

Sadržaj

- 1 Literatura
- 2 Arhitektura sistema baza podataka
- 3 Upitni jezik SQL

Literatura za drugu sedmicu

- C.J.Date: An Introduction to Database Systems, VIII ed, Addison Wesley Inc, 2004.
Poglavlje 2
- pisani materijal koji prati predavanja

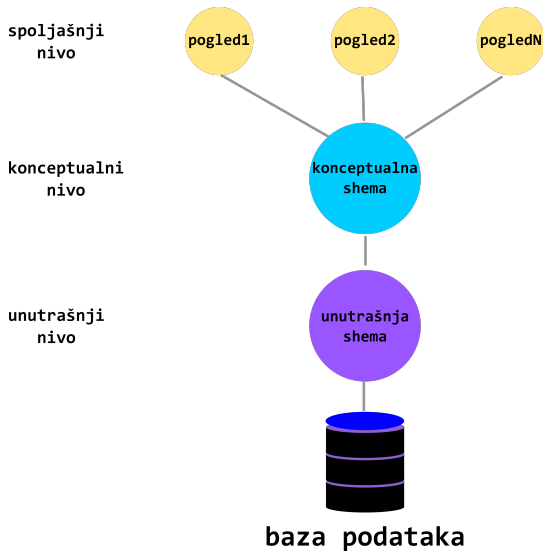
Definicija

Arhitektura sistema baza podataka je apstraktni opis njegovih komponenti i njihovih interakcija

ANSI/SPARC arhitektura

- Unutrašnji nivo (interni nivo, fizički izgled)
- Konceptualni nivo (zajednicki logički izgled, model podataka)
- Spoljašnji nivo (eksterni nivo, individualni korisnički izgled)
- ANSI - American National Standards Institute
- ANSI/SPARC - ANSI/System Planning and Requirements Committee

ANSI/SPARC arhitektura



Konceptualni nivo

- Predstavlja informacioni kontekst celokupne baze podataka
- Podaci su nezavisni od jezika i hardvera
- Konceptualni izgled je definisan konceptualnom shemom
- Konceptualna shema uključuje definicije svakog od tipova konceptualnih slogova
- Zapisuje se pomocu konceptualnog DDL-a

Unutrašnji nivo

Reprezentacija baze podataka na niskom nivou

- Sastoji se od pojava različitih tipova unutrašnjih slogova (ANSI/SPARC termin za sačuvani slog)
- Njihove karakteristike su definisane unutrašnjom shemom i zapisane pomoću unutrašnjeg DDL-a

Jos uvek je iznad fizičkog nivoa (ne radi sa adresama, blokovima podataka ili stranicama u memoriji)

Baza podataka i unutrašnji izgled

Termini

- sačuvana baza podataka = unutrašnji izgled
- definicija sačuvanih struktura = unutrašnja shema

Neki (aplikativni) programi mogu da rade nad unutrašnjim izgledom baze

- ne preporučuje se zbog sigurnosti i integriteta
- ovaj način obično koriste *utility* programi

Preslikavanje nivoa

- Preslikavanje je opis povezanosti dva nivoa
- Jedno konceptualno/unutrašnje preslikavanje
 - kako su konceptualni slogovi i polja predstavljani na unutrašnjem nivou
- Više spoljašnje/konceptualnih preslikavanja

Promene strukture

- Konceptualno/unutrašnje preslikavanje je ključno za nezavisnost podataka od promene fizičke strukture
- Spoljašnje/konceptualno preslikavanje je ključno za nezavisnost podataka od promene logičke strukture

Koraci u pristupu bazi

- Korisnik ispostavlja zahtev (npr. SQL upit)
- SUBP prihvata zahtev i analizira ga
- Da bi odredio potrebne operacije SUBP proverava spoljašnju shemu korisnika, odgovarajuće spoljašnje/konceptualno preslikavanje, konceptualnu shemu, konceptualno/unutrašnje preslikavanje i definiciju sačuvane baze podatka
- SUBP izvršava potrebne operacije (tj. zahtev korisnika) nad bazom

