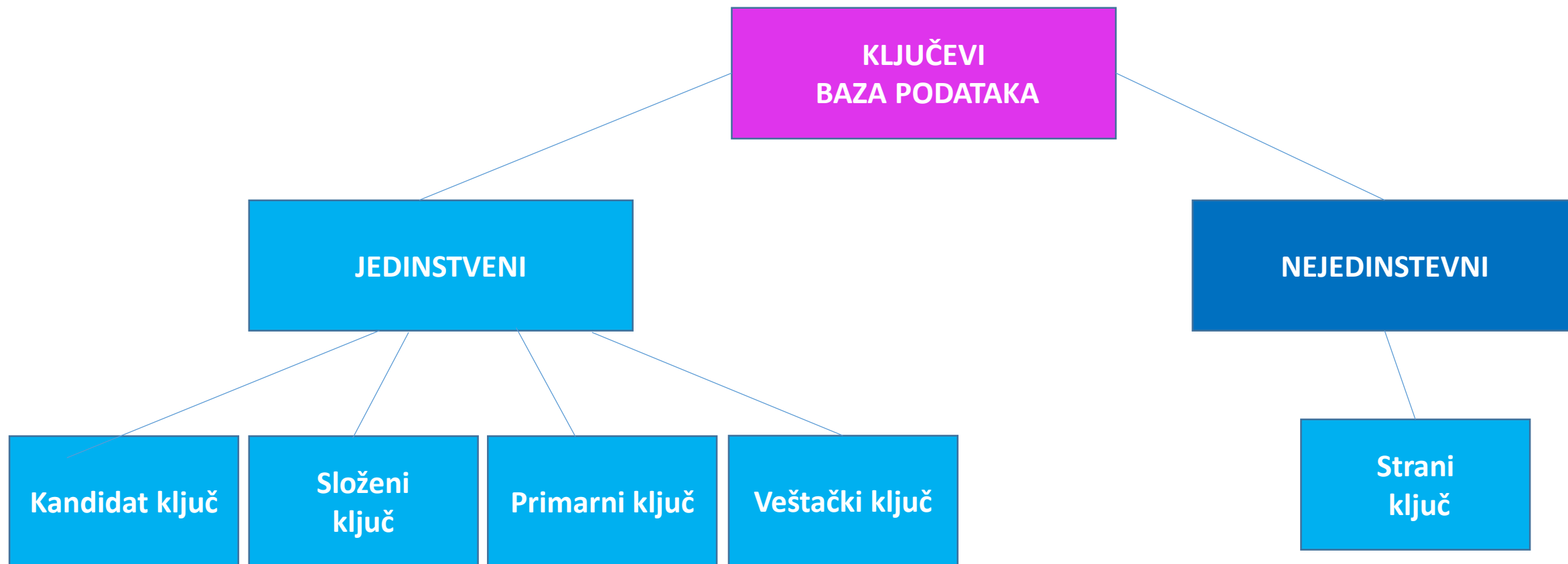


KLJUČEVI

- **Ključ** relacije predstavlja jednu ili više kolona koje svojim vrednostima identifikuju red.

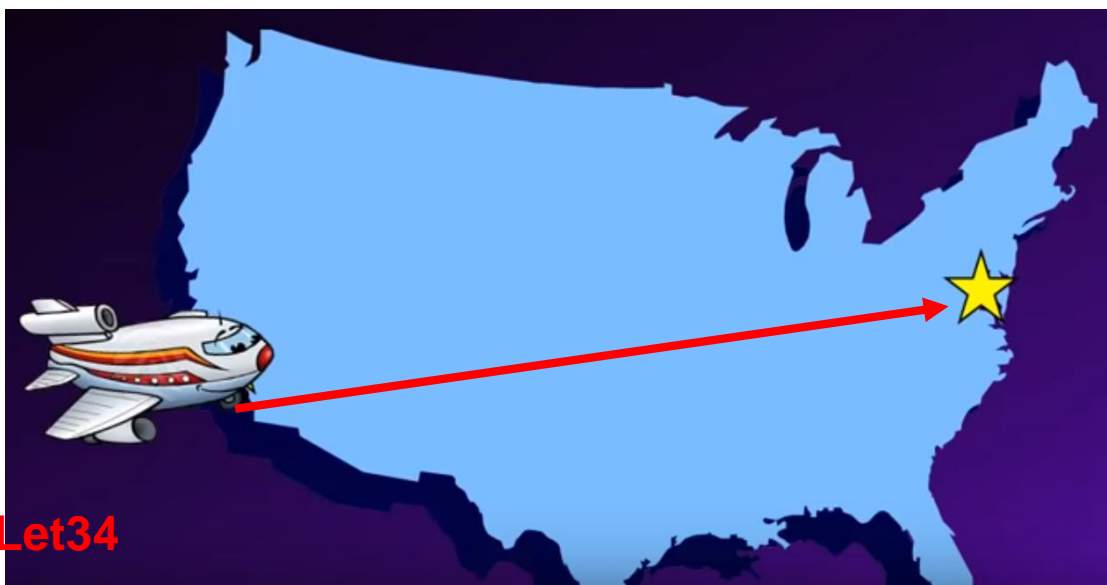
Jedinstveni ključ	Nejedinstven ključ
Vrednost je jedinstvena za svaki red.	Iste vrednosti mogu biti u nekoliko redova.
Jedinstveno i jednoznačno identifikuje red.	Identifikuje skup redova.

