Uvod u relacione baze podataka

Upitni jezik SQL.

Predavanja, sedmica 5

6. novembar 2022

Sadržaj

Sadržaj

Napomena

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

Sadržaj

Skalarne funkcije

Funkcija	**Opis**
TRANSLATE (niska_arg1, zamenisa_arg2, zameni_arg3, podrazumevano_arg4)	vraća nisku niska_arg1 u kojoj su karakteri navedeni u zameni_arg3 zamenjena sa zamenisa_arg2. Karakter koji se menja i njegova zamena se uparuju prema poziciji. Karakterom podrazumevano_arg4 se zamenjuju karakteri u zameni_arg3 koji nemaju svog para u zamenisa_arg2

Primer

Primer: Prikazati šifre predmeta i šifre u kojima je

- 0 zamenjena sa 9
- 1 zamenjena sa 8
- 5 i 8 zamenjeni sa -

```
select sifra, translate (sifra,  '98', '0158', '-')
from predmet
```

Primer rezultata M105 M89-

Funkcije za rad sa datumima

Funkcija	**Opis**
DATE(arg1)	vraća vrednost tipa date za arg1, a datum se zadaje kao niska
CAST(arg1 as arg2)	vraća vrednosti arg1 tipa arg2
TIMESTAMP(arg1, arg2)	vraća vrednost tipa timestamp definisan argumentima od kojih arg1 predstavlja datum, a arg2 vreme
CHAR(datetime_arg, format_arg)	vraća datum ili vreme zadatih sa datetime_arg u željenom formatu koji se zadaje sa format_arg

Primer

Primer: Prikazati današnji datum u različitim formatima.

```
values (char(current date, EUR), 'EUR'),
       (char(current date, USA), 'USA'),
       (char(current date, ISO),  'ISO'),
       (char(current date, JIS), 'JIS'),
       (char(current date, LOCAL), 'LOCAL')
```

Primer rezultata
30.10.2022 EUR
10/30/2022 USA
2022-10-30 ISO
2022-10-30 JIS
10-30-2022 LOCAL

Primer

Primer: Zapisati datum 5.5.2023.

```
values date('5.5.2023')
```

ili

```
values cast('5.5.2023' as date)
```

Primer rezultata 2023-05-05

Primer

Primer: Zapisati kombinaciju datuma 5.5.2023. i vremena 1.02.

```
values timestamp('05/05/2023','1.02')
```

Primer rezultata 2023-05-05 01:02:00.0

Funkcije za rad sa datumima

Funkcija	**Opis**
YEAR(datum_arg1)	vraća godinu iz datum_arg1
MONTH (datum_arg1)	vraća mesec iz datum_arg1
DAY (datum_arg1)	vraća dan iz datum_arg1
WEEK(datum_arg1)	vraća sedmicu u godini za datum_arg1 (rezultat je u intervalu 1- 54, sedmica se broji od nedelje)

Primer

Primer: Izdvojiti za danasnji datum: redni broj nedelje u godini, godinu, mesec i dan.

```
select week(current date) redni_broj_nedelje ,
       year(current date) godina,
       month(current date) mesec,
       day(current date) dan
from sysibm.sysdummy1
```

Funkcije za rad sa datumima

Funkcija	**Opis**
DAYOFYEAR(datum_arg1)	vraća dan u godini za datum_arg1
DAYOFWEEK(datum_arg1)	vraća redni broj dana u sedmici za datum_arg1 (rezultat u intervalu od 1 do 7, gde je 1 nedelja)
DAYNAME(datum_arg1)	vraća naziv dana u sedmici za datum_arg1
MONTHNAME(datum_arg1)	vraća naziv meseca za datum_arg1
MONTHNAME(datum_arg1, zapis_arg2)	vraća naziv meseca za datum_arg1 u zapisu zapis_arg2 videti cldr.unicode.org

Primer

Primer: Odrediti redni broj dana u nedelji i ime za današnji datum.

```
select dayofweek_iso(current date) dan_u_nedelji_iso,
       dayofweek(current date) dan_u_nedelji,
       dayname(current date) ime_dana
from sysibm.sysdummy1
```

Primer rezultata: 7 1 Sunday

Primer

Primer: Ispisati ime dana za današnji datum.

```
select dayname(current_date),
       dayname(current_date,'CLDR181_sr_SR'),
       dayname(current_date,'CLDR181_sr_Latn_SR')
from sysibm.sysdummy1;
```

