Vsebina

[UVOD 2](#_Toc8816631)

[Concurrency control 8](#_Toc8816632)

UVOD

**1) Kaj je SUPB?**

Sistem za Upravljanje s Podatkovnimi Bazami. Je programski sistem za shranjevanje podatkov in upravljanje z podatkovnimi bazami.  
  
**2) Datoteke vs. SUPB? Str. 8 -> Pdf. 43**

Težave pri tem, da bi imeli podatke shranjene v datotekah so naslednje:

* Nimamo dovolj velikega spomina, da lahko shranimo vse podatke. Datoteke moramo shraniti na kakšen disk ali trak, nato pa prenesti datoteke v glavni spomin, da jih lahko obdelujemo.
* Tudi če imamo dovolj velik glavni spomin z 32-bitnim sistemom lahko dostopamo samo do 4GB podatkov na enkrat. Moramo sprogramirati metodo za prepoznavanje vseh datotečnih elementov.
* Spisati moramo programe, da lahko odgovorimo vsako vprašanje, ki bi jih naj uporabnik zastavil, glede podatkov. Programi bodo zahtevni, zaradi velike količine podatkov.
* Podatke moramo zaščititi pred nedoslednimi spremembami, ki jih naredijo drugi uporabniki, ki hkrati dostopajo do podatkov. Če mora program obravnavati podatke s sočasnim dostopom in hkrati zaščiti ostale, to zelo prispeva k njihovi kompleksnosti.
* Zagotoviti moramo, da se podatki obnovijo v prvotno stanje, če se sistem zruši medtem, ko se podatki spreminjajo.
* Operacijski sistemi imajo samo geslo za zaščito. To ni dovolj prilagodljivo, da bi lahko uveljavili varnost in kontrolo dostopa, da bi lahko dovolili, da uporabniki dostopajo do svojih datotek, ki so v različnih podskupinah

SUPB, je programska oprema, ki je narejena zato, da so zgoraj naštete težave lahke za obravnavo. Ko shranimo v SUPB sistem podatke, lahko uporabimo SUPB lastnosti za njihovo obravnavo. Bolj ko naša baza raste bolj nepogrešljiv je SUPB sistem.

**3) Kaj je podatkovni model?**

Podatkovni model je zbirka konceptualnih gradnikov za opis podatkov.

Shema ja opis konkretne zbirke podatkov z uporabo danega podatkovnega modela.

Relacijski podatkovni model je najbolj pogosto uporabljan model danes.

Osnovni koncepti:

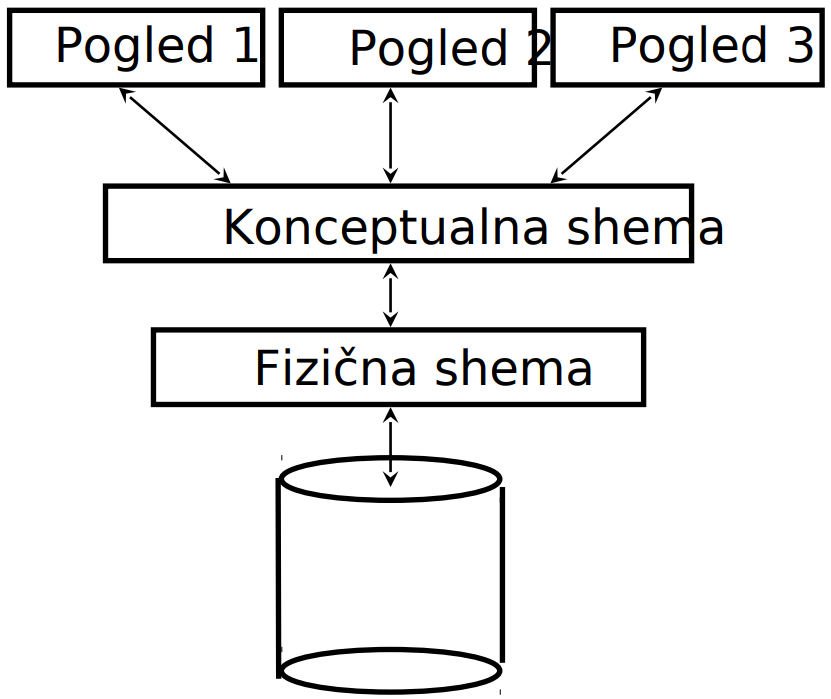
* Relacija, ki je v osnovi tabela s stolpci in vrsticami.
* Vsaka relacija ima shemo, ki opisuje stolpce in vrstice.

