

Baze podataka

Koncepcija baze podataka

*Istorijat razvoja postupaka za
upravljanje podacima*

Sadržaj

- Motivacija
- Klasična organizacija datoteka
- Baze podataka i SUBP
- Šema baze podataka
- Sistemi baza podataka

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

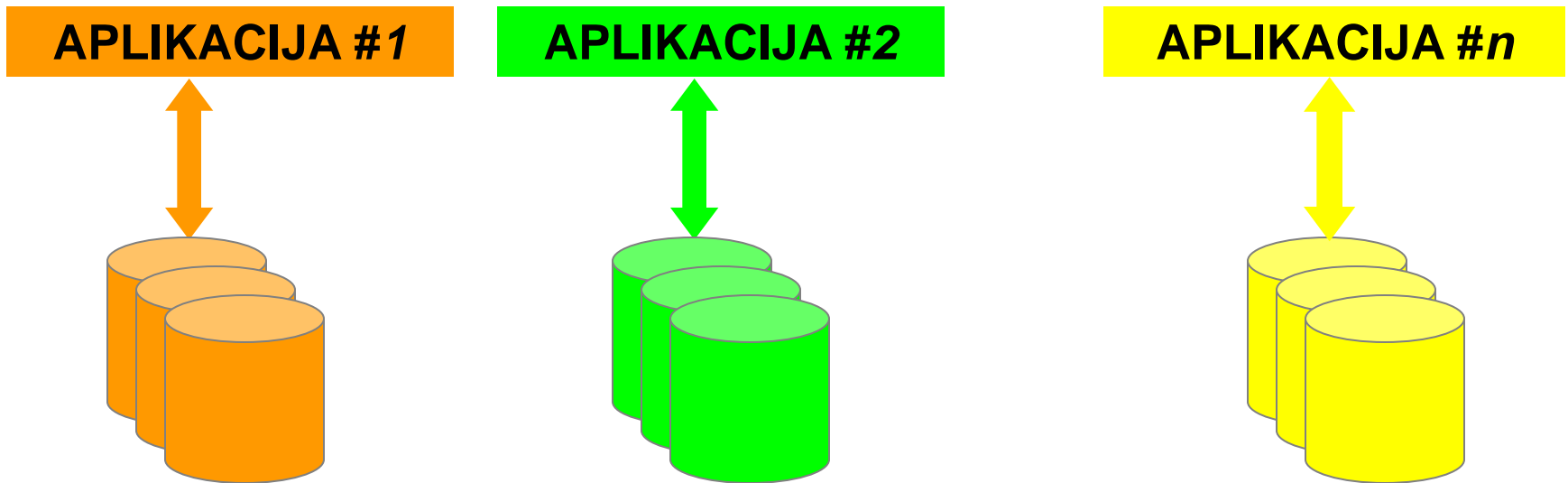
Sadržaj

- Motivacija
- Klasična organizacija datoteka
- Baze podataka i SUBP
- Šema baze podataka
- Sistemi baza podataka

