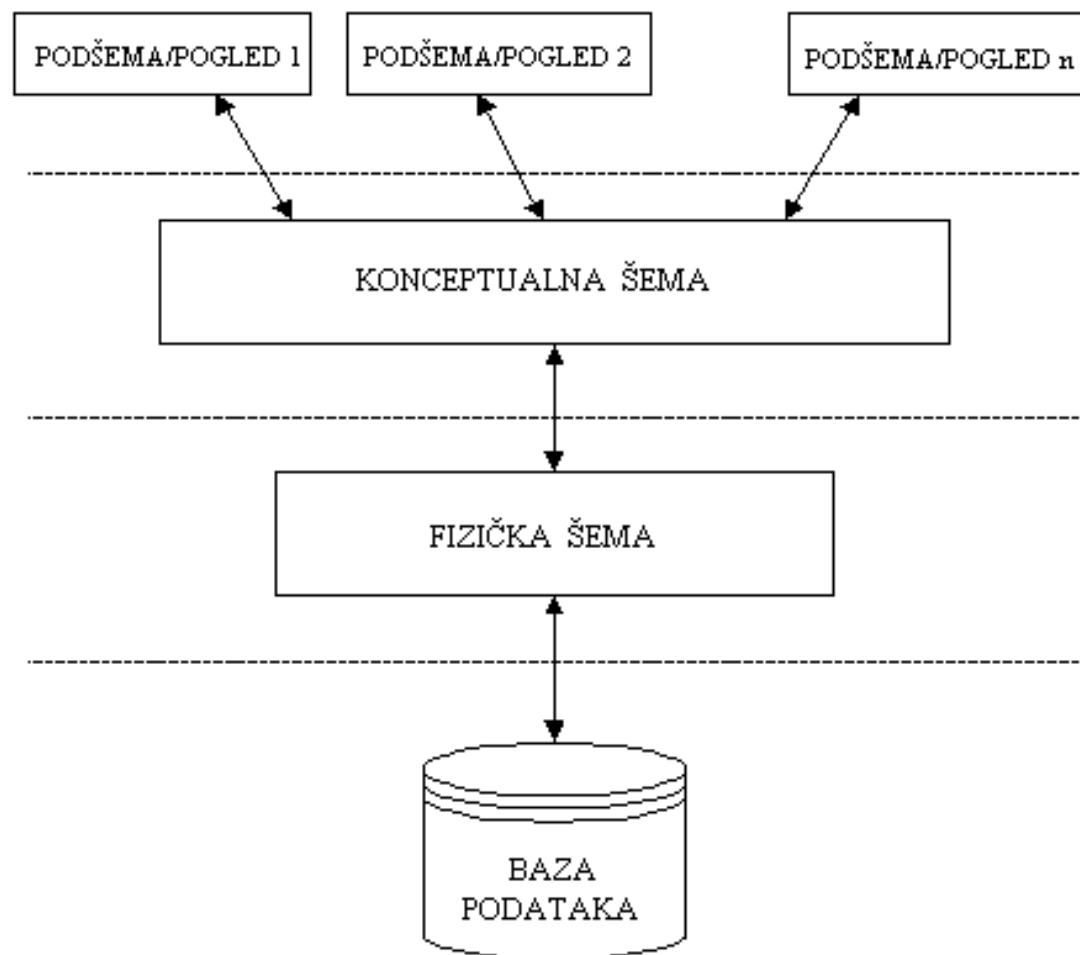


# Faze projektovanja BP

---

1. Prikupljanje i analiza zahteva
  2. Logičko projektovanje baze podataka
  3. Izbor sistema za upravljanje bazom podataka
  4. Prevođenje modela podataka
  5. **Fizičko projektovanje BP**
  6. Implementacija BP
-