**4) Kaj je podatkovna neodvisnost? Str. 15 -> Pdf. 50**

Podatkovna neodvisnost je to, da so programi izolirani od sprememb, kako so podatki strukturirani in shranjeni. Podatkovna neodvisnost je pridobljena skozi tri nivoje abstrakcije podatkov: konceptualno shemo, fizično shemo in zunanjo shemo. Predvsem vlogo pri podatkovni neodvisnosti imata konceptualna in zunanja shema, ki na tem področju prineseta veliko korist.

**5) Opišite ravni abstrakcije v SUPB. Str. 12 -> Pdf. 47**

Obstajajo tri ravni abstrakcije: konceptualna, fizična in zunanja. Konceptualna shema definira logično strukturo. Fizična shema opisuje uporabljene datoteke in indekse. vPogledi" Zunanja shema opisuje kako uporabnik vidi podatke. Shema je definirana z uporabo DDL (Data definition language).

  
  
**6) Kaj je transakcija?**

Transakcija jeosnovni koncept, ki je atomarna sekvenca akcij SUPB (branje / pisanje). Vsaka transakcija, ki se izvrši mora pustiti podatkovno bazo v konsistentnem stanju, če je podatkovna baza konsistentna, ko se je transakcija začela izvajati.

* Uporabniki lahko specificirajo enostavne integritetene omejitve nad podatki in podatkovna baza bo zagotovila veljavnost omejitev.
* SUPB zares ne razume pomena podatkov (npr., ne razume kako se računajo obresti na bančnem računu).
* Torej, zagotavljanje, da transakcija ohranja konsistentnost podatkovne baze je odgovornost uporabnika.

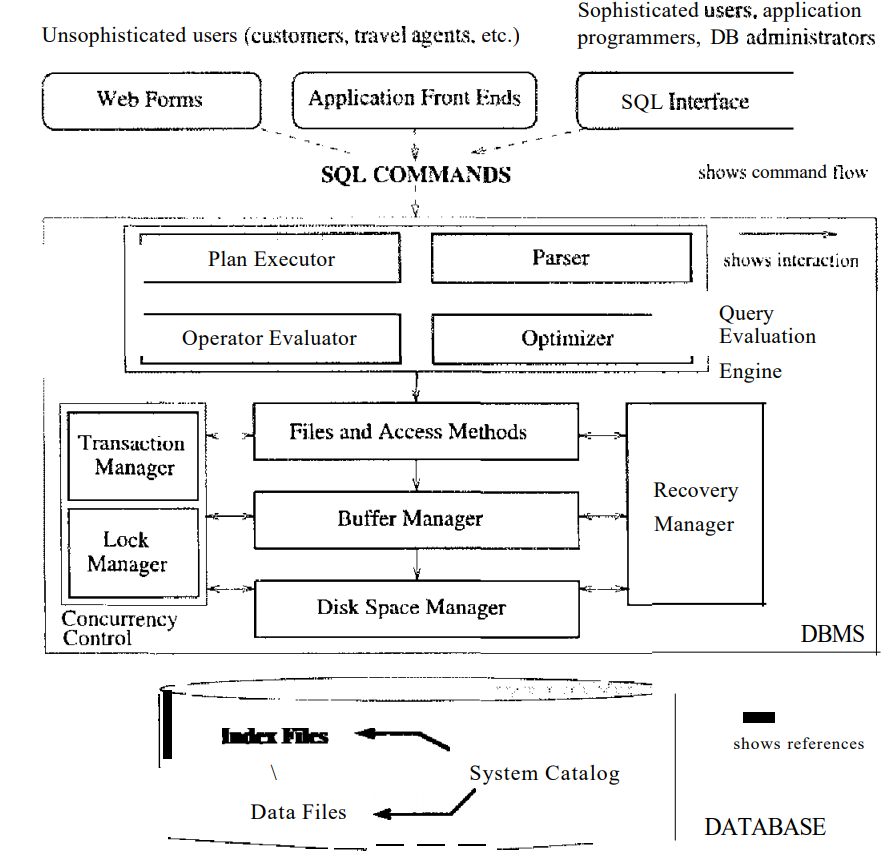
**7) Opiši strukturo SUPB. Str. 19 -> Pdf. 54**

SUPB sprejme SQL komande, ki so generirane iz več različnih uporabniških vmesnikov, uporabnik izdeluje načrte za ocenjevanje poizvedb, izvaja te načrte proti bazi podatkov in vrne odgovore. Ko uporabnik izda poizvedbo, je razčlenjena poizvedba predstavljena optimizatorju poizvedb, ki uporablja informacije o tem, kako so shranjeni podatki, da izdela učinkovit načrt izvajanja za ocenjevanje poizvedbe.

Koda, ki izvaja operacije relacij, se nahaja na vrhu datoteke in dostopa metode. Ta plast podpira koncept datoteke, ki je v SUPB zbiranje strani ali zbirka zapisov. Datotečne kopice ali datoteke neurejenih strani, kot tudi indeksi. Poleg sledenja strani v datoteki, ta plast organizira informacije na strani.