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

Klasična organizacija datoteka

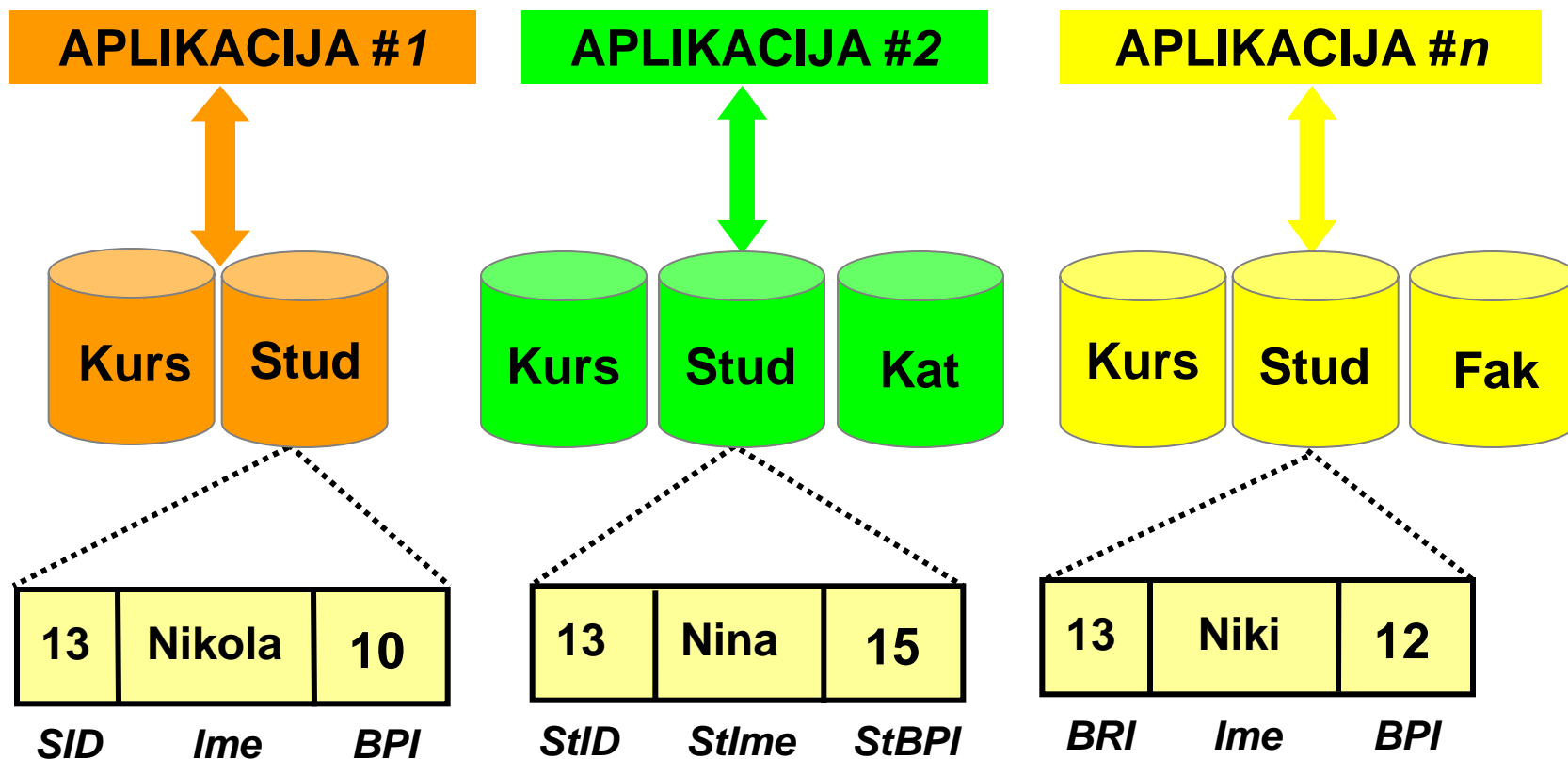


Klasična organizacija datoteka

- 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 fizičkoj strukturi podataka (FSP) datoteke, kako u opisu, tako i u proceduri
- Posledice
 - otežano održavanje IS-a
 - otežan dalji razvoj IS-a