Primer rezultata: Sunday недеља nedelja

Funkcije za rad sa datumima

Funkcija	Opis
LAST_DAY(datum_arg1)	vraća datum poslednjeg dana meseca u kome je datum_arg1
NEXT_DAY(datum_arg1, dan_arg2)	vraća datum koji odgovara prvom datumu koji je posle datum_arg1 i čije je ime dana zadato sa dan_arg2, koji može biti MON, TUE, WED ...
DAYS(datum_arg1)	vraća celobrojnu reprezentaciju datuma

Primer

Primer: Izdvojiti

- dan u godini za današnji datum
- poslednji dan u mesecu za tekući mesec
- poslednji dan u mesecu za datum 31.12.2022.
- datum za prvu nedelju nakon današnjeg dana

```
select dayofyear(current date) dan_u_godini,
       last_day(current date) poslednji_u_mesecu,
       last_day('31.12.2022') poslednji_u_mesecu2,
       next_day(current date,'SUN') sledeca_nedelja
from sysibm.sysdummy1
```

Primer rezultata: 303 2022-10-31 2022-12-31 2022-11-06

Primer

Primer: Odrediti koliko je dana prošlo od 1.1.0001. za 1.1.0001. i za današnji dan.

```
values (days('1.1.0001') , days(current date))
```

Primer rezultata:1 738458

Funkcije za rad sa datumima

- Dva datuma se mogu oduzimati pri čemu je rezultat u obliku ymmdd
- Na neki datum je moguće dodati neki broj kojem se dodeljuje značenje pomoću reči
  - DAY, DAYS
  - MONTH, MONTHS
  - YEAR, YEARS
- Na neko vreme je moguće dodati neki broj kojem se dodeljuje značenje pomoću reči
  - SECOND, SECONDS
  - MINUTE, MINUTES
  - HOUR, HOURS

Primer

Primer: Odrediti

- koji je datum bio pre godinu dana
- koje ce vreme biti za 3 sata 20 minuta i 10 sekund

```
select current date - 1 year,
       current time + 3 hour + 20 minute + 10 seconds
from sysibm.sysdummy1
```

Primer rezultata: 2021-10-30 17:12:31

Primer

Primer: Koliko vremena je prošlo od 5.5.2020?

```
values current date - date('5.5.2020')
```

Primer rezultata:20525

Primer

Primer: Prikazati trenutno vreme u različitim formatima.

```
values (char(current time, EUR), 'EUR'),
       (char(current time, USA), 'USA'),
       (char(current time, ISO),  'ISO'),
       (char(current time, JIS), 'JIS'),
       (char(current time, LOCAL), 'LOCAL')
```

Primer rezultata:
13.44.04 EUR
01:44 PM USA
13.44.04 ISO
13:44:04 JIS
13:44:04 LOCAL

Funkcije za rad sa vremenima

Funkcija	**Opis**
TIME(arg1)	vraća vrednost tipa time, a vreme se zadaje kao niska
HOUR(arg1)	izdvaja sate iz vremena zadatog sa arg1
MINUTE (arg1)	izdvaja minute iz vremena zadatog sa arg1
SECOND (arg1)	izdvaja sekunde iz vremena zadatog sa arg1
MICROSECOND(arg1)	izdvaja mikrosekunde iz vremena zadatog sa arg1

Primer

Primer: Izdvojiti trenutno vreme i prikazati vreme 10:24.

```
select time(current timestamp), time('10:24:00')
from sysibm.sysdummy1
```

Primer rezultata: 14:07:09 10:24:00

Primer

Primer: Izdvojiti trenutne minute, sekunde i mikrosekunde.

```
select minute(current time) minuta,
       second(current time) sekundi,
       microsecond(current timestamp) mikrosekundi
from sysibm.sysdummy1
```

Primer rezultata: 7 48 501459

Funkcije za konverziju i rad sa brojevima

Funkcija	**Opis**
INTEGER(arg1)	konvertuje vrednost arg1 (broj ili niska) u tip integer
DECIMAL(arg1, arg2, arg3)	vraća decimalnu reprezentaciju arg1 (broj ili niska) pri čemu arg2 predstavlja ukupan broj cifara, a arg3 broj mesta iza decimalne tačke
DIGITS(arg1)	vraća arg1 kao nisku koja sadrži vrednost arg1 bez znaka ili decimalne tačke. Vodeće 0 će biti dodate ako je potrebno da bi rezultat imao minimalnu dužinu

