



## TEMATSKA CJELINA

Baze podataka

### Ciljevi nastavne cjeline



- Dati definiciju baze podataka
- Pokazati prednost sustava zasnovanih na bazama podataka u odnosu na klasični datotečni pristup
- Navesti zadatke sustava za upravljanje bazom podataka (DBMS)
- Definirati sustav baze podataka (Database system) i njegove komponente
- Pokazati organizaciju podataka u bazama podataka
- Istaknuti razliku između administratora podataka i administratora baze podataka
- Ukratko opisati arhitekturu klijent - server
- Opisati svojstva relacijskih baza podataka
- Definirati pojam entiteta

2

## Ciljevi nastavne cjeline



- ❑ Prikazati proces dizajniranja relacijske baze podataka
  - ☞ Konceptualni model
  - ☞ Dijagram Entiteti Veze (ERD)
    - ❑ Objasniti vrste veza u tom modelu
  - ☞ Detaljni ERD
  - ☞ Strukturalni prikaz relacijske baze podataka (tablice → zapisi i polja tablice)
- ❑ Navesti čemu služe primarni i strani ključevi u relacijskim bazama podataka
- ❑ Ukazati na važnost poslovnih pravila u procesu dizajniranja baze podataka
- ❑ Pokazati kako upitni jezici proširuju funkcionalnost relacijskih baza podataka
  - ☞ SQL
  - ☞ QBE

3

## PRIMJER



- ❑ Imate prodavaonicu računalne opreme
- ❑ U početku vašeg poslovanja prodavali ste samo brand name desktop računala i to samo od jednog proizvođača
- ❑ Situacija je prilično jasna i niste imali nikakvih problema s "ručnim" vođenjem evidencije vašeg poslovanja (prodaja i stanje na skladištu)
- ❑ S vremenom ste proširili svoje poslovanje i počeli nuditi sve računalne komponente, prijenosna računala, gotove konfiguracije (brand name i no name) itd.
- ❑ Lako vam je predočiti novonastalu situaciju i problem u kojem ste se našli ako bi nastavili "ručno" obrađivati podatke

4

## PRIMJER

---



- ❑ S vremenom ste planirali unaprijediti vaše poslovanje uvođenjem rješenja e – poslovanja itd.
- ❑ Razumno je da ste s klasičnog načina poslovanja prešli na poslovanje zasnovano na bazi podataka
- ❑ Prednosti koje ćete odmah uočiti su sljedeće:
  - ☞ Izbjegava se potreba za prekomjernim gomilanjem ispisa poslovanja
  - ☞ Povećava se brzina poslovanja – strojevi su puno brži od čovjeka ("Imate li SATA II disk WD s 16MB cachea kapaciteta 250GB na skladištu – treba mi 10 komada → prodavač koji radi u vašoj prodavaonici će na ovo pitanje u najboljem slučaju dati odgovor nakon nekoliko minuta, a često i nakon nekoliko desetaka minuta. S druge strane računalo će odgovor dati gotovo istog trena.

5

## PRIMJER

---



- ❑ Prednosti koje ćete odmah uočiti su sljedeće (nastavak):
  - ☞ Stalno su vam na raspolaganju informacije u realnom vremenu
  - ☞ Manje zamornog posla oko održavanja podataka – ponavljajuće rutine su puno bolje za računala
  - ☞ Zaštita podataka od neželjenog pristupa i gubitka važnih informacija
- ❑ Sada se samo još zamislite u situaciji u kojoj ste otvorili podružnice u Splitu, Puli, Rijeci, Zagrebu, Osijeku itd.
- ❑ Glavno skladište za sve prodavaonice se nalazi u Zagrebu, a svaka prodavaonica ima i vlastito skladište
- ❑ Sve prodavaonice će trebati pristupati istoj bazi podataka → višekorisnički sustav (**multi – user system**)

6

## PRIMJER

---



- ❑ Pristup bazi je centraliziran i kontroliran od strane administratora baze (**DBA – DataBase Administrator**)
- ❑ Ovaj pristup je u potpunosti različit od načina rada bez jedinstvene baze podataka, gdje svaka aplikacija koristi vlastite datoteke (na svom disku)
  - ☞ Sve to uzrokuje da su podaci razasuti, te se teško mogu sistematski kontrolirati
- ❑ Navedimo još neke specifične prednosti pristupa s bazama podataka:
  - ☞ Podaci se mogu lako dijeliti
  - ☞ Izbjegava se redundancija
  - ☞ Izbjegava se nekonzistentnost podataka

7

## PRIMJER

---



- ❑ Navedimo još neke specifične prednosti pristupa s bazama podataka (nastavak):
  - ☞ Osigurava se podrška transakcijama
  - ☞ Integritet podataka je zajamčen – ispravnost podataka
  - ☞ Sigurnost pristupa podacima
  - ☞ Mogućnost primjene novih standarda
  - ☞ Itd.