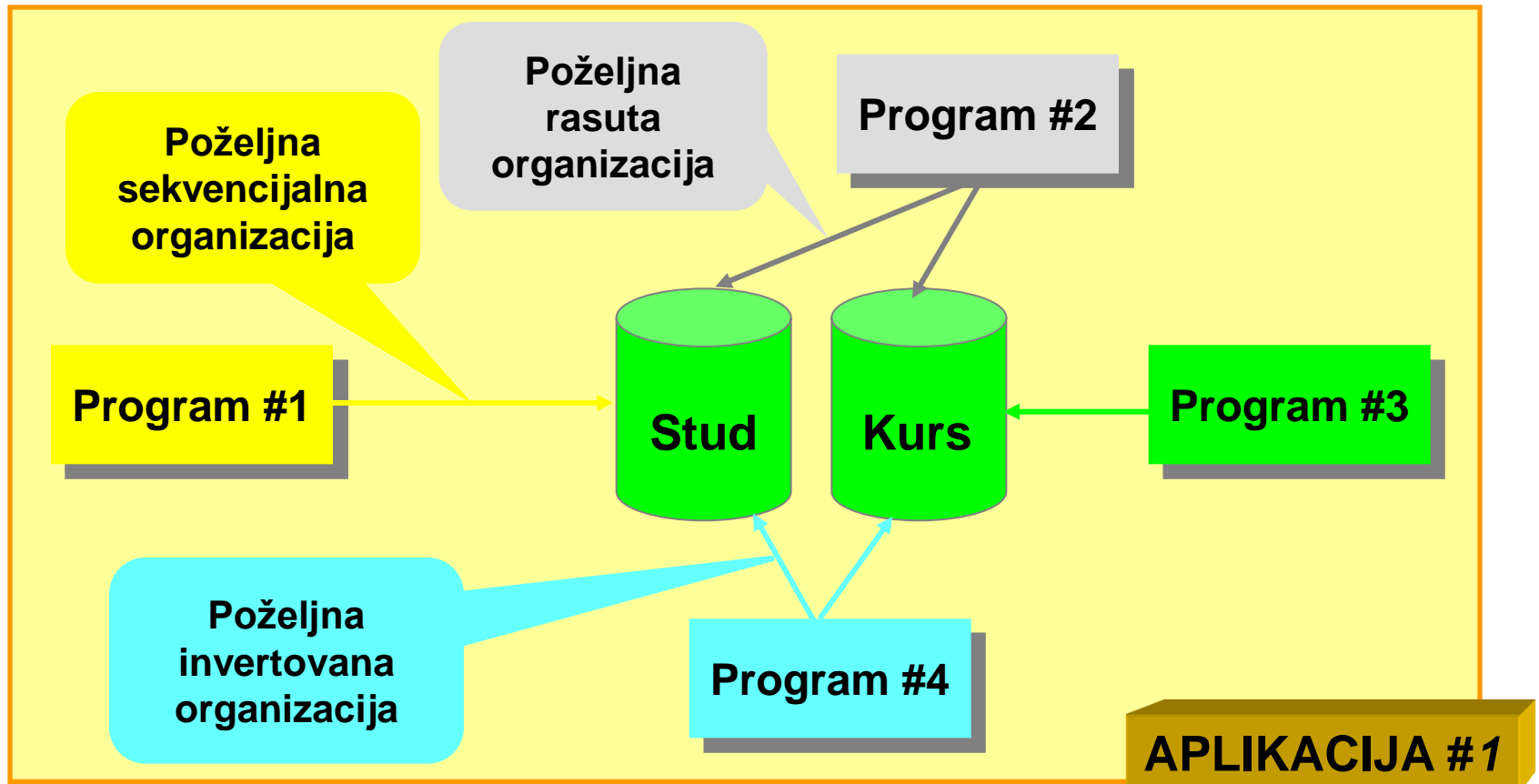
Klasična organizacija datoteka

- Primer - nepovezanost i redundantnost



Klasična organizacija datoteka

- Čvrsta povezanost programa i podataka



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.
```

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.
```

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. 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.
```

Klasična organizacija datoteka

- Čvrsta povezanost – primer COBOL

WORKING-STORAGE SECTION.

01 SupplierStatus PIC X(2).

01 SupplierKey PIC 99.