Datoteka in koda načina dostopa se nahaja na vrhu upravljalnika medpomnilnika. Prenaša strani iz diska v glavni pomnilnik, kot je potrebno za odgovor na zahtevano branje. Najnižji nivo SUPB programske opreme se ukvarja z upravljanjem prostora na disku, kjer so shranjeni podatki.

SUPB podpira vzporednost in "crash recovery" tako, da previdno načrtuje in izvaja vse kar uporabnik vnese v bazo. Baza ima tudi protokol, da se zaklene in izvaja vzdrževalna dela dokler ne konča. Sledi tudi vsem zahtevam uporabnika in jih skrbno načrtuje kdaj jih bo lahko izvedla.

  
  
RELACIJSKI PODATKOVNI MODEL  
  
**1) Kaj je relacijski model baze podatkov?**

Relacijski model predstavlja bazo podatkov kot zbirko "odnosov" oz. relacij. Relacija ni nič drugega kot tabela vrednosti. Vsaka vrsta v tabeli predstavlja zbirko povezanih podatkovnih vrednosti. Vrstice v tabeli označujejo subjekt ali razmerje v realnem svetu.

Ime tabele in imena stolpcev so v pomoč pri razlagi pomena vrednosti v vsaki vrstici. Podatki so predstavljeni kot niz odnosov. V relacijskem modelu so podatki shranjeni kot tabele. Vendar pa je fizično shranjevanje podatkov neodvisno od načina, kako so podatki logično organizirani.  
**2) Kaj so integritetne omejitve?**

Integritetne omejitve (IC) so pogoj, ki je podan v shemi baze podatkov in omejujejo podatke, ki jih je mogoče shraniti v bazo podatkov. Legalna instanca relacije, je takšna, ki zadovoljuje vse specificirane integritetne omejitve (SUPB ne bi smel dopuščati nelegalnih instanc). Če SUPB preverja integritetne omejitve, potem so shranjeni podatki bližje pomenu v realnem svetu (izogibanje napakam pri vnosu).

**3) Opiši koncept primarnega ključa.**

Primarni ključ je poseben stolpec tabele relacijske baze podatkov (ali kombinacije stolpcev), ki je narejena da enkratno identificira vse zapise tabele. Glavne značilnosti primarnega ključa so, da mora vsebovati edinstveno vrednost za vsako vrstico podatkov in ne more vsebovati ničelnih vrednosti.

Primarni ključ je množica atributov relacije, če:

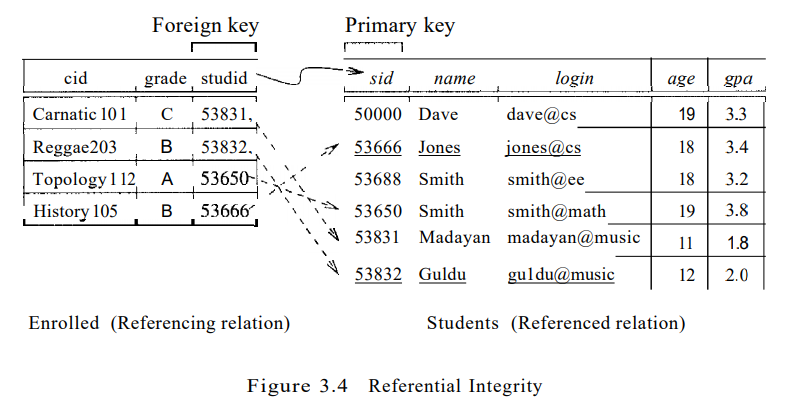
* Neobstajata dva enaka zapisa z isto vrednostjo ključa
* To ne velja za nobeno podmnožico ključa
  + Če obstaja potem temu pravimo superključ?
  + Če obstaja več kot en ključ za relacijo potem izberemo enega, ki ga imenujemo primarni ključ.

**4) Kaj je referenčna integriteta?**

Referenčna integriteta se nanaša na tuji ključ. Tuj ključ je množica atributov neke relacije, ki referencira zapise druge relacije. Izbrana množica atributov mora ustrezati primarnem ključu druge relacije. Neke vrste “logični kazalec”.

Primer: sid je tuj ključ (Studenti):

* Vpis(sid: string, pid: string, ocena: string)
* Če je integritetna omejitev tujih ključev upoštevana v podatkovni bazi potem pravimo, da dosežemo **referenčno integriteto**; ni “visečih referenc”.

