

Koncepcija baze podataka

Istorijat razvoja postupaka za upravljanje podacima

Baze podataka 1

1

Sadržaj

- OM i klasična organizacija datoteka
- Baze podataka i SUBP
- Šema baze podataka
- Podšema / Eksterna šema
- Pogled
- Sistemi baza podataka

2

2

OM i klasična organizacija datoteka

3

Motivacija

- Vrednost svakog sistema, pa i sistema baza podataka, najbolje se shvata
 - ne samo na osnovu poznavanja samog sistema
 - već na osnovu činjenice da taj sistem predstavlja korak u evoluciji rešavanja onih problema, koje prethodni sistemi nisu mogli da reše
- Da bi se stekla precizna slika o bazama podataka
 - nije dovoljno samo definisati pojam baze podataka
 - potrebno je prvo baze podataka sagledati u kontekstu njihovog istorijskog razvoja

4

4

Operativna memorija

- Podaci se mogu skladištiti u OM
 - brza i mogućnost *random* pristupa
 - mogućnost koju nudi svaki programski jezik
- Nedostaci:
 - mali kapacitet
 - privremena

5

5

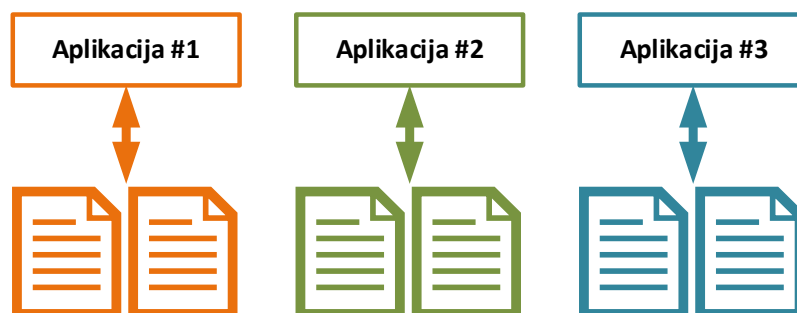
Klasična organizacija datoteka

- Najstarije rešenje
 - IS "na karticama", bez podrške računara
 - medijum za trajno memorisanje struktura podataka - papir
- Naprednije rešenje
 - IS organizovani nad sistemima datoteka
 - medijum za trajno memorisanje struktura podataka – sistem diskova
 - IS je sačinjavao skup nezavisnih aplikacija
 - svaka aplikacija - sopstvene datoteke
 - "skladište podataka" - skup datoteka
 - podaci o istom entitetu u različitim datotekama
 - vremenom, takav IS dolazi u kontradikciju sa samim sobom

6

6

Klasična organizacija datoteka



7

7

Klasična organizacija datoteka

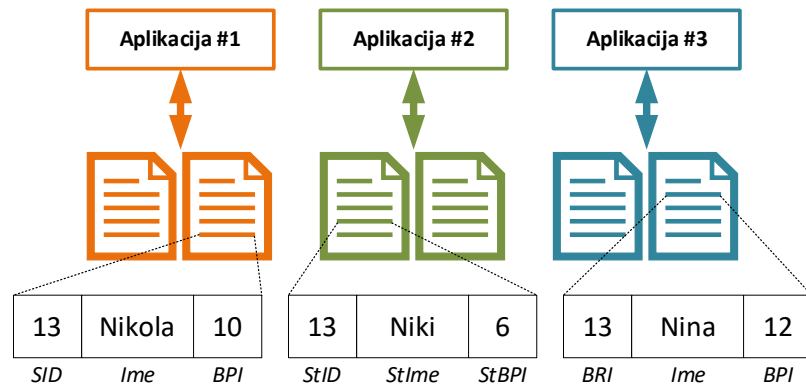
- Osnovni nedostaci
 - nepovezanost aplikacija
 - potreba ručnog prepisivanja istih ili sličnih podataka
 - redundantnost podataka
 - potreba višestrukog memorisanja istih ili sličnih podataka
 - čvrsta povezanost programa i podataka
 - program vodi računa o FSP datoteke, kako u opisu, tako i u proceduri
 - konkurentni pristup više korisnika
- Posledice
 - otežano održavanje IS-a
 - otežan dalji razvoj IS-a

