Elektrotehnički fakultet Sarajevo
Odsjek za računarstvo i informatiku

Zadaća 2

VAŽNO: FORMAT ZADAĆE PAŽLJIVO PROČITATI

Zadataci zahtjevaju da popunite c2 kviz i pošaljete na zamger text SQL upita. Sve SQL upite i ostale SQL skripte treba sastaviti u jedan tekst i ubaciti u tekstualno polje zadatka na zamgeru.
Primjer izgleda upita:

```
--1.
Select * from drzava;
--2.
Select * from drzava;
--3.
--4.
--5.
--6.
--7.
--8.
--9.
--10.
```

Da bi zadatak bio bodovan potrebno je da je popunjen c2 dio i poslan zamger dio. Odgovaranje na pitanje c2 dijela bez zamger dijela će biti kažnjavano negativnim poenima. Poslije slanja dobro provjeriti da li su sve datoteke ispravno poslane. Slanje zadaće van roka se neće uzeti u razmatranje. Zadaću je moguće popuniti-slati više puta i rok je dovoljno dug tako da nema potrebe čekati 23:58 da se pošalje zadaća. Pošaljite je dan ranije kako bi stigli ispraviti sve eventualne tehničke poteškoće na vrijeme. Rok zadaće je objavljen na informacionom sistemu zamger.

Zadatak 1 (5b)

Ako se prebacite na schema Erd komandom ALTER SESSION SET CURRENT_SCHEMA = erd; pronaći ćete bazu podataka za prodaju proizvoda. Za datu bazu podataka poznato je da ima tabele:
KONTINENT, DRZAVA, GRAD, LOKACIJA, FIZICKO_LICE, PRAVNO_LICE, PROIZVODJAC, KURIRSKA_SLUZBA, UGOVOR_ZA_PRAVNO_LICE, KUPAC, UPOSLENIK, ODJEL, UGOVOR_ZA_UPOSLENIKA, SKLADISTE, KATEGORIJA, POPUST, PROIZVOD, KOLICINA, GARANCIJA, ISPORUKA, NARUDZBA_PROIZVODA i FAKTURA
i poznato je da:
PROIZVODJAC i KURIRSKA_SLUZBA imaju primary key koji je ujedno i foreign key na PRAVNO_LICE
KUPAC i UPOSLENIK imaju primary key koji je ujedno i foreign key na FIZICKO_LICE.

Napisati upit koji će:

1. Ispisati nazive bez ponavljanja svih pravnih lica za koje postoji fizičko lice na istoj lokaciji.
2. Ispisati bez ponavljanja datum potpisivanja ugovora (u formatu dd.MM.yyyy) i naziv pravnog lica za sve ugovore kod kojih je datum potpisivanja poslije prvog datuma kupoprodaje faktura koje sadrže proizvod kod kojeg broj mjeseci garancije nije null.
3. Ispisati nazive proizvoda čija je kategorija jednaka bar jednoj kategoriji proizvoda čija je ukupna količina jednaka maksimalnoj od ukupnih količina svakog proizvoda posebno.

4. Ispisati nazive proizvoda i nazive proizvođača za sve proizvode za čijeg proizvođača postoji proizvod čija je cijena proizvoda veća od prosjeka cijena svih proizvoda.
5. Ispisati ime i prezime kupaca koji su istovremeno uposlenici i sumu potrošenog iznosa na fakture koje su platili za svakog od njih (grupisani po imenu PA prezimenu) čija je suma iznosa veća od prosjeka (zaokruženog na dvije decimale) suma iznosa faktura fizičkih lica (grupisanih po imenu PA prezimenu).
6. Ispisati naziv kurirske službe čija je suma količine jednog proizvoda u njenim narudžbama gdje ima popusta jednaka maksimalnoj sumi količine jednog proizvoda u narudžbama svih kurirskih službi (suma grupisano po id kurirske službe) gdje ima popusta.
7. Ispisati ime i prezime svakog kupca (grupisano po imenu i prezimenu zajedno) i uštedu ostvarenu na sve popuste u njegovim fakturama (izračunatu preko količine jednog proizvoda).
8. Ispisati sve isporuke bez ponavljanja isporuke čije fakture imaju popust i broj mjeseci garancije.
9. Ispisati naziv i cijenu proizvoda čija je cijena veća od prosjeka (zaokruženog na dvije decimale) maksimalnih cijena proizvoda iz svake kategorije.
10. Ispisati naziv i cijenu svih proizvoda čija je cijena manja od svih prosječnih cijena svake kategorije koja nije podkategorija kategorije tog proizvoda.

