

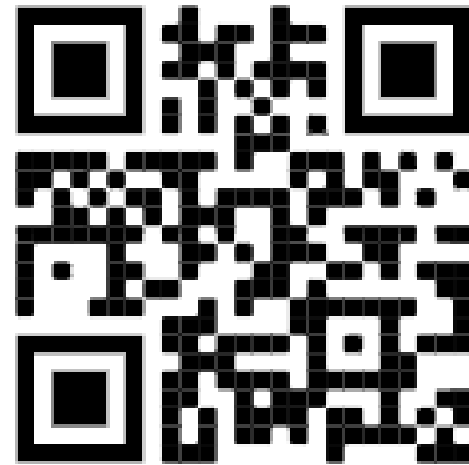


OBLIKOVANJE BAZA PODATAKA

Predavanje 06

Kviz

- <https://bit.ly/3W8uxgi>



Fizički smještaj podataka

Fizički smještaj podataka

- Logički, podaci su smješteni u tablicama
- Fizički, podaci se mogu nalaziti:
 - U radnoj memoriji
 - Na disku
- Dohvat podataka iz radne memorije je višestruko brži nego s diska
 - Analogija:
 - Ako želite popiti kavu koja je u radnoj memoriji, samo trebate otići do obližnjeg automata za kavu
 - Ako želite popiti kavu koja je na disku, trebate otići do automata u Ogulin

Princip komunikacije s diskom

- Osnovni princip rada RDBMS-a s diskom:
 1. RDBMS dohvati podatke s diska u radnu memoriju
 2. RDBMS obrađuje zadane naredbe **nad podacima u radnoj memoriji**
 - SELECT
 - INSERT
 - UPDATE
 - DELETE
 3. RDBMS vraća promijenjene podatke (ako ih ima) natrag na disk

Smještaj na disku

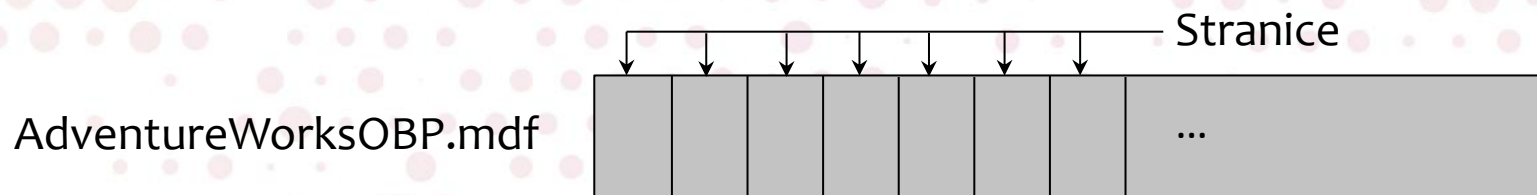
- Ovisi o RDBMS-u i operacijskom sustavu
- SQL Server podatke na disku smješta u:
 - **Primarna podatkovna** datoteka
 - Obavezna, sadržava sve objekte i podatke iz baze podataka
 - Ima ekstenziju **mdf**
 - **Zapisnička** datoteka (engl. *database log*)
 - Obavezna, koristi se za podršku transakcijama
 - Često se naziva i transakcijska datoteka (engl. *transaction log*)
 - Ima ekstenziju **ldf**
 - Opcionalne **sekundarne podatkovne** datoteke
 - Omogućavaju razmještanje podataka na više diskova i optimizaciju izrade sigurnosnih kopija
 - Imaju ekstenziju **ndf**

Organizacija podataka (1/2)

- Podatkovna datoteka je podijeljena na **stranice** (engl. *pages*)
- Dio radne memorije (engl. *buffer pool*) RDBMS-a se sastoji od **odsječaka** (engl. *slot*)
 - Svaki odsječak je toliko velik da u njega strane jedna stranica
- Stranica je osnovna količina podataka u RDBMS-u
 - Rad s diskom se naziva **IO operacija** (engl. *input output*)
 - Postoje IO operacije **čitanja** (engl. *read*) i **pisanja** (engl. *write*)
 - Čitanje kopira jednu ili više stranica iz podatkovne datoteke u odsječke u radnoj memoriji
 - Pisanje kopira jednu ili više stranica iz odsječaka iz radne memorije u podatkovnu datoteku

Organizacija podataka (2/2)

- Sve stranice i odsječki u nekoj bazi podataka imaju jednaku veličinu:
 - Na Oracleu se može birati između 4, 8, 16 ili 32 KiB
 - Na SQL Serveru je **8 KiB** (8.192 bajta)