# Arhitektura tronivojskog SUBP



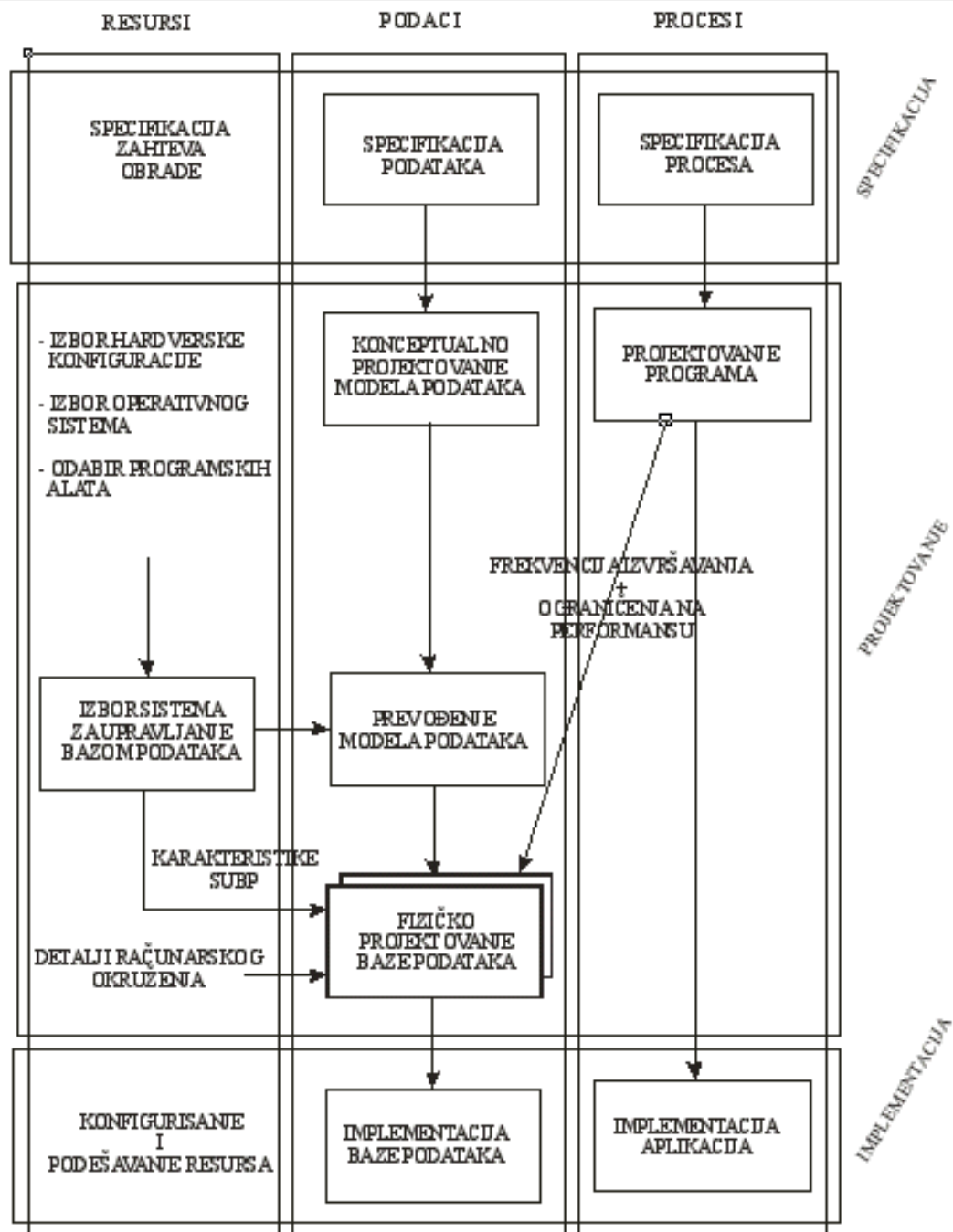
# **Osnovna karakteristika savremenih SUBP:**

