

BAZE PODATAKA - JEDANAESTI TERMIN VJEŽBI (TRIGERI)

1. Kreirati tabelu FAKULTET koja treba da sadrži šifre (primarni ključ) i imena svih nastavnika iz tabele dostupne preko javnog sinonima NASTAVNICI, koji su angažovani na univerzitetskoj jedinici *Mašinski fakultet*, kao i ukupan broj časova nastave koju oni izvode na Univerzitetu. Tabela treba da sadrži i kolonu PREKO_NORME, koja je predviđena za unošenje numeričkih vrijednosti, i čija je podrazumijevana vrijednost 0. Prije unosa podataka u novokreiranu tabelu potrebno je napraviti triger TR1 koji će prilikom unošenja imena novog nastavnika, u slučaju da broj časova njegovog angažmana prelazi 12, što predstavlja potrebnu normu, u kolonu PREKO_NORME upisati broj časova koji je prelaze. Za unos podataka koristiti tabele dostupne preko javnih sinonima NASTAVNICI, OPT i JEDINICE.
2. Kreirati triger TR_ZAD2 koji će prilikom ažuriranja tabele iz prethodnog zadatka, u slučaju da je novi broj časova 12 ili manje u kolonu PREKO_NORME upisati vrijednost 0, a u slučaju da ovaj broj prelazi 12 upisati za koliko je ova norma prekoračena.
3. Napraviti log tabelu FAKULTET_LOG koja će služiti za nadgledanje svake promjene u tabeli FAKULTET. Podaci koji se čuvaju u ovoj tabeli su: datum promjene, korisničko ime osobe koja je u bazi promjenu izvršila, kao i opis promjene – da li je bilo u pitanju unošenje novih podataka, ili ažuriranje ili brisanje starih. Kreirati triger TR_ZAD3 koji služi za automatsko popunjavanje ove log tabele.
4. Kreirati tabelu KATALOG_SF koja treba da sadrži spisak knjiga u biblioteci koje pripadaju žanru SF. Tabela treba da ima kolone sa nazivom knjige, prezimenom i imenom autora i identifikatorom koji se sastoji od dijela 'K330SF' na koji se nadovezuje jedinstvena kombinacija od 7 brojeva, koja se unosi prilikom unošenja ostalih informacija. Napraviti triger koji će omogućiti automatsko kreiranje ovakvog identifikatora, odnosno, koji će dio 'K330SF' automatski dodati ispred kombinacije brojeva, i koji će istovremeno omogućiti da prezime autora bude uvijek ispisano isključivo velikim slovima. U slučaju da identifikator nije unijet, u odgovarajućem polju treba da stoji upis: 'Nezavedeno'.
5. Za tabelu iz prethodnog zadatka postojeći triger modifikovati tako da omogućava i da kombinacija koja se nadovezuje na 'K330SF' i služi kao identifikator bude isključivo kombinacija cifara i da tih cifara bude tačno 7. Ukoliko su unijete isključivo cifre, ali ih nema tačno 7, u odgovarajućoj koloni treba da stoji upis: 'Nezavedeno'.
6. Prethodnoj tabeli dodati kolonu STATUS, koja ima podrazumijevanu vrijednost 'Neiznajmljeno' i ograničenje NOT NULL. Napraviti triger koji će omogućiti da jedine dvije vrijednosti koje se mogu unijeti u ovu kolonu budu 'Neiznajmljeno' i 'Iznajmljeno'.
7. Napraviti triger koji omogućava da kombinacija brojeva koja se nadovezuje na string 'K330SF' bude jedinstvena za svaku vrstu tabele, to jest, triger treba da spriječi unos kombinacije u tabelu, ukoliko ona već postoji. *Napomena:* ovdje nećemo koristiti ograničenje kolone UNIQUE pošto se podrazumijeva da je moguće da tabela sadrži više upisa 'Nezavedeno' u posmatranoj koloni, što znači da triger treba da funkcioniše isključivo prilikom unošenja kombinacije brojeva koja je validan identifikator.

