Elektrotehnički fakultet
Univerzitet u Beogradu
Katedra za računarsku tehniku i informatiku

Baze podataka 1

SQL – Zadaci

Autori:
Miloš Cvetanović
Stefan Tubić
Filip Hadžić
Tamara Šekularac

2018/2019

Veleprodajni lanac prodavnica

1. Data je šema relacione baze podataka veleprodajnog lanca prodavnica:

   PRODAVNICA(IdProd, Adresa, IdMes);

   MESTO(IdMes, Naziv);

   KLIJENT(IdKli, Naziv, IdMes);

   RACUN(IdRac, IdKli, IdProd, IdRad, Datum);

   PROIZVOD(IdProi, Naziv, Cena);

   STAVKA_RACUNA(IdSta, IdRac, IdProi, RedniBr, Kolicina, Iznos);

   RADNIK(IdRad, Ime, IdProd);

Veleprodajni lanac prodavnica

Sastaviti SQL skript koji za svaki datum, za koji je izdat bar jedan račun, daje ukupan iznos računa izdatih do tog datuma (uključujući i posmatrani datum).

```
CREATE VIEW IznosPoDatumu (Datum, Total) AS
SELECT R.Datum, SUM(S.Iznos)
FROM Racun R, StavkaRacuna S
WHERE R.IdRac=S.IdRac
GROUP BY R.Datum;

SELECT A.Datum, SUM(B.Total)
FROM IznosPoDatumu A, IznosPoDatumu B
WHERE A.Datum >= B.Datum
GROUP BY A.Datum;
```

Skladište robe

2. Data je šema relacione baze podataka za potrebe skladišta robe u toku jedne godine:

   ROBA(IdRoba, Naziv, Opis, IdDob);

   DOBAVLJAC(IdDob, Naziv, Adresa);

   NABAVKA(IdNab, Datum, Kolicina, Cena, IdRoba);

   Sastaviti SQL skript koji vraća cenu po kojoj bi trebalo prodati 40 jedinica robe sa id 3, ukoliko te robe ima u dovoljnoj količini. Cena se formira tako što se najpre rasprodaju količine koje su najskorije nabavljene (LIFO).

   Smatrati da ne postoji dve nabavke iste robe u jednom danu.

Skladište robe

```
CREATE VIEW LIFO (Proizvod, Datum, JedCena, TotalKol, TotalCena)
AS SELECT R1.IdRoba, R1.Datum, R1.Cena, SUM(R2.Kolicina), SUM(R2.Kolicina*R2.Cena)
FROM Nabavka R1, Nabavka R2
WHERE R2.Datum >= R1.Datum AND R2.IdRoba = R1.IdRoba
GROUP BY R1.IdRoba, R1.Datum, R1.Cena;

SELECT (TotalCena - ((TotalKol - 40) * JedCena))
FROM LIFO
WHERE Proizvod = 3 AND Datum = (
        SELECT MAX (Datum)
        FROM LIFO
        WHERE TotalKol >= 40
        AND Proizvod = 3
);
```

Fudbalski savez

3. Data je šema relacione baze podataka fubalskog saveza za potrebe evidencije utakmica jedne sezone (pretpostavka je da fudbaleri ne mogu da menjaju tim u kome igraju, u toku sezone):

   FUDBALER (IdFud, Ime, IdTim)

   TIM (IdTim, Naziv, Mesto)

   UTAKMICA (IdUta, IdDomaci, IdGost, Kolo, Ishod, Godina)

   IGRAO (IdFud, IdUta, PozicijaIgraca)

   GOL (IdGol, IdUta, IdFud, RedniBrGola, Minut)

   KARTON (IdKar, IdUta, IdFud, Tip, Minut)

Napomena: Ishod (1-pobeda domaćih, 2-pobeda gostiju, X-nerešeno)

Fudbalski savez

a) Sastaviti SQL skript koji kao rezultat daje šifre i imena fudbalera kao i prosečan broj golova po utakmici koji su ti fudbaleri dali, ali samo za one fudbalere koji su dali bar jedan gol nakon što su dobili žuti karton, i to u utakmici gde njihov tim bio gostujući.

Fudbalski savez

```
CREATE VIEW BrUtakmica(IdFud, BrUtakmica)
AS SELECT IdFud, COUNT(*) FROM Igrao GROUP BY IdFud;

CREATE VIEW BrGolova(IdFud, BrGolova)
AS SELECT IdFud, COUNT(*) FROM Gol GROUP BY IdFud;

CREATE VIEW DaliGolNakonKartona (IdFud)
AS SELECT F.IdFud
FROM Fudbaler F, Utakmica U, Gol G, Karton K
WHERE F.IdFud=G.IdFud AND F.IdFud=K.IdFud AND G.IdUta=K.IdUta AND G.IdUta=U.IdUta
      AND G.Minut > K.Minut  AND F.IdTim=U.IdGost AND K.Tip='zuti karton' ;

SELECT F.IdFud, F.Ime, (BG.BrGolova * 1.0) / BU.BrUtakmica
FROM Fudbaler F, BrGolova BG, BrUtakmica BU
WHERE F.IdFud=BG.IdFud AND F.IdFud=BU.IdFud
AND F.IdFud IN (SELECT IdFud FROM DaliGolNakonKartona);
```

Fudbalski savez

b) Sastaviti SQL skript koji kao rezultat daje id i imena fudbalera, ali samo za one fudbalere koji su dali bar dva gola i to na nekoj od utakmica na kojoj je njihov tim pobedio, i pri tom dva gola tog fudbalera bili ujedno i poslednji golovi na utakmici.