Funkcije SUBP-a

- Definisanje podataka (preko DDL procesora)
- Obrada podataka (preko DML procesora)
- Optimizacija izvršavanja upita
- Obezbeđivanje zaštite i integriteta podataka

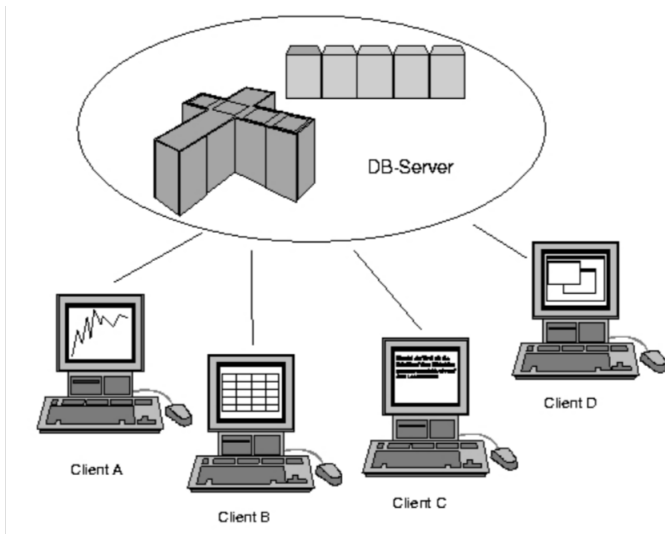
Funkcije SUBP-a

- Obezbeđivanje konkurentnog pristupa podacima i oporavka
- Formiranje rečnika podataka (repozitorijuma podataka, kataloga)
 - sadrži informacije o definiciji svih objekata (shema, preslikavanja, ograničenja, ...)
 - rečnik sadrži metapodatke (podatke o podacima)
- Obezbeđivanje što efikasnijeg rada
- SUBP takođe predstavlja korisnički interfejs ka sistemu baza podataka

Klijent-server arhitektura

- Sistem baza podataka može da se posmatra kao da ima dve komponente
 - server (u suštini SUBP)
 - klijent (ispostavlja zahteve serveru)
 - aplikacije razvijene od strane korisnika
 - alati

Klijent-server arhitektura



Utility programi

- Koriste se za različite administratorske poslove
 - spoljašnji - aplikacije specijalne namene
 - unutrašnji - deo servera
- Primeri
 - LOAD/UNLOAD/RELOAD
 - programi za statistiku, analizu, ...

Sadržaj

- 1 Literatura
- 2 Arhitektura sistema baza podataka
- 3 **Upitni jezik SQL**

Primeri

Primer 1: Prikazati jedinstvene identifikacije predmeta koji su polagani u januarskom ispitnom roku 2015. godine

```
select distinct id_predmeta
from ispit
where godina_roka=2015 and oznaka_roka='jan'
```

Primer 2: Prikazati jedinstvene identifikacije predmeta i jedinstvene ocene dobijene na njima ako su predmeti polagani u januarskom ispitnom roku 2015. godine

```
select distinct id_predmeta, ocena
from ispit
where godina_roka=2015 and oznaka_roka='jan'
```


Primeri

Primer 1: Prikazati jedinstvene identifikacije predmeta koji su polagani u januarskom ispitnom roku 2015. godine

```
select distinct id_predmeta
from ispit
where godina_roka=2015 and oznaka_roka='jan'
```

Primer 2: Prikazati jedinstvene identifikacije predmeta i jedinstvene ocene dobijene na njima ako su predmeti polagani u januarskom ispitnom roku 2015. godine

```
select distinct id_predmeta, ocena
from ispit
where godina_roka=2015 and oznaka_roka='jan'
```

Da li može: `select distinct id_predmeta, distinct ocena ... ?`

Primeri

Primer 1: Prikazati jedinstvene identifikacije predmeta koji su polagani u januarском ispitnom roku 2015. godine

```
select distinct id_predmeta
from ispit
where godina_rocka=2015 and oznaka_rocka='jan'
```