8. Tabeli KATALOG_SF dodati kolonu ID_KNJIGE koja će sadržati identifikator koji je jedinstven za sve knjige sa istim nazivom, i koje su od istog autora. Treba napraviti trigger koji će ovu kolonu automatski popunjavati. Identifikator može maksimalno biti šestocifreni broj. Inicijalno, trigger postavlja nulu kao prvu vrijednost identifikatora. Ukoliko se radi o knjizi (istog autora) koja već postoji u bazi, njoj se dodjeljuje isti identifikator kao postojećoj. Ukoliko je knjiga različita, najveći postojeći identifikator se uvećava za jedan.
9. Za vježbu napraviti tabele koje se tiču knjiga iz *beletristike*, *istorije umjetnosti*, *filosofije*, *istorije* i koje će imati odgovarajuće triggere, analogno tabeli KATALOG_SF.
10. Napraviti tabelu KATALOG_KNJIGA koja sadrži informacije o broju i dostupnosti knjiga kojima biblioteka raspolaže. Potrebno je napraviti kolone koje sadrže informacije o žanru, ukupnom broju knjiga, broju iznajmljenih knjiga kao i identifikatoru knjige. Napraviti trigger koji će služiti za ažuriranje ove tabele, i koji će kolonu sa žanrom automatski popunjavati na osnovu identifikatora iz zadatka 4. - na primjer: ukoliko identifikator u sebi sadrži skraćenicu 'SF' trigger postavlja 'Science fiction' u odgovarajuću kolonu tabele KATALOG_KNJIGA.

PREDLOG RJEŠENJA

1. Tabela može biti kreirana na sljedeći način:

```
create table fakultet(sifra number primary key, ime
varchar2(50),broj_casova number, preko_norme number default(0))
```

Triger se može kreirati na sljedeći način:

```
create or replace trigger tr1
before insert on fakultet
for each row
when (new.broj_casova>12)
declare
preko_norme fakultet.preko_norme%type;
begin
    preko_norme:=:new.broj_casova-12;
    :new.preko_norme:=preko_norme;
end;
```

Unos podataka može se obaviti na sljedeći način:

```
insert into fakultet(sifra, ime, broj_casova) select n.nid, n.ime,
sum(cp+gv*cv+gl*cl) from nastavnici n, jedinice j, opt o where
n.jid=j.jid and o.nid=n.nid and upper(j.naziv) like upper('%Mašinski
fakultet%')group by n.nid, n.ime
```

2. Triger može biti napravljen na sljedeći način:

```
create or replace trigger tr_zad2
before update on fakultet
for each row
declare
preko_norme fakultet.preko_norme%type;
begin
    if (:new.broj_casova>12) then
        preko_norme:=:new.broj_casova-12;
        :new.preko_norme:=preko_norme;
    else
        :new.preko_norme:=0;
    end if;
end;
```

Provjera ažuriranjem:

```
update fakultet set broj_casova=12.5 where sifra=140351
update fakultet set broj_casova=5 where sifra=140016
```

3. Napravimo log tabelu:

```
create table fakultet_log (datum date, korisnik varchar2(40), opis
varchar2(100))
```

Triger može biti kreiran na sljedeći način:

```
create or replace trigger tr_zad3
after insert or update or delete on fakultet
for each row
declare a number;
begin
    a:=:new.sifra;
    if inserting then
        insert into fakultet_log values (sysdate, user,
        'unosenje reda nastavnika sa sifrom' || ' ' || a);
    end if;
    if updating then
        insert into fakultet_log values (sysdate, user,
        'azuriranje nastavnika sa sifrom' || ' ' || a);
    end if;
    if deleting then
        insert into fakultet_log values (sysdate, user,
        'brisanje nastavnika sa sifrom' || ' ' ||
        :old.sifra);
    end if;
end;
```

4. Triger može biti kreiran na sljedeći način:

```
create or replace trigger kat_tr
before insert on katalog_sf
for each row
begin
    if (:new.sifra is not null) then
        :new.sifra:='K330SF' || :new.sifra;
    else
        :new.sifra:='NEZAVEDENO';
    end if;
    :new.prezime:=upper(:new.prezime);
end;
```