```
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.
```

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. 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.
```

Klasična organizacija datoteka

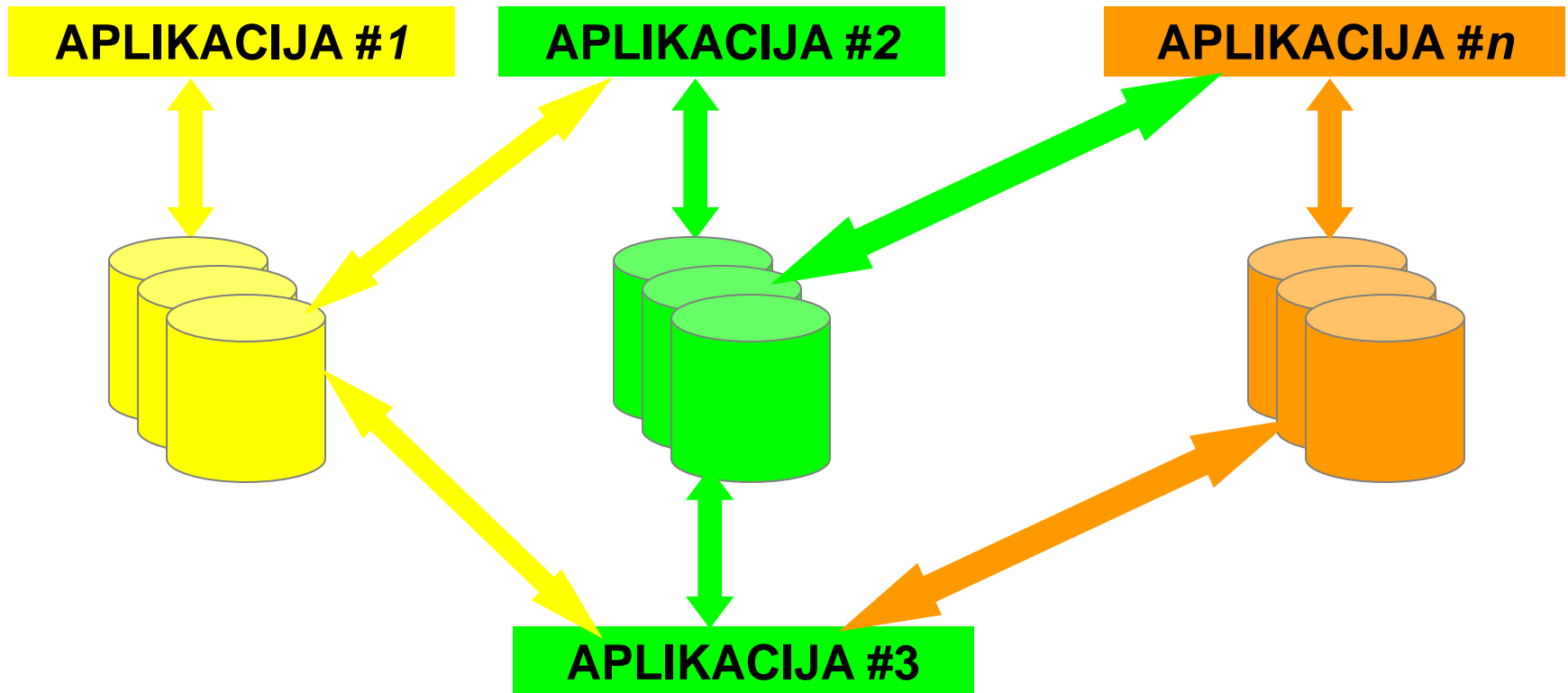
- Čvrsta povezanost – primer COBOL