VRSTE KLJUČEVA



SLOŽENI KLJUČ

- **Složeni ključ** je ključ koji sačinjen od dva ili više atributa jedinstveno identifikuje svaki red u relaciji.
- Let34 svakog dana leti iz Beograda do Niša. Svakog dana realizuje se više različitih letova. Broj leta se ne menja bez obzira na datum leta. Kako identifikovati putnike za određeni let?



ACCOUNTS

BANK_NO	ACCT_NO	BALANCE	DATE_OPENED
104	75760	12,0050.00	21-OCT-89
104	77956	100.10	
105	89570	55,775.00	15-JAN-85
103	55890	15,001.85	10-MAR-91
105	75760	5.00	22-SEP-03

↑ ↑
Složeni ključ

Broj leta	Datum
Let34	01.10.2010.
Let35	01.10.2010.
Let34	02.10.2010.
Let36	01.10.2010.
Let37	05.10.2010.

KANDIDAT KLJUČ

- Kandidat ključ se zove „kandidat“ jer potencijalno može biti primarni ključ.
- Tabela može imati kolone koja mogu da budu kandidati za primarni ključ.
- “Kandidat” ključ je jedinstven ključ.
- Koji su sve kandidat ključevi za relaciju osoba?

EMPLOYEES

EMP_NO	LNAME	FNAME	DEPT_NO	PAYROLL_ID
100	SMITH	DANA	10	21215
310	ADAMS	TYLER	15	59877
210	CHEN	LAWRENCE	10	1101
405	GOMEZ	CARLOS	10	52
378	LOUNGANI	NEIL	22	90386

KANDIDAT KLJUČ

KANDIDAT KLJUČ

PRIMARNI KLJUČ

- ***Primarni ključ*** je kandidat ključ izabran da bude **jedinstveni identifikator redova u tabeli koja je relacija.**
- **Primarni ključ mora biti jedinstven.**
- **Svaka tabela mora da ima primarni ključ.**
- **Kolona (kolone) koje predstavlja primarni ključa ne sme biti bez vrednosti.**
- **Na osnovu primarnog ključa može se identifikovati svaki red u tabeli.**

PRIMARNI KLJUČ

Primarni ključ



RadnikID	Ime	Prezime
100	Miroslav	Mikić
101	Igor	Petrović
102	Lazar	Lazić
107	Ana	Simić

VEŠTAČKI - SUROGAT KLJUČ


- **Veštački ključ** predstavlja **jedinstvenu numeričku vrednost** koja se **dodaje u relaciju** kako bi bila definisana kao **primarni ključ**.
- Definiše se nova posebna kolona (atribut) specijalno za namenu primarnog ključa.
- Vrednosti veštačkog ključa nemaju značenje za korisnika
- Uglavno predstavljaju sakrivena polja u formama i izveštajima aplikacija.
- **Veštački ključ** se često koristi **umesto složenog ključa**.

RadnikID je primarni ključ, ali s obzirom da vrednosti ove kolone nemaju značenje za korisnika ovaj tip primarnog ključa zovemo surogat ključ.

RadnikID	Ime	Prezime
100	Miroslav	Mikić
101	Igor	Petrović
102	Lazar	Lazić
107	Ana	Simić