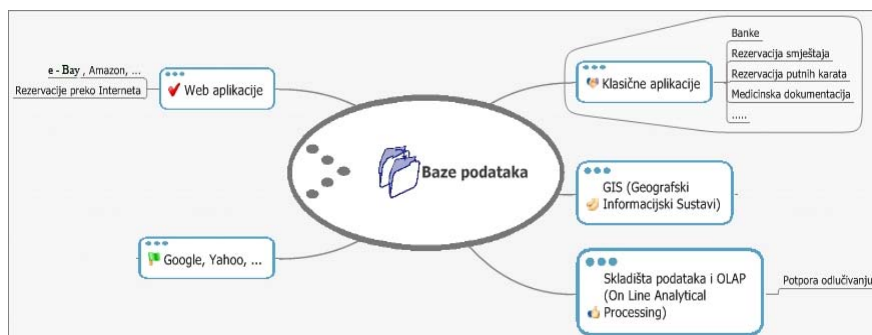
8



**Bodily exercise, when compulsory, does no harm to the body; but knowledge which is acquired under compulsion obtains no hold on the mind.**

Platon

## Područja primjene baza podataka



## Datoteke vs Baze podataka



- Uzmimo za primjer da imamo neku excel datoteku

Podružnica	Aranžman skijanje	Aranžman Nova godina	Aranžman Božić	Aranžman Uskrs	Aranžman Rafting
Podružnica 1	35	75	67	26	11
Podružnica 2	25	105	78	32	17
Podružnica 3	54	57	35	44	25
Podružnica 4	23	120	111	23	32
Podružnica 5	77	150	142	10	5

	Aranžman skijanje	Aranžman Nova godina	Aranžman Božić	Aranžman Uskrs	Aranžman Rafting
cijena aranžmana po osobi	200 €	450 €	390 €	270 €	50 €

11

## Datoteke vs Baze podataka



- Pošaljemo tu datoteku iz centrale Turističke agencije u podružnicu
- U centrali se promijeni cijena aranžmana po osobi za skijanje i potom se datoteka pohrani pod istim imenom i na istu lokaciju

	Aranžman skijanje	Aranžman Nova godina	Aranžman Božić	Aranžman Uskrs	Aranžman Rafting
cijena aranžmana po osobi	200 €	450 €	390 €	270 €	50 €

Ovo se promijeni na 250€ i potom → CTRL + S

12

## Datoteke vs Baze podataka



- Na računalu u centrali sada imamo:

	Aranžman skijanje	Aranžman Nova godina	Aranžman Božić	Aranžman Uskrs	Aranžman Rafting
cijena aranžmana po osobi	250 €	450 €	390 €	270 €	50 €

- Na računalu u podružnici i dalje imamo:

	Aranžman skijanje	Aranžman Nova godina	Aranžman Božić	Aranžman Uskrs	Aranžman Rafting
cijena aranžmana po osobi	200 €	450 €	390 €	270 €	50 €

13

## Datoteke vs Baze podataka



- Problem → neažurirani podaci
- Mogli bi ga “riješiti” ponovnim mailanjem → očigledno nije dobro rješenje
- Ovo je samo jedan od primjera nedostatka datotečnog pristupa u odnosu na pristup temeljen na bazama podataka
- Drugi nedostatak kojeg lako možemo predočiti je problem višestrukih unosa → redundancija podataka
- Treći lako uočljiv nedostatak → pronalaženje podataka
- Pristup temeljen na bazama podataka eliminira mnoge probleme koji se javljaju kod klasičnog načina pohranjivanja podataka u odvojene datoteke

14

## Baze podataka podržane računalom



- ❑ Nas isključivo zanimaju baze podataka podržane računalom
- ❑ Postojale su i još uvijek postoje baze podataka bez računalne potpore
  - ☞ Klasični telefonski imenik
  - ☞ Kartotečni sustav u Gradskoj knjižnici
  - ☞ Red vožnje autobusa
  - ☞ Itd.
- ❑ Osnovno svojstvo podataka pohranjenih u bazu podataka na nekom računalu je njihova "trajnost" u smislu da se nakon konfiguriranja baze preko DBMS – a osnovni podaci naknadno mogu promijeniti samo eksplicitnim zahtjevom DBMS – a, a nikako kao rezultat rada aplikacije koja pristupa toj bazi podataka

15

## Baze podataka podržane računalom



**Baza podataka** – je skup podataka, logički povezanih u jedinstvenu cjelinu koje koriste aplikacije u nekoj organizaciji.

**DataBase Management System (DBMS)** – skup programa koji omogućavaju kreiranje i održavanje baze podataka, te kontrolu pristupa podacima.