**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.**

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

Klasična organizacija datoteka



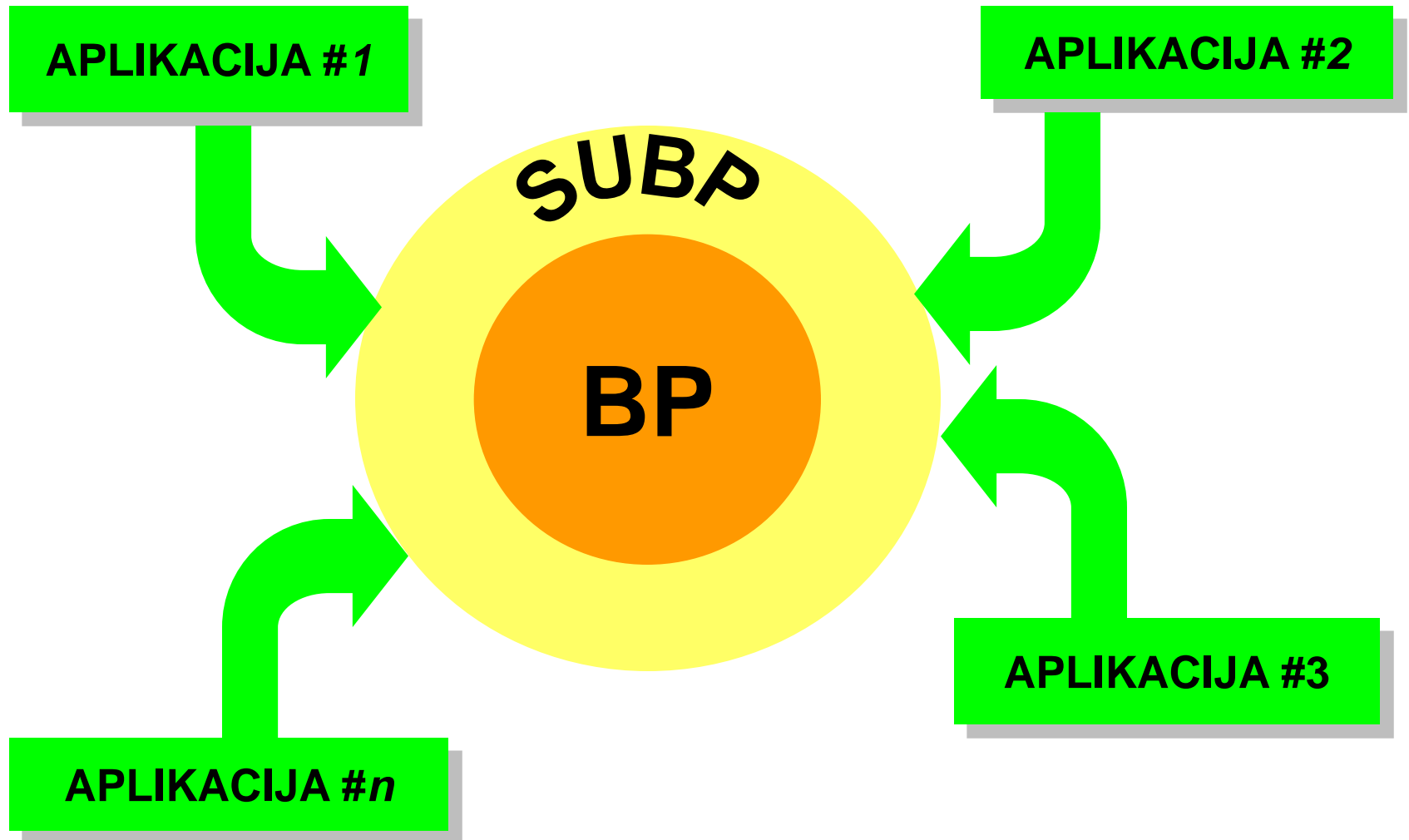
Sadržaj

- Motivacija
- Klasična organizacija datoteka
- Baze podataka i SUBP
- Šema baze podataka
- Sistemi baza podataka