5. Da bi kombinacija koja služi za kreiranje identifikatora bila isključivo kombinacija cifara, deklarisaćemo promjenljivu *a* koja je tipa *number(7)*. Vrijednost *:new.id* smještaćemo u ovu promjenljivu prije konkatencije sa stringom 'K330SF'. Ukoliko string *:new.id* sadrži bilo kakav karakter koji nije cifra, Oracle će vratiti grešku i spriječiti unos. Funkcija *length()* vraća dužinu sadržaja (broj karaktera) varijable *a*.

```
create or replace trigger kat_tr
before insert or update on katalog_sf
for each row
declare
a number(7);
```

```

begin
a:=:new.id;
if (:new.id is null) then
:new.id:='NEZAVEDENO';
elsif (length(a)=7) then
:new.id:='K330SF' || a;
else
:new.id:='NEZAVEDENO';
end if;
:new.prezime:=upper(:new.prezime);
end
/

```

6. Dodavanje nove kolone izvršićemo na sljedeći način:

```

alter table katalog_sf add (status varchar2(13) default
'neiznajmljeno' not null);

```

Traženi triger ćemo kreirati na sljedeći način:

```

create or replace trigger kat_tr2
before insert or update on katalog_sf
for each row
begin
if (:new.status <> 'neiznajmljeno' and :new.status <>'iznajmljeno')
then
raise_application_error(-20000, 'Moguće vrijednosti kolone status su
''iznajmljeno'' ili ''neiznajmljeno''');
end if;
end
/

```

U ovom zadatku iskorišćena je funkcija *raise_application_error(-kod_greske, 'objašnjenje greške')*, pri čemu *kod_greske* predstavlja broj iz opsega: 20000-20999, ispred kojeg je neophodno staviti tire (-). Funkcija omogućava ispisivanje prigodnog obavještenja u slučaju zadovoljavanja/nezadovoljavanja odgovarajućih uslova, i sprečava dalje izvršavanje upita koji je izazvao grešku. Dvostruki apostrofi u stringu koji predstavlja objašnjenje greške su korišćeni u cilju ispisivanja jednostrukih apostrofa na njihovom mjestu (isto kao i u SQL upitima).

7. Zadatak se može riješiti korišćenjem sljedećeg trigera:

```

create or replace trigger kat_tr3
before insert or update on katalog_sf
for each row
declare
a number(1);
cursor c_id is
select id from katalog_sf;
begin
for n in c_id loop
a:=0;
if (n.id='K330SF'||:new.id) then
a:=1;
end if;
exit when (n.id='K330SF'||:new.id);
end loop;
if (a=1) then
raise_application_error(-20001, 'Kombinacija brojeva koju ste unijeli
vec postoji');
end if;
end

```

Kurzor u ovom primjeru sadrži sve moguće ID-eve oblika K330SFxxxxxxx, gdje je xxxxxxxx sedmocifrena kombinacija. Ponovimo neka pravila vezana za kurzore. Deklaracija kurzora obavlja se prije ključne riječi **begin**, u **declare** dijelu posmatranog bloka naredbi na sljedeći način:

```

cursor <ime_kurzora> [(<lista parametara>)] is <select upit>;

```

<lista parametara> se sastoji od: <ime_parametra> <tip_parametra>. Mogući tipovi parametara su **char**, **varchar2**, **number**, **date** i **boolean** (radiće i sa **integer**), i oni se mogu upotrijebiti kao parametri u SELECT upitu (npr. vrijednosti kolone u WHERE dijelu upita treba da budu veće od neke zadate vrijednosti parametra). Ovaj dio nije obavezan u deklaraciji kurzora, i u našem primjeru nije upotrijebljen. U opštem slučaju kurzor se otvara pomoću:

```

OPEN ime_kurzora(parametri);

```

jedan red iz kurzora se uzima sa:

```

FETCH ime_kurzora INTO lista_varijabli;

```

Informisati se o tome kako se vrši provjera da li je dodjela pomoću FETCH bila uspješna ili ne.

Zatvaranje se vrši pomoću:

```

CLOSE ime_kurzora;

```

Ukoliko se kurzor upotrebljava u okviru FOR petlje, otvaranje, uzimanje redova iz kurzora i zatvaranje kurzora obavlja se automatski. U našem konkretnom zadatku, pri svakoj iteraciji se promjenljivoj *n* dodjeljuje po jedan red (vrsta) iz kurzora, obavlja se set naredbi iz same petlje, a zatim se postupak ponavlja, ukoliko nijesu zadovoljeni uslovi za njegovo prekidanje. Iterativni postupak se prekida u dva slučaja: kada više nema neispitanih redova u kurzoru (kraj

petlje), ili kada je zadovoljen uslov u naredbi `exit when (n.id='K330SF' || :new.id);` Ovaj uslov je zadovoljen kada je **id** sadržan u promjenljivoj *n* (red kurzora u datoj iteraciji) jednak stringu dobijenom konkatencijom stringa 'K330SF' i stringa iz INSERT ili UPDATE upita koji aktivira triger. Ovaj dio koji se odnosi na prekidanje petlje je opcion, i realizovan je u okviru trigeru da bi se spriječile nepotrebne iteracije ukoliko se pronade vrijednost identifikatora u tabeli *katalog_sf* čije ponavljanje želimo spriječiti trigerom.