- ❑ DBMS ne upravljaju samo jednom bazom, nego je češće riječ o nekoliko baza podataka

16



## DBMS



- DataBase Management system je software opće namjene koji ima višestruku ulogu:
  - ☞ Definiranje baze podataka – proces u kojem se specificiraju tipovi podataka, struktura baze, te ograničenja nad podacima
  - ☞ Konstrukcija baze podatka – spremanje podataka na odabranu lokaciju (server) koja se kontrolira preko DBMS
  - ☞ Manipuliranje bazom podataka – postavljanje upita prema bazi kako bi dobili željene informacije, ažuriranje baze podataka kako bi se pratile promijene u stvarnom svijetu i kreiranje izvješća iz postojećih podataka
  - ☞ Upravlja istovremenim pristupima podacima kako bi se spriječili simultane promijene koje su u konfliktu
  - ☞ Upravljanje transakcijama po principu sve ili ništa (**all – or – nothing**)  
→ uspješna transakcija vrši promjenu, a neuspješna podatke ne mijenja
  - ☞ Kreira sigurnosne kopije (backup)
  - ☞ Posjeduje sigurnosne mehanizme kojima štiti od neovlaštenog pristupa podacima

17

## Database system



- Zajednički naziv za bazu podataka i DBMS je sustav baze podataka (**database system**)
- Taj sustav minimizira sljedeće probleme:
  - ☞ Redudanciju podataka – nema višestrukog unosa istih podataka (podaci se unose samo jednom na jednom mjestu)
  - ☞ Izolaciju podataka – jedna aplikacija ne može pristupiti podacima dodijeljenim nekoj drugoj aplikaciji
  - ☞ Nekonzistentnost podataka – mogućnost da se kopije podataka ne slažu
- Sustav baze podataka maksimizira sljedeće:
  - ☞ Sigurnost podataka
  - ☞ Integritet podataka – ograničenja nad podacima (npr. broj bankovne kartice ne može biti tekstualnog tipa)
  - ☞ Nezavisnost podataka – aplikacija i podaci su međusobno nezavisni (Različite aplikacije mogu pristupiti istim podacima)

18

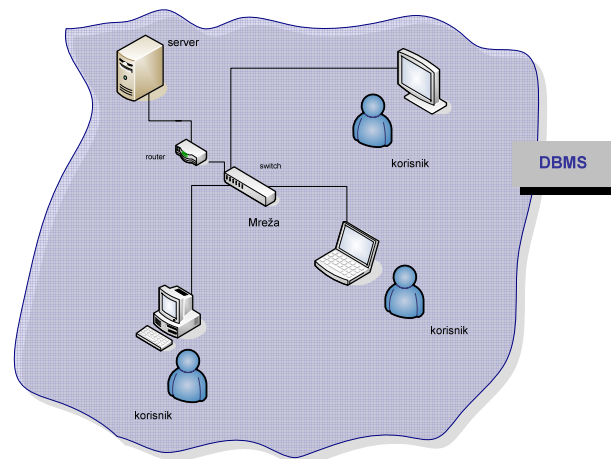
## Database system



- ❑ Možemo reći da se radi o računalnom sustavu za pohranu i manipuliranje podacima.
- ❑ Takvi sustavi omogućuju pohranu podataka i dopuštanje pristupa tim podacima korisnicima (najčešće registriranim) na način da ih mogu po potrebi dohvaćati i mijenjati – ovisno o ovlastima koje su im dodijeljene
- ❑ Database system se sastoji od četiri osnovne komponente:
  - ☞ hardwarea
  - ☞ softwarea
  - ☞ podataka
  - ☞ korisnika

19

## Database system – pojednostavljeni prikaz



20

## Database system – hardware i software



- ❑ **Hardware**
  - ☞ Pod hardverskim komponentama podrazumijevamo:
    - ❑ servere i klijentska računala
    - ❑ mrežnu opremu
- ❑ **Software**
- ❑ Softversku komponentu predstavlja
  - ☞ DBMS (DataBase Management System)
  - ☞ samostojeće aplikacije,
  - ☞ korisne alate,
  - ☞ alate za razvoj aplikacija,
  - ☞ alate za izvješća
  - ☞ najvažniji u poslovnim aplikacijama TP - Transaction manager

21

## Database system – podaci



- ❑ **Podaci**
  - ☞ Prema pristupu podacima razlikujemo:
    - ❑ single – user system → sustav u kojem samo jedan korisnik može pristupiti bazi
    - ❑ multi – user system → je sustav u kojem veći broj korisnika istovremeno može pristupiti bazi
  - ☞ U praksi je uobičajen ovaj drugi pristup, te ćemo se na njega i ograničiti
  - ☞ U praksi je ponekad zgodnije podatke podijeliti u više različitih baza