8

8

Klasična organizacija datoteka

- Primer - nepovezanost i redundantnost

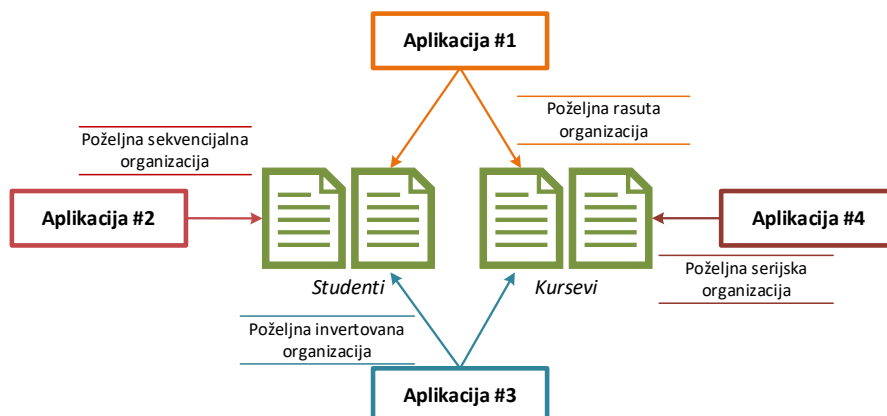


9

9

Klasična organizacija datoteka

- Primer - čvrsta povezanost programa i podataka



10

10

Klasična organizacija datoteka

- Čvrsta povezanost – primer COBOL
**IDENTIFICATION DIVISION.
 PROGRAM-ID. Seq2Rel.
 AUTHOR. MC**

```
IDENTIFICATION DIVISION.
PROGRAM-ID. Seq2Rel.
AUTHOR. MC

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
  SELECT SupplierFile ASSIGN TO "RELSUPP.DAT"
  ORGANIZATION IS RELATIVE
  ACCESS MODE IS RANDOM
  RELATIVE KEY IS SupplierKey
  FILE STATUS IS Supplierstatus.

  SELECT SupplierFileSeq ASSIGN TO "SEQSUPP.DAT"
  ORGANIZATION IS LINE SEQUENTIAL.

DATA DIVISION.
FILE SECTION.
FD SupplierFile.
  01 SupplierRecord.
  02 SupplierCode PIC 99.
  02 SupplierName PIC X(20).
  02 SupplierAddress PIC X(50).

FD SupplierFileSeq.
  01 SupplierRecordSeq.
  88 EndOfFile VALUE HIGH-VALUES.
  02 SupplierCodeSeq PIC 99.
  02 SupplierNameSeq PIC X(20).
  02 SupplierAddressSeq PIC X(50).

WORKING-STORAGE SECTION.
  01 SupplierStatus PIC X(2).
  01 SupplierKey PIC 99.

PROCEDURE DIVISION.
Begin.
  OPEN OUTPUT SupplierFile.
  OPEN INPUT SupplierFileSeq.

  READ SupplierFileSeq
  AT END SET EndOfFile TO TRUE
  END-READ
  PERFORM UNTIL EndOfFile
  MOVE SupplierCodeSeq TO SupplierKey
  MOVE SupplierRecordSeq TO SupplierRecord
  WRITE SupplierRecord
  INVALID KEY DISPLAY "Supplier status = " SupplierStatus
  END-WRITE
  READ SupplierFileSeq
  AT END SET EndOfFile TO TRUE
  END-READ
  END-PERFORM.

  CLOSE SupplierFile, SupplierFileSeq.
  STOP RUN.
```

11

11

Klasična organizacija datoteka

- Čvrsta povezanost – primer COBOL
**ENVIRONMENT DIVISION.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
 SELECT SupplierFile ASSIGN TO
 "RELSUPP.DAT"
 ORGANIZATION IS RELATIVE
 ACCESS MODE IS RANDOM
 RELATIVE KEY IS SupplierKey
 FILE STATUS IS Supplierstatus.
 SELECT SupplierFileSeq ASSIGN TO
 "SEQSUPP.DAT"
 ORGANIZATION IS LINE SEQUENTIAL**

```
IDENTIFICATION DIVISION.
PROGRAM-ID. Seq2Rel.
AUTHOR. MC

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
  SELECT SupplierFile ASSIGN TO "RELSUPP.DAT"
  ORGANIZATION IS RELATIVE
  ACCESS MODE IS RANDOM
  RELATIVE KEY IS SupplierKey
  FILE STATUS IS Supplierstatus.

  SELECT SupplierFileSeq ASSIGN TO "SEQSUPP.DAT"
  ORGANIZATION IS LINE SEQUENTIAL.

DATA DIVISION.
FILE SECTION.
FD SupplierFile.
  01 SupplierRecord.
  02 SupplierCode PIC 99.
  02 SupplierName PIC X(20).
  02 SupplierAddress PIC X(50).

FD SupplierFileSeq.
  01 SupplierRecordSeq.
  88 EndOfFile VALUE HIGH-VALUES.
  02 SupplierCodeSeq PIC 99.
  02 SupplierNameSeq PIC X(20).
  02 SupplierAddressSeq PIC X(50).

WORKING-STORAGE SECTION.
  01 SupplierStatus PIC X(2).
  01 SupplierKey PIC 99.

PROCEDURE DIVISION.
Begin.
  OPEN OUTPUT SupplierFile.
  OPEN INPUT SupplierFileSeq.

  READ SupplierFileSeq
  AT END SET EndOfFile TO TRUE
  END-READ
  PERFORM UNTIL EndOfFile
  MOVE SupplierCodeSeq TO SupplierKey
  MOVE SupplierRecordSeq TO SupplierRecord
  WRITE SupplierRecord
  INVALID KEY DISPLAY "Supplier status = " SupplierStatus
  END-WRITE
  READ SupplierFileSeq
  AT END SET EndOfFile TO TRUE
  END-READ
  END-PERFORM.

  CLOSE SupplierFile, SupplierFileSeq.
  STOP RUN.
```

12

12

Klasična organizacija datoteka

- Čvrsta povezanost – primer COBOL

DATA DIVISION.

FILE SECTION.

FD SupplierFile.

01 SupplierRecord.

02 SupplierCode PIC 99.
 02 SupplierName PIC X(20).
 02 SupplierAddress PIC X(50).

FD SupplierFileSeq.