Funkcije za konverziju i rad sa brojevima

Funkcija	**Opis**
DOUBLE(arg1)	konvertuje vrednost arg1 (broj ili niska) u tip double (zapis u pokretnom zarezu dvostruke tačnosti)
REAL(arg1)	konvertuje vrednost arg1 (broj ili niska) u tip real (zapis u pokretnom zarezu jednostruke tačnosti)
ROUND(arg1, arg2)	vraća vrednost arg1 zaokruženu na arg2 broj decimala ako je arg2 pozitivan broj, a ako je negativan vraća zaokružen broj arg1 na arg2 broj mesta u levo
CEIL(arg1)	vraća najmanji ceo broj koji je jednak ili veći od arg1
FLOOR(arg1)	vraća najveći ceo broj koji je jednak ili manji od arg1

Primer

Primer: Prikazati broj u formatu 8.2.

```
select   decimal('1234.5656',8,2)
from sysibm.sysdummy1
```

Rezultat: 1234.56

Primer

Primer: Prikazati samo cifre brojeva 1234.56 i broj 101 kao vrednost tipa double.

```
select digits(1234.56),  float(101), double(101)
from sysibm.sysdummy1
```

Rezultata: 123456 101.0 101.0

Primer

Primer: Prikazati rezultat primene funkcija floor i ceil na broj 65.6.

```
select floor(65.6), ceil(65.6)
from sysibm.sysdummy1
```

Rezultata: 65 66

Primer

Primer: Zaokružiti broj 873.726 na različite decimale.

```
select round(873.726,2), round(873.726,1),
       round(873.726,0),round(873.726,-1),
       round(873.726,-2), round(873.726,-3)
from sysibm.sysdummy1
```

Rezultata: 873.730 873.700 874.000 870.000 900.000 1000.000

Agregatne funkcije

Agregatne funkcije se koriste kada je potrebno izvršiti neke operacije nad svim entitetima koji ulaze u rezultat upita. Neke agregatne funkcije:

- count - za prebrojavanje entiteta
  - count(*) ili count
  - count(ime-kolone) ili count(distinct ime-kolone)
- sum(ime-kolone) - sabira vrednosti kolone numeričkog tipa
- max(ime-kolone) - računa najveću vrednost kolone
- min(ime-kolone) - računa najmanju vrednost kolone
- avg(ime-kolone) - računa srednju vrednost kolone
- stddev (ime-kolone) - računa standardnu devijaciju kolone
- correlation(ime-kolone1, ime-kolone2) - računa korelaciju vrednosti zadatih kolona

Primer

Primer: Naći ukupan broj studenata koji su upisani školske 2013/2014. godine.

```
select count(*)
from   dosije
where indeks/10000=2013
```

ili

```
select 'Ukupan broj upisanih studenata skolske 2013/2014 godine je',
       count(*)
from   dosije
where indeks/10000=2013
```

Primer

Primer: Naći ukupan broj studenata koji su upisani školske 2013/2014. godine.

ali ne može

```
select indeks,
       count(*)
from   dosije
where indeks/10000=2013
```

kao ni

```
select indeks/10000,
       count(*)
from   dosije
where indeks/10000=2013
```

Grupisanje rezultata

Klauzula GROUP BY - za grupisanjei entitete koji zadovoljavaju uslove upita prema vrednostima jedne ili više kolona, tako da u jednoj grupi budu svi entiteti koji imaju iste vrednosti u kolonama po kojima se vrši grupisanje.

4 select lista-kolona, agregatne-funkcije
1 from tabele
2 where uslovi
3 group by lista-kolona
5* order by lista-kolona

Primer

Primer: Naći ukupan broj studenata koji su upisani školske 2013/2014. godine.

```
select indeks, count(*)
from   dosije
where indeks/10000=2013
group by indeks
```

Primer

Primer: Prikazati datume kada su polagani ispiti i za svaki datum ukupan broj studenata koji su tog datuma polagali neki od ispita.

```
select datum_ispita, count(*) as "Broj studenata koji su polagali"
from   ispit
group  by datum_ispita
```

Primer

Primer: Prikazati ime i prezime studenta i ukupan broj bodova koje je student do sada sakupio.

```
select ime as "Ime", prezime as "Prezime",
       sum(p.bodovi) as "Položio bodova"
from   dosije d join ispit i on  d.indeks=i.indeks
       join predmet p on i.id_predmeta=p.id_predmeta
where  ocena>5
group  by ime,prezime
```

