Uvod u relacione baze podataka

Relacioni račun. Upitni jezik SQL.

Predavanja, sedmica 4

6. novembar 2022

Sadržaj

1. Literatura
2. Relacioni račun
3. Upitni jezik SQL

Sadržaj

1 Literatura

2 Relacioni račun

3 Upitni jezik SQL

Literatura za drugu sedmicu

- C.J.Date: An Introduction to Database Systems, VIII ed, Addison Wesley Inc, 2004.
  Poglavlje 8
- pisani materijal koji prati predavanja

Napomena

Slajdovi su nastali obradom slajdova profesora Nenada Mitića za predmet Uvod u relacione baze podataka

Sadržaj

1 Literatura

2 Relacioni račun

3 Upitni jezik SQL

Relaciona račun

- Relaciona račun je drugi formalizam kojim se može opisati manipulativni deo relacionog modela
- Opisan, neproceduralan jezik
- Logički ekvivalent relacione algebre ako se posmatra deo relacionog modela podataka za obradu podataka
- Zasnovan na predikatskom računu
- Dve varijante:
  - Relaciona račun n-torki
  - Relaciona račun domena – osnova za QBE

Relaciona račun

- Opseg promenljivih
  - Promenljiva torki ima opseg iz skupa navedenih relacija i dopuštene vrednosti koje pripadaju torkama iz tih relacija
  - Promenljiva domena ima opseg iz skupa navedenih domena i dopuštene vrednosti koje pripadaju tim domenima.

Relaciona račun n-torki

- Deklaracija promenljive

RANGE OF imePromenljive IS Relacija

- Na primer
  - range of d is dosije
  - range of i is ispit

Relacioni izraz

<promenljiva>.<ime_atributa> [, <promenljiva>.<ime_atributa>]*
[WHERE <uslovni_izraz>]?

uslovni_izraz

- poređenje: $a \Theta b$, a i b - atributi ili konstante , Θ - $<, >, \leq, \geq, =, <>$
- uslovni_izraz AND uslovni_izraz
- uslovni_izraz OR uslovni_izraz
- NOT (uslovni izraz)

Primer

Primer: Izdvojiti ime i prezime za sve studente.

```
range of d is dosije
d.ime, d.prezime
```

Primer

Primer: Izdvojiti podatke o studenatima.

```
range of d is dosije
d.*
```

Primer

Primer: Izdvojiti ime i prezime studenata koji su rođeni u Beogradu.

```
range of d is dosije
d.ime, d.prezime
where mesto_rodjenja='Beograd'
```

Primer

Primer: Izdvojiti ime i prezime studenata koji su rođeni u Beogradu posle 1.1.1995.

```
range of d is dosije
d.ime, d.prezime
where mesto_rodjenja='Beograd' and datum_rodjenja > '1.1.1995'
```

Primer

Primer: Izdvojiti indekse studenata i nazive predmeta.

```
range of d is dosije
range of p is predmet
d.indeks, p.naziv
```

Primer

Primer: Izdvojiti ime i prezime studenta i identifikatore predmeta koje su položili.

```
range of d is dosije
range of i is ispit
d.ime, d.prezime, i.id_predmeta
where d.indeks=i.indeks and ocena > 5
```

Relacioni račun torki

- Kvantifikatori
  - EXISTS

    EXISTS promenljiva (uslovni_izraz)
  - FORALL

    FORALL promenljiva (uslovni_izraz)

Primer

Primer: Izdvojiti ime i prezime studenta koji ima položen neki ispit.

```
range of d is dosije
range of i is ispit
d.ime, d.prezime
where exists i (d.indeks=i.indeks and ocena > 5)
```

Primer

Primer: Izdvojiti ime i prezime studenta koji je položio sve predmete.

```
range of d is dosije
range of i is ispit
range of p is predmet
d.ime, d.prezime
where forall p ( exists i (p.id_predmeta=i.id_predmeta
                                and d.indeks=i.indeks
                                and ocena > 5))
```

Relacioni račun torki

- Slobodne promenljive (iz liste atributa)
- Vezane promenljive (uvedene uz kvantifikator)