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

Baze podataka i SUBP



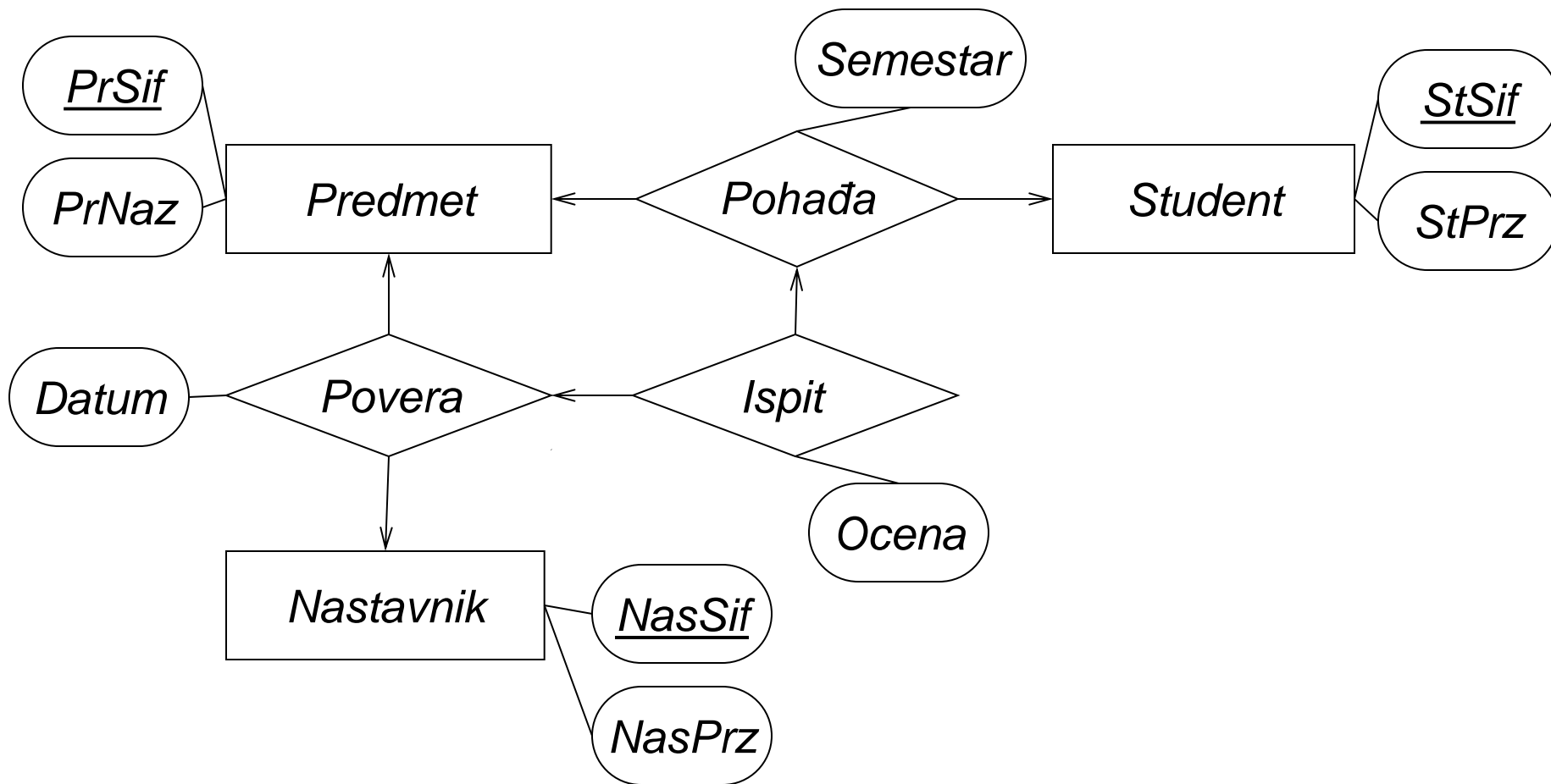
Baze podataka i SUBP

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

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

Logička struktura obeležja



Reprezentacija LSP putem tabela

Radnik

<i>MBR</i>	<i>IME</i>	<i>PRZ</i>	<i>GRD</i>	<i>SEF</i>
159	Ivo	Ban	1940	081
081	Eva	Pap	1948	
013	Ana	Ras	1962	081
015	Ena	Kon	1975	013

Zgrada

<i>SZG</i>	<i>ADR</i>	<i>BRS</i>
003	Puškinova 8	3
013	Andrićeva 13	8
015	Tolstojeva 1	4
113	Balzakova 44	8