01 SupplierRecordSeq.

88 EndOfFile VALUE HIGH-VALUES.
 02 SupplierCodeSeq PIC 99.
 02 SupplierNameSeq PIC X(20).
 02 SupplierAddressSeq PIC X(50).

```
IDENTIFICATION DIVISION.
PROGRAM-ID. SeqRel.
AUTHOR. MC

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT SupplierFile ASSIGN TO "RELSUPP.DAT"
        ORGANIZATION IS RELATIVE
        ACCESS MODE IS RANDOM
        RELATIVE KEY IS SupplierKey
        FILE STATUS IS SupplierStatus.

    SELECT SupplierFileSeq ASSIGN TO "SEQSUPP.DAT"
        ORGANIZATION IS LINE SEQUENTIAL.

DATA DIVISION.
FILE SECTION.

FD SupplierFile.
01 SupplierRecord.
02 SupplierCode PIC 99.
02 SupplierName PIC X(20).
02 SupplierAddress PIC X(50).

FD SupplierFileSeq.
01 SupplierRecordSeq.
88 EndOfFile VALUE HIGH-VALUES.
02 SupplierCodeSeq PIC 99.
02 SupplierNameSeq PIC X(20).
02 SupplierAddressSeq PIC X(50).

WORKING-STORAGE SECTION.
01 SupplierStatus PIC X(2).
01 SupplierKey PIC 99.

PROCEDURE DIVISION.
Begin.
    OPEN OUTPUT SupplierFile.
    OPEN INPUT SupplierFileSeq.

    READ SupplierFileSeq
        AT END SET EndOfFile TO TRUE
    END-READ
    PERFORM UNTIL EndOfFile
        MOVE SupplierCodeSeq TO SupplierKey
        MOVE SupplierRecordSeq TO SupplierRecord
        WRITE SupplierRecord
        INVALID KEY DISPLAY "Supplier status = " SupplierStatus
    END-WRITE
    READ SupplierFileSeq
        AT END SET EndOfFile TO TRUE
    END-READ
    END-PERFORM.

CLOSE SupplierFile, SupplierFileSeq.
STOP RUN.
```

13

13

Klasična organizacija datoteka

- Čvrsta povezanost – primer COBOL

WORKING-STORAGE SECTION.

01 SupplierStatus PIC X(2).

01 SupplierKey PIC 99.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. SeqRel.
AUTHOR. MC

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT SupplierFile ASSIGN TO "RELSUPP.DAT"
        ORGANIZATION IS RELATIVE
        ACCESS MODE IS RANDOM
        RELATIVE KEY IS SupplierKey
        FILE STATUS IS SupplierStatus.

    SELECT SupplierFileSeq ASSIGN TO "SEQSUPP.DAT"
        ORGANIZATION IS LINE SEQUENTIAL.

DATA DIVISION.
FILE SECTION.

FD SupplierFile.
01 SupplierRecord.
02 SupplierCode PIC 99.
02 SupplierName PIC X(20).
02 SupplierAddress PIC X(50).

FD SupplierFileSeq.
01 SupplierRecordSeq.
88 EndOfFile VALUE HIGH-VALUES.
02 SupplierCodeSeq PIC 99.
02 SupplierNameSeq PIC X(20).
02 SupplierAddressSeq PIC X(50).

WORKING-STORAGE SECTION.
01 SupplierStatus PIC X(2).
01 SupplierKey PIC 99.

PROCEDURE DIVISION.
Begin.
    OPEN OUTPUT SupplierFile.
    OPEN INPUT SupplierFileSeq.

    READ SupplierFileSeq
        AT END SET EndOfFile TO TRUE
    END-READ
    PERFORM UNTIL EndOfFile
        MOVE SupplierCodeSeq TO SupplierKey
        MOVE SupplierRecordSeq TO SupplierRecord
        WRITE SupplierRecord
        INVALID KEY DISPLAY "Supplier status = " SupplierStatus
    END-WRITE
    READ SupplierFileSeq
        AT END SET EndOfFile TO TRUE
    END-READ
    END-PERFORM.

CLOSE SupplierFile, SupplierFileSeq.
STOP RUN.
```

14

14

Klasična organizacija datoteka

- Čvrsta povezanost – primer COBOL

PROCEDURE DIVISION.

BEGIN.

**OPEN OUTPUT SupplierFile.
OPEN INPUT SupplierFileSeq.**

**READ SupplierFileSeq
AT END SET EndOfFile TO TRUE**

END-READ

PERFORM UNTIL EndOfFile

**MOVE SupplierCodeSeq TO SupplierKey
MOVE SupplierRecordSeq TO SupplierRecord
WRITE SupplierRecord
INVALID KEY DISPLAY "Supplier status = " SupplierStatus**

END-WRITE

READ SupplierFileSeq

AT END SET EndOfFile TO TRUE

END-READ

END-PERFORM.

**CLOSE SupplierFile, SupplierFileSeq.
STOP RUN.**

```
IDENTIFICATION DIVISION.
PROGRAM-ID. SeqRel.
AUTHOR. MC

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT SupplierFile ASSIGN TO "RELSUPP.DAT"
ORGANIZATION IS RELATIVE
ACCESS MODE IS RANDOM
RELATIVE KEY IS SupplierKey
FILE STATUS IS SupplierStatus.

SELECT SupplierFileSeq ASSIGN TO "SEQSUPP.DAT"
ORGANIZATION IS LINE SEQUENTIAL.

DATA DIVISION.
FILE SECTION.

FD SupplierFile.
01 SupplierRecord.
02 SupplierCode PIC 99.
02 SupplierName PIC X(20).
02 SupplierAddress PIC X(50).

FD SupplierFileSeq.
01 SupplierRecordSeq.
88 EndOfFile VALUE HIGH-VALUES.
02 SupplierCodeSeq PIC 99.
02 SupplierNameSeq PIC X(20).
02 SupplierAddressSeq PIC X(50).

WORKING-STORAGE SECTION.
01 SupplierStatus PIC X(2).
01 SupplierKey PIC 99.

PROCEDURE DIVISION.
Begin.
OPEN OUTPUT SupplierFile.
OPEN INPUT SupplierFileSeq.

READ SupplierFileSeq
AT END SET EndOfFile TO TRUE
END-READ
PERFORM UNTIL EndOfFile
MOVE SupplierCodeSeq TO SupplierKey
MOVE SupplierRecordSeq TO SupplierRecord
WRITE SupplierRecord
INVALID KEY DISPLAY "Supplier status = " SupplierStatus
END-WRITE
READ SupplierFileSeq
AT END SET EndOfFile TO TRUE
END-READ
END-PERFORM.