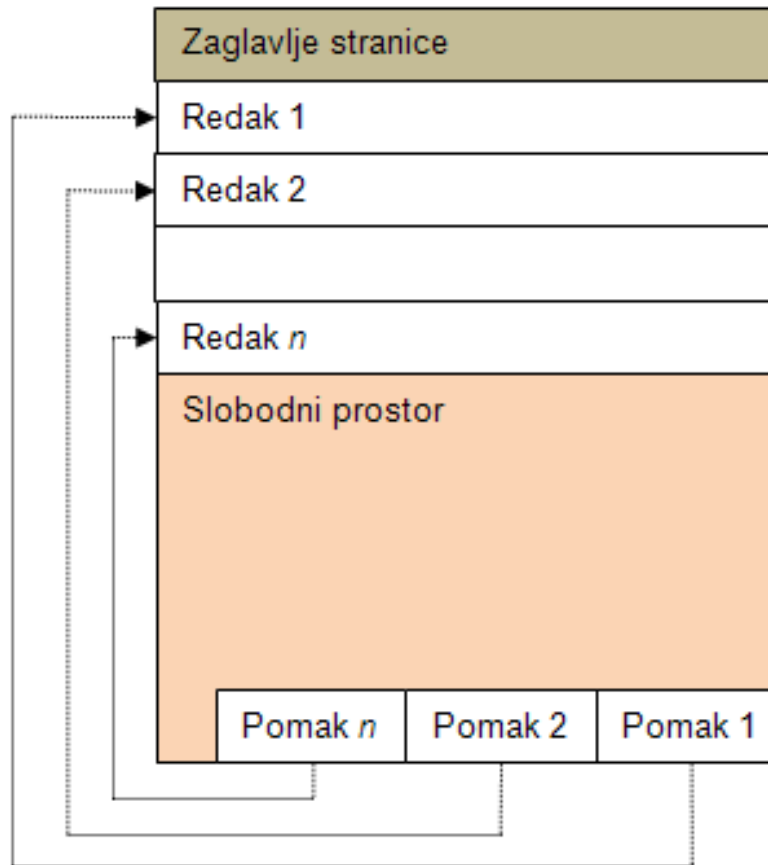
Stranice i skupovi stranica

- Na SQL Serveru u 1 MiB memorije stane 128 stranica
 - $1024 \text{ KiB} / 8 \text{ KiB} = 128$
- Na svakoj stranici se tih 8.192 bajta dijeli na:
 - Zaglavlje stranice (132 bajta)
 - **Podaci na stranici (8.060 bajtova)**
- Podaci koje stranica sadržava ovise o tipu stranice
 - Mogu biti reci iz tablice, dijelovi indeksa, ...
- 8 stranica koje se nalaze jedna iza druge u memoriji nazivamo **skup stranica** (engl. *extent*)

Tipovi stranica

- SQL Server stranica po tipu može biti:
 - **Data Page** – stranica koja na sebi čuva retke iz tablice
 - **Index Page** – stranica koja na sebi čuva ključeve indeksa
 - Large Objects (LOB) Page
 - Page Free Space
 - Index Allocation Map
 - GAM i SGAM
 - Bulk Changed Map
 - Differential Changed Map

Primjer podatkovne stranice



Podjela tipova podataka

- Podjela tipova podataka:

1. **Poznate duljine:** int, bit, datetime, ...

- Uvijek se čuvaju na podatkovnoj stranici

2. **Varijabilne duljine:** nvarchar(*n*), varchar(*n*) i varbinary(*n*)

- Ako stanu, čuvaju se na podatkovnoj stranici
- Inače se čuvaju na LOB stranici

3. **LOB:** nvarchar(max), varchar(max), varbinary(max), xml

- Čuvaju podatke do veličine od **2 GiB**
- Uvijek se čuvaju se na LOB stranicama
- Podatkovna stranica čuva pokazivače na LOB stranice

Smještaj redaka na stranicama

- Pravilo: **Svaki redak mora cijeli stati na jednu stranicu**
=> duljina svakog retka ograničena na najviše 8.060 bajtova
- Duljinu retka **određuju upisani podaci** u redak, a ne definicije stupaca
- Primjer: tablica s tri nvarchar(4000) stupca
 - Ako umetnemo jedan redak s vrijednostima 'a', 'b' i 'c'
 - Ukupna duljina retka je 6 bajtova (svako slovo po 2 bajta)
 - Sve vrijednosti ostaju na podatkovnoj stranici
 - Ako umetnemo jedan redak s vrijednostima 'aaa...a', 'bbb...b' i 'ccc...c' (svako slovo 4.000 puta)
 - Ukupna duljina retka je 24.000 bajtova
 - Sve vrijednosti se izmještaju na LOB stranice

Naredbe za analizu stranica

- Za analizu ćemo koristiti DBCC naredbe koje su proširenje T-SQL-a:

- Potrebno uključiti ispis svih DBCC poruka na ekran:

`DBCC TRACEON(3604)`

- Prikaz stranica koje pripadaju tablici:

`DBCC IND('naziv_baze', 'naziv_tablice', -1)`

- Vrijednost -1 označava podatkovne i indeksne stranice

- Prikaz detalja stranice:

`DBCC PAGE('naziv_baze', FID, PID, 3) WITH TABLERESULTS`

- Vrijednost 3 označava najviše detalja

Korištenje naredbi

- DBCC TRACEON(3604) vrijedi za cijelu konekciju
- DBCC IND će nam vratiti sljedeće bitne podatke:
 - PageFID – redni broj podatkovne datoteke
 - Kod nas će uvijek biti 1
 - PagePID – redni broj stranice unutar podatkovne datoteke
 - PageType – tip stranice
 - 1 = podatkovna stranica
 - 2 = indeksna stranica
- Za DBCC PAGE koristimo gornje podatke
 - Vraća zaglavlje i niz *slotova* koji sadržavaju retke

Primjeri

1. Odgovorite na pitanja:

- a. Na koliko stranica su smješteni podaci iz tablice dbo.Proizvod?
- b. Koji su redni brojevi tih stranica?
- c. Po kojem stupcu su poredani zapisi na stranicama?
- d. Pronađite *slot* na kojem je redak za IDProizvod = 858 (Half-Finger Gloves, S). Koja je boja tog proizvoda?
- e. Koji je zadnji redak na prvoj stranici?
- f. Koji je prvi redak na drugoj stranici?
- g. Umetnite redak u tablicu. Na koju stranicu je dodan i zašto? Pronađite ga.

B-stabla

B-stabla

- B-stablo (engl. *B-tree*) je najčešća struktura podataka za implementaciju indeksa
- Ne zna se što znači B u nazivu, moguće:
 - Balansirano
 - Široko (engl. *broad*)
 - Rudolf Bayer i Ed McCreight su smislili B-stabla dok su 1971. radili u Boeing Research Labs:
 - Boeing
 - Bayer

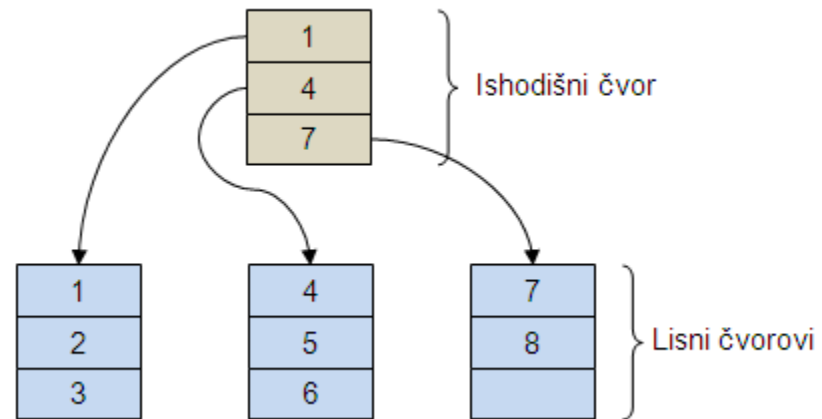
Opis B-stabla

- B-stablo je stablasta struktura se sastoji od tri tipa **čvorova**:
 - Ishodišni (engl. *root*)
 - Unutarnji (engl. *internal*)
 - Lisni (engl. *leaf*)
- Svi lisni čvorovi su na istoj dubini \Leftrightarrow stablo je balansirano
 - Dubina je udaljenost od ishodišta
- Cilj stabla: omogućiti brzo dohvaćanje **lisnih čvorova**
- Svi ostali čvorovi su samo **putokazi**