Primer

Primer: Naci stvarnu dužinu januarskog ispitnog roka 2015. godine, tj. interval od kada do kada su polagani ispiti u tom roku.

```
select 'Ispiti u januarskom ispitnom roku 2015. godine su polagani od',
       min(datum_ispita),
       ' do ',
       max(datum_ispita)
from   ispit
where  godina_roka=2015
and    oznaka_roka='jan'
```

Primer

Primer: Ispitati da li postoji korelacija između broja indeksa studenta i ocene koju je dobio na ispitu.

```
select correlation(indeks,ocena)
from ispit
```

Primer

Primer: Za svakog studenta prikazati ime, prezime, broj indeksa, najmanju ocenu, najveću ocenu, prosečnu ocenu i standardnu devijaciju ocena koje je dobio na ispitima u školskoj 2013/2014, uključujući i ispite koje nije položio.

```
select ime, prezime, a.indeks, max(ocena) as "Najveca ocena",
       min(ocena) as "Najmanja ocena",
       avg(ocena) as "Prosecna ocena",
       avg(ocena*1.0) as "Prosek ocena*1.0",
       dec(avg(ocena),7,2) as "dec(prosek ocena),7,2)",
       round(avg(dec(ocena,7,2)),2) as "Zaokruzena prosecna decimalna
                                        vrednost ocene" ,
       dec(round(avg(dec(ocena,7,4)),2),7,2) as "Zaokruzena prosecna
                                                 decimalna vrednost ocene
                                                 prikazana na dve decimale"
       stddev(ocena) as "Stddev(ocena)"
from   dosije a join ispit b on  a.indeks=b.indeks
group  by ime,prezime,a.indeks
```

Izdvajanje rezultata za željene grupe

Klauzula HAVING - za izbor grupa koje su od interesa, tj. grupa za koje će se prikazati podaci u rezultatu

5 select lista-kolona, agregatne-funkcije
1 from tabele
2 where uslovi
3 group by lista-kolona
4 having uslovi-za-grupe (mogu se postavljati uslovi sa agr. f-jama)
6* order by lista-kolona

Primer

Primer: Prikazati imena i prezimena studenata koji su položili bar dva ispita sa ocenom većom od 7. Izveštaj urediti po imenima i prezimenima studenata.

```
select ime, prezime, a.indeks
from   dosije a, ispit b
where  a.indeks=b.indeks and ocena>7
group  by ime, prezime, a.indeks
having count(*)>1
order by ime, prezime
```

Primer

Primer: Prikazati broj indeksa, ime i prezime studenta koji je u 2015. godini imao prosečnu ocenu na ispitima veću od 7,5.

```
select indeks,ime,prezime
from   dosije a
where  7.5 < (
               select avg(ocena*1.0)
               from   ispit b
               where  b.indeks=a.indeks and godina_roka=2015
              )
```

Primer

Primer: Prikazati uređene četrvorke (naziv_predmeta_1, naziv_roka_1, naziv_predmeta_2, naziv_roka_2) tako da važi da je predmet sa nazivom naziv_predmeta_1 u ispitnom roku naziv_roka_1 položilo više studenata nego predmet sa nazivom naziv_predmeta_2 u ispitnom roku naziv_roka_2

```
select a.naziv, c.naziv, b.naziv, d.naziv
from   predmet a, predmet b, ispitni_rok c, ispitni_rok d
where  (select count(*) from   ispit
        where  id_predmeta=a.id_predmeta and oznaka_roka=c.oznaka_roka
               and godina_roka=c.godina_roka and ocena>5
        having count(*)>0
       )
        >
       (select count(*) from   ispit
        where id_predmeta=b.id_predmeta and oznaka_roka=d.oznaka_roka
              and godina_roka=d.godina_roka and ocena>5
        having count(*)>0
       )
```

Agregatna funkcija LISTAGG

LISTAGG(ime-kolone[, separator]*) WITHIN GROUP (ORDER BY lista-kolona sa poretkom) - funkcija pravi jedan niz spajanjem niski iz zadate kolone (ili izraza).