FORALL preko EXISTS

Univerzalni kvantifikator se može izraziti preko egzistencijalnog

FORALL X (uslovni_izraz) = NOT EXISTS X (NOT uslovni_izraz)

Primer

Primer: Izdvojiti ime i prezime studenta koji je položio sve predmete.

```
range of d is dosije
range of i is ispit
range of p is predmet
d.ime, d.prezime
where not exists p ( not exists i (
                                p.id_predmeta=i.id_predmeta and
                               d.indeks=i.indeks and
                               ocena > 5))
```

Implikacija

IF <uslovni_izraz1> THEN <uslovni_izraz2>

Primer

Primer: Izdvojiti ime i prezime studenta koji je položio sve predmete od 6 espb.

```
range of d is dosije
range of i is ispit
range of p is predmet
d.ime, d.prezime
where forall p ( if p.bodovi=6 then
                                exists i (
                                     p.id_predmeta=i.id_predmeta and
                                     d.indeks=i.indeks
                                      and ocena > 5))
```

Implikacija

IF <uslovni_izraz1> THEN <uslovni_izraz2> =
NOT <uslovni_izraz1> OR <uslovni_izraz2>

Primer

Primer: Izdvojiti ime i prezime studenta koji je položio sve predmete od 6 espb.

```
range of d is dosije
range of i is ispit
range of p is predmet
d.ime, d.prezime
where forall p ( not  p.bodovi=6 or
                      exists i (
                            p.id_predmeta=i.id_predmeta and
                           d.indeks=i.indeks and
                           ocena > 5))
```

Primer

Neka je

- *uslovni_izraz1*: p.bodovi=6
- *uslovni_izraz2*: exists i (p.id_predmeta=i.id_predmeta and d.indeks=i.indeks and ocena > 5)

Primer

U prethodnom zadatku uslovni izraz možemo napisati na jedan od sledećih načina

1. forall p (if uslovni_izraz1 then uslovni_izraz2)
2. forall p (not uslovni_izraz1 or uslovni_izraz2)
3. not exists p (not (not uslovni_izraz1 or uslovni_izraz2))
4. not exists p (uslovni_izraz1 and not uslovni_izraz2)

Primer

Primer: Izdvojiti ime i prezime studenta koji je položio sve predmete od 6 espb.

```
range of d is dosije
range of i is ispit
range of p is predmet
d.ime, d.prezime
where not exists p (p.bodovi=6 or
                           not  exists i (
                                p.id_predmeta=i.id_predmeta and
                                d.indeks=i.indeks and
                                ocena > 5))
```

Relacioni račun domena

- Opseg važenja promenljivih su domeni a ne relacije
- Moguće je definisati uslov pripadnosti koji je oblika

  R (lista_parova)

  gde je
  - R je naziv relacije
  - svaki par u listi je oblika *A x* gde je *A* naziv atributa u R, a *x* ili domenska promenljiva ili ili konstanta iz domena atributa A
- Uslov je tačan akko postoji torka u relaciji R takva da je za svaki par iz liste parova poređenje A=x tačno

Primer

Neka je promenljiva

- **imex** definisana nad domenom atributa ime u relaciji dosije
- **indeksx** nad domenom atributa indeks u relaciji dosije
- **idx** nad domenom atributa id_predmeta u relaciji predmet

Primer

Primer: Naći imena studenata iz Beograda.

```
imex
where dosije( ime imex, mesto_rodjenja 'Beograd')
```

Primer

Primer: Pronaći imena i indekse studenata koji su polagali Analizu 1.

```
imex, indeksx
where exists idx ( predmet(id_predmeta idx, naziv 'Analiza 1') and
                   ispit(id_predmeta idx, indeks indeksx) and
                   dosije(indeks indeksx, ime imex))
```

Relaciona algebra ili relacioni račun

- Algebra i račun su semantički ekvivalentni
- Kodov algoritam redukcije (prikaz da je algebra moćna bar koliko i račun)
- Upitni jezik je relaciono kompletan ako je moćan bar koliko i relacioni račun (ili relaciona algebra)
- Neki upitni jezici su više zasnovani na algebri, a neki na računu
- SQL ima osobine i relacione algebre i relacionog računa