VEŠTAČKI - SUROGAT KLJUČ: PRIMER

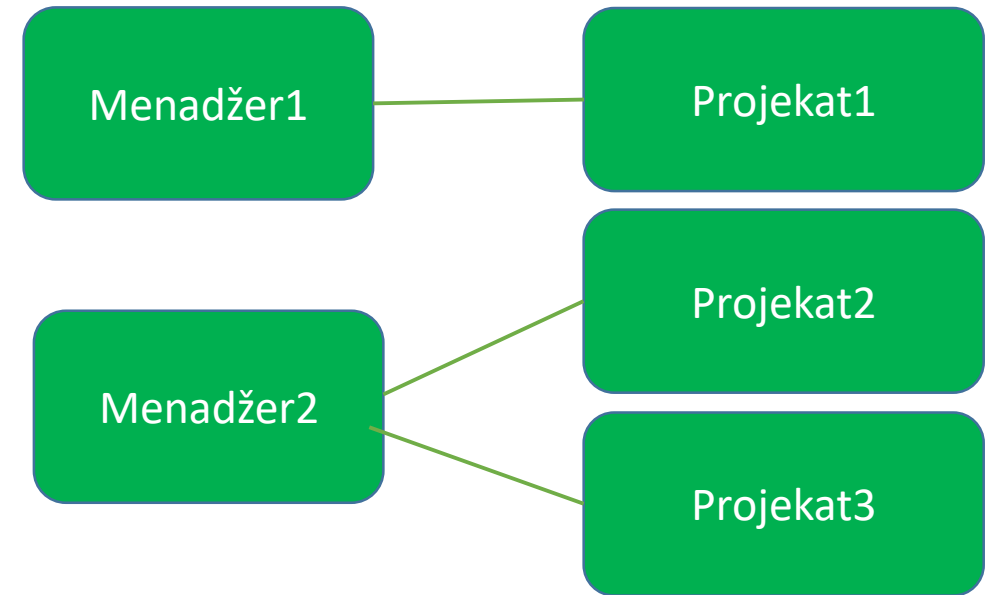
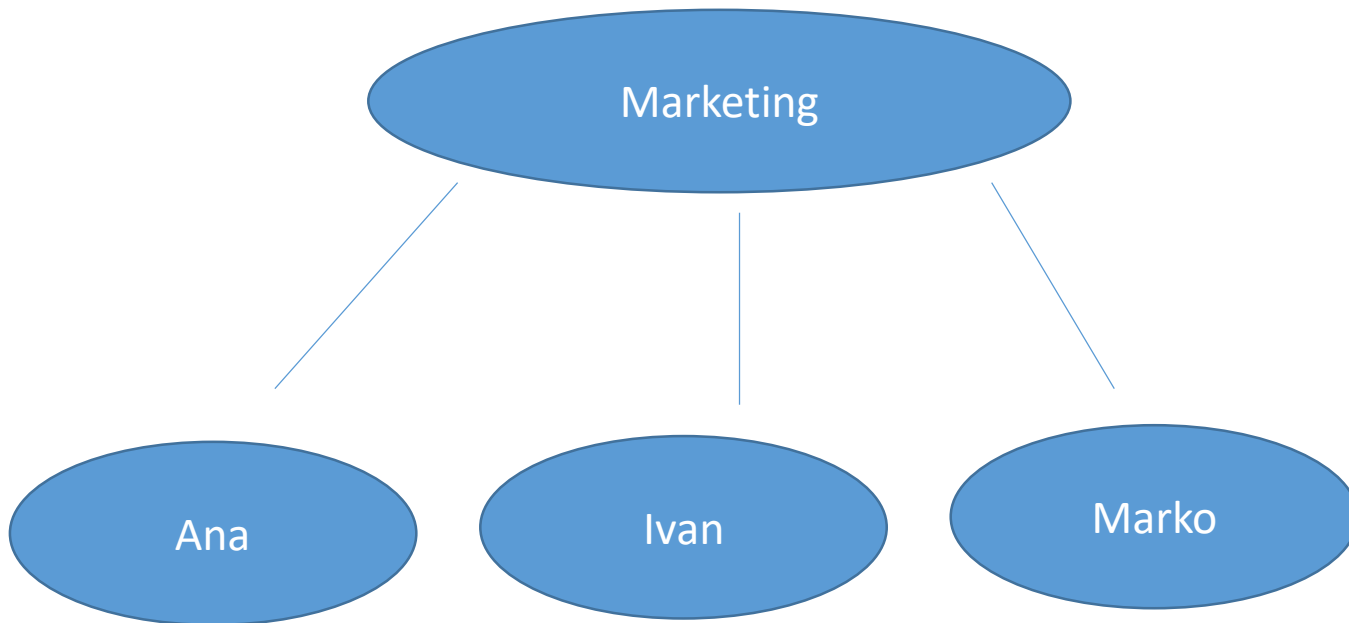
Veštački ključ kao primarni ključ



LetID	Broj leta	Datum
1	BG 36	03.10.2010
2	WN 45	03.10.2010.
3	WN 45	04.10.2010.
4	BN 12	04.10.2010.

VEZE IZMEĐU TABELA

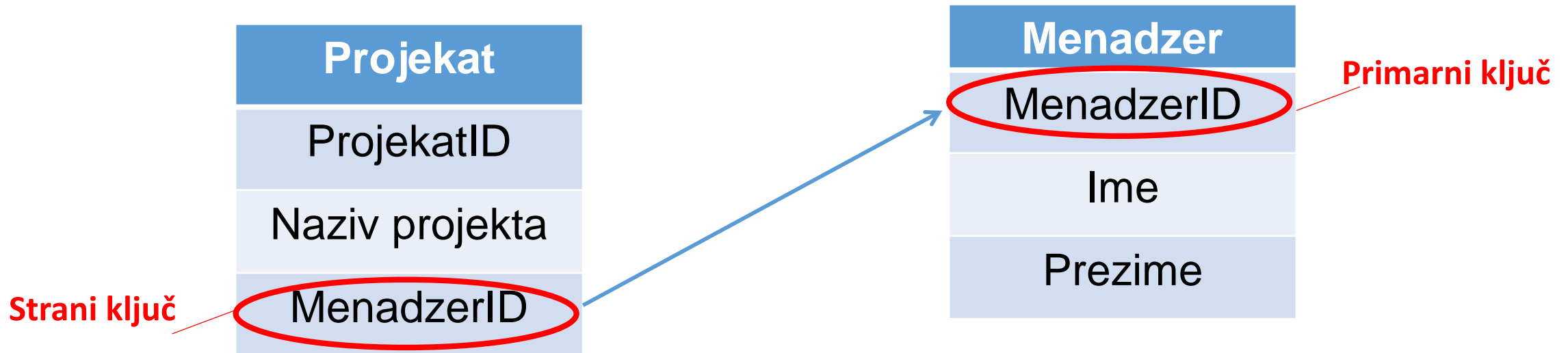
- Postoje veze između poslovnih objekata (konceptata).
- Primer:
 - Radnici rade u Odeljenju
 - Menadžer upravlja Projektima



STRANI KLJUČ

- ***Strani ključ*** je primarni ključ jedne tabele koji se dodaje u drugu tabelu sa ciljem povezivanja – uspostavljanje veze između tih tabela.
- Za **uspostavljanje veza** između tabela neophodno je **implementirati strani ključ**.
- **U tabeli u kojoj se dodaje kao kolona** kojom se uspostavlja veza predstavlja strani ključ.
- Omogućava realizaciju fizičke povezanosti između dve tabele.