CLOSE SupplierFile, SupplierFileSeq.
STOP RUN.
```

15

15

Klasična organizacija datoteka

- Čvrsta povezanost – primer COBOL

**SELECT STUDENT ASSIGN TO "STUD.DAT"
ORGANIZATION IS INDEXED
ACCESS MODE IS RANDOM
RECORD KEY IS BRI
ALTERNATE RECORD KEY IS PREZIME
WITH DUPLICATES
FILE STATUS IS StudStatus.**

16

16

Klasična organizacija datoteka

- Problemi koji se mogu ublažiti, ili čak razrešiti u klasičnoj organizaciji
 - nepovezanost aplikacija
 - redundantnost
- Problem koji je gotovo nemoguće ublažiti ili razrešiti u klasičnoj organizaciji
 - čvrsta povezanost programa i podataka

17

17

Baze podataka i SUBP

18

Baze podataka i SUBP

- Osnovne ideje:
 - **da se svi podaci jednog IS integrišu u jednu veliku "datoteku"**
 - nastanak pojma baze podataka
 - **nereduntantno memorisanje podataka**
 - izbegavanje nepotrebnog višestrukog memorisanja istih ili sličnih podataka
 - **da se uvede poseban softverski proizvod za podršku kreiranja i korišćenja baze podataka**
 - **sistem za upravljanje bazama podataka (SUBP)**
 - da svi programi koriste podatke iz baze podataka, ili je ažuriraju koristeći isključivo usluge SUBP
 - transakcijska obrada
 - teška implementacija nad sistemom datoteka
 - SUBP ima ugrađenu podršku
 - višekorisnički konkurentni pristup
 - autorizacija korisnika

19

19

Definicija BP

- *James Martin (IBM):*
 - **Skup međusobno povezanih podataka, pohranjenih zajedno, bez bespotrebne redundanse, koji mogu zadovoljiti različite primene.**
 - Podaci su pohranjeni na način nezavisan od programa koji ih koriste.
 - Prilikom dodavanja novih podataka, menjanja i pretraživanja postojećih podataka, primenjuje se zajednički i kontrolisani pristup.
 - Podaci su strukturirani tako da služe kao osnova za razvoj budućih primena.

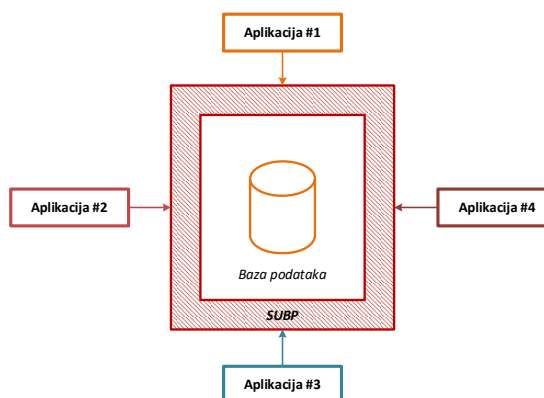


20

20

Baze podataka i SUBP

- **Sistem za upravljanje bazama podataka (SUBP)**
- **Database Management System (DBMS)**
 - softverski proizvod, namenjen da omogući izgradnju i korišćenje baza podataka
 - prilagodljiv i lak za korišćenje
 - Sadrži:
 - jezik za opis podataka
 - Data Definition Language - DDL
 - jezik za manipulisanje podacima
 - Data Manipulation Language - DML
 - upitni jezik
 - Query Language - QL



21

21

Baze podataka i SUBP

- Jezgro SUBP
 - obezbeđenje fizičke organizacije podataka
 - rutine za upravljanje podacima