Primer 2: Prikazati jedinstvene identifikacije predmeta i jedinstvene ocene dobijene na njima ako su predmeti polagani u januarском ispitnom roku 2015. godine

```
select distinct id_predmeta, ocena
from ispit
where godina_rocka=2015 and oznaka_rocka='jan'
```

Da li može: `select distinct id_predmeta, distinct ocena ... ?`

Odgovor: Ne

Primeri

Primer 3: Za studente čije su ocene na ispitima u intervalu [6,8] prikazati broj indeksa, identifikaciju predmeta i dobijenu ocenu

```
select indeks, id_predmeta, ocena  
from ispit  
where ocena between 6 and 8
```

Primer 4: Za studente koji su na ispitima koje nisu polagali u ispitnom roku sa oznakom *jan* dobili ocene 7,8 ili 9 prikazati broj indeksa, identifikaciju predmeta i dobijenu ocenu

```
select indeks, id_predmeta, ocena  
from ispit  
where ocena in (7,8,9) and oznaka_rocka <> 'jan'
```

Uslovi restrikcije

Operator **like** se koristi za pretraživanje po nisci karaktera u obliku

izaz **like** maska

- maska se navodi između jednostrukih navodnika
- posebno značenje karaktera u okviru maske:
 - `_` označava pojavljivanje jednog karaktera
 - `%` označava pojavljivanje 0 ili više karaktera
- pravi se razlika između malih i velikih slova u okviru maske
- ukoliko se `_` ili `%` mora upotrebiti u masci onda se ispred njega navodi prekidački simbol koji se definiše sa `escape`

izaz **like** maska **escape** prekidač

Primeri

Primer 5: Prikazati imena, prezimena i mesta rođenja svih studenata ako naziv mesta njihovog rođenja

- se završava na *ad*

```
select ime, prezime, mesto_rodjenja  
from dosije  
where mesto_rodjenja like
```

- ima *r*

```
select ime, prezime, mesto_rodjenja  
from dosije  
where mesto_rodjenja like
```

Primeri

Primer 5: Prikazati imena, prezimena i mesta rođenja svih studenata ako naziv mesta njihovog rođenja

- se završava na *ad*

```
select ime, prezime, mesto_rodjenja
from dosije
where mesto_rodjenja like '%ad'
```

- ima *r*

```
select ime, prezime, mesto_rodjenja
from dosije
where mesto_rodjenja like
```

Primeri

Primer 5: Prikazati imena, prezimena i mesta rođenja svih studenata ako naziv mesta njihovog rođenja

- se završava na *ad*

```
select ime, prezime, mesto_rodjenja
from dosije
where mesto_rodjenja like '%ad'
```

- ima *r*

```
select ime, prezime, mesto_rodjenja
from dosije
where mesto_rodjenja like '%r%'
```


Primeri

Primer 5: Prikazati imena, prezimena i mesta rođenja svih studenata ako naziv mesta njihovog rođenja

- kao drugo slovo sadrži *r*

```
select ime, prezime, mesto_rodjenja
from dosije
where mesto_rodjenja like
```

- ima podnisku *m%k*

```
select ime, prezime, mesto_rodjenja
from dosije
where mesto_rodjenja like
```

Primeri

Primer 5: Prikazati imena, prezimena i mesta rođenja svih studenata ako naziv mesta njihovog rođenja

- kao drugo slovo sadrži *r*

```
select ime, prezime, mesto_rodjenja  
from dosije  
where mesto_rodjenja like '_r%'
```

- ima podnisku *m%k*

```
select ime, prezime, mesto_rodjenja  
from dosije  
where mesto_rodjenja like
```

Primeri

Primer 5: Prikazati imena, prezimena i mesta rođenja svih studenata ako naziv mesta njihovog rođenja