---

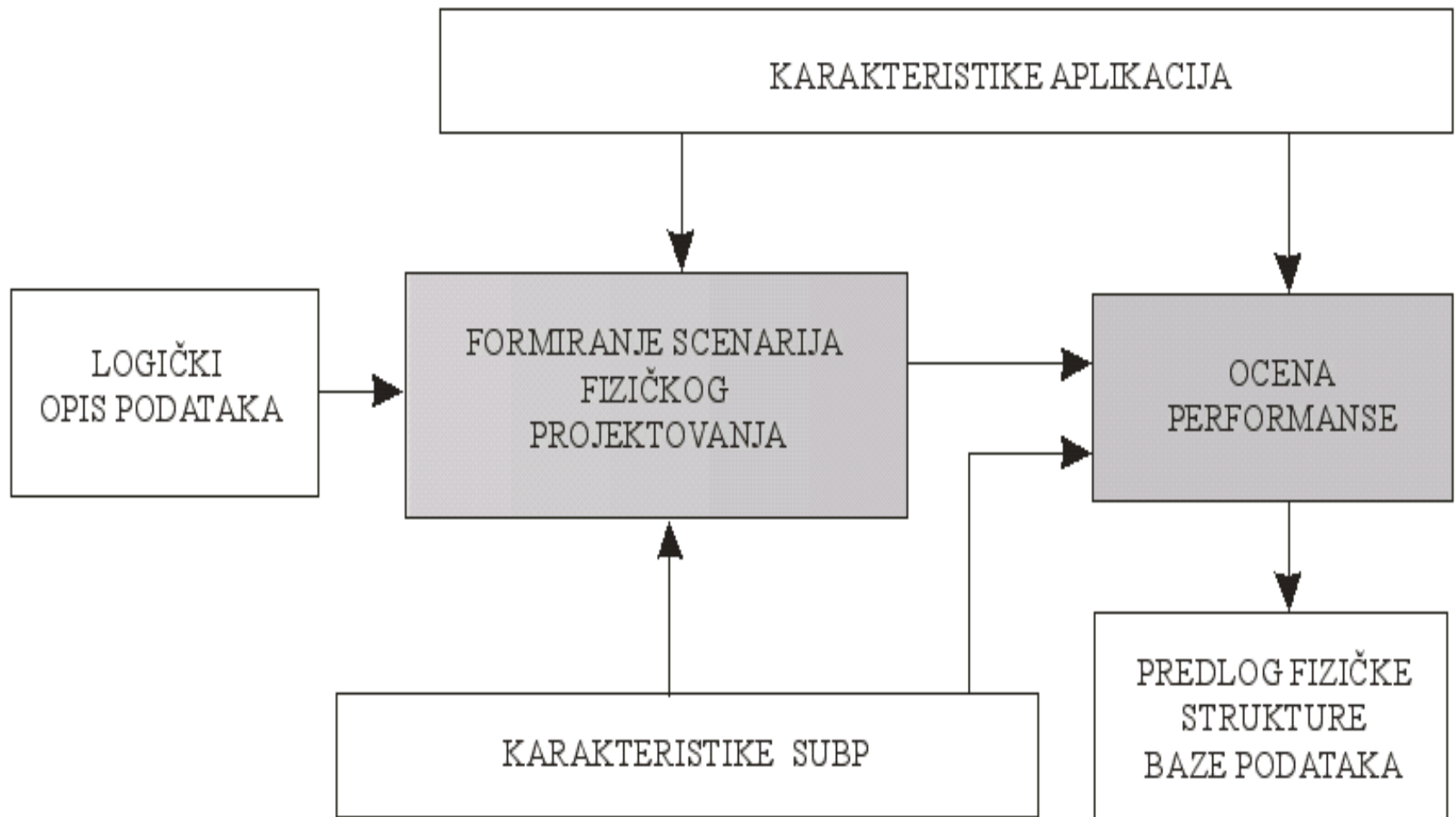
- ❑ Fizička nezavisnost podataka -  
razdvajanje logičkih tipova podataka i  
njima pridruženih operacija od fizičke  
reprezentacije

## **Posledica:**

- ❑ Logički model podataka može se  
predstaviti sa više različitih fizičkih  
struktura podataka
-