 - zaštita od neovlašćenog pristupa i od uništenja
 - obezbeđenje višekorisničkog režima rada
 - obezbeđenje distribuirane organizacije BP
 - obezbeđenje zadavanja šeme baze podataka
 - nad skupom obeležja ranijih datoteka formira se struktura šeme BP
 - nad šemom BP se kreira, koristi i ažurira baza podataka

22

22

Primena BP

- Bankovne aplikacije
 - tekući računi
 - transakcije
 - štednja



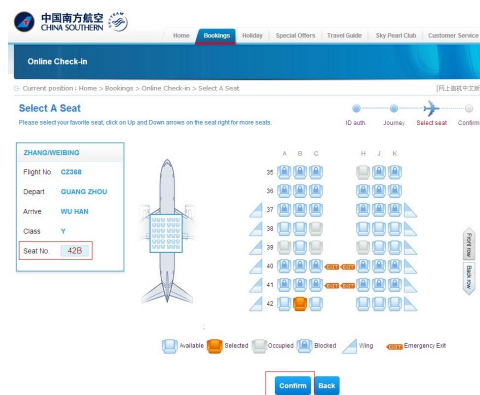
23

23

Primena BP

- Transport
 - red vožnje/letenja
 - rezervacija i kupovina karata
 - odabir mesta za sedenje

| THAI AIRWAYS FLIGHT SCHEDULE | | | | | | | |
|------------------------------|-----------|----------|-----------|-------------------------|-----------|----------|-----------|
| ROUTE: BANGKOK - PHUKET | | | | ROUTE: PHUKET - BANGKOK | | | |
| FLIGHT NUMBER | DEPARTURE | ARRIVAL | FREQUENCY | FLIGHT NUMBER | DEPARTURE | ARRIVAL | FREQUENCY |
| TG 201 | 7:35 AM | 9:00 AM | DAILY | TG 226 | 7:25 AM | 8:50 AM | DAILY |
| TG 203 | 8:30 AM | 9:55 AM | DAILY | TG 202 | 10:00 AM | 11:30 AM | DAILY |
| TG 205 | 10:00 AM | 11:20 AM | DAILY | TG 206 | 10:55 AM | 12:25 AM | DAILY |
| TG 207 | 10:55 AM | 12:15 AM | DAILY | TG 208 | 1:05 PM | 2:30 PM | DAILY |
| TG 211 | 12:35 AM | 1:45 PM | DAILY | TG 212 | 3:40 PM | 4:05 PM | DAILY |
| TG 215 | 2:15 PM | 3:35 PM | DAILY | TG 214 | 5:10 PM | 4:35 PM | DAILY |
| TG 217 | 4:00 PM | 5:20 PM | DAILY | TG 216 | 4:25 PM | 5:50 PM | DAILY |
| TG 221 | 6:20 PM | 7:40 PM | DAILY | TG 218 | 6:25 PM | 7:50 PM | DAILY |
| TG 223 | 7:20 PM | 8:40 PM | DAILY | TG 222 | 8:50 PM | 10:15 PM | DAILY |
| TG 225 | 10:40 PM | 11:59 PM | DAILY | TG 224 | 9:30 PM | 10:55 PM | DAILY |



24

24

Primena BP

- Telekomunikacije
 - podaci o pozivima
 - telefonski računi
 - praćenje kvarova
 - model mreže

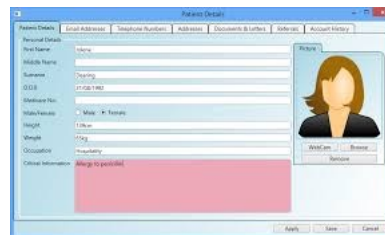


25

25

Primena BP

- Medicina
 - podaci o pacijentima
 - istorija bolesti
 - pomoć pri dijagnostikovanju

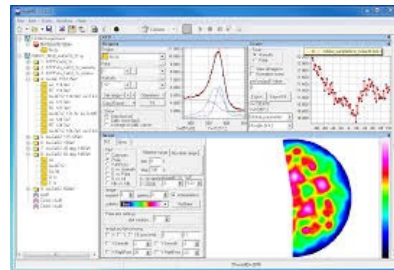
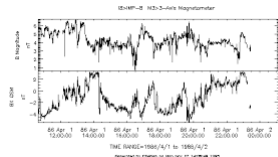


26

26

Primena BP

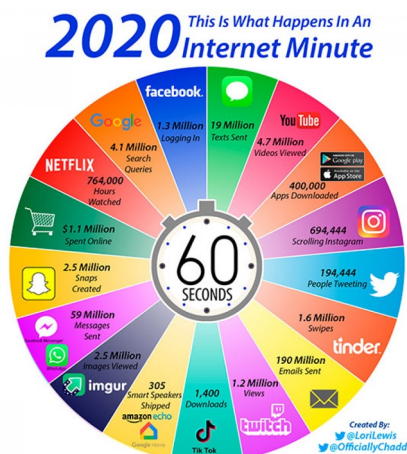
- Nauka
 - prikupljanje i obrada podataka



27

27

Velika količina podataka

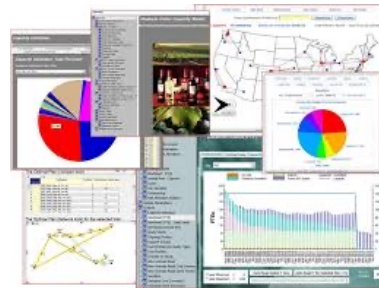
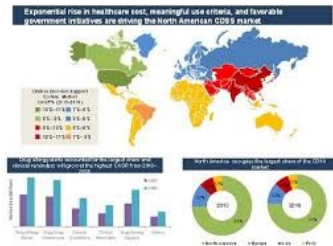


28

28

Primena BP

- Sistemi za pomoć pri odlučivanju
 - podaci o poslovanju
 - izveštaji
 - višedimenzionalni pogledi
 - data mining

