

Adatbázis rendszerek 2 féléves egyéni PL/SQL feladat

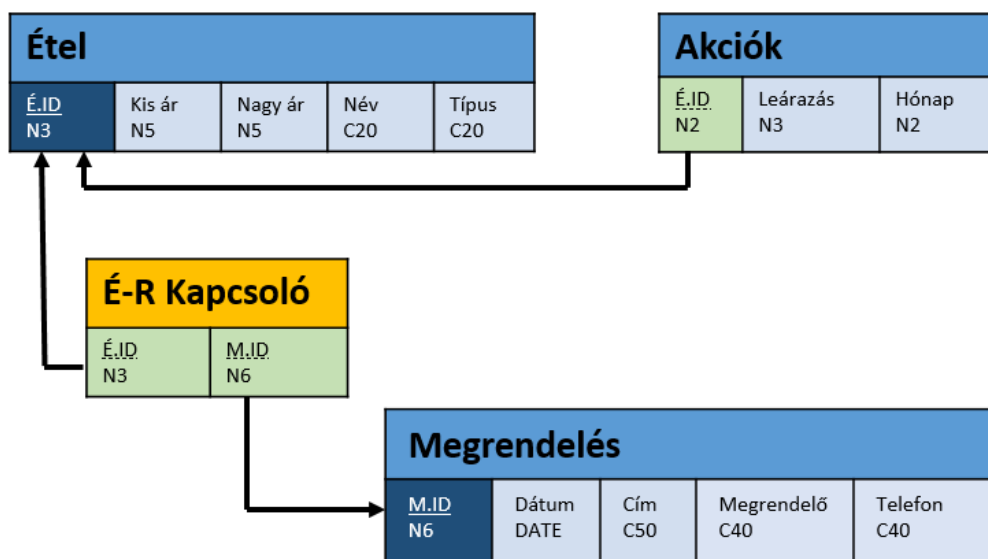
Juhász Ákos

F58KQ8

Gyakorlatvezető: Polonkai Dávid

Feladat leírása:

A beadandó egy étteremhez tartozó megrendelések adatbázisát modellezi. Az adatbázisban nyilvántartjuk egy-egy megrendelés alapvető adatait, illetve külön a fizető vendégek által rendelhető ételek étlapját is. Továbbá el van tárolva, hogy melyik ételek mikor vannak leárazva az évben. Ezek az akciók egy hónapig tartanak, az adatbázis letárolja melyik hónapban érvényesek, és mekkora mértékű a kedvezmény.



Táblákat létrehozó SQL parancsok:

- **Etel tábla:** CREATE TABLE etel(eid int primary key, kisar int, nagyar int, tipus varchar2(200), nev varchar2(200))
- **Akciok tábla:** CREATE TABLE akciok(eid int,foreign key(eid) references etel(eid),learazas int,honap int)
- **Megrendeles tábla:** CREATE TABLE megrendeles(mid int primary key,datum date,megrendelo varchar2(200),cim varchar2(200),telefon varchar2(200))
- **E-R Kapcsoló tábla:** CREATE TABLE er_kapcsoló(eid int,foreign key(eid) references etel(eid),mid int,foreign key(mid) references megrendeles(mid))

A táblákat feltöltő SQL parancsok:

Etel tábla:

```
INSERT INTO etel VALUES(1, 1610, 3310, "Pizza", "Margareta")
INSERT INTO etel VALUES(2, 2050, 4310, "Pizza", "4sajtos")
INSERT INTO etel VALUES(3, 1550, 2500, "Salata", "Cezar Salata")
INSERT INTO etel VALUES(4, 2120, 3160, "Husetel", "Gyros tál")
INSERT INTO etel VALUES(5, 2120, 3160, "Husetel", "Cordon Blue box")
INSERT INTO etel VALUES(6, 750, 1120, "Desszert", "Turogomboc")
```