U datoteci Provjera.sql se nalaze upiti koji vam pomažu da provjerite da li je vaš upit tačan (Obratite pažnju da morate tačno ispoštovati kako se trebaju zvati kolone). Zadatak provjeravate tako što dio ZAMJENA zamjenite sa vašim upitom i pokrenete upit. Kao rezultat upita dobiti ćete broj. Npr ako je dat upit:

```
SELECT SUM(LENGTH(DRZAVA)*3)
FROM (ZAMJENA);
```

A trebate ispisati nazive država onda je potrebno napisati upit koji ako vraća rezultat 183 upit je ispravan:

```
SELECT SUM(LENGTH(DRZAVA)*3)
FROM (SELECT naziv AS Drzava FROM Drzava);
```

Rezultati upita su:
1. 207
2. 402
3. 51
4. 504
5. 6897
6. 18
7. 17709
8. 243
9. 9210
10. 2448

Za slanje na c2 trebate dobiti brojeve na isti način samo sto koristite drugu datoteku pod nazivom SLANJE.sql

Zadatak 2 (5b.)

Zadatak 2 se postavlja odmah ispod zadatka 1 sa komentarom --Zadatak 2. Na slici 1 su date tri tabele pri čemu su null vrijednosti prikazane simbolom "-". Potrebno je kreirati ove tri tabele i popuniti ih odgovarajućim podacima. Uslovi za tabele su da je FKTabelaA foreign key na id tabele *tabelaA* i ne smije biti null a FkTabelaB foreign key na id tabele *tabelaB* i smije biti null. *RealniBroj* od tabelaA mora biti veći od 5 i *cijeliBroj* ne smije biti između 5 i 15. *CijeliBroj* od tabelaB se ne smije ponavljati. *Naziv* i *cijeliBroj* tabele tabelaC ne smiju biti null. Foreign key ograničenje na tabelu *TabelaB* je potrebno nazvati "FkCnst".

TabelaA

ID	NAZIV	DATUM	CIJELIBROJ	REALNIBROJ
1	tekst	-	-	6.2
2	-	-	3	5.26
3	tekst	-	1	-
4	-	-	-	-
5	tekst	-	16	6.78

TabelaB

ID	NAZIV	DATUM	CIJELIBROJ	REALNIBROJ	FKTABELAA
1	-	-	1	-	1
2	-	-	3	-	1
3	-	-	6	-	2
4	-	-	11	-	2
5	-	-	22	-	3

TabelaC

ID	NAZIV	DATUM	CIJELIBROJ	REALNIBROJ	FKTABELAB
1	YES	-	33	-	4
2	NO	-	33	-	2
3	NO	-	55	-	1

Slika1

Pozivanjem sljedećih 10 insert komandi jednom za drugom utvrditi za svaku komandu da li se može izvršiti ili ne i zašto. Svaka komanda se poziva nad tabelom koja ima rezultujuće podatke prethodnih komandi koje su se uspješno izvršile.

```
INSERT INTO TabelaA (id,naziv,datum,cijeliBroj,realniBroj) VALUES (6,'tekst',null,null,6.20);
INSERT INTO TabelaB (id,naziv,datum,cijeliBroj,realniBroj,FkTabelaA) VALUES (6,null,null,1,null,1);
INSERT INTO TabelaB (id,naziv,datum,cijeliBroj,realniBroj,FkTabelaA) VALUES (7,null,null,123,null,6);
INSERT INTO TabelaC (id,naziv,datum,cijeliBroj,realniBroj,FkTabelaB) VALUES (4,'NO',null,55,null,null);
Update TabelaA set naziv = 'tekst' Where naziv is null and cijeliBroj is not null;
Drop table tabelaB;
Delete from TabelaA where realniBroj is null;
Delete from TabelaA where id = 5;
Update TabelaB set fktabelaA = 4 where fktabelaA = 2;
```

```
Alter Table tabelaA add Constraint cst Check (naziv like 'tekst');

Rezultati za provjeru su.
Select Sum(id) From TabelaA
--Rezultat 16
Select Sum(id) From
--Rezultat 22
Select Sum(id) From TabelaC;
--Rezultat 10
```