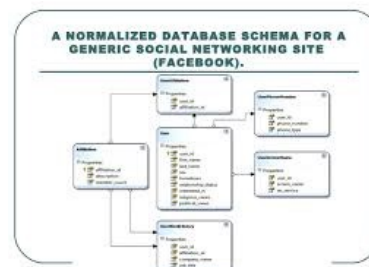


29

29

Primena BP

- Društvene mreže
 - korisnički nalozi
 - poruke, objave
 - video klipovi, slike

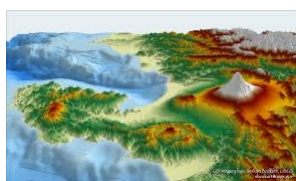


30

30

Primena BP

- Geoinformacioni sistemi
 - zaštita životne sredine
 - upravljanje resursima



31

31

Mane SUBP

- Da li je SUBP rešenje bez mane?
 - Visoka cena
 - Komplikovana instalacija i administracija (ali mnogostruko manja nego u slučaju sistema baziranim na datotekama)
 - Proizvod opšte namene.
 - npr. nije namenjen za skladištenje vremenski serija ili pretragu po tekstu

32

32

Tipovi baza podataka

- Dominantne paradigme
 - Relacione baze podataka
 - zasnovane na relacionoj algebri i relacionom računu
 - tradicionalan i dominantno korišćen pristup
 - ***izučavaju se na ovom predmetu***
 - NoSQL baze podataka
 - nisu zasnovane na relacionim principima
 - objektne (binarne, JSON) baze
 - Orijentisane ka dokumentima
 - Orijentisane ka familijama kolona
 - Skladišta tipa ključ-vrednost
 - XML baze
 - graf baze podataka
 - *izučavaju se na višim godinama (izborni predmeti)*

33

33

Šema baze podataka

34

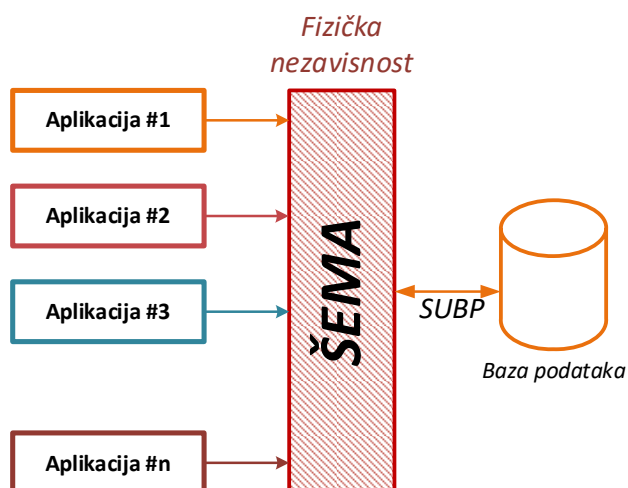
Šema baze podataka

- Program koji koristi usluge SUBP
 - poznaje samo šemu BP, kao logičku kategoriju
 - nad šemom BP, koristi logičku strukturu podataka (LSP), saglasno konkretnom zadatku
 - ne sme da vodi računa o fizičkoj strukturi podataka (FSP)
 - koja, po pravilu, može biti veoma kompleksna
- Preslikavanje LSP ↔ FSP - zadatak SUBP
 - primeri potreba različitih programa za istim podacima
 - pristup saglasno rastućim vrednostima primarnog ključa
 - pristup saglasno vrednostima sekundarnog ključa
 - direktni pristup, saglasno zadatoj vrednosti primarnog ključa

35

35

Šema baze podataka



36

36

Šema baze podataka

- Efekti uvođenja koncepta šeme baze podataka
 - smanjenje zavisnosti programa i šeme BP od promena FSP
 - smanjenje redundantnosti
 - povećanje konzistentnosti podataka
 - uvođenje uloga
 - projektant baze podataka (šeme i FSP)
 - administrator baze podataka (DBA)

37

37

Projektant baze podataka

- Definiše šemu BP
- Ima potpuno razumevanje činilaca i poslovnih procesa projektovanog IS
- Zaduženja:
 - Analiza zahteva
 - Identifikacija tipova entiteta i poveznika
 - Definisavanje pomoćnih struktura

38

38

Administrator baze podataka

- Upravlja aktivnostima SUBP
- Ima dobro razumevanje resursa i potreba organizacije