# Blok šema fizičkog projektovanja



# **Karakteristike relacionih SUBP**

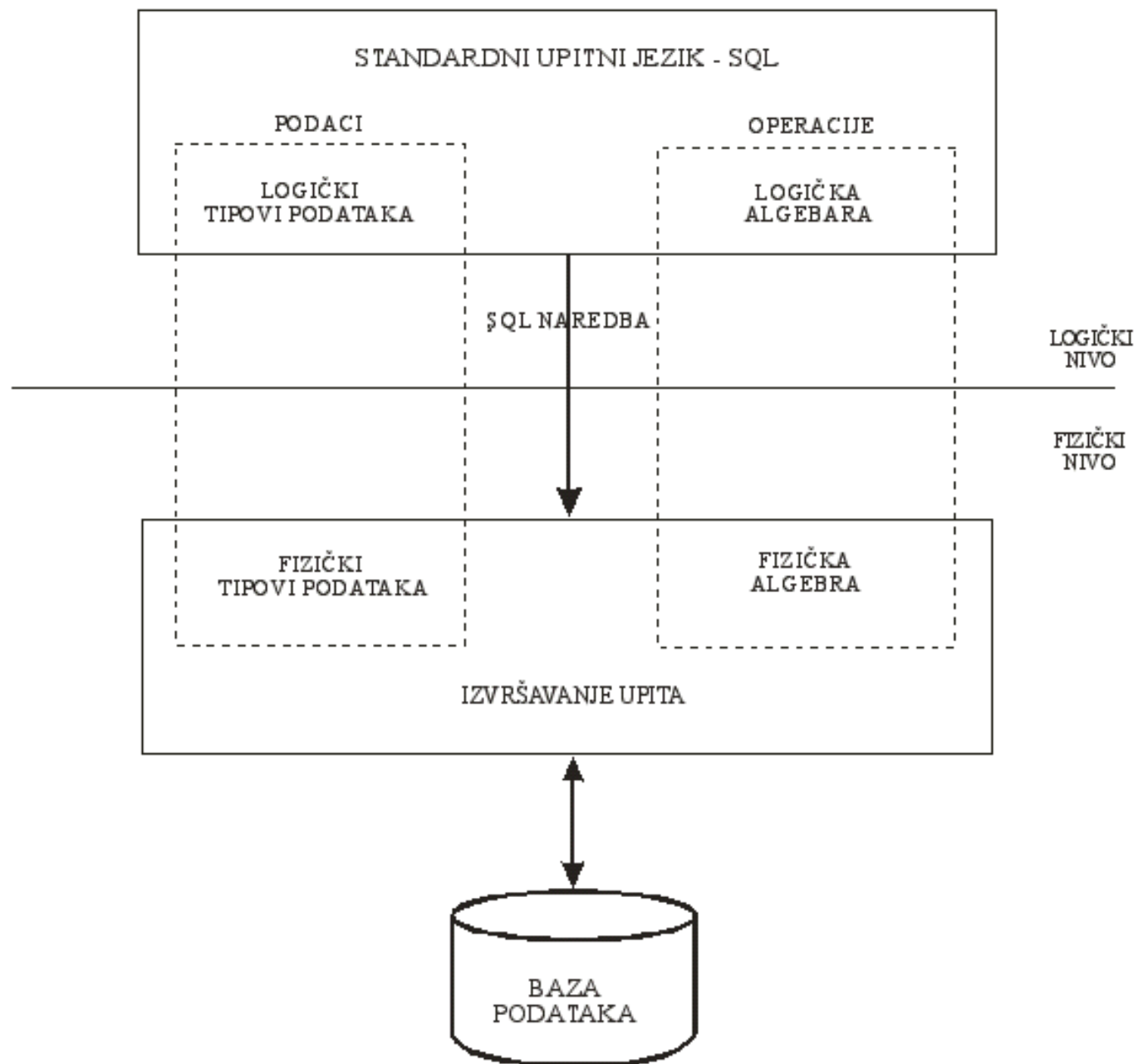
---

- ❑ Neproceduralni upitni jezik**
  - ❑ SUBP obezbeđuje mehanizam da se dođe do traženih podataka**
  - ❑ Algoritmi obrade**
    - implementacija relacionih operacija
    - implementacija funkcija (agregacija, grupisanja, eliminacija duplikata ...)
-

# **Karakteristike relacionih SUBP**

---

- **Zadatak sistema je izbor optimalnog načina izvršavanja upita** (za dati skup algoritama obrade i fizičku strukturu baze podataka)
-





# Karakteristike relacionih SUBP bitne za fizičko projektovanje

---

- fizičke strukture podataka
  - implementacija osnovnih relacionih operacija
  - implementacija operacije spoja
  - način optimizacije upita
  - ažuriranje ( $n$ -torki i *indeksa*)
-