STRANI KLJUČ - PRIMER



ProjekatId	Naziv projekta	Sredstva	Početak	MenadzerID
100	DB upgrade	8000	03.05.2010.	2
101	Novi mail server	98000	12.06.2010	2
102	3D printeri	15000	01.09.2010.	1

STRANI KLJUČ - PRIMER

Primarni ključ

Odeljenje
OdeljenjeID
Naziv odeljenja
Mesto

Radnik
RadnikID
Ime
Prezime
OdeljenjeID

Strani ključ

RadnikID	Ime	Prezime	Datum_zap	OdeljenjeID
1	Laza	Lazić	03.05.2001.	10
2	Ana	Savić	12.06.2002	10
3	Marko	Marković	01.09.2010.	20

REFERENCIJALNI INTEGRITET

- ***Referencijalni integritet*** znači da svaka vrednost stranog ključa jedne tabele mora da odgovara nekoj od vrednosti primarnog ključa druge tabele na koju ukazuje.
- ***Strani ključ*** omogućava uspostavljanje referencijalnog integriteta.
- Primenom referencijalnog integriteta DBMS utiče na kvalitet podataka u bazi podataka.

REFERENCIJALNI INTEGRITET - PRIMER

Tabela Projekat

ProjekatId	Naziv projekta	Sredstva	Početak	MenadzerID
100	DB upgrade	8000	03.05.2010.	2
101	Novi mail server	98000	12.06.2010	2
102	3D printeri	15000	01.09.2010.	1
103	Strukturno kabliranje	13456	23.09.2010.	5

Tabela Menadzer

MenadzerID	Ime
1	Ana
2	Luka

Referencijalni integritet znači da prilikom dodavanja novog zapisa u tabelu Projekat baza podataka proveriti da li postoji red u tabeli Menadzer označen vrednošću primarnog ključa MenadzerID=5

Ako ne postoji odgovarajući zapis u tabeli Menadzer baza podataka neće dozvoliti unos navedenog reda u tabelu Projekat!!

NULL vrednost

- ***Vrednost NULL*** označava da **ne postoji vrednost** ili da je **trenutno nepoznata vrednost atributa**.
- Vrednost NULL je **univerzalna vrednost** koja se može **definirati za attribute različitih domena**.
- NULL vrednost je **specijalno ograničenje domena atributa**.
- **To nije vrednost** koja je **predstavljena nulom, nizom nula, praznim prostorom, praznim stringom, tab karakterom**.

NULL vrednost

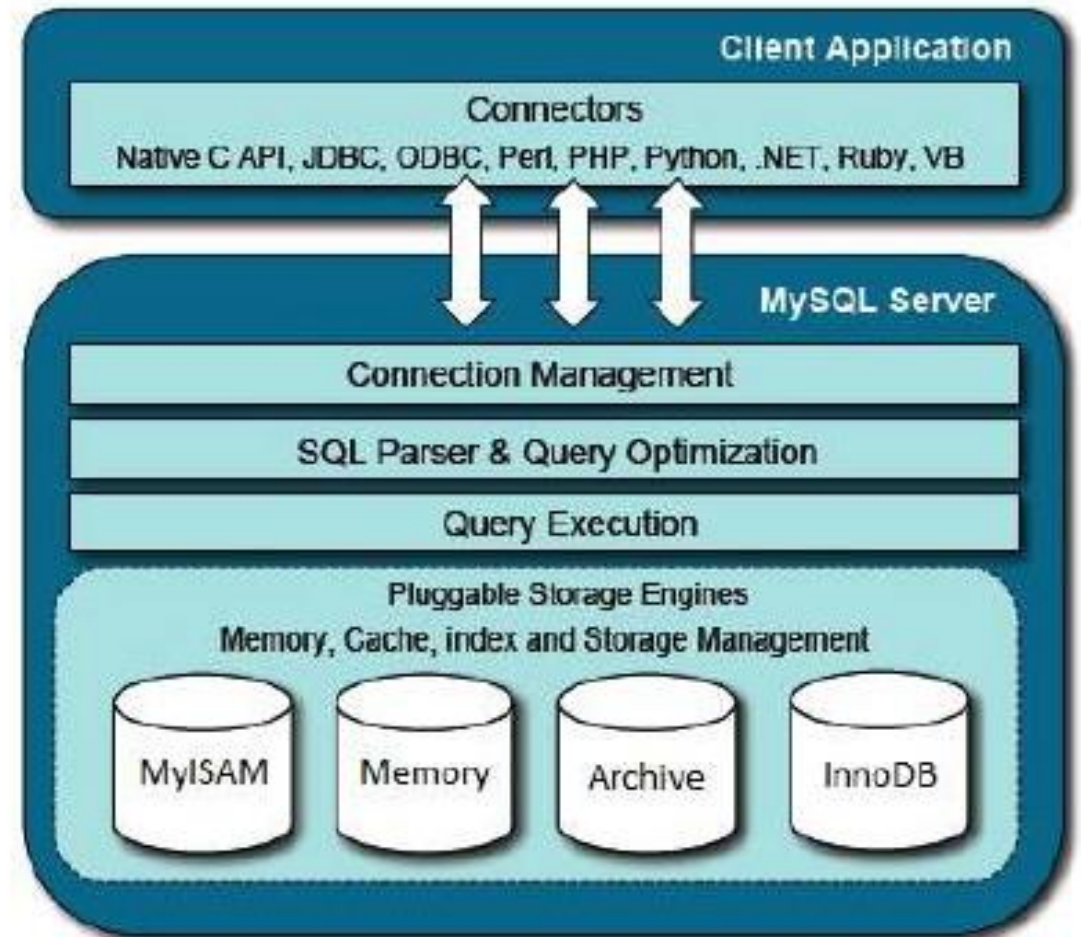
- Kod upisa studenata unosimo sve podatke izuzev podataka o telefonu, koji se može naknadno uneti jer student možda nema telefon
 - STUDENT (BrInd,Ime,Prezime,Telefon)
- Kod unosa podatka o datumu kada je knjiga vraćena u biblioteku jer ne znamo prilikom zaduženja kada će knjiga biti vraćena
 - ZADUZENJE (datum_zaduzenja_knjige, datum_vracanja_knjige)
- Kod unosa podatka o e-mail adresi zaposlenih
 - RADNIK (radnikID, ime, adresa, email)