Fudbalski savez

```
CREATE VIEW Uslov(IdFud, IdUta) AS
  SELECT F.IdFud, U.IdUta
  FROM Fudbaler F, Utakmica U, Gol G
  WHERE F.IdFud=G.IdFud AND G.IdUta=U.IdUta
  AND ( (F.IdTim=U.IdDomacin AND Ishod='1') OR (F.IdTim=U.IdGost AND Ishod='2') )
  AND NOT EXISTS (
          SELECT IdFud
          FROM Gol L
          WHERE L.IdFud != F.IdFud AND L.IdUta = U.IdUta AND L. RedniBrGola > G. RedniBrGola
          )
  GROUP BY F.IdFud, U.IdUta
  HAVING COUNT(*) >= 2;
```

```
SELECT DISTINCT F.IdFud, F.Ime
FROM Fudbaler F, Uslov U
WHERE F.IdFud=U.IdFud;
```

Parovi za ples

4. Neka je data tabela Osoba (IdOso, Ime, Pol), pri čemu atribut Pol može imati dve vrednosti (Z-žena, M-muškarac). Potrebno je napisati SQL upit koji će vratiti na bilo koji način mešovite parove osoba koji će plesati (tj. svaki par treba da se sastoji od jedne žene i jednog muškarca). Svaka osoba sme da se pojavi najviše u jednom paru (osobe koji nemaju svoj par, nije potrebno ispisivati).

Parovi za ples

```
SELECT O1.IdOso, O1.Ime, O2.IdOso, O2.Ime
FROM Osoba O1, Osoba O2
WHERE O1.Pol = 'Z' AND O2.Pol = 'M' AND (
        SELECT COUNT(*)
        FROM Osoba O3
        WHERE O3.IdOso < O1.IdOso AND O3.Pol = 'Z'
    ) = (
        SELECT COUNT(*)
        FROM Osoba O4
        WHERE O4.IdOso < O2.IdOso AND O4.Pol = 'M'
    );
```

Matrica

5. Data je šema relacione baze podataka, za potrebe izračunavanja pomoću matrica:

   MATRICA (SifM, Naziv, BrVrsta, BrKolona);

   PODACI (SifP, I, J, Vrednost, SifM);

Smatrati da je u tabeli PODACI definisan svaki element matrice.

Sastaviti SQL skript koji vraća vrstu, kolonu i vrednost svakog od elemenata matrice nastale množenjem matrica sa nazivima ‘A’ i ‘B’.

Matrica

```
SELECT A.I, B.J, SUM(A.Vrednost * B.Vrednost)
FROM Matrica MA, Podaci A, Matrica MB, Podaci B
WHERE    MA.Naziv = 'A' AND MA.SifM = A.SifM AND
         MB.Naziv = 'B' AND MB.SifM = B.SifM AND
         A.J = B.I
GROUP BY A.I, B.J;
```

Raspored časova

6. Data je šema relacione deo baze podataka fakulteta:

STUDENT (SifS, Ime, BrIndeksa)

PROFESOR (SifP, Ime, SifO)

ODSEK (SifO, Naziv)

KURS (SifK, Naziv, BrKredita, SifO)

UCIONICA (SifU, BrMesta)

PREDUSLOV (SifK, SifKP)

POHADJA(SifS, SifR)

RASPORED (SifR, SifP, SifK, SifU, Termin, Dan, BrPrijavljenih)

Raspored časova

Sastaviti SQL skript koji proverava kvalitet rasporeda studenata i pored broja indeksa ispisuje i odgovarajuću poruku.

Raspored studenta je loš ukoliko u svom rasporedu bar jednog dana ima prekid u terminima u kojima prati predavanje, u suprotnom raspored se smatra dobrim. Za sve studente sa lošim rasporedom treba ispisati broj indeksa i poruku LOS, a pored broja indeksa studenata sa dobrim rasporedom poruku DOBAR.

Smatrati da su termini uvek dva sata i da su raspoređeni na sledeći način:

Termin	Vreme
1	08:00-10:00
2	10:00-12:00
3	12:00-14:00
...	...

Raspored časova

```
WITH LosRaspored (SifS, Dan) AS(
	SELECT SifS, Dan
	FROM Pohadja NATURAL JOIN Raspored
	GROUP BY SifS, Dan
	HAVING COUNT(Termin) < (MAX(Termin)-MIN(Termin)+1)
)
SELECT S.BrIndeksa, CASE
                        WHEN S.SifS IN (SELECT SifS FROM LosRaspored) THEN 'LOS'
                        ELSE 'DOBAR'
                    END AS Status
FROM STUDENT S
WHERE S.SifS IN (SELECT SifS FROM Pohadja);
```