# Implementacija operacije SELEKCIJE

---

- **za jedan uslov selekcije** koristi se
    - pristupna putanja, ili
    - linearno pretraživanje
  - **za složeni konjuktivan uslov** izbor putanje se vrši na osnovu *faktora selektivnosti* pojedinačnih uslova
  - **za složeni disjunktivan uslov** koristi se
    - linearno pretraživanje
    - više putanja i formira se unija *IDN*-ova
-

# Implementacija operacije PROJEKCIJE

---

- jednostavna kad *<lista obeležja>* sadrži primarni ključ;
  - u suprotnom mora se vršiti eliminacija duplikata; vrlo često se koristi heširanje
-

# Implementacija operacija nad skupovima

- 
- sortiranje relacija + linearni “prolaz” kroz relacije
  - može se koristiti i heširanje
  - Pri implementaciji pristupnih rutina vrši se kombinacija operacija
-

# Implementacija operacije SPOJ

---

## □ Metod ugnježenih petlji

- najjednostavniji
  - koristi se kad ne postoje pristupne strukture
  - vreme izvršavanja proporcionalno ( $n \times m$ )
-

# Implementacija operacije SPOJ

---

## □ ***Sort-Merge metod***

- zahteva uređenost relacija po vrednosti obeležja spajanja
  - kad su relacije unapred sortirane ima značajnu prednost u odnosu na metod ugnježdenih petlji
  - troškovi proporcionalni ( $n \times \log n$ )
-

# Implementacija operacije SPOJ

---

## □ Heš metod

- jedan od najefikasnijih načina implementacije
  - performansa zavisi od *heš funkcije*
  - troškovi proporcionalni  $(n + m)$
-

# Implementacija operacije SPOJ

---

## ☐ Korišćenjem posebnih struktura podataka

- efikasne indeksne strukture
  - strukture za podršku samo operacije spoja
    - ☐ *join indeksi*
    - ☐ *Bc - stablo*
    - ☐ *Kd -stablo*
    - ☐ *T - stablo*
-



# Zadatak optimizatora upita

---

- Određivanje alternativnih planova za izvršavanje upita
    - radi smanjenja troškova optimizacije
      - uglavnom se razmatra samo podskup svih mogućih planova
  - Estimacija troškova alternativnih planova i izbor plana sa najnižim troškovima
-

# **Osnovne tehnike optimizacije**

---

- ☐ Heuristička optimizacija
  - ☐ Optimizacija estimacijom troškova izvršavanja upita
  - ☐ Semantička optimizacija
-

# Heuristička optimizacija

---

- zasniva se na transformacionim pravilima;
  - polazi se od kanoničkog stabla upita;
  - vrši se transformacija polaznog, kanoničkog, stabla upita u konačno stablo upita koje ima bolju performansu;
  - vrši se izbor pristupnih rutina i algoritama za operacije upita koje je moguće primeniti za raspoložive pristupne putanje baze podataka
  - osnovno pravilo je unarne operacije izvršavati pre binarnih;
-

# Optimizacija estimacijom troškova izvršavanja upita

---

- vrši se procena i poređenje troškova različitih planova izvršavanja upita;
  - odabira se plan sa najnižim očekivanim troškovima;
  - obično se ograničava broj planova koji se pri optimizaciji razmatra da bi se trošilo manje vremena;
  - način optimizacije pogodan za kompilirane upite;
-

# **Optimizacija estimacijom troškova izvršavanja upita**

---

- kod nekih SUBP se vrši “potpuna” optimizacija kompiliranih i “delimična” optimizacija interpretativnih upita, koja zahteva manje vremena;
  - funkcija troškova je aproksimativna, pa je moguć i izbor plana izvršenja koji nije optimalan;
-

# Ažuriranje relacionih baza podataka

---

- ❑ neophodno je analizirati troškove pristupa podacima i troškove samog ažuriranja
  - ❑ ograničavajući faktor na broj pristupnih struktura je odnos smanjenja troškova koje neki indeks donosi i troškova njegovog održavanja;
-

# Ažuriranje relacionih baza podataka

---

- ❑ troškovi ažuriranja  $n$ -torki i indeksa imaju važan uticaj na rezultat procesa selekcije indeksa i na odluke optimizatora;
  - ❑ u zavisnosti od toga da li se objektima BP pristupa istim redosledom kojim su smešteni u bazu podataka ili ne primenjuju se različite formule za računanje troškova;
-