Promjenljiva *a* ima ulogu indikatora. Prije iterativnog postupka inicijalizuje se na vrijednost 0, a ukoliko je zadovoljen uslov `n.id='K330SF' || :new.id`, dodjeljuje joj se vrijednost 1 koja indicira da u tabeli *katalog_sf* već postoji kombinacija brojeva koju je korisnik pokušao unijeti pomoću svog INSERT ili UPDATE upita.

Nakon izlaska iz FOR petlje, ukoliko promjenljiva *a* u sebi sadrži vrijednost 1, triger sprečava unos (ili izmjenu vrste) i obavještava korisnika o tome (funkcija `raise_application_error`).

Napomena: brisanje trigeru se obavlja pomoću: `drop trigger ime_trigera.`

8. Dodavanje nove kolone izvršićemo sljedećim upitom:

```
alter table katalog_sf add(id_knjige varchar2(6));
```

Triger koji služi za popunjavanje ove nove vrste može se napraviti na sljedeći način:

```
create or replace trigger kat_tr4
before insert or update on katalog_sf
for each row
declare
idk number(6);
m number(6);
a number(1);
cursor max_id is
select max(id_knjige) id_knjige from katalog_sf;
cursor red is
select id_knjige, ime, prezime, naziv from katalog_sf;
begin
a:=0;
open max_id;
fetch max_id into m;
close max_id;
if (m is null) then
idk:=0;
end if;
for n in red loop
if(n.ime=:new.ime and n.prezime=:new.prezime and n.naziv=:new.naziv)
then
:new.id_knjige:=n.id_knjige;
a:=1;
exit when (n.ime=:new.ime and n.prezime=:new.prezime and
n.naziv=:new.naziv);
end if;
```

```

end loop;
if (idk=0) then
:new.id_knjige:=idk;
elsif(a=0) then
:new.id_knjige:=m+1;
end if;
end
/

```

Kurzor **max_id** sadrži maksimalnu vrijednost identifikatora. Kurzor **red** sadrži vrijednosti kolona tabele *katalog_sf* navedene u SELECT upitu pri njegovoj deklaraciji. U zadatku je demonstrirano otvaranje kurzora **max_id**, dodjela njegove vrijednosti promjenljivoj *m* i njegovo zatvaranje. Ukoliko unijeta vrsta ima isti naziv knjige, ime i prezime autora kao i neka postojeća u tabeli, dio naredbi u FOR petlji će omogućiti pridruživanje postojećeg identifikatora novoj vrsti. IF naredbe nakon FOR petlje omogućavaju da inicijalni identifikator bude 0, a ukoliko IF uslov u FOR petlji nije zadovoljen, da najveći postojeći identifikator bude uvećan za jedan i pridružen novoj vrsti. Promjenljiva *a* ovdje ima ulogu indikatora, koji ima vrijednost 1 isključivo kada je IF uslov u FOR petlji zadovoljen, dok vrijednost 0 indicira da knjiga ne postoji u tabeli i da joj je potrebno dodijeliti novi indikator (najveći postojeći uvećan za jedan).

Za vježbu zadatak odraditi bez upotrebe kurzora **max_id**. Pri dodjeljivanju vrijednosti SELECT upita (rezultat upita mora biti tačno jedna vrijednost!) promjenljivoj *m*, koristi se sintaksa: SELECT *x* INTO *ime_promjenljive*.... gdje *x* mora biti tačno jedna vrijednost.

Za vježbu pokušati takvu modifikaciju trigera koja će omogućiti da kolona *id_knjige* sadrži uvijek šestocifrene brojeve. Ukoliko je cifara manje od šest, ispred njih treba dodati potreban broj nula.

10. Pomoć: koristiti ključnu riječ LIKE.