Primer

Primer: Prikazati nisku koja sadrži imena studenata uređena prema godini njihovog rođenja.

```
select listagg(ime,' ,') within group (order by datum_rodjenja)
from   dosije a
```

Primer

Primer: Izdvojiti parove imena i prezimena studenata čije su prosečne ocene veće od polovine prosečne ocene svih studenata u tabeli ispit.

```
select A.ime,A.prezime, B.ime, B.prezime, C.prosek
from
    (select x.indeks,ime, prezime, avg(ocena*1.0) as prosek
     from   ispit x, dosije y  where  x.indeks=y.indeks
     group  by x.indeks, ime, prezime
    ) as    A,
    (select x.indeks,ime, prezime, avg(ocena*1.0) as prosek
     from   ispit x, dosije y  where  x.indeks=y.indeks
     group  by x.indeks, ime, prezime
    ) as    B,
    (select avg(ocena*1.0) as prosek from   ispit
    ) as    C
where  A.prosek>C.prosek/2 and B.prosek>C.prosek/2 and A.indeks<B.indeks
```

Imenovanje međurezultata

Klauzula WITH - za imenovanje rezultati podupita

1 with ime-rezultata1 (imena-kolona-rezultata) as (podupit1)
[,ime-rezultataN (imena-kolona-rezultata) as (podupit)]*

6 select lista-kolona, agregatne-funkcije

2 from tabele (može da se koristi ime-rezultata)

3 where uslovi

4 group by lista-kolona

5 having uslovi-za-grupe (mogu se postavljati uslovi sa agr. f-jama)

7* order by lista-kolona

Primer

Primer: Izdvojiti parove imena i prezimena studenata čije su prosečne ocene veće od polovine prosečne ocene svih studenata u tabeli ispit.

```
with  student_prosek(indeks,ime,prezime,prosek)
as
(select x.indeks,ime, prezime, avg(ocena*1.0)
 from   ispit x, dosije y
 where  x.indeks=y.indeks
 group  by x.indeks, ime, prezime
),
prosek as
(select avg(ocena*1.0) as prosek
 from   ispit
)
select A.ime,A.prezime, B.ime, B.prezime, C.prosek
from   student_prosek A, student_prosek B, prosek C
where  A.prosek>C.prosek/2 and  B.prosek>C.prosek/2 and A.indeks<B.indeks
```

Izrazi CASE

CASE izrazi omogućavaju da se na osnovu provere važenja jednog ili više uslova izabere odgovarajuća vrednost ili rezultat navedenog izraza koja će se vratiti za svaki entitet

Prvi oblik:

case <izraz>
when <vrednost1> then <rezultat1>
[when <vrednost> then <rezultat>]*
[else <rezultat>]?
end

Izrazi CASE

Drugi oblik:

case
when <uslov> then < rezultat>
[when <uslov> then < rezultat>]*
[else <rezultat>]?
end

Primer

Primer: Prikazati tabelu koja sadrži spisak predmeta i identifikacije grupa kojoj ti predmeti pripadaju pri čemu predmeti čija šifra počinje sa

- M pripadaju grupi matematičkih
- P i R grupi računarskih
- O grupi opšteobrazovnih predmeta.

Ostali predmeti pripadaju grupi predmeta sa identifikacijom 'Nepoznato'.

```
select naziv, case substr(sifra,1,1)
                  when 'M'  then 'Matematicki'
                  when 'P'  then 'Racunarki'
                  when 'O'  then 'Opsteobrazovni'
                  when 'R'  then 'Racunarski'
                  else           'Nepoznato'
              end as "Grupe Predmeta"
from   predmet
```

Primer

Primer: Prikazati tabelu koja sadrži spisak imena i prezimena studenata i godine upisa. Ako je godina upisa tekuća ili prošla godina, prikazati trenutni prosek ocena. U slučaju da je godina upisa za 6 manja od tekuće godine, prijaviti grešku. U svim ostalim slučajevima prikazati ukupan broj položenih ispita.

```
with  student_prosek(indeks,ime,prezime,prosek)
as (select x.indeks,ime, prezime, avg(ocena*1.0)
    from   ispit x, dosije y  where  x.indeks=y.indeks
    group  by x.indeks, ime, prezime )
select ime, prezime,
case
when (year(current_date)> indeks/10000 - 1)  then decimal(prosek)
when (year(current_date)> indeks/10000+6)
then RAISE_ERROR('70014','Provera da li je izgubljeno pravo na studiranje')
else  (select count(*) from ispit where indeks=a.indeks and ocena>5)
end as "Podatak"
from student_prosek a
```

Skalarna funkcija RAISE_ERROR

Funkcija RAISE_ERROR obezbeđuje da se vrati greška sa zadatim sql stanjem (eng. SQLSTATE) i zadatim objašnjenjem greške u obliku niske, dok je SQLCODE -438.