Sadržaj

1 Literatura

2 Relacioni račun

3 Upitni jezik SQL

Kvantifikovana poređenja

- skalarna vrednost se može porediti sa tabalom skalara ako se poređenje kvantifikuje sa
  - ALL
  - SOME ili ANY

Kvantifikovana poređenja sa ALL

izraz relacija ALL (podupit)

- poređenje je tačno ako rezultat podupita nije prazan i vrednost izraza je u datoj relaciji sa svim redovima iz podupita

Kvantifikovana poređenja sa SOME ili ANY

izraz relacija ANY (podupit)

- kvantifikatori SOME i ANY imaju isto značenje
- poređenje je tačno ako rezultat podupita nije prazan i vrednost izraza je u datoj relaciji sa bar jednim redom iz podupita

Primeri

Primer: Pronaći ispit koji je položen sa najvećim brojem bodova.

```
select *
from ispit
where bodovi >= all (select bodovi
                     from ispit)
```

Da li postoji problem?

Primeri

Primer: Pronaći ispit koji je položen sa najvećim brojem bodova.

```
select *
from ispit
where bodovi >= all (select bodovi
                     from ispit
                     where bodovi is not null)
```

Primeri

Primer: Pronaći ispit koji nije položen sa najvećim brojem bodova.

```
select *
from ispit
where bodovi < any (select bodovi
                    from ispit
                    where bodovi is not null)
```

Egzistencijalni kvantifikator EXISTS

EXISTS (podupit)

- izraz je tačan ako rezultat podupita nije prazan

NOT EXISTS (podupit)

- izraz je tačan ako je rezultat podupita prazan

Primeri

Primer: Prikazati broj indeksa, ime i prezime studenta koji na bar jednom ispitu dobio ocenu 10.

```
select indeks,ime,prezime
from   dosije a
where  exists (
                select *
                from   ispit b
                where  b.indeks=a.indeks
                       and ocena=10
              )
```

Primeri

Primer: Prikazati broj indeksa, ime i prezime studenta koji ni na jednom ispitu nije dobio ocenu 10.

```
select indeks,ime,prezime
from   dosije a
where not  exists (
                   select *
                   from   ispit b
                   where  b.indeks=a.indeks
                          and    ocena=10
                  )
```

Primeri

Primer: Prikazati nazive predmeta koji su polagani u nekom od ispitnih rokova iz 2015. godine, i za koje važi da niko od studenata koji su ih polagali nije pao.

```
select naziv
from predmet a
where exists ( select *
               from   ispit b
               where  b.id_predmeta=a.id_predmeta and godina_roka=2015)
and not exists ( select *
                 from   ispit b
                 where  b.id_predmeta=a.id_predmeta
                        and godina_roka=2015 and ocena=5)
```

Univerzalni kvantifikator

- U SQL-u ne postoji univerzalni kvantifikator (FORALL)
- izražava se pomoću negacije egzistencijalnog kvantifikatora

  (FORALL x) uslov = NOT EXISTS x (NOT uslov)

Primeri

Primer: Prikazati nazive predmeta koje su položili svi studenti.

```
select naziv
from   predmet p
where  not exists  (select *  from   dosije d
                    where  not exists (select * from ispit i
                                       where p.id_predmeta = i.id_predmeta
                                             and d.indeks=i.indeks
                                             and i.ocena>5
                                       )
                   )
```

Primeri

Primer: Prikazati nazive predmeta koje su položili svi studenti.

```
select naziv
from   predmet p
where  not exists  (select 1  from   dosije d
                    where  not exists (select 1 from ispit i
                                       where p.id_predmeta = i.id_predmeta
                                             and d.indeks=i.indeks
                                             and i.ocena>5
                                       )
                   )
```

Primeri

Primer: Prikazati nazive predmeta koje su položili svi studenti.

```
select naziv
from   predmet p
where  not exists  ( select *
                     from dosije d
                     where indeks not in (
                                          select indeks
                                          from ispit i
                                          where i.id_predmeta = p.id_predmeta
                                                   and i.ocena>5
                                             )
                   )
```