- kao drugo slovo sadrži *r*

```
select ime, prezime, mesto_rodjenja  
from dosije  
where mesto_rodjenja like '_r%'
```

- ima podnisku *m%k*

```
select ime, prezime, mesto_rodjenja  
from dosije  
where mesto_rodjenja like '%m+%k%' escape '+'
```

Primeri

Primer 5: Prikazati imena, prezimena i mesta rođenja svih studenata ako naziv mesta njihovog rođenja

- nema slovo o

```
select ime, prezime, mesto_rodjenja  
from dosije  
where mesto_rodjenja not like '%o%'
```

Izvedene kolone

- U select klauzuli se pored kolona mogu pojaviti i računski izrazi
- Za svaki red u rezultatu se posebno izvršava računski izraz
- Vrednosti dobijene računanjem izraza podrazumevano nemaju ime (dodeljen je redni broj kolone)
- Koloni se može dodeliti (novo) ime tako što se iza izraza (kolone) navede i ime
 - izraz ime - rezultat je kolona sa nazivom IME
 - izraz AS ime - rezultat je kolona sa nazivom IME
 - izraz as "Ime" - rezultat je kolona sa nazivom Ime
 - izraz "Ime" - rezultat je kolona sa nazivom Ime

Primeri

Primer 6: Prikazati identifikaciju i šifru predmeta, kao i dvostruku vrednost broja bodova koju taj predmet nosi

```
select id_predmeta, sifra, bodovi* 2  
from predmet
```

Primeri

Primer 7: Prikazati identifikaciju i šifru predmeta, kao i dvostruku vrednost broja bodova koju taj predmet nosi. Novodobijenu vrednost označiti kao Dvostruko

```
select id_predmeta, sifra, bodovi* 2 as dvostruko
from predmet
```

Napomena: ime kolone u rezultatu je DVOSTRUKO

ili

```
select id_predmeta, sifra, bodovi* 2 as "Dvostruko"
from predmet
```

Primeri

Primer 7: Prikazati identifikaciju i šifru predmeta, kao i dvostruku vrednost broja bodova koju taj predmet nosi. Novodobijenu vrednost označiti kao Dvostruko

```
select id_predmeta, sifra, 'Dvostruka vrednost broja bodova je =' as "Opis", bodovi* 2
as "Dvostruko"
from predmet
```


Primeri

Primer 8: Prikazati identifikaciju i šifru predmeta, kao i dvostruku vrednost broja bodova koju taj predmet nosi. Novodobijenu vrednost označiti kao Dvostruko. Izdvojiti samo predmete za koje je dvostruki broj bodova veći od 15.

```
select id_predmeta, sifra, 'Dvostruka vrednost broja bodova je =' as "Opis", bodovi* 2  
as " Dvostruko"  
from predmet  
where bodovi* 2 > 15
```

Primeri

Da li može

```
select id_predmeta, sifra, 'Dvostruka vrednost broja bodova je =' as "Opis", bodovi* 2  
as "Dvostruko"  
from predmet  
where "Dvostruko" > 15
```

Uređivanje rezultata upita

Oblik **select** naredbe je

```
select lista-kolona 3  
from ime-tabele 1  
where uslovi 2  
order by lista-kolona 4
```

Rezultat upita može biti uređen uz pomoć **order by** klauzule

Uređivanje rezultata upita

- Rezultat upita može biti uređen
 - po jednoj ili više kolona
 - u rastućem (asc) ili opadajućem (desc) redosledu
- Kolone se navode imenom ili rednim brojem iz select klauzule
- Smer uređenja se navodi iza imena kolone

Primeri

Primer 9: Prikazati sadržaj tabele predmet ureden po broju bodova u rastućem i šifri predmeta u opadajućem redosledu

```
select *  
from predmet  
order by bodovi, sifra desc
```

ili

```
select id_predmeta, sifra, bodovi, naziv  
from predmet  
order by bodovi, sifra desc
```