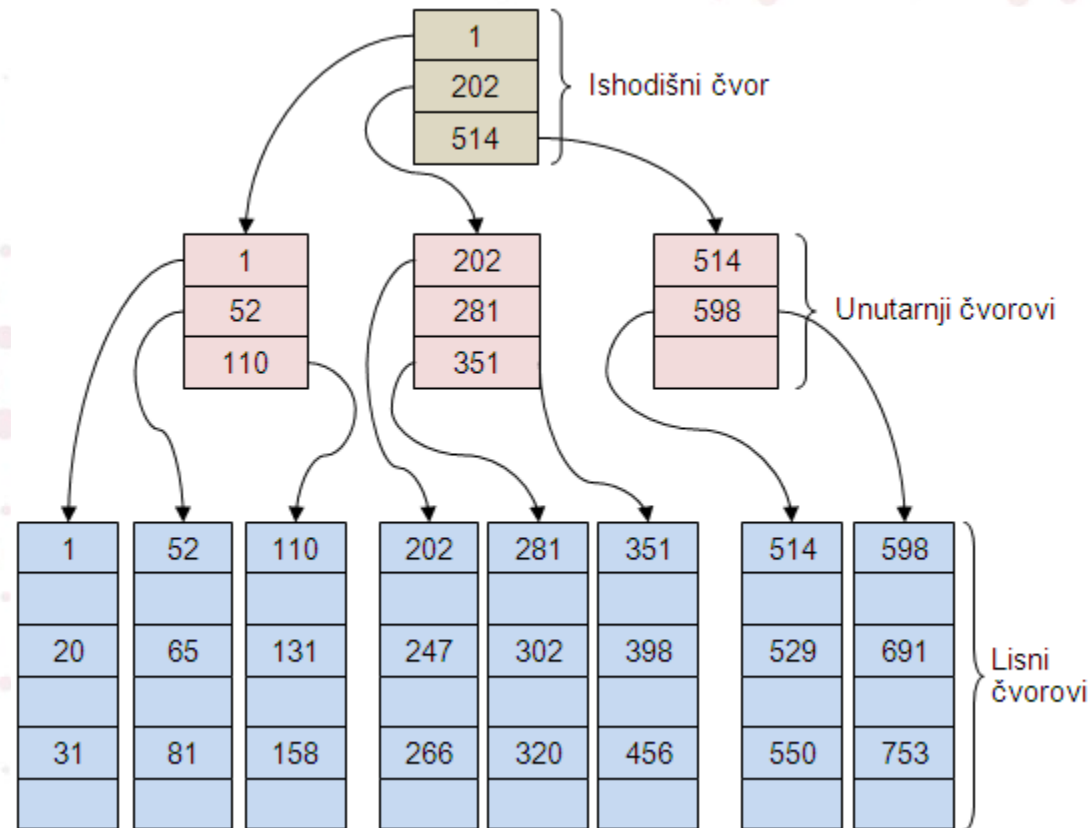
Čvorovi

- Svaki čvor B-stabla sadržava jedan ili više elemenata
- Svaki element sadržava točno jednu vrijednost
- Vrijednost elementa ovisi o vrsti čvora:
 - Elementi ishodišnog i unutarnjih čvorova sadrže putokaze
 - Svaki putokaz sadrži vrijednost prvog elementa nekog čvora razine ispod i adresu tog elementa
 - Elementi lisnih čvorova sadrže podatke
 - Podaci su uvijek sortirani prema nekom kriteriju
 - **Ti podaci su razlog postojanja stabla**

Primjer B-stabla



Primjer složenijeg B-stabla



Konstrukcija stabla

- Stablo konstruiramo od lisnih čvorova prema ishodištu:
 1. Odabrati podatke i njihov redoslijed
 - Primjerice, želimo abecedno smjestiti imena 14 zaposlenika
 2. Definirati koliko elemenata može biti u svakoj vrsti čvora
 - Primjerice, definiramo da lisni čvorovi mogu imati po tri elementa, a svi ostali po dva
 3. Smjestiti podatke u **lisne čvorove** prema odabranom kriteriju
 4. Graditi **unutarnje čvorove** prema potrebi
 5. Ponavljati korak 4 dok ne izgradimo **ishodišni čvor**

Pretraživanje stabla

- Cilj pretraživanja: pronaći podatak smješten u lisnom čvoru
- Stablo se **gradi od listova**, ali **pretražuje od ishodišta**
 - Koristimo činjenicu da su podaci u B-stablama uvijek sortirani
- Algoritam pretrage:
 1. Pretraga počinje od ishodišnog čvora.
 2. U trenutnom nelisnom čvoru pronaći najveći element koji još ima vrijednost manju/jednaku (uzlazno) ili veću/jednaku (silazno) od tražene.
 3. Pratiti putokaz.
 4. Ako putokaz vodi na nelisni čvor, ići na korak 2.
 5. Pronaći traženi podatak u lisnom čvoru.

Primjeri

1. Rasporedite brojeve od 1 do 20 u B-stablo u kojem svaki čvor može sadržavati **2 elementa**. Opišite postupak pronalaska brojeva 7, 15 i 20. Koliko je čvorova potrebno posjetiti za pronalazak svakog broja?
2. Rasporedite brojeve od 1 do 20 u B-stablo u kojem svaki čvor može sadržavati **3 elementa**. Opišite postupak pronalaska brojeva 7, 15 i 20. Koliko je čvorova potrebno posjetiti za pronalazak svakog broja?
3. Rasporedite brojeve od 1 do 20 u B-stablo u kojem svaki čvor može sadržavati **5 elemenata**. Opišite postupak pronalaska brojeva 7, 15 i 20. Koliko je čvorova potrebno posjetiti za pronalazak svakog broja?