NULL vrednost

- **Atributi definisani kao mandatorni ne mogu imati NULL vrednost!!!**
- Kada se u jednoj relaciji pojavljuje veliki broj podataka sa NULL vrednostima to ukazuje na:
 - loš model podataka
 - lošu strukturu relacija među podacima
- Rad sa zapisima u kojima postoje NULL vrednosti je posebno osetljiv:
 - poređenje i računске operacije atributa sa NULL vrednostima specifična.
- Prednost relacionog modela je u tome što pomaže da se dizajnira struktura podataka na takav način koja minimizira ili čak potpuno eliminiše postojanje praznih ćelija (NULL vrednost) u tabelama.

Elementi MySql okruženja

- Kada se baza podataka koristi u nekoj aplikaciji, bila ona web, desktop ili mobilna, tada sama aplikacija postaje klijent baze podataka.
- Da bi aplikacija koje samostalno razvijamo bila u stanju da se sporazume sa serverom MySQL-a, tj. da bi poznavala njegov jezik koristi konektor
- Konektori se razlikuju u zavisnosti od platforme i programskog jezika u kome će biti upotrebljavani.
- To su zapravo biblioteke napisane na izvornom jeziku za koji su namenjene, koje sadrže gotove metode za komunikaciju.



Sistemske baze podataka

Podrazumevano, nakon instalacije, mysql server sadrži tri baze podataka.

information_schema – rečnik podataka

- Omogućava pristup informacijama o podacima, tj. podacima o podacima. (metadata).

mysql

- sadrži sve serverske informacije. Korisnici, relacije, privilegije i slično.

performance_schema

- Podešavanja i statistike praćenja događaja niskog nivoa

MySql Workbench

MySql Workbench je oficijelni alat za upravljanje MySql serverom.

MySQL Workbench omogućava administratorima i programerima integrisano okruženje sa svim potrebnim alatima za:

- Dizajn i modelovanje baze podataka,
- SQL razvoj (zamena za MySQL Query Browser),
- Administriranje (zamena za MySQL Administrator)
- Migraciju baze.

Kreiranje konekcije

Pod konekcijom se podrazumeva prijavljivanje (logovanje) na MySQL server sa adekvatnim pristupnim podacima.

Setup New Connection

Connection Name: MySQL Type a name for the connection

Connection Method: Standard (TCP/IP) Method to use to connect to the RDBMS

Parameters SSL Advanced

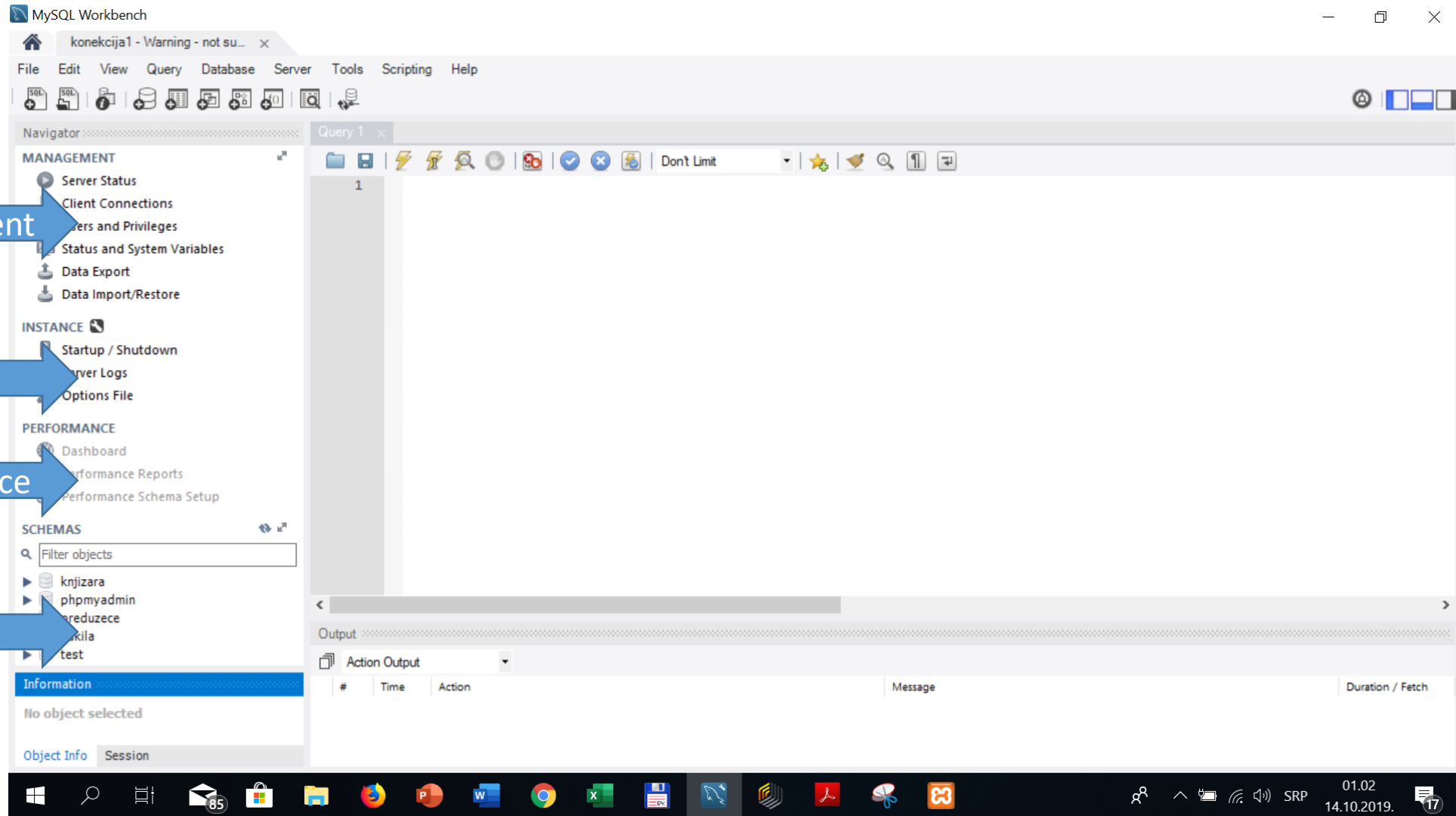
Hostname: 127.0.0.1 Port: 3306 Name or IP address of the server host. - TCP/IP port.