- Zaduženja:
 - Implementacija i održavanje šeme
 - Podešavanje parametara servera
 - Instalacija i održavanje SUBP
 - Dodela prava korisnicima SUBP i baza podataka
 - Praćenje i unapređivanje performansi servera

39

39

Podšema / Eksterna šema

40

Podšema / Eksterna šema

- Šema BP je, po pravilu,
 - kompleksna
 - podložna češćim modifikacijama
- Motivacija
 - promene šeme BP mogu izazivati česte i nepotrebne promene postojećih programa
 - kada bi program direktno koristio koncepte šeme BP
 - posledica: otežano održavanje softverske podrške IS
- Rešenje
 - uvođenje novog koncepta i novog sloja - podšema

41

41

Podšema / Eksterna šema

- **Podšema** ili **eksterna šema**
 - logička struktura obeležja (LSO), dobijena na osnovu dela šeme BP
 - potrebna i dovoljna za realizaciju zadataka jednog, ili grupe sličnih transakcionih programa
 - sličnih sa stanovišta modelovanih procesa poslovanja i korisničkih zahteva
 - model dela BP realnog sistema
 - za razliku od šeme BP koja predstavlja model cele BP realnog sistema
 - projektuje se, kao i šema BP, u procesu razvoja IS
 - treba da egzistira kao projektantska specifikacija
 - poželjna takva organizacija transakcionih programa
 - da koriste BP isključivo putem podšema

42

42

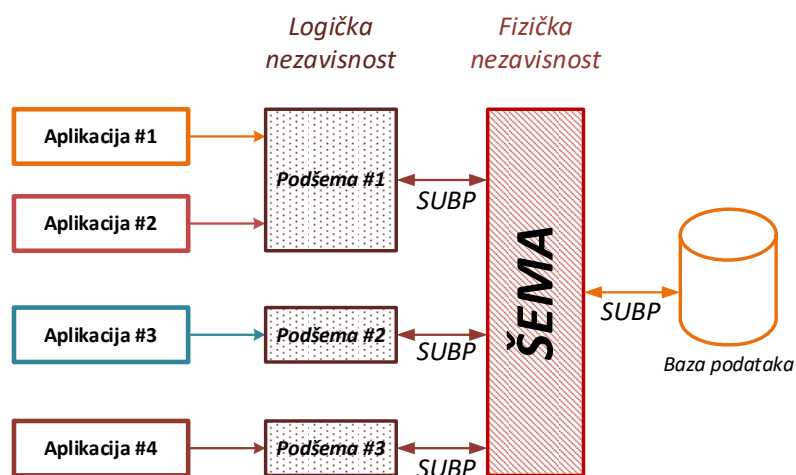
Podšema / Eksterna šema

- Preslikavanje Podšema ↔ Šema BP
 - moguće i poželjno da bude zadatak SUBP
 - SUBP prevodi
 - zahtev programa, definisan s obzirom na koncepte podšeme, u zahtev definisan s obzirom na koncepte šeme BP
 - podatke strukturirane **s obzirom na koncepte šeme BP** u podatke strukturirane **s obzirom na koncepte podšeme** i obratno
 - alternativno, to može biti delimično ili u celosti zadatak samog transakcionog programa
 - danas, često, onog dela transakcionog programa koji upravlja logičkim strukturama podataka
 - nalazi se u višenivoovskim arhitekturama na tzv. "donjem" sloju – sloju modela podataka

43

43

Podšema / Eksterna šema



44

44

Podšema / Eksterna šema

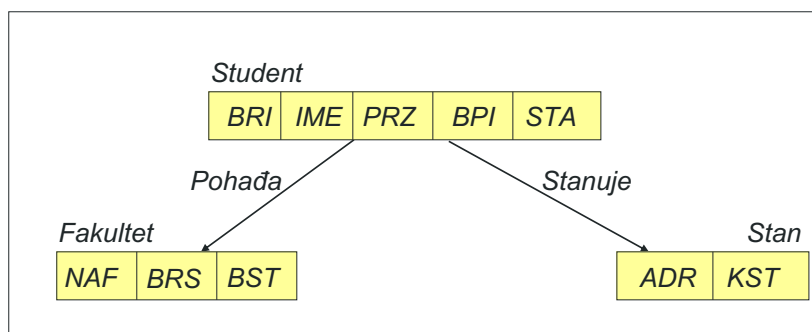
- Efekat uvođenja koncepta podšeme
 - **logička nezavisnost** programa od podataka
 - promene šeme ne izazivaju promene podšeme i programa
- Efekat uvođenja koncepta šeme BP
 - **fizička nezavisnost** programa od podataka
 - promene FSP ne izazivaju promene šeme, podšeme i programa
- Fizička i logička nezavisnost su uslovne, a ne apsolutne kategorije!