RAISE_ERROR(sql-stanje, opis-greške)

sql-stanje mora biti niska dužine 5 koja zadovoljava sledeće uslove:

- svaki karakter može biti cifra od 0 do 9 ili veliko slovo od A do Z
- klasa stanja (prva dva karaktera) ne može biti neka od već definisanih, npr. 00, 01 ili 02; može se npr. koristiti sql-stanje čija su prva dva karaktera 7, 8 ili 9.

Korisnički definisane funkcije

U SQL je moguće definisati i korisničke funkcije.
Osnovni oblik:

create function ime-funkcije (arg1 tip-arg1, arg2 tip-arg2, ... , argN tip-argN)
returns tip-povratne-vrednosti
return izraz-koji-racuna-rezultat

Primer

Primer: Napisati korisnički definisanu funkciju koja za predmet sa zadatim broj espb bodova kao argument računa cenu slušanja za samofinansirajuće studente. Cena jednog espb boda je 2000 rsd.

```
create function cena(espb_predmeta smallint)
returns float
return espb_predmeta*2000.0
```

Primer upotrebe

```
values cena(10)
```

Primer

Primer: Napisati korisnički definisanu funkciju koja za zadati id predmeta vraća broj studenata koji su taj predmet položili.

```
create function br_polozenih (id int)
returns integer
return select count(distinct indeks)
       from ispit
       where ocena>5 and id=id_predmeta
```

Primer upotrebe

```
select id_predmeta, br_polozenih(id_predmeta)
from predmet
```

Primer rezultata: 1001 7

Dodavanje entiteta

Za dodavane novih entiteta koristi se naredba INSERT koja se upotreblajva u dva oblika:

- za dodavanje konstantnih entiteta
  INSERT INTO <tabela> (<lista-kolona>)
  VALUES (<lista-vrednosti>)
- za dodavanje novih redova na osnovu rezultata upita
  INSERT INTO <tabela> (<lista-kolona>)
  [with <ime-rezultata>...]*
  select lista-kolona [, agregatne-funkcije]*
  from tabele
  [where uslovi-za-redove]*
  [group by lista-kolona]*
  [having uslovi-za-grupe]*
  [order by lista-kolona]*

Primer

Primer: Uneti u tabelu predmet podatke o predmetu Razvoj softvera, koji ima id 4005, šifru R103 i 6 espb.

```
insert into predmet
values (4005, 'R103', 'Razvoj softvera', 6)
```

Primer

Primer: Uneti u tabelu predmet podatke o predmetu Uvod u relacione baze podataka, koji ima id 4006, šifru R105 i 6 espb.

```
insert into predmet (sifra, naziv, id_predmeta)
values ('R105','Uvod u relacione baze podataka',  4006)
```

Izvršavanje prvog rešenja - greška: pokušaj dodele null vrednosti koloni koja ne može da je sadrži

```
insert into predmet (sifra, naziv, id_predmeta, bodovi)
values ('R105','Uvod u relacione baze podataka',  4006, 6)
```

Primer

Primer: U tabelu ispit uneti podatke o polaganju ispita iz predmeta Razvoj softvera (id 4005) za studente iz Beograda koji su polagani u poslednjem održanom roku i na kojima su studenti dobili ocenu 9.

```
insert into ispit (indeks, id_predmeta, godina_roka, oznaka_roka, ocena)
with poslednji_rok as(
select distinct godina_roka, oznaka_roka
from ispit
where datum_ispita = (select max(datum_ispita)
                      from ispit))
select indeks, 4005, godina_roka, oznaka_roka, 9
from dosije, poslednji_rok
where mesto_rodjenja='Beograd'
```

Promena vrednosti entiteta

Vrednosti kolona u tabelama se menjaju naredbom UPDATE koja ima oblik:

UPDATE <tabela> [AS <alias>]*
SET <dodela>
[WHERE <uslov>]*

<dodela> ima oblik

<kolona>=<izraz>

Primer

rimer: Ažurirati ispite na kojima je polagan predmet Razvoj softvera (id 4005) i postaviti da je dobijeni broj bodova 85.

```
update ispit
set bodovi = 85
where id_predmeta=4005
```