22

## Database system – korisnici



### □ Korisnici

- ☞ Razlikujemo tri različita tipa korisnika:
  - Programeri aplikacija – koji razvijaju aplikacije za baze u nekom od viših programskih jezika → COBOL, PL/I, C++, Java
  - Krajnji korisnici – pristupaju bazi interaktivno preko interfeasa u vidu formi ili komandne linije (upitni jezici)
  - Administratori baze (DBA – DataBase Administrators) – dodjeljuju ovlasti i prate "promet" na bazi

23

## Administratori baze i Administratori podataka



- Princip centralizirane kontrole zasniva se na postojanju administratora baze podataka.
  - ☞ U biti može se raditi i o više osoba koje imaju administratorske ovlasti
- Podaci su za svaku organizaciju najvrijednija imovina, pa je potrebno da postoje osobe koje ih razumiju, kao i potreba organizacije za njima
  - ☞ Radi se o osobama na najvišim nivoima organizacije (**senior management level**) – Administratori podataka (Data Administrators)
  - ☞ Njihov posao je da odluče koji podaci će se pohranjivati u bazi, a potom da uvedu pravila za njihovo održavanje i korištenje

24

## Administratori baze i Administratori podataka

---



- ❑ Administrator podataka je menadžer i treba ga razlikovati od administratora baze podataka (DBA) koji je čisto tehničar koji osigurava centralizaciju baze isključivo na tehničkom nivou
- ❑ Tehničara odgovornog za implementiranjem podataka i pravila koje je odredio DA nazivamo administrator baze podataka (**DBA – Database Administrator**)
- ❑ DBA će često imati pomoć sistem programera i drugog tehničkog osoblja
- ❑ U praksi funkciju DBA vrši tim stručnjaka, a ne samo jedna osoba

25

## Administratori baze i Administratori podataka

---



- ❑ Zbog jednostavnosti ćemo pisati DBA u jedini, a na umu ćemo imati činjenicu da se iza toga krije tim IT stručnjaka

**DBA (Database Administrator) predstavlja skupinu IT stručnjaka koji s tehničke strane implementiraju zamisli administratora podataka (DA – Data Administrator)**

26

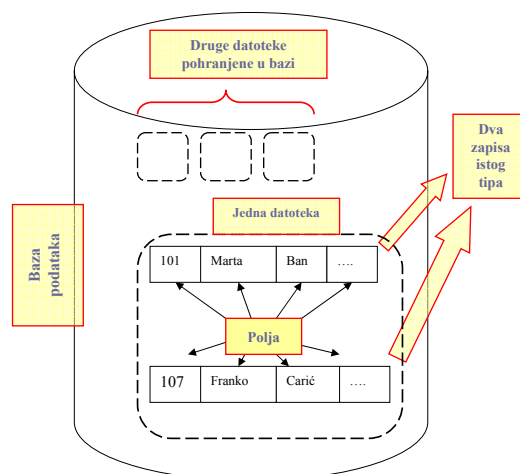
## Organizacija podataka u bazama



- ❑ **Polje (Field)** je najmanja jedinica pohrane podataka u bazi. Svaka baza će posjedovati veliki broj predstavnika svakog polja ili čak veliki broj tipova polja. Na primjer u bazi će se pohranjivati podaci o studentima i svako polje "matični broj" bit će različite vrijednosti, a istog tipa.
- ❑ **Zapis (Record)** je skup međusobno zavisnih polja. Za našeg studenta ime, prezime, mjesto rođenja, godina rođenja, matični broj, ... itd. predstavlja jedan zapis. Ovdje isto možemo govoriti o tipu zapisa. Isti tip zapisa sastoji se od istih polja u istom redoslijedu.
- ❑ **Datoteka baze (Database file)** je skup svih zapisa istog tipa. Zbog jednostavnosti smatramo da se svaka datoteka baze sastoji samo od jednog tipa zapisa.