Primeri

Primer 9: Prikazati sadržaj tabele predmet ureden po broju bodova u rastućem i šifri predmeta u opadajućem redosledu

```
select id_predmeta, sifra, bodovi, naziv  
from predmet  
order by 3, 2 desc
```

ili

```
select id_predmeta, sifra, bodovi, naziv  
from predmet  
order by 3, sifra desc
```

Primeri

Primer 10: Prikazati identifikaciju i šifru predmeta, kao i dvostruku vrednost broja bodova koju taj predmet nosi. Novodobijenu vrednost označiti kao dvostruko. Rezultat urediti po dobijenoj vrednosti dvostrukog broja bodova u rastućem i šifri predmeta u opadajućem redosledu.

```
select id_predmeta, sifra, bodovi*2 as dvostruko
from predmet
order by 3 asc, 2 desc
```

Može li ovako?

```
select id_predmeta, sifra, bodovi* 2 as dvostruko
from predmet
order by dvostruko asc, 2 desc
```

Proizvod tabela

- Navođenjem dve (ili više) tabele u from klauzuli u obliku
from tabela1, tabela2
- Kombinuju se svi redovi jedne tabele sa svim redovima druge tabele
- U select klauzi mogu se navoditi samo imena kolona bez navođenja imena tabele ako je ime kolone jedinstveno, u sprotnom se mora navesti i ime tabele (tabela.kolona)
- kvalifikovano ime: tabela.kolona

Primeri

Primer 11: Prikazati sve moguće kombinacije studenata i predmeta koje oni mogu da odaberu.

```
select *  
from dosije, predmet
```

ili

```
select *  
from dosije CROSS JOIN predmet
```

Kako izgleda zaglavlje tabele koja se dobija kao rezultat, a kako telo?

Spajanje tabela

I način: proizvod tabela + navođenje uslova spajanja u where klauzuli

Primeri

Primer 12: Prikazati sve ispite koje su polagali pojedinačni studenti. Izveštaj urediti po broju indeksa studenta i oznaci ispitnog roka u kome je ispit polagan.

```
select *  
from dosije, ispit  
where dosije.indeks=ispit.indeks  
order by dosije.indeks, ispit.oznaka__roka  
ili
```

Primeri

```
select dosije.indeks, dosije.ime, dosije.prezime, dosije.datum_rodjenja,  
dosije.mesto_rodjenja, dosije.datum_upisa,  
ispit.id_predmeta, ispit.godina_roka, ispit.oznaka_roka,  
ispit.ocena, ispit.datum_ispita, ispit.bodovi  
from dosije, ispit  
where dosije.indeks=ispit.indeks  
order by dosije.indeks, ispit.oznaka_roka
```

ili

```
select dosije.indeks, ime, prezime, datum_rodjenja,  
mesto_rodjenja, datum_upisa,  
id_predmeta, godina_roka, oznaka_roka, ocena, datum_ispita, bodovi  
from dosije, ispit  
where dosije.indeks=ispit.indeks  
order by dosije.indeks, ispit.oznaka_roka
```

Spajanje tabela

II način: sa operatorom **join** u okviru from klauzule u obliku

```
from tabela1 [inner] join tabela2 on uslov-spajaja
```

inner join se naziva i unutrašnje spajanje

Primeri

Primer 12: Prikazati sve ispite koje su polagali pojedinačni studenti. Izveštaj urediti po broju indeksa studenta i oznaci ispitnog roka u kome je ispit polagan.

```
select *  
from dosije join ispit on dosije.indeks=ispit.indeks  
order by dosije.indeks, ispit.oznaka__roka
```

Spajanje tabela

tabela A	
a	c
1	m
2	n
3	l

tabela B	
b	c
4	m
5	n
6	r

Spajanje tabela

Šta je rezultat

```
select tabelaA.a, tabelaA.c, tabelaB.b  
from tabelaA join tabelaB on tabelaA.c=tabelaB.c
```