Akciok tábla:

```
INSERT INTO akciok VALUES(1, 40, 1)
INSERT INTO akciok VALUES(2, 50, 6)
INSERT INTO akciok VALUES(4, 50, 12)
INSERT INTO akciok VALUES(5, 20, 2)
INSERT INTO akciok VALUES(3, 30, 1)
```

Megrendeles tábla:

```
INSERT INTO megrendels VALUES(1, Date.valueOf("2022-03-08"), "Juhász Akos",
"Miskolc_xy_utca-1", "062011111")
INSERT INTO megrendels VALUES(2, Date.valueOf("2021-12-20"), "Kis Pista",
"Miskolc_Ady_Endre_utca-13", "062012222")
INSERT INTO megrendels VALUES(3, Date.valueOf("2021-12-02"), "Nagy Greta",
"Miskolc_Kuruc_utca-22", "06203145160")
INSERT INTO megrendels VALUES(4, Date.valueOf("2021-01-01"), "Kis Pista",
"Miskolc_Ady_Endre_utca-13", "0630667708")
INSERT INTO megrendels VALUES(5, Date.valueOf("2022-3-10"), "Kozepes Karoly",
"Miskolc_Kuruc_utca-11", "0620198765")
INSERT INTO megrendels VALUES(6, Date.valueOf("2022-06-13"), "Kis Pista",
"Miskolc_Harsfa_utca-14", "0630611220")
```

ER Kapcsoló tábla:

```
INSERT INTO er_kapcsoló VALUES( 5, 1)
INSERT INTO er_kapcsoló VALUES( 2, 2)
INSERT INTO er_kapcsoló VALUES( 1, 2)
INSERT INTO er_kapcsoló VALUES( 4, 3)
INSERT INTO er_kapcsoló VALUES( 1, 3)
INSERT INTO er_kapcsoló VALUES( 3, 4)
INSERT INTO er_kapcsoló VALUES( 4, 5)
INSERT INTO er_kapcsoló VALUES( 2, 6)
```

PLSQL program:

Tárolt eljárás adatok felvitelére:

```
create or replace procedure EtelFel(EID number, kisar number, nagyar number, tipus varchar2, nev varchar2) is
begin
insert into etel values(EID, kisar, nagyar, tipus, nev);
dbms_output.put_line('Etel adatok felveve!');
end;

create or replace procedure AkciokFel(EID number, lear number, honap number) is
begin
insert into akciok values(EID, lear, honap);
dbms_output.put_line('Akcio adatok felveve!');
end;

create or replace procedure MegrendelesFel(MID number, datum date, rendelo varchar2, cim varchar2, telefon varchar2) is
begin
insert into megrendeles values(MID, datum, rendelo, cim, telefon);
dbms_output.put_line('Megrendelo adatok felveve!');
end;

create or replace procedure ERKapsoloFel(EID number, MID number) is
begin
insert into erkapcsolo values(EID, MID);
dbms_output.put_line('ERKapsolo adatok felveve!');
end;
```

Minden táblának megvan a maga külön tárolt eljárása az adatok felvitelére.
Futtatási képek:

```
begin
    EtelFel(52, 100, 201, 'Pizza', 'Tonhalas');
end;
```

PL/SQL procedure successfully completed.

Etel adatok felveve!

```
begin
    AkciokFel(52, 10, 10);
end;

PL/SQL procedure successfully completed.

Akcio adatok felveve!
```

Tárolt eljárás adatok módosítására:

```
create or replace procedure arValt(ujEID number, ujkisar number, ujnagyar number) is
begin
    update etel set etel.kisar=ujkisar, etel.nagyar=ujnagyar where etel.EID=ujEID;
    dbms_output.put_line('Etel tabla modositva!');
end;

create or replace procedure etelMod(ujEID number, ujkisar number, ujnagyar number, ujtipus varchar2, ujnev varchar2) is
begin
    update etel set etel.kisar=ujkisar, etel.nagyar=ujnagyar, etel.tipus=ujtipus, etel.nev=ujnev where etel.EID=ujEID;
    dbms_output.put_line('Etel tabla modositva!');
end;
```

```
create or replace procedure akciokMod(ujEID number, ujlear number, ujhonap number) is
begin
    update akciok set akciok.lear=ujlear, akciok.honap=ujhonap where akciok.EID=ujEID;
    dbms_output.put_line('Akciok tabla modositva!');
end;

create or replace procedure megrendelesMod(ujMID number, ujdatum date, ujrendelo varchar2, ujcim varchar2, ujtelefon varchar2) is
begin
    update megrendeles set megrendeles.datum=ujdatum, megrendeles.megrendelo=ujrendelo, megrendeles.cim=ujcim, megrendeles.telefon=ujtelefon where megrendeles.MID=ujMID;
    dbms_output.put_line('Megrendeles tabla modositva!');
end;
```