27

## Organizacija podataka u bazama



28

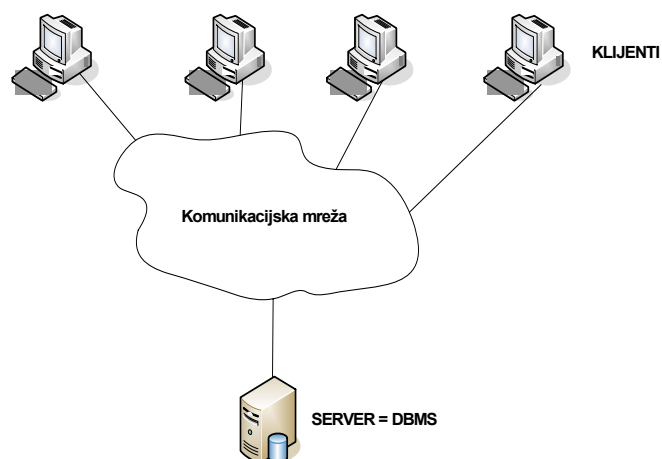
## Arhitektura klijent - server



- ❑ Serverski dio se često naziva i **back end**, dok se klijentski dio naziva **front end**.
- ❑ Server je u biti sam DBMS – m i podržava sve funkcije koje smo naveli pod funkcijama DBMS –a → definiranje podataka, manipuliranje podacima, sigurnost podaka i njihov integritet itd.
- ❑ Klijenti su razne korisničke aplikacije čiji se rad oslanja na DBMS – m.
  - ☞ aplikacije koje su razvijene od strane korisnika
  - ☞ ugrađenim aplikacijama (aplikacijama koje se nude s DBMS – om ili pak koje razvijaju neke druge profesionalne tvrtke)

29

## Arhitektura klijent - server



30

## Relacijske baze podataka



- ❑ Pravi procvat područja primjene baza podataka započinje 70 – tih godina prošlog stoljeća od uvođenja **relacijskog modela** baza podataka
- ❑ Osnovna svojstva koja taj model posjeduje su sljedeća:
  - ☞ Podaci se korisniku predstavljaju kao zasebni entiteti
  - ☞ Svaki entitet je opisan svojstvima
  - ☞ Entiteti se prikazuju tablicama
  - ☞ Entiteti su međusobno povezani – kažemo da postoje relacije među entitetima
  - ☞ Podaci su jednostavno dohvatljivi – SQL

31

## Relacijske baze podataka



- ❑ Osnovna ideja ovog modela leži u činjenici da korisnik ne može unaprijed znati sve moguće načine korištenja podataka u bazi → ne postoje predefimirani putovi kretanja kroz podatke
- ❑ Upitni jezici – prije svega tu se misli na **SQL (Structure Query Language)**, operiraju sa skupom zapisa, a ne samo s jednim u danom trenutku (što je karakteristika hijerarhijskog i mrežnog modela)
- ❑ Npr. želimo izdvojiti sve kupce koji nisu podmirili dugovanja za tekuću godinu...
- ❑ Podaci se, kao što je već rečeno, u ovom modelu na strukturnom nivou prikazuju kao dvodimenzionalne tablice uz osiguranje fizičke i logičke nezavisnosti

32



## Drugi tipovi baza podataka



- ❑ Hijerarhijske baze podataka
  - ❑ Mrežne baze podataka
  - ❑ Objektno orijentirane baze podataka
- } Više se ne koriste
- ☞ Smatraju se pogodnom platformom za multimedijske Web aplikacije u skoroj budućnosti
    - ❑ Enkapsulacija (učahurivanje) → omogućuje da se lakše radi s kompleksnim tipovima podataka (grafika, slike, tekst, zvuk, video) nego što to mogu klasične relacijske baze podataka
    - ❑ Inheritance (nasljeđivanje) → mogućnost kreiranja novog objekta koji nasljeđuje sva dobra svojstva roditelja, a istovremeno ih nadopunjava novima
      - Moguće je kreirati objekt koji sadrži dizajn proizvoda i potom ga pohraniti u objektno orijentiranu bazu podataka
      - Novi dizajn se dobije nadopunjavanjem ili izmjenom svojstava postojećeg objekta

33

## Baze znanja



- ❑ Baze znanja sadrže znanje prikazano u različitim oblicima:
  - ☞ Pravila
  - ☞ Semantičke mreže
  - ☞ Scenariji
- ❑ Upotreba tako prikazanog znanja → pomoću različitih mehanizama zaključivanja
- ❑ Primjeri:
  - ☞ Financijska predviđanja
  - ☞ Dijagnoza uzroka kvara
  - ☞ Planiranje akcija
  - ☞ Itd.

34

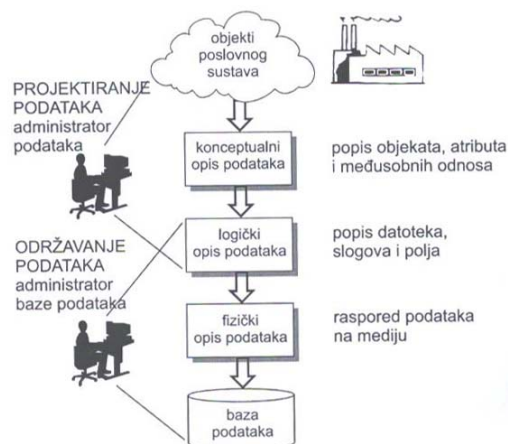
## Dizajniranje relacijske baze podataka