Klauzula VALUES

Klauzula VALUES se koristi za navođenje tabele sa konstantnim entitetima u obliku

VALUES <entitet1>, <entitet2>, ... <entitetN>

Klauzula VALUES

VALUES ('a'), ('b'), ('c')

1
a
b
c

VALUES ('a', 1), ('b', 2), ('c', 3)

1	2
a	1
b	2
c	3

Specijalni registri

U DB2 sistemu postoje specijalni registri koji čuvaju informacije koje se mogu koristiti u SQL naredbama Neki od specijalnih registara

- CURRENT SCHEMA
- CURRENT DATE (ili CURRENT_DATE)
- CURRENT TIME (ili CURRENT_TIME)
- CURRENT TIMESTAMP (ili CURRENT_TIMESTAMP) - yyyy-mm-dd-hh.mm.ss.nnnnnn
- CURRENT TIMEZONE - za vrednost koja predstavlja razliku između UTC (koordinisano univerzalno vreme) i lokalnog vremena koje je definisano na serveru
- USER

Primeri

Primer: Izlistati trenutni datum.

```
select current date  from   dosije
```

ili

```
select current date  from sysibm.sysdummy1
```

ili

```
values(current date)
```

Skalarne funkcije

Funkcija	Opis
COALESCE(arg1, arg2, ...)	vraća prvi argument iz liste argumenata čija vrednost nije NULL
VALUE(arg1, arg2, ...)	isto kao COALESCE(arg1, arg2, ...)

Primer

Primer: Za svaki ispit izdvojiti dobijenu ocenu i datum polaganja ispita. Ako je na ispitu nepoznata ocena, umesto null vrednosti ispisati -10, a ako je datum polaganja nepoznat umesto null vrednosti ispisati današnji datum.

```
select value(ocena,-10), coalesce(datum_ispita, current date)
from ispit
```

Skalarne funkcije za rad sa niskama

Funkcija	**Opis**
SUBSTR(niska_arg, početak_arg, dužina_arg)	vraća podnisku iz niska_arg dužine dužina_arg koja počinje od pozicije početak_arg
CONCAT(arg1, arg2) ili sa operatorom arg1 || arg2	spajanje niski
SPACE(arg1)	vraća arg1 praznih mesta
POSSTR(arg1, arg2)	traži gde počinje niska arg2 u arg1
REPEAT(arg1, arg2)	ponavlja nisku arg1 arg2 puta
REPLACE(arg1, arg2, arg3)	zamenjuje sva pojavljivanja niske arg2 u arg1 sa arg3

Primer

Primer: Za svaki predmet izdvojiti drugo i treće slovo iz naziva i na njih nadovezati šifru predmeta.

```
select substr(naziv,2,2)  CONCAT sifra
from predmet
```

ili

```
select concat(substr(naziv,2,2), sifra)
from predmet
```

ili

```
select substr(naziv,2,2)  || sifra
from predmet
```

Primer rezultata: naM113

Primer

Primer: Ispisati nisku koja reč *Tekst* sadrži 3 puta.

```
values (repeat('Tekst',3))
```

ili

```
select repeat('Tekst',3)
from sysibm.sysdummy1
```

Rezultat: TekstTekstTekst

Primer

Primer: Ispisati mesta rođenja studenata i umesto *Beograd* ispisati *Bg*.

```
select replace(mesto_rodjenja,'Beograd','Bg')
from dosije
```

Skalarne funkcije za rad sa niskama

Funkcija	**Opis**
LENGTH(arg1)	vraća dužinu niske arg1
LCASE(arg1) ili LOWER(arg1)	vraća nisku arg1 u kojoj su velika slova zamenjena malim slovima
UCASE(arg1) ili UPPER(arg1)	vraća nisku arg1 u kojoj su mala slova zamenjena velikim slovima
LEFT(arg1, arg2)	vraća prefiks niske arg1 dužine arg2
RIGHT(arg1, arg2)	vraća sufiks niske arg1 dužine arg2