Van egy külön módosító tárolt eljárás, ami csak az étel árait írja felül.

Futtatási képek:

```
begin
    etelMod(52,200,402,'Pizza','Szardíniás');
end;
begin
    akciokMod(52,11,11);
end;
```

PL/SQL procedure successfully completed.
Etel tabla modositva!
Procedure AKCIOKMOD compiled
Akciok tabla modositva!

Tárolt eljárás adatok törlésére:

```
create or replace procedure DelEtel(tEID number) is
begin
    delete from etel where etel.EID=tEID;
    dbms_output.put_line('Etel torolve!');
end;

create or replace procedure DelAkciok(tEID number) is
begin
    delete from etel where akciok.EID=tEID;
    dbms_output.put_line('Akcio torolve!');
end;

create or replace procedure DelRendeles(tMID number) is
begin
    delete from etel where megrendeles.MID=tMID;
    dbms_output.put_line('Rendeles torolve!');
end;

create or replace procedure DelErkapcsolo(tEID number) is
begin
    delete from etel where erkapcsolo.EID=tEID;
    dbms_output.put_line('erkapcsolo torolve!');
end;
```

Futtatási képek:

```
begin
    DelAkciok(52);
end;
begin
    DelEtel(52);
end;
```

Procedure DELERKAPCSOLO compiled
Akcio torolve!

Tárolt függvény adott rekord mezőinek lekérdezésére:

```
create or replace function lekerEtelNev(azon number) return varchar2 as
    x varchar2(30);
begin
    select etel.nev into x from etel where etel.eid=azon;
    return x;
end;

create or replace function lekerEtelek(azon number) return etel%rowtype as
    x etel%rowtype;
begin
    select * into x from etel where etel.eid=azon;
    return x;
    dbms_output.put_line(x.nev);
end;
```

```

declare
    c_eid etel.eid%type;
    c_kisar etel.kisar%type;
    c_nagyar etel.nagyar%type;
    c_tipus etel.tipus%type;
    c_nev etel.nev%type;
    cursor c_etel is select eid, kisar, nagyar, tipus, nev from etel;
begin
    open c_etel;
    loop
        fetch c_etel into c_eid, c_kisar, c_nagyar, c_tipus, c_nev;
        exit when c_etel%notfound;
        dbms_output.put_line(c_eid || ' - ' || c_kisar || ' - ' || c_nagyar || ' - ' || c_tipus || ' - ' || c_nev);
    end loop;
    close c_etel;
end;

```

Itt az utolsó példa az explicit kurzor használatát mutatja be.
Futtatási képek:

```

declare
    x varchar2(30);
begin
    x:=lekerEtelNev(1);      PL/SQL procedure successfully completed.
    dbms_output.put_line(x);
end;                        Margareta
PL/SQL procedure successfully completed.

```

```

1 - 1610 - 3310 - Pizza - Margareta
2 - 2050 - 4310 - Pizza - 4sajtos
3 - 1550 - 2500 - Salata - Cezar Salata
4 - 2120 - 3160 - Husetel - Gyros tal
5 - 2120 - 3160 - Husetel - Cordon Blue box
6 - 750 - 1120 - Desszert - Turogomboc

```

Tárolt függvény adott feltételű rekordok aggregált értékének lekérdezésére:

```

create or replace function lekerAtlagAr return number as
    x number;
begin
    select sum(etel.kisar)/count(*) into x from etel;
    return x;
end;

```

A felvitt ételek kisárának átlagát adja vissza.