- Dizajniranje baze podataka podrazumijeva:
  - ▣ Projektiranje podataka
  - ▣ Održavanje podataka
- Projektiranje podataka → konceptualno i logičko modeliranje podataka → vrše ga **administratori podataka**
- Održavanje podataka → fizičko modeliranje podataka, a potom fizička uspostava i održavanje baze podataka → vrše **administratori baze podataka**

35

## Dizajniranje relacijske baze podataka



Izvor: V. Čerić, M. Varga - Informacijska tehnologija u poslovanju

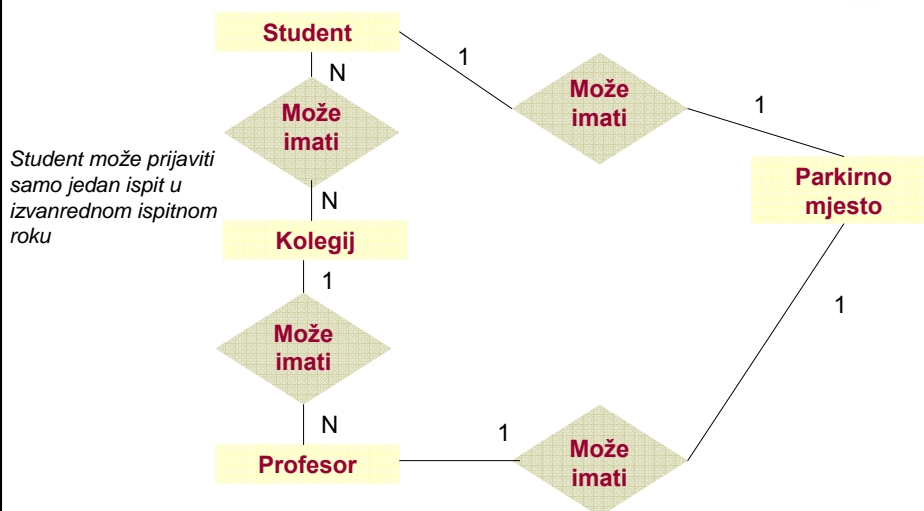
## Dizajniranje relacijske baze podataka – konceptualni model



- Polazi se od entiteta
  - ▣ Bilo koja stvar (stvarna ili apstraktna) u poslovnom sustavu o kojoj možemo bilježiti podatke
  - ▣ Primjeri:
    - Osobe
    - Kupci
    - Poduzeća
    - Odjeli
    - Proizvodi
    - Računi
    - Narudžbe
    - Studenti
    - Profesori
    - Kolegiji

37

## Dizajniranje relacijske baze podataka – konceptualni model

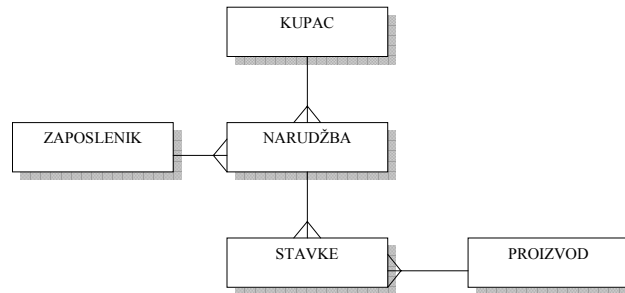


38

## Dizajniranje relacijske baze podataka



- Iz konceptualnog modela slijedi model entiteta – veze (definira sve entitete i veze među njima)



Ovakav način prikaza podataka naziva se **dijagram ili model entiteta – veze** → **ERD (Entity – Relationship Diagram)**

39

## Entiteti



- Entitete smo već susreli → ponovimo što bi bio entitet?
- U ovom primjeru entiteti su Kupac, Narudžba, Zaposlenik, Stavke i Proizvod
  - ☞ Sve što postoji u stvarnom ili virtualnom svijetu o čemu možemo bilježiti neke podatke
  - ☞ **Entitet može biti** osoba, mjesto, stvar ili događaj
  - ☞ Svaki entitet je opisan određenim brojem **atributa** (svojstava)

**ENTITET** – je dio stvarnog ili apstraktnog svijeta koji je opisan određenim brojem svojstava koje predstavljamo podacima (osoba, ustanova, predmet, dokument,...)

40

## Veze (relacije)



- Veze između svih entiteta su iste
  - ▣ jedan kupac može imati više narudžbi;
  - ▣ jedan zaposlenik može obraditi veći broj narudžbi,
  - ▣ na svakoj narudžbi može biti više stavki,
  - ▣ svaki proizvod se može pojaviti više puta kao stavka.
- Ovakav tip veza naziva se jedan – prema više (**one – to – many**).
  - ▣ Jednostruka linija označava stranu jedan
  - ▣ trokraka linija označava stranu više.
    - U žargonu se ta trokraka linija često naziva crow's foot.