# **Korisničko fizičko projektovanje BP**

---

- ❑ Sledi posle projektovanja konceptualne šeme baze podataka;
  - ❑ Cilj je obezbediti dobru performansu često zahtevanih upita i operacija ažuriranja nad BP koja se projektuje;
-



# Korisničko fizičko projektovanje BP

---

- ❑ Polazna osnova FP je logički opis podataka koji je rezultat *logičkog projektovanja*;
  - ❑ Svaki SUBP podržava određeni skup fizičkih struktura podataka, pa se FP može definisati kao proces stvaranja efikasne fizičke strukture BP za konkretni SUBP, na osnovu kanoničke šeme;
-

# Korisničko fizičko projektovanje BP

- 
- ❑ Rezultat fizičkog projektovanja je *fizička šema* BP za konkretni (odabrani) SUBP
  - ❑ Pri FP nije cilj samo doći do fizičke strukture BP, već to uraditi na način koji garantuje dobru performansu IS
  - ❑ Pošto se korisnički zahtevi proširuju i menjaju tokom vremena, da bi se obezbedila dobra performanse uglavnom je neophodno *podešavanje* parametara BP pri uvođenju sistema u rad, a kasnije tokom eksploatacije

# Zahtevi pri fizičkom projektovanju

---

Da bi se minimizirali troškovi U/I operacija u sistemu baze podataka važno je:

- a) da strukture na sekundarnoj memoriji obezbeđuju pretraživanje relevantnih podataka preko efikasnih pristupnih putanja;
  - b) da su podaci organizovani i smešteni na sekundarnu memoriju na način koji minimizira U/I troškove pri pristupu podacima
-

# Opcije pri fizičkom projektovanju

---

- Najniži nivo FP baza podataka obuhvata:
    - izbor formata datoteka; definiše se broj datoteka BP, veličina datoteka, veličina stranice odnosno bloka, način grupisanja stranica u jedinice U/I prenosa, veličina kontinualnih proširenja ...
    - korišćenje pokazivača; podržavaju navigaciju kroz BP; unapređuju performansu pri uparivanju skupova podataka; implementiraju se kao identifikatori  $n$ -torki (*IDN*-ovi)
-

# Opcije pri fizičkom projektovanju

---

- vertikalnu podelu relacija; rezultuje smanjenjem broja pristupa sekundarnoj memoriji; grupe obeležja iste relacije smeštaju se u različite datoteke BP
  - kompresiju podataka; od interesa je iz dva razloga: smanjenje potrebnog prostora na sekundarnoj memoriji i poboljšanja performanse obrade;
-

# Opcije pri fizičkom projektovanju

---

## □ Asocijativno pretraživanje;

- koristi se da bi se smanjio broj pristupa sekundarnoj memoriji;
  - postoji u svim SUBP; što je razlog da bez obzira na odabrani SUBP **izbor indeksa** predstavlja aktivnost FP koja se uvek sprovodi bilo da se radi o projektovanju jednostavnih ili složenih baza podataka;
  - najpoznatija i najčešće korišćena indeksna struktura u relacionim bazama podataka je B-stablo
-

# Opcije pri fizičkom projektovanju

---

- mogu se koristiti tako da redosled i organizacija indeksa određuje redosled  $n$ -torki u datoteci podataka; *grupišući (klaster) indeksi*;
  - ostali indeksi se nazivaju *negrupišući*; moraju biti gusti, odnosno postoji isti broj ulaznih tačaka u indeksnoj strukturi koliko ima  $n$ -torki;
  - najveći broj SUBP pri pretraživanju indeksa ne pristupa  $n$ -torkama, pa je u nekim slučajevima do podataka moguće doći samo pretraživanjem indeksa
-

# Opcije pri fizičkom projektovanju

---

- ❑ Kontrolisano uvođenje redudantnih podataka
    - **Replikacija;** opcija FP kojom se definiše čuvanje istih delova baze podataka na više uređaja sekundarne memorije;
    - **Izvedene informacije;** odnosno materijalizovani relacioni pogledi, nastaju kao posledica smeštanja rezultata operacije spoja na sekundarnu memoriju;
-