Futtatási képek:

```

declare
    x number;
begin
    x:=lekerAtlagAr();      Function LEKERATLAGAR compiled
    dbms_output.put_line(x);
end;                        1700

```

Tárolt csomag készítése egy tábla funkcióinak összefoglalására:

```
create or replace package etelPack as
    procedure EteIFel(EID number, kisar number, nagygar number, tipus varchar2, nev varchar2);
    function lekerEtelNev(azon number) return varchar2;
    function lekerEtelek(azon number) return etel%rowtype;
    function lekerAtlagAr return number;
    procedure DelEtel(tEID number);
    procedure arValt(ujEID number, ujkisar number, ujnagyar number);
    procedure etelMod(ujEID number, ujkisar number, ujnagyar number, ujtipus varchar2, ujnev varchar2);
end etelPack;
```

Az etelPack tárolt csomag tartalmazza az összes etel táblához kapcsoló függvényt és eljárást. A package bodyjába be is másoltam az odatartozó függvényeket, eljárásokat:

```
create or replace package body etelPack is
    procedure EteIFel(EID number, kisar number, nagygar number, tipus varchar2, nev varchar2) is
    begin
        insert into etel values(EID, kisar, nagygar, tipus, nev);
        dbms_output.put_line('Etel adatok felveve!');
    end;

    function lekerEtelNev(azon number) return varchar2 as
    x varchar2(30);
    begin
        select etel.nev into x from etel where etel.eid=azon;
        return x;
    end;

    function lekerEtelek(azon number) return etel%rowtype as
    x etel%rowtype;
    begin
        select * into x from etel where etel.eid=azon;
        return x;
        dbms_output.put_line(x.nev);
    end;

    function lekerAtlagAr return number as
    x number;
    begin
        select sum(etel.kisar)/count(*) into x from etel;
        return x;
    end;

    procedure DelEtel(tEID number) is
    begin
        delete from etel where etel.EID=tEID;
        dbms_output.put_line('Etel torolve!');
    end;

    procedure arValt(ujEID number, ujkisar number, ujnagyar number) is
    begin
        update etel set etel.kisar=ujkisar, etel.nagyar=ujnagyar where etel.EID=ujEID;
        dbms_output.put_line('Etel tabla modositva!');
    end;

    procedure etelMod(ujEID number, ujkisar number, ujnagyar number, ujtipus varchar2, ujnev varchar2) is
    begin
        update etel set etel.kisar=ujkisar, etel.nagyar=ujnagyar, etel.tipus=ujtipus, etel.nev=ujnev where etel.EID=ujEID;
        dbms_output.put_line('Etel tabla modositva!');
    end;
end etelPack;
```

Az így elmentett eljárások, függvények meghívhatók a packen keresztül:

```
declare
    x etel%rowtype;
begin
    x:=etelPack.lekerEtelek(1);
    dbms_output.put_line(x.eid || ' - ' || x.kisar || ' - ' || x.nagyar || ' - ' || x.nev || ' - ' || x.tipus);
end;
```

1 - 1610 - 3310 - Margareta - Pizza

PL/SQL procedure successfully completed.

Trigger készítése módosítási események naplózására:

```
create table naplo (muvelet char(50),datum date, felhasznalo char(20));

create or replace trigger trig_etel_del after delete on etel for each row
begin
    insert into naplo values('Delete',sysdate,user);
end;

create or replace trigger trig_etel_fel after insert or update on etel for each row
begin
    if updating then
        insert into naplo values('Update',sysdate,user);
    elsif inserting then
        insert into naplo values('Insert',sysdate,user);
    end if;
end;
```

A kód itt három részre van osztva: első maga napló tábla létrehozása, ahova az események adatait fogjuk menteni, a második csak a törlés funkciót naplózza, míg a harmadik az update és insert parancsokat figyeli.