Username: root Name of the user to connect with.

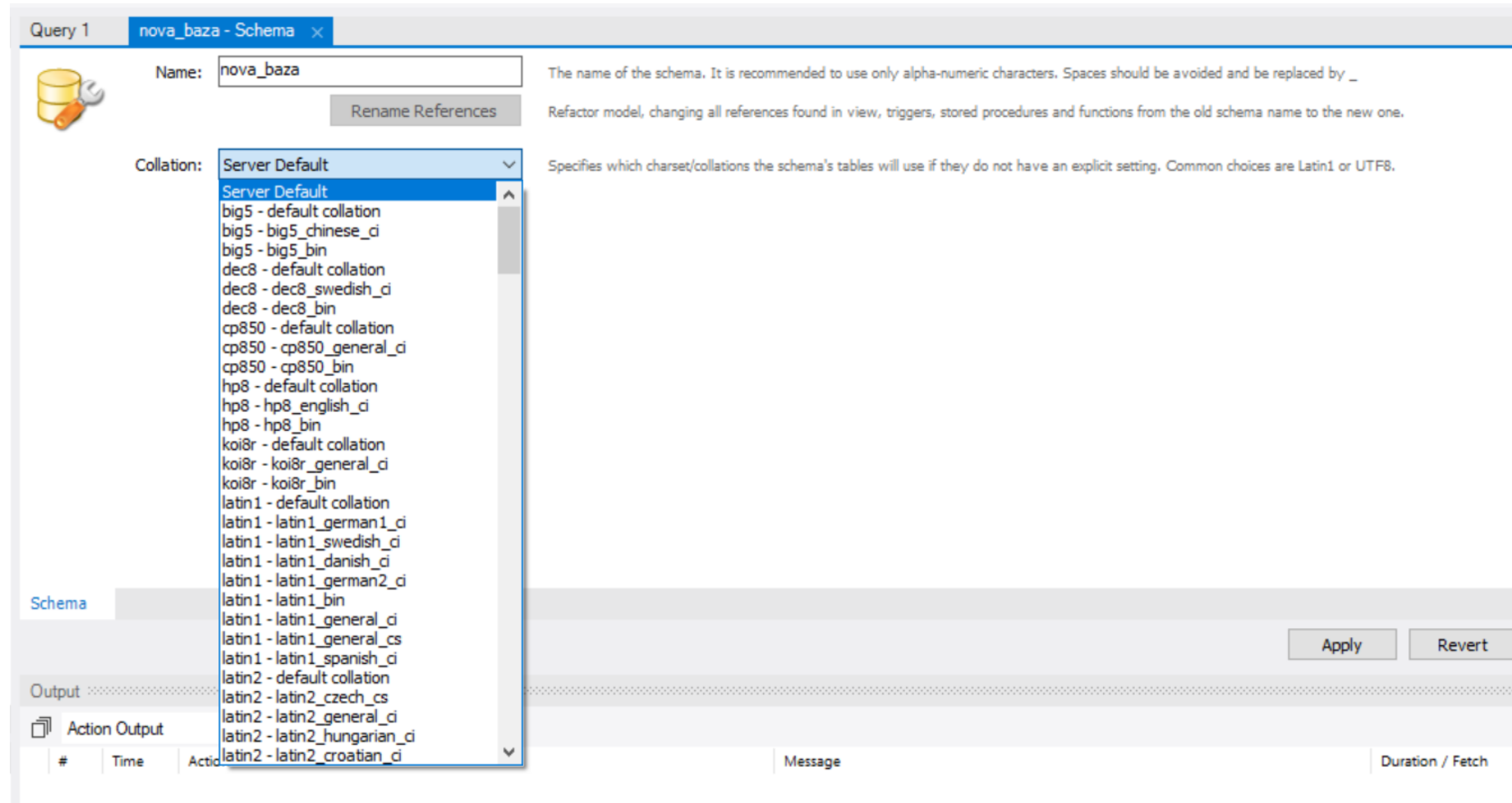
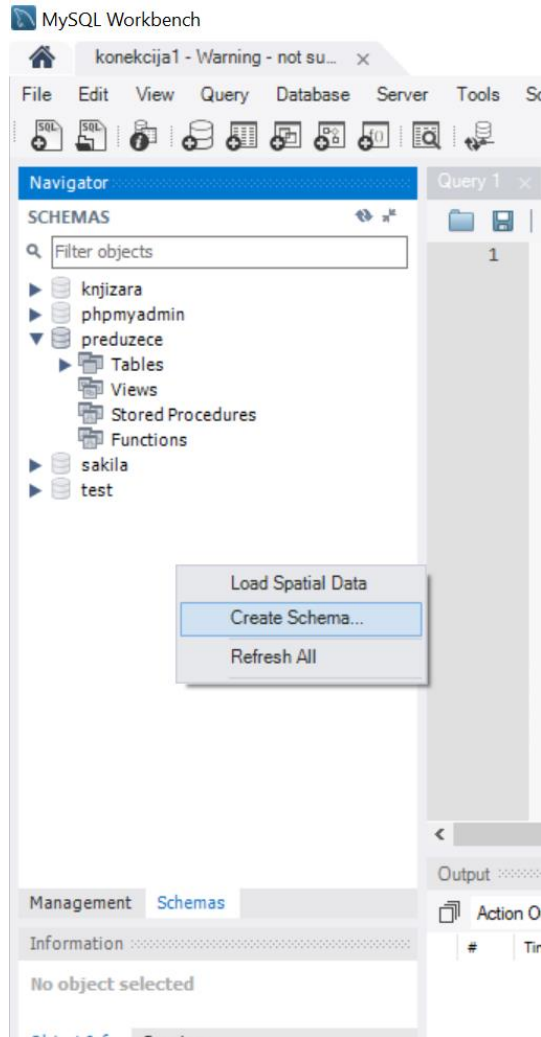
Password: The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