Zadatak 3 (5b)

Obrisati sve tabele iz zadatka2. Na slici 1 su date tri tabele pri čemu su null vrijednosti prikazane simbolom "-". Potrebno je ponovo kreirati ove tri tabele i popuniti ih odgovarajućim podacima. Uslovi za tabele su isti (da je FKTabelaA foreign key na id tabeleA i ne smije biti null a FkTabelaB foreign key na id tabelaB i smije biti null. RealniBroj tabeleA mora biti veći od 5 i cijeliBroj ne smije biti između 5 i 15. CijeliBroj tabeleB se ne smije ponavljati. Naziv i cijeliBroj tabele tabelaC ne smiju biti null. Foreign key ograničenje na tabelu b je potrebno nazvati "FkCnst").

Potrebno je kreirati sekvence seq1 i seq2 koje startaju sa 1 i povećavaju se za 1 i tabelu *TabelaABekap* koja je identična tabeli tabelaA samo što ima dodatne kolone *cijeliBrojB* tipa INTEGER i *sekvenca* tipa INTEGER. Zatim je potrebno kreirati dva trigera i jednu proceduru.

Prvi triger se okida poslije inserta na tabelu tabelaB za svaki unešeni red tako što spašava referencirani red tabeleA i u kolonu cijeliBrojB upisuje cijeliBroj od unešenog reda tabele TabelaB. Ako je red već ranije unešen u tabelu *TabelaABekap* potrebno je samo dodati *cijeliBroj* unešenog reda tabele tabelaB na *cijeliBrojB*. Pod *sekvenca* se unosi sljedeća vrijednost sekvence seq pri prvom insertu reda tabele TabelaABekap.

Drugi trigger u tabelu **CREATE TABLE TabelaBCheck(sekvenca INTEGER PRIMARY KEY)** unosi sljedeću vrijednost sekvence seq2 za svaku Delete komandu nad tabelom TabelaB.

Procedura prima jednu number varijablu i u tabelu TabelaC unosi redove onoliko puta kolika je suma kolone cijeliBroj tabele *TabelaA*. Pod cijeli broj unijeti parametar proslijeđen u proceduru a pod id unijeti redni broj unesenog reda u tabelu TabelaC tako što se nastavi od zadnjeg unesenog id iz tabele TabelaC.

Nakon Poziva komandi:

```
INSERT INTO TabelaB (id,naziv,datum,cijeliBroj,realniBroj,FkTabelaA) VALUES (6,null,null,2,null,1);
INSERT INTO TabelaB (id,naziv,datum,cijeliBroj,realniBroj,FkTabelaA) VALUES (7,null,null,4,null,2);
INSERT INTO TabelaB (id,naziv,datum,cijeliBroj,realniBroj,FkTabelaA) VALUES (8,null,null,8,null,1);
INSERT INTO TabelaB (id,naziv,datum,cijeliBroj,realniBroj,FkTabelaA) VALUES (9,null,null,5,null,3);
INSERT INTO TabelaB (id,naziv,datum,cijeliBroj,realniBroj,FkTabelaA) VALUES (10,null,null,7,null,3);
INSERT INTO TabelaB (id,naziv,datum,cijeliBroj,realniBroj,FkTabelaA) VALUES (11,null,null,9,null,5);
Delete From TabelaB where id not in (select FkTabelaB from TabelaC);
Alter TABLE tabelaC drop constraint FkCnst;
Delete from TabelaB where 1=1;
--izvrsenje procedure na kraju sa call NazivProcedure(1);
Rezultati poziva su:
Select SUM(id*3 + cijeliBrojB*3) from TabelaABekap; --138
Select Sum(id*3 + cijeliBroj*3) from TabelaC; --1251
Select Sum(MOD(sekvenca,10)*3) from TabelaBCheck; --9
Potrebno je utvditi rezultate poziva:
Select SUM(id*7 + cijeliBrojB*7) from TabelaABekap;
Select Sum(id*7 + cijeliBroj*7) from TabelaC;
Select Sum(MOD(sekvenca,10)*7) from TabelaBCheck;
```