Példa a triggererek által készített napló bejegyzésekre:

MUVELET	DATUM	FELHASZNALO
1 Insert	22-MÁJ. -07	H22_F58KQ8

Trigger készítése kulcs érték automatikus megadására:

```
CREATE SEQUENCE etel_seq;

CREATE OR REPLACE TRIGGER etel_insert
BEFORE INSERT ON etel
FOR EACH ROW
BEGIN
    SELECT etel_seq.nextval
    INTO :new.EID
    FROM dual;
END;
```


Trigger készítése a módosítások kontrollálására

```
create or replace trigger trig_etel_mod before update on etel for each row
begin
    if (:new.kisar < :old.kisar) then
        :new.kisar := :old.kisar;
        raise_application_error (-20001, 'A kisarat csak novelni szabad!');
    elsif (:new.nagyar < :old.nagyar) then
        :new.nagyar := :old.nagyar;
        raise_application_error (-20002, 'Az nagyarat csak novelni szabad!');
    end if;
end;
```

Ez a trigger, annyit figyel, hogy az ételek módosításakor az árak mindig csak növekedni tudjanak. Ellenkező esetben pedig hibát jelez és visszaállítja az értéket az előző állapotra.

Error report -

ORA-20001: A kisarat csak novelni szabad!

Egyéb kötelező funkciók:

Explicit kurzor használata:

```
declare
    c_eid etel.eid%type;
    c_kisar etel.kisar%type;
    c_nagyar etel.nagyar%type;
    c_tipus etel.tipus%type;
    c_nev etel.nev%type;
    cursor c_etel is select eid, kisar, nagy, tipus, nev from etel;
begin
    open c_etel;
    loop
        fetch c_etel into c_eid, c_kisar, c_nagyar, c_tipus, c_nev;
        exit when c_etel%notfound;
        dbms_output.put_line(c_eid || ' - ' || c_kisar || ' - ' || c_nagyar || ' - ' || c_tipus || ' - ' || c_nev);
    end loop;
    close c_etel;
end;
```

Futtatás eredménye:

```
1 - 1610 - 3310 - Pizza - Margareta
2 - 2050 - 4310 - Pizza - 4sajtos
3 - 1550 - 2500 - Salata - Cezar Salata
4 - 2120 - 3160 - Husétel - Gyros tal
5 - 2120 - 3160 - Husétel - Cordon Blue box
6 - 750 - 1120 - Desszert - Turogomboc
```

Kivételkezelés:

```
create or replace trigger trig_etel_mod before update on etel for each row
begin
    if (:new.kisar < :old.kisar) then
        :new.kisar := :old.kisar;
        raise_application_error (-20001, 'A kisarat csak novelni szabad!');
    elsif (:new.nagyar < :old.nagyar) then
        :new.nagyar := :old.nagyar;
        raise_application_error (-20002, 'Az nagyarat csak novelni szabad!');
    end if;
end;
```

Error report -

ORA-20001: A kisarat csak novelni szabad!

Extra funkciók:

Rowtype visszatérésű függvény:

```
create or replace function lekerEtelek(azon number) return etel%rowtype as
    x etel%rowtype;
begin
    select * into x from etel where etel.eid=azon;
    return x;
    dbms_output.put_line(x.nev);
end;
```

Futtatás:

```
declare
    x etel%rowtype;
begin
    x:=lekerEtelek(1);
    dbms_output.put_line(x.eid || ' - ' || x.kisar || ' - ' || x.nagyar || ' - ' || x.nev || ' - ' || x.tipus);
end;
```

1 - 1610 - 3310 - Margareta - Pizza

PL/SQL procedure successfully completed.

Több esemény kezelése egyetlen triggerben:

```
create or replace trigger trig_etel_fel after insert or update on etel for each row
begin
    if updating then
        insert into naplo values('Update',sysdate,user);
    elsif inserting then
        insert into naplo values('Insert',sysdate,user);
    end if;
end;
```

Egyszerre kezeli az update és insert parancsot is