SQL editor



Kreiranje baze podataka

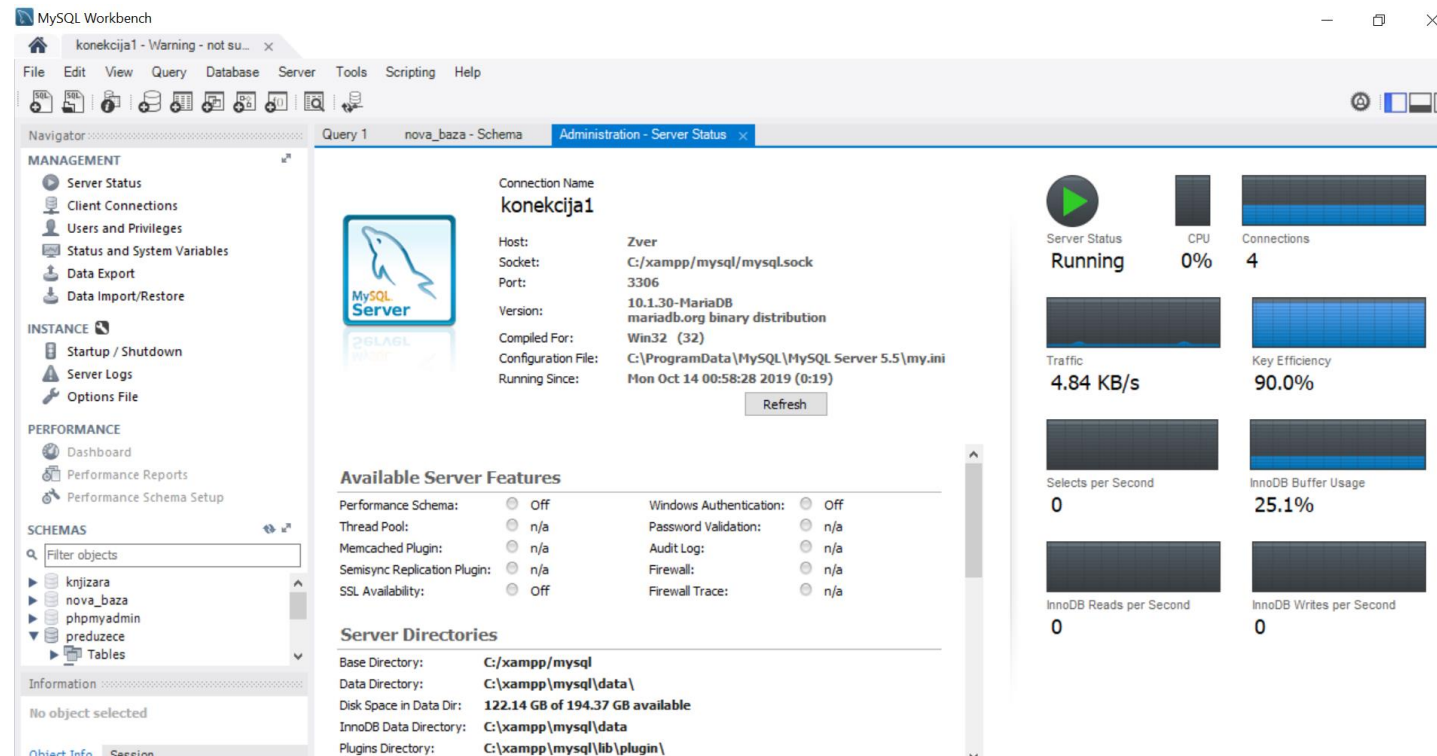


Character Sets and Collations

- Ako treba da uporedimo karaktere A i B. Najlakši način da se to uradi je poređenjem njihovih brožanih vrednosti, odnosno enkodinga.
- **Collation** upravo predstavlja pravila poređenje enkodinga.
- **Character setovi** i **kolacije** moraju odgovarati jedni drugima.
- U padajućem meniju, koji se može otvoriti u prozoru za dodavanje baze, moguće odabrati kombinaciju karakter seta i kolacije.
- Na krajevima kolacija možemo da primetimo određene sufikse. Sufiksi i njihovo značenje su sledeći:
 - ci – Case Insensitive
 - cs – Case Sensitive
 - bin – Binary
- Ostavljanjem opcije Server Default zapravo se za karakter set i kolaciju podešava onaj karakter set i kolacija koji su podešeni na nivou servera.