41

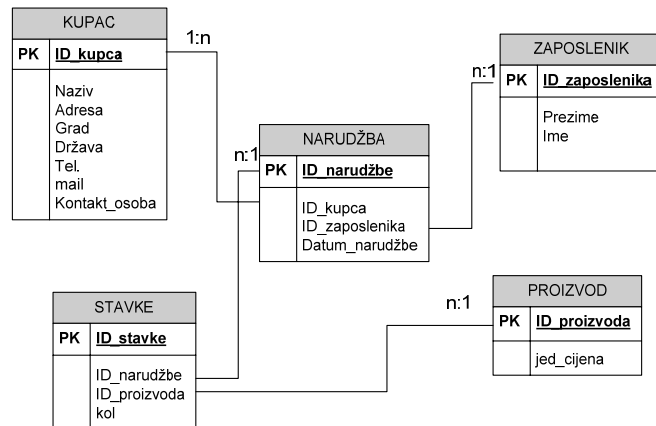
## Veze (relacije)



- Postoje još dvije vrste veza u relacijskom modelu, a to su:
  - ▣ više – prema – više (**many – to – many**) i
  - ▣ veza jedan – prema – jedan (**one – to – one**)
- Oba ova tipa veza se nastoje izbjeći, pa kažemo da želimo relacijski model baze podataka u kojem postoje samo one – to – many tipovi veza

42

## Detaljni dijagram entiteti - veze



43

## Strukturni model – dvodimenzionalne tablice



TABLICA KUPAC							
ID_kupca	Naziv	Adresa	Grad	Država	Tel.	mail	Kontakt_osoba
10001	Frki doo.	Blavorska bb	Gromila gomja	Deluzija	447 33 456789	frks@frki.del	Frksan Novodrp
10002	Karić & comp.	Astralska 33	Skrapinjan	Jamijana	567 35 897323	nu_me@kaco.ast	Prawko Beuzd

TABLICA ZAPOSLENIK		
ID_zaposlenika	Prezime	Ime
100	Centić	Zočmar
200	Ostav	Svenam
300	Preuzman	Arestina

TABLICA NARUDŽBA			
ID_narudžbe	ID_kupca	ID_zaposlenika	Datum_narudžbe
10	10001	100	11.8.2006
11	10001	100	15.9.2006
12	10001	200	10.10.2006
13	10002	200	11.10.2006
14	10002	300	17.10.2006

44

## Zapisi i polja



TABLICA NARUDŽBA				POLJE
ID_narudžbe	ID_kupca	ID_zaposlenika	Datum_narudžbe	
10	10001	100	11.8.2006	
11	10001	100	15.9.2006	
12	10001	200	10.10.2006	
13	10002	200	11.10.2006	
ZAPIS	14	10002	300	17.10.2006

- ❑ Redci u tablicama baze predstavljaju **zapise** (slogove ili records), a **stupci polja** (fields)
- ❑ Redci predstavljaju informacije, a stupci podatke

45

## Primarni i strani ključevi



- ❑ Pretpostavimo da imamo dva kupca
  - ☞ istog imena i prezimena
  - ☞ koji se nalaze u istom gradu na istoj adresi
- ❑ Istina da je mala vjerojatnost za tako nešto, ali zamislimo takav slučaj čisto za ilustraciju
- ❑ Kako će se oni razlikovati, tj. kako se osigurava jedinstvenost zapisa u relacijskom modelu?
- ❑ Iz priloženih tablica vidimo da svaki kupac ima jedinstveni identifikacijski broj čime se izbjegava zamjena informacija

46

## Primarni i strani ključevi



- ❑ Polja koja osiguravaju jedinstvenost zapisa nazivaju se **primarni ključevi (primary keys)**
  - ☞ poželjno je da svaka tablica baze ima barem jedan primarni ključ
- ❑ Pravilnim planiranjem i dizajniranjem baze može se osigurati da svaka tablica sadrži isključivo po jedan primarni ključ
- ❑ Povezivanje između različitih tablica baze ostvaruje upravo preko tih ključeva
- ❑ Kada se primarni ključ pojavi kao polje u nekoj drugoj tablici u cilju ostvarivanja relacije s primarnom tablicom nazivamo ga **strani ključ** ili eng. **foreign key**
  - ☞ Broj stranih ključeva nije ničim ograničen

47

## Poslovna pravila



- ❑ Poslovna pravila (**Bussines Rules**) su pravila, zakoni, procedure ili standardi koje je u svoj rad ugradila organizacija
- ❑ Poslovna pravila igraju vrlo važnu ulogu u kasnijem dizajniranju baze, jer najčešće određuju neka ograničenja nad podacima.
- ❑ Za primjer, korisnik tekućeg računa ima dozvoljeni minus u visini jedne plaće
  - ☞ kada prekorači taj iznos više ne može vršiti transakcije podizanja novca s tog računa sve dok se ne uplati odgovarajući iznos – da pojednostavnimo dok ne sjedeća plaća.
- ❑ Poslovna pravila se ni ne prikazuju ERD - om, već dolaze kao popratna dokumentacija uz njega
- ❑ Poslovna pravila i međupodaci su uključeni u konceptijski model