# Opcije pri fizičkom projektovanju

---

- **Join indeksi;** u izvedene relacije se smeštaju samo neophodne informacije za efikasno izvršavanje operacije spoja; eliminišu problem ažuriranja;
  - **Denormalizacija;** replikacija pojedinačnih vrednosti obeležja radi bržeg pristupa; značajna opcija za sisteme koji ne podržavaju grupisanje po različitim tipovima podataka;
-

# Opcije pri fizičkom projektovanju

- Način smeštanja i fizički redosled podataka utiču na performansu pristupa podacima:
- **Grupisanje podataka;** organizacija elemenata podataka na sekundarnoj memoriji vrši se na takav način da se minimizira količina relevantnih podataka koji se čitaju jednom U/I operacijom i na taj način smanji broj potrebnih U/I operacija da bi se zadovoljio zahtev obrade BP;
- **Razdvajanje relacija;** a kod nekih SUBP i stranica datoteka BP po više uređaja sekundarne memorije omogućava veći broj U/I operacija u jedinici vremena, a time i veću U/I propustnost

# Završna razmatranja opcija FP

---

- ❑ Većina analiziranih opcija egzistira nezavisno od modela podataka;
  - ❑ Zbog velikog broja opcija FP predstavlja kompleksan zadatak;
  - ❑ Izvor kompleksnosti FP je i to što je većina odluka međuzavisne;
-

# **Završna razmatranja opcija FP**

---

- ❑ Još uvek mali broj istraživanja koja se odnose na formalno i automatizovano rešavanja problema FP;
  - ❑ U komercijalnim SUBP nisu sve opcije FP podjednako zastupljene i nisu sve podjednako važne;
-

# Izbor indeksa za relacionu BP

---

## □ Obuhvata:

- izbor redosleda  $n$ -torki u relaciji
- izbor obeležja za definisanje skupa indeksa

## □ Zavisi od:

- karakteristika podataka
  - karakterističnog skupa aplikacija
-

# Karakteristike podataka

---

- broj relacija;
  - očekivani broj  $n$ -torki svake relacije;
  - broj obeležja svake relacije;
  - za svako obeležje:
    - tip obeležja,
    - dužina u bajtima,
    - očekivani broj mogućih vrednosti koje obeležje dobija u  $n$ -torkama relacije;
  - veličina svake relacije izražena u broju osnovnih jedinica U/I prenosa.
-

# Opis karakterističnog skupa aplikacija

---

- ❑ spisak upita i frekvencije njihovog izvršavanja;
  - ❑ spisak ažuriranja i njihove frekvencije izvršavanja;
  - ❑ ciljnu performansu za svaki tip upita i ažuriranja
-

# **Opis upita u skupu aplikacija**

---

- kojim relacijama se pristupa;
  - koja obeležja se zahtevaju (u SELECT klauzuli);
  - koja obeležja učestvuju u izrazima za uslove selekcije i spajanja relacija (u WHERE klauzuli) i očekivanu selektivnost tih uslova.
-



# Opis ažuriranja u skupu aplikacija

---

- nad kojim obeležjima je definisan uslov selekcije ili spajanja (u WHERE klauzuli) i očekivanu selektivnost tih uslova;
  - tip ažuriranja (INSERT, DELETE ili UPDATE), i relacije koje se ažuriraju;
  - za naredbu UPDATE, obeležja čije vrednosti se modifikuju.
-

# Obeležja za definisanje pristupnih putanja

---

- ❑ Konstantni indeksi
  - ❑ Pogodna obeležja
    - obeležje nad kojim postoji predikat u naredbi, i ako sistem može koristiti (neki) indeks za obradu tog predikata (WHERE klauzula)
    - obeležja koja se pojavljuju u okviru GROUP BY i ORDER BY klauzula
-

# **Preporuke za selekciju indeksa**

---

- Postupak selekcije
  - Kriterijumi za selekciju
-

# Kriterijumi za selekciju indeksa

---

- ☐ Kada indeksirati;
  - ☐ Izbor ključa pretraživanja;
  - ☐ Složeni ključ pretraživanja;
  - ☐ Grupisanje;
  - ☐ Heširanje ili stabla indeksa;
  - ☐ Uravnoteženje troškova održavanja indeksa;
-