Fizičko smeštanje podataka baze

- Fajlovi baze, fizički su smešteni u direktorijumu koji je konfigurisan u fajlu my.ini, odnosno my.cnf
- Ovaj i slične podatke možemo dobiti direktnom analizom pomenutog fajla, ili opcijom Server Status u MySQL Workbenchu



The screenshot displays the MySQL Workbench interface, specifically the 'Administration - Server Status' window. The window is titled 'konekcija1 - Warning - not su...' and shows the following information:

- Connection Name:** konekcija1
- Host:** Zver
- Socket:** C:/xampp/mysql/mysql.sock
- Port:** 3306
- Version:** 10.1.30-MariaDB mariadb.org binary distribution
- Compiled For:** Win32 (32)
- Configuration File:** C:\ProgramData\MySQL\MySQL Server 5.5\my.ini
- Running Since:** Mon Oct 14 00:58:28 2019 (0:19)

Available Server Features:

Performance Schema:	<input type="radio"/> Off	Windows Authentication:	<input type="radio"/> Off
Thread Pool:	<input type="radio"/> n/a	Password Validation:	<input type="radio"/> n/a
Memcached Plugin:	<input type="radio"/> n/a	Audit Log:	<input type="radio"/> n/a
Semisync Replication Plugin:	<input type="radio"/> n/a	Firewall:	<input type="radio"/> n/a
SSL Availability:	<input type="radio"/> Off	Firewall Trace:	<input type="radio"/> n/a

Server Directories:

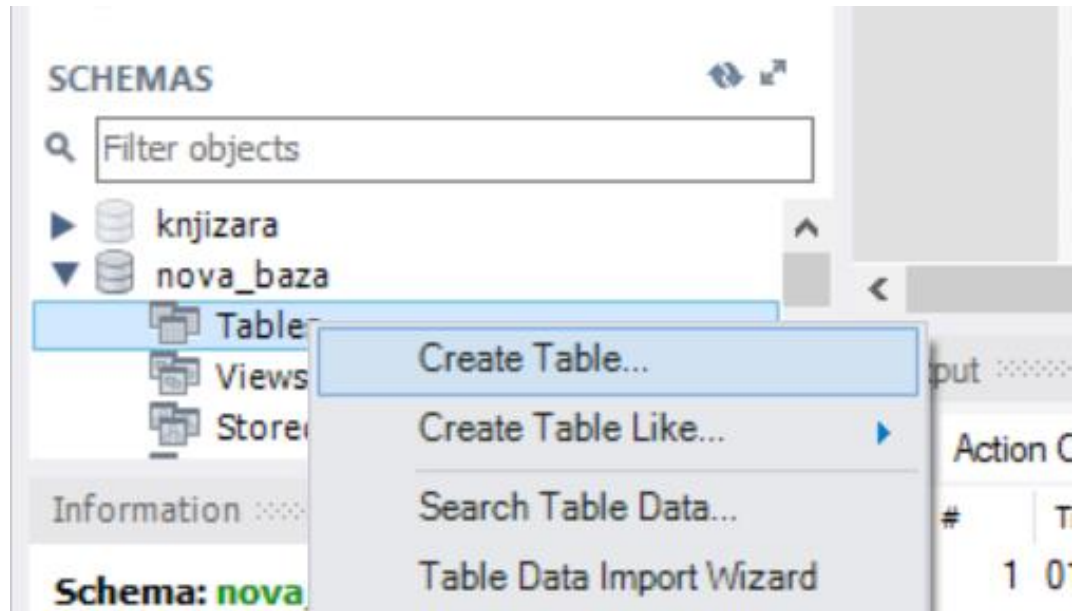
Base Directory:	C:/xampp/mysql
Data Directory:	C:\xampp\mysql\data\
Disk Space in Data Dir:	122.14 GB of 194.37 GB available
InnoDB Data Directory:	C:\xampp\mysql\data
Plugins Directory:	C:\xampp\mysql\lib\plugin\

Performance Metrics (Right Panel):

- Server Status: Running
- CPU: 0%
- Connections: 4
- Traffic: 4.84 KB/s
- Key Efficiency: 90.0%
- Selects per Second: 0
- InnoDB Buffer Usage: 25.1%
- InnoDB Reads per Second: 0
- InnoDB Writes per Second: 0

Kreiranje tabele

- Da bi baza imala smisla, mora imati tabele.
- Tabela se može kreirati pomoću GUI-ja ili SQL upita



Mehanizmi skladištenja

- MySQL može tretirati svaku tabelu na različit način
- Načini tretiranja tabela, nazivaju se **mehanizmi skladištenja** (*storage engines*)
- Mehanizam skladištenja je moguće zameniti naknadno.
- **Podrazumevani** mehanizam skladištenja je **InnoDB**
- Nemaju sve distribucije MySQL-a podršku za sve mehanizme skladištenja
- Spisak dostupnih mehanizama mozemo dobiti naredbom: *show engines*
- Najčešće korišćeni mehanizmi skladištenja su InnoDB i MyISAM