Primer

Primer: Ispisati nazive predmeta samo sa malim slovima, a zatim samo sa velikim slovima.

```
select lcase(naziv),lower(naziv),
       ucase(naziv),upper(naziv)
from predmet
```

Primer rezultata: analiza 1 analiza 1 ANALIZA 1 ANALIZA 1

Primer

Primer: Izdvojiti prva dva slova iz naziva predmeta i poslednja dva slova iz naziva predmeta.

```
select left(naziv,2), right(naziv,2)
from predmet
```

Primer rezultata za Algoritmi i strukture podataka
Al ka

Skalarne funkcije za rad sa niskama

Funkcija	**Opis**
RTRIM(arg1)	briše beline sa kraja niske arg1 i vraća nisku koju dobije kao rezultat
LTRIM(arg1)	briše beline sa početka niske arg1 i vraća nisku koju dobije kao rezultat
TRIM(arg1)	briše beline sa početka i kraja niske arg1 i vraća nisku koju dobije kao rezultat
CHAR(arg1)	pretvara vrednosti različitih tipova u nisku koju vraća kao rezultat. Niska će imati dužinu koja zavisi od tipa argumenta
CHAR(niska_arg1, dužina_arg2)	vraća prvih dužina_arg2 karaktera niske niska_arg1

Primer

Primer:Prikazati

- broj 45 kao nisku i odrediti duzinu tako dobijene niske
- broj 475.6 kao nisku i odrediti duzinu tako dobijene niske
- nisku *Tekst* sa 10 karaktera
- nisku *Ovo je niska karaktera* sa 10 karaktera

```
select char(45), length(char(45)), length(rtrim(char(45))),
       char(475.6),length(char(475.6)),  length(rtrim(char(475.6))),
       char('Tekst',10), length(char('Tekst',10)),
       char('Ovo je niska karaktera',10)
from sysibm.sysdummy1
```

Rezultat: 45 11 2 475.6 6 5 Tekst 10 Ovo je nis

Skalarne funkcije za rad sa niskama

Funkcija	Opis
INSERT (niska_arg, početak_arg, dužina_arg, zamena_arg)	podnisku dužine dužina_arg koja počinje na poziciji početak_arg u okviru niske niska_arg zamenjuje se zamena_arg
LOCATE_IN_STRING(niska_arg1, podniska_arg2, počevšiod_arg3, brpojavljivanja_arg4)	vraća poziciju brpojavljivanja_arg4. pojavljivanja niske podniska_arg2 u okviru niske niska_arg1 počevši od pozicije počevšiod_arg3
LOCATE(podniska_arg1, niska_arg2, pozicija_arg3)	vraća poziciju prvog pojavljivanja niske podniska_arg1 u okviru niske niska_arg2 počevši od pozicije pozicija_arg3

Primer

Primer: Prikazati šifre predmeta i šifre u kojima je

- podniska od 2. do 4. pozicije zamenjena sa niskom *abab*
- na pocetak dodata niska *ss*

```
select sifra, insert(sifra,2,2,'abab'), insert(sifra,1,0,'ss')
from predmet
```

Primer rezultata:
M111 Mabab1 ssM111

Primer

Primer: Odrediti na kojoj poziciji počinje niska *ra* u reči *Abrakadabra*.

```
values posstr('Abrakadabra','ra')
```

ili

```
values locate('ra','Abrakadabra',1)
```

ili

```
values locate('ra','Abrakadabra')
```

ili

```
values locate_in_string('Abrakadabra','ra',1,1)
```

ili

```
values locate_in_string('Abrakadabra','ra')
```

Rezultat: 3

Primer

Primer:Odrediti poziciju prvog pojavljivanja niske *ra* u reči *Abrakadabra* počevši od 4. pozicije.

```
values locate('ra','Abrakadabra',4)
```

ili

```
values locate_in_string('Abrakadabra','ra',4,1)
```

ili

```
values locate_in_string('Abrakadabra','ra',4)
```

Rezultat: 10

Primer

Primer:Odrediti poziciju trećeg pojavljivanja niske *ra* u reči *Abrakadabrara*.

```
values locate_in_string('Abrakadabrara','ra',1,3)
```

Rezultat: 12