48



## Upitni jezici - Query Languages



- Proširuju mogućnosti relacijskih baza podataka – tek uz njih su relacijske baze podataka dobile na važnosti
- Povezivanjem podataka iz nekoliko tablica ili izvlačenjem podataka iz jedne tablice uz zadovoljenje jednog ili više uvjeta dobivaju se nove informacije koje se eksplicite nisu nigdje unosile
  - ☞ **Structured query language (SQL)** je najpopularniji upitni jezik koji se koristi za dobivanje dodatnih informacija
  - ☞ **Query by example (QBE)** je mreža ili predložak kojeg korisnik popunjava kako bi dobio uzorak ili opis željenih informacija

49

## Virtualne baze podataka - Virtual Databases



- Aplikacije koje osiguravaju upravljanje i manipulaciju podacima iz većeg broja baza podataka na način kao da se radi samo o jednoj velikoj bazi podataka
- Dobre strane virtualnih baza podataka:
  - ☞ Niži troškovi razvoja
  - ☞ Brže vrijeme razvoja
  - ☞ Manje posla oko održavanja
  - ☞ Jedinstvena točka unosa podataka

50

## Sažetak



- Upravljanje podacima → sustavi zasnovani na bazama podataka
- Datotečni način pohrane podataka ima čitav niz nedostataka u odnosu na sustav pohrane podatak u baze podataka
- Baza podataka – skup međusobno logički povezanih podataka o nekoj temi, predmetu i sl.
- Tipovi baza podataka (povijesni razvoj)
  - ☞ Hijerarhijske
  - ☞ Mrežne
  - ☞ Relacijske
  - ☞ Objektno orijentirane
- Danas su najzastupljenije relacijske baze podataka, dok objektno orijentirane imaju budućnost u Web aplikacijama
  - ☞ Najpoznatiji proizvođači relacijskih DBMS Oracle i IBM DB2 u njih ugrađuju posebne module za podršku objektnom pristupu

51

## Sažetak



- Relacijske baze podataka → prikaz u obliku 2D tablica
  - ☞ Svaka tablica predstavlja po jedan ENTITET
  - ☞ Svi entiteti su opisani skupom svojstava (atributa)
  - ☞ Entiteti su međusobno logički povezani
  - ☞ Proširena funkcionalnost uvođenjem upitnih jezika (SQL i QBE)
- ENTITET – sve što postoji u stvarnom ili virtualnom svijetu, a o čemu možemo bilježiti podatke
- DBMS DataBase Management System – skup programa koji omogućavaju kreiranje i održavanje baze podataka, te kontrolu pristupa podacima
- Baza podataka + DBMS = Sustav baze podataka
- Dizajniranje baze podataka → konceptualni model → ERD dijagram → detaljni ERD dijagram → strukturni model

52

## Sažetak



- Strukturni model – 2D tablice
  - ▣ Redci tablica → zapisi (records) – informacije
  - ▣ Stupci tablica → polja – podaci
- Tablice se povezuju preko ključeva
  - ▣ Primarni ključ – jedinstvenost zapisa
  - ▣ Sekundari (strani) ključ – primarni ključ koji se nalazi u nekoj drugoj (sekundarnoj tablici) → preko njega se ostvaruje veza

53

## Pitanja za provjeru znanja



- Navedite razloge zbog kojih je potrebno upravljati podacima u nekoj organizaciji.
- Što svaka organizacija treba zadovoljiti da bi mogla kvalitetno upravljati podacima?
- Objasnite nedostatke klasičnog datotečnog sustava u odnosu na baze podataka (dovoljno je navesti najveće nedostatke – potkrijepite primjerom).
- Što je baza podataka?
- Što je DataBase Management System (DBMS)?
- Koji su zadaci sustava za upravljanje bazama podataka (DBMS – a)?
- Koji je zajednički naziv za bazu podataka i DBMS?

54

## Pitanja za provjeru znanja



- Što su primarni ključevi i čemu služe?
- Što su strani ključevi i čemu služe?
- Kako se entiteti prikazuju u strukturalnom prikazu relacijskih baza podataka?
- Navedite barem dva primjera poslovnih pravila?
- Gdje se uključuju poslovna pravila?
- Na koji način upitni jezici proširuju funkcionalnost relacijskih baza podataka?
- Koje upitne jezike poznajete?

55



**KRAJ**

TEMATSKE CJELINE