Spajanje tabela

Šta je rezultat

```
select tabelaA.a, tabelaA.c, tabelaB.b  
from tabelaA join tabelaB on tabelaA.c=tabelaB.c
```

a	tabelaA.c	b
1	m	4
2	n	5

Spoljašnje spajanje tabela

- rezultat unutrašnjeg spajanja - upareni redovi koji postoje u obe tabele i zadovoljavaju uslov
- spoljašnje spajanje - uključuje i vrednosti koje postoje samo u jednoj tabeli

Spoljašnje spajanje tabela

Levo spoljašnje spajanje

```
from tabelaA left [outer] join tabelaB on uslov-spajanja
```

Rezultat se sastoji od

- uparenih redova obe tabele
- redova koji se nalaze u levoj tabeli, a nemaju svog para po uslovu u desnoj tabeli
 - vrednostima kolonama iz desne tabele se dodaju nedefinisane vrednosti

Spajanje tabela

Šta je rezultat

```
select tabelaA.a, tabelaA.c, tabelaB.c, tabelaB.b  
from tabelaA left join tabelaB on tabelaA.c=tabelaB.c
```

Spajanje tabela

Šta je rezultat

```
select tabelaA.a, tabelaA.c, tabelaB.c, tabelaB.b  
from tabelaA left join tabelaB on tabelaA.c=tabelaB.c
```

a	tabelaA.c	tabelaB.c	b
1	m	m	4
2	n	n	5
3	l	null	null

Spoljašnje spajanje tabela

Desno spoljašnje spajanje

```
from tabelaA right [outer] join tabelaB on uslov-spajanja
```

Rezultat se sastoji od

- uparenih redova obe tabele
- redova koji se nalaze u desnoj tabeli, a nemaju svog para po uslovu u levoj tabeli
 - vrednostima kolonama iz leve tabele se dodaju nedefinisane vrednosti

Spajanje tabela

Šta je rezultat

```
select tabelaA.a, tabelaA.c, tabelaB.c, tabelaB.b
from tabelaA right join tabelaB on tabelaA.c=tabelaB.c
```

Spajanje tabela

Šta je rezultat

```
select tabelaA.a, tabelaA.c, tabelaB.c, tabelaB.b  
from tabelaA right join tabelaB on tabelaA.c=tabelaB.c
```

a	tabelaA.c	tabelaB.c	b
1	m	m	4
2	n	n	5
null	null	r	6

Spoljašnje spajanje tabela

Potpuno spoljašnje spajanje

```
from tabelaA full [outer] join tabelaB on uslov-spajanja
```

Rezultat se sastoji od

- uparenih redova obe tabele
- redova koji se nalaze u levoj tabeli, a nemaju svog para po uslovu u desnoj tabeli
 - vrednostima kolonama iz desne tabele se dodaju nedefinisane vrednosti
- redova koji se nalaze u desnoj tabeli, a nemaju svog para po uslovu u levoj tabeli
 - vrednostima kolonama iz leve tabele se dodaju nedefinisane vrednosti

Spajanje tabela

Šta je rezultat

```
select tabelaA.a, tabelaA.c, tabelaB.c, tabelaB.b
from tabelaA full join tabelaB on tabelaA.c=tabelaB.c
```


Primeri

Primer 13: Prikazati informacije o svim predmetima i ispitima na kojima su polagani. Izdvojiti i predmete iz kojih nisu polagani ispiti.

ili

Primeri

Primer 13: Prikazati informacije o svim predmetima i ispitima na kojima su polagani. Izdvojiti i predmete iz kojih nisu polagani ispiti.

```
select *  
from predmet left join ispit on predmet.id_predmeta=ispit.id_predmeta  
ili
```

Primeri

Primer 13: Prikazati informacije o svim predmetima i ispitima na kojima su polagani. Izdvojiti i predmete iz kojih nisu polagani ispiti.

```
select *  
from predmet left join ispit on predmet.id_predmeta=ispit.id_predmeta
```

ili

```
select * from ispit right join predmet on predmet.id_predmeta=ispit.id_predmeta
```