Poslovnica

<i>SPR</i>	<i>NAZ</i>	<i>DIR</i>
03	Lim 1	013
13	Matica	081
23	Lim 3	013

Zaposlen

<i>MBR</i>	<i>SPR</i>
159	23
081	23
013	03

Nalazi_se

<i>SPR</i>	<i>SZG</i>
03	013
13	003
23	015

Stanuje

<i>MBR</i>	<i>SZG</i>
159	003
081	113
013	113

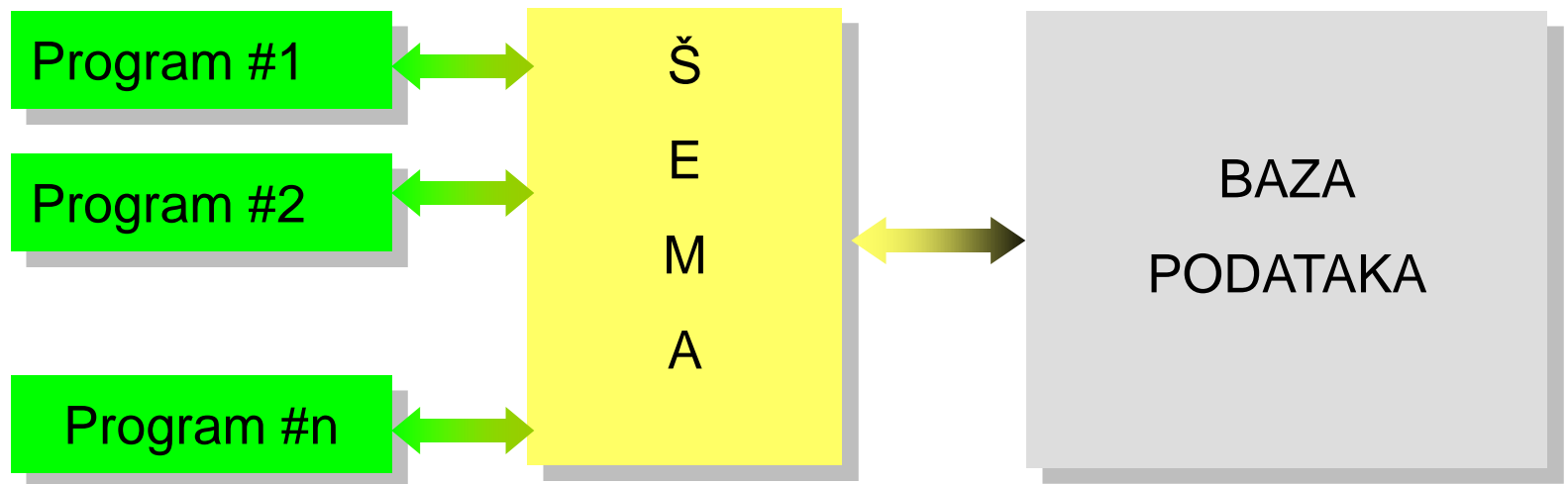
Sadržaj

- Motivacija
- Klasična organizacija datoteka
- Baze podataka i SUBP
- Šema baze podataka
- Sistemi baza podataka

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 $LSP \leftrightarrow FSP$ - 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

Šema baze podataka



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)

Sadržaj

- Motivacija
- Klasična organizacija datoteka
- Baze podataka i SUBP
- Šema baze podataka
- Sistemi baza podataka

Sistemi baza podataka

- Sistem baze podataka
 - obuhvata
 - bazu podataka
 - SUBP, sistemski softver i računare (servere) na kojima je BP kreirana
 - šemu baze podataka, implementiranu na SUBP
 - jezike i operacije za kreiranje, ažuriranje i korišćenje BP

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

Sadržaj

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

Pitanja i komentari



Kraj prezentacije

Baze podataka

Koncepcija baze podataka

*Istorijat razvoja postupaka za
upravljanje podacima*