  
**5) Opiši relacijsko algebro. Str. 102 -> Pdf. 137**

Relacijska algebra je ena od dveh formalnih poizvedbenih jezikov, povezanih z relacijskim modelom. Poizvedbe v algebri so sestavljene iz zbirke operaterjev. Temeljna lastnost je, da vsak operater v algebri sprejema (en ali dva) primera relacij kot argumenta in vrne primer razmerja rezultata (Vsaka operacija vrne relacijo kot rezultat.) . Ta lastnost olajša sestavljanje operaterjev za tvorbo kompleksne poizvedbe - izraz relacijske algebre je rekurzivno definiran kot relacija, operater eno člene algebre, ki se uporablja za en sam izraz, ali operator binarne algebre za dva izraza. Osnovne operacije: Selekcija ( Π ), Projekcija ( δ ), Produkt ( X ), Unija ( U ), Razlika ( - ). Dodatne operacije (niso nujne, ampak so koristne): Presek, Stik, Deljenje, Preimenovanje.  
**6) Opišite relacijski račun.**

Relacijski račun je alternativa relacijski algebri**.** V nasprotju z relacijsko algebro, ki je proceduralni relacijski račun ni nepredelan ali deklarativen, ker nam omogoča, da opišemo niz odgovorov, ne da bi bili eksplicitni o tem, kako jih je treba izračunati.  
**7) Present the basic syntax of SQL query language.**  
**8) Describe the concept of join in SQL.**  
**9) Describe aggregation functions in SQL.**  
**10) Describe the query language QBE.**  
  
Data model ER  
  
**1) Describe the data model ER.**  
**2) What is the cardinality of a relationship?**  
**3) What is a weak entity?**  
**4) Present the Chen's notation of the ER model.**  
**5) Present the constructs for modeling the generalization/specialization hierarchy of entities.**  
**6) How to model the aggregation of entities?**  
**7) Present the rules for translating the ER model into the relational database model.**  
  
Disks and files  
  
**1) Describe the architecture and the use of disks.**  
**2) Present the organization of DBMS data on disks.**  
**3) Present the main principles of RAID disks.**  
**4) Present the memory hierarchy in DBMS.**  
**5) Describe the function of a DBMS buffer pool.**  
**6) What are the possible implementations of records?**  
**7) How a table is stored in DBMS?**  
  
Indexes  
  
**1) Name and describe a few external storage devices.**  
**2) What is a file organization? What alternatives we have?**  
**3) What is an index? Describe the concepts a search key, a data entry, an index entry, and a data record.**  
**4) What are the alternatives for a data entry k\*?**  
**5) What is a primary/secondary index? What is a clustered / unclustered index?**  
**6) Present the ISAM index.**  
**7) Present the B+ tree index.**  
**8) Describe the B+ tree operations insert and delete.**  
**9) Describe hash-based indexes. What are the alternatives?**  
  
Query evaluation  
  
**1) What is the access method? What kind of access methods do you know?**  
**2) Describe common techniques used for the evaluation of relational operations.**  
**3) Present the general external merge sort algorithm. What is the complexity of the external merge sort?**  
**4) How to implement the selection operation?**  
**5) Present the methods for the implementation of the projection.**  
**6) Describe the nested loops join, the index nested loops join, and the block nested loops join.**  
**7) Present the sort-merge join algorithm.**  
**8) Describe the hash-based join algorithm.**  
  
Query optimization  
  
**1) How to estimate the cost of a query?**  
**2) Present the relational algebra equivalences?**  
**3) How to obtain all equivalent query expressions for a given query expressed in the relational algebra?**  
**4) Describe cost estimation for single-relation query plans.**  
**5) Describe left-deep and bushy join trees?**

Concurrency control  
  
**1) What is the transaction?**  
**2) Explain possible anomalies of the interleaved execution of transactions.**  
**3) What is a conflict serializable schedule of a transaction?**  
**4) Describe the strict and non-strict two-phase locking protocols.**  
**5) What happens when a transaction is aborted?**  
**6) Describe the deadlock prevention methods.**  
**7) What is index locking? How is it implemented?**  
**8) Describe the optimistic concurrency control.**   
  
Crash recovery  
  
**1) Explain the ACID properties of a relational DBMS.**  
**2) Explain the trade-offs of "stealing" pages from the buffer pool and "forcing" pages to be stored to a disk.**  
**3) Describe the Write-Ahead Logging (WAL) protocol.**  
**4) Describe what data is stored and where it is stored for the execution of the WAL protocol.**  
**5) Present the analysis, redo and undo phases of the crash recovery.**  
  
Logical design  
  
**1) Why redundancy can appear in relational databases?**  
**2) What is the functional dependency?**  
**3) How can we reason about the functional dependencies?**  
**4) What is the purpose of the normalization of a relation? What is a normal form?**  
**5) Present the Boyce-Codd normal form of a relation.**  
**6) Present the 3rd normal form (3NF) of a relation.**  
**7) What is lossless-join decomposition of a relation?**  
**8) What is dependency preserving decomposition of a relation?**  
**9) Present the algorithm for the decomposition of a relation into the BCNF.**