Primer

Primer: Ažurirati ispite na kojima je polagan predmet Razvoj softvera (id 4005) od strane studenata koji su rođeni pre više od 20 godina i postaviti da je na tim ispitima nepoznat broj dobijenih bodova i da su ispiti polagani 3 dana nakon posledjnjeg ispita u roku u kome su polagani.

```
update ispit as i
set (bodovi, datum_ispita) =
    (null, ( select max(datum_ispita)+ 3 days
             from ispit i1
             where i1.oznaka_roka=i.oznaka_roka
                   and i1.godina_roka=i.godina_roka))
where id_predmeta=4005
      and indeks in ( select indeks
                      from dosije
                      where datum_rodjenja < current date - 20 years)
```

Uklanjanje entiteta

Postojeći entiteti u tabeli se uklanjaju naredbom DELETE

DELETE FROM <tabela>
WHERE <uslov>

Primer

Primer: Obrisati sve ispite na kojima je polagan predmet Razvoj softvera (id 4005).

```
delete from ispit
where id_predmeta=4005
```

Primer

Primer: Obrisati sve ispite na kojima je polagan predmet Razvoj softvera (id 4005) od strane studenata koji su rođeni pre više od 20 godina.

```
delete from ispit
where id_predmeta=4005
      and indeks in ( select indeks
                      from dosije
                      where datum_rodjenja < current date - 20 years)
```

Promena vrednosti tabele na osnovu sadržaja druge tabele/tabela

Naredba MERGE ažurira ciljnu tabelu koristeći podatke iz druge tabele ili tabela (u nastavku označini kao upit). Sintaksa:

MERGE INTO <ciljna-tabela> [<alias>]*
USING <upit> [<alias>]*
ON <uslov-spajanja-redova-iz-cilja-i-upita>
[WHEN MATCHED [AND <uslov>]* THEN
<akcija1>]*
[WHEN NOT MATCHED [AND <uslov>]* THEN
<akcija2>]*

Promena vrednosti tabele na osnovu sadržaja druge tabele/tabela

<akcija1> može biti

- brisanje, i navodi se samo

  delete

- ažuriranje, kada se koristi sintaksa

  update
  set <izraz-dodele>

Promena vrednosti tabele na osnovu sadržaja druge tabele/tabela

<akcija2> može biti za unos redova i koristi sintaksu

insert [<lista-kolona>]*
values (<lista-vrednosti>)

Primer

Primer: Napraviti tabelu dosije_prosek koja ima tri kolone: indeks studenta, prosek studenta i broj položenih ispita.

```
create table dosije_prosek (

       indeks integer not null,
       prosek float,
       broj_polozenih smallint
)
```

Primer

U tabelu dosije_prosek uneti indekse studenata iz Beograda.

```
insert into dosije_prosek (indeks)
select indeks
from dosije
where mesto_rodjenja='Beograd'
```

Primer

Napisati naredbu za menjanje sadržaja tabele dosije_prosek koja

- za studente o kojima već postoje podaci i imaju prosek bar 8,5 ažurira prosek i broj položenih ispita
- briše podatke o studentima o kojima već postoje podaci u tabeli dosije_prosek, a prosek im je manji od 8,5
- unosi podatke o studentima koji imaju prosek bar 8,5 i o njima nema podataka u tabeli dosije_prosek.

Primer

```
merge into dosije_prosek dp
using ( select d.indeks, count polozeno, avg(ocena*1.0) prosek
        from dosije d join ispit i
             on d.indeks=i.indeks and ocena>5
        group by d.indeks ) as t
on dp.indeks=t.indeks
when matched and t.prosek>=8.5 then
     update
     set (prosek, broj_polozenih)=(t.prosek, t.polozeno)
when matched and t.prosek<8.5 then
     delete
when not matched and t.prosek>=8.5 then
     insert
     values(indeks, prosek, polozeno)
```

Izdvajanje željenog broja redova

Klauzula FETCH FIRST se koristi za zadavanje maksimalnog broja redova koji će se prikazati u rezultatu. Sintaksa:

FETCH FIRST <broj-redova> ROWS ONLY

i zadaje se posle order by klauzule

Primer

Primer: Prikazati prva dva sloga iz tabele dosije.

```
select *
from   dosije
fetch first 2 rows only
```