45

45

Podšema / Eksterna šema

- Primer male šeme BP u mrežnom modelu podataka

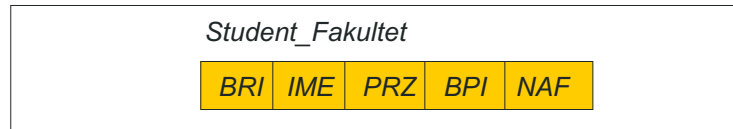


46

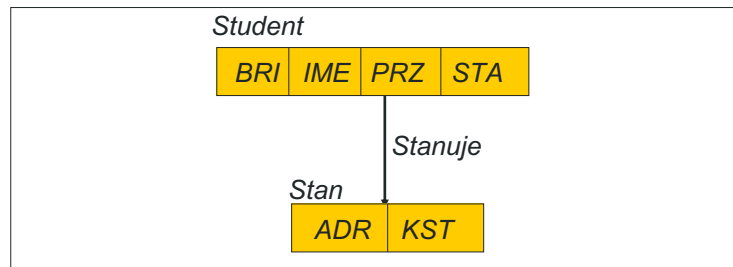
46

Podšema / Eksterna šema

PODŠEMA #1



PODEŠEMA #2



47

47

Pogled

48

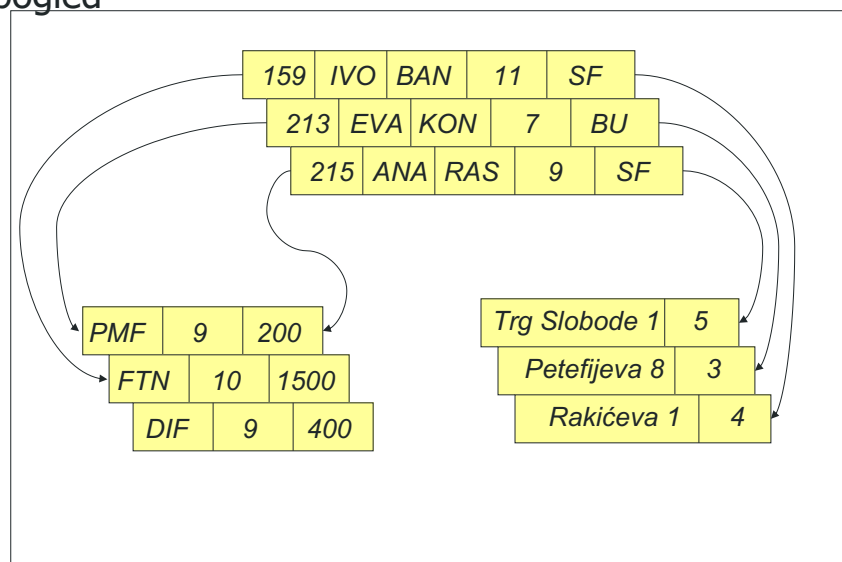
Pogled

- Šema i podšema
 - modeli na nivou apstrakcije obeležja
- Globalni pogled i pogled
 - modeli na nivou apstrakcije podataka
- **Pogled**
 - pojava (LSP) nad podšemom
 - slika dela BP kako je vidi programer, ili korisnik
- Globalni pogled
 - pojava (LSP) nad šemom BP – baza podataka
 - slika stanja modelovanog dela sistema

49

49

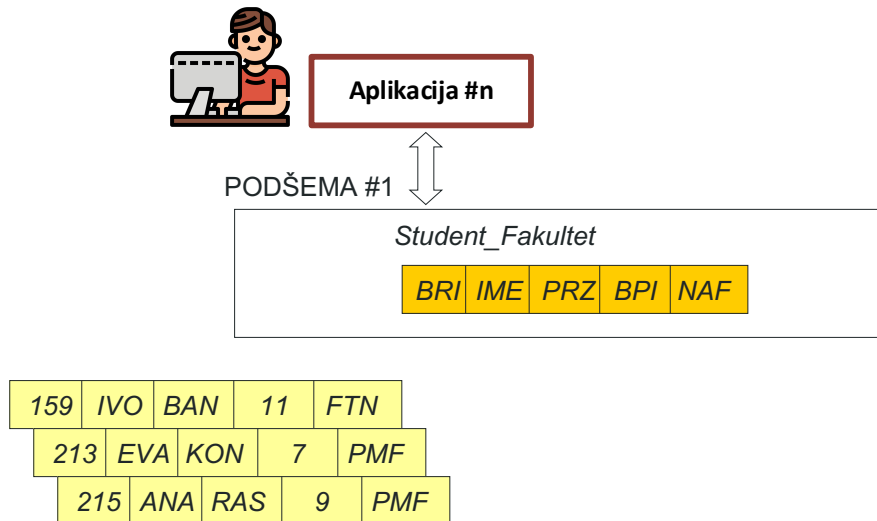
Globalni pogled



50

50

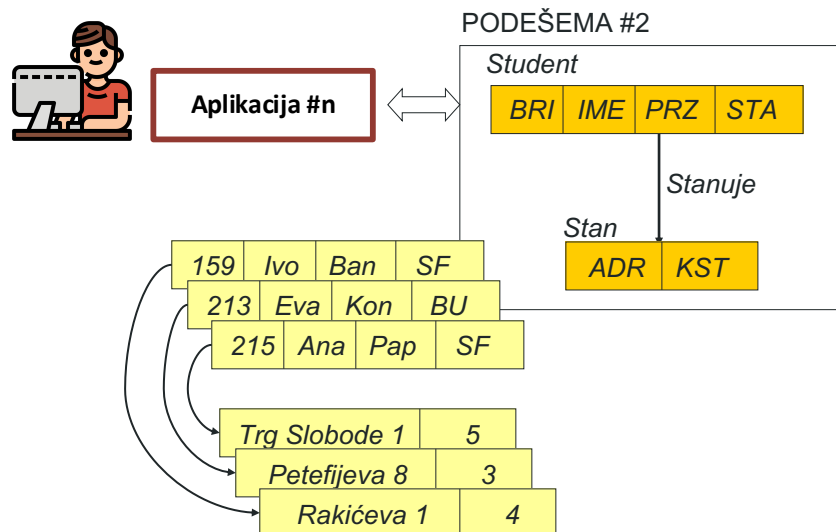
Pogled



51

51

Pogled



52

52

Sistemi baza podataka

53

Sistemi baza podataka

- Sistem baze podataka
 - u užem smislu obuhvata
 - bazu podataka
 - implementiranu na jednom uređaju ili distribuiranu na više računara
 - sistem za upravljanje bazom podataka (SUBP)
 - šemu baze podataka, implementiranu na SUBP
 - jezike i operacije za kreiranje, ažuriranje i korišćenje BP
 - sistemski softver i računare (servere) na kojima je BP kreirana

54

54

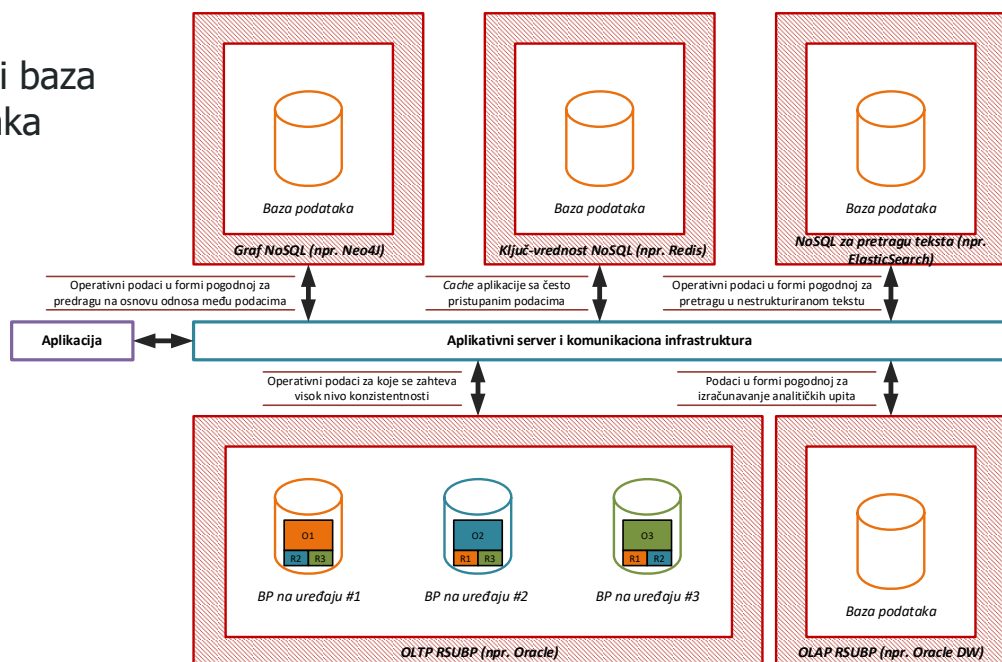
Sistemi baza podataka

- Sistem baze podataka
 - u širem smislu obuhvata
 - više baza podataka integrisanog informacionog sistema
 - potencijalno zasnovanih na različitim modelima podataka
 - više SUBP-ova koji predstavljaju logičku celinu
 - **više instanci istog SUBP = homogen SBP**
 - **više instanci različitih SUBP = heterogen SBP**
 - sa svim pratećim softverom i hardverom
 - koji omogućava rad i korišćenje SBP

55

55

Sistemi baza podataka



56

56

Sistemi baza podataka

- Rezime
 - opisane karakteristike sistema BP predstavljaju ciljeve kojima treba težiti
 - u kojoj meri će ti ciljevi biti ostvareni, zavisi od
 - projektanta BP
 - kvalitet / znanje raspoloživih ljudskih resursa
 - izabranih koncepata, metoda i tehnika projektovanja
 - kvalitet procesa razvoja sistema
 - karakteristika SUBP
 - kvalitet primenjenih informacionih tehnologija
 - razvoj postupaka za organizovanje i upravljanje podacima vodi ka:
 - povećanju produktivnosti razvojnog tima
 - izgradnji integrisanih IS

58

58

Literatura

- Pavle Mogin, Ivan Luković: Principi baza podataka
 - Poglavlja: 3.1, 3.2, 3.3 i 3.4

59

59