

Oracle adatbázis ORDBMS műveletek

Készítsen egy lakcím leíró T_LAKCIM osztályt, amely a fontosabb postai cím komponenseket tartalmazza.

```
CREATE OR REPLACE TYPE t_lakcim AS OBJECT (  
    irsz NUMBER(4),  
    varos CHAR(20),  
    utca CHAR(20),  
    hazszam NUMBER(3)  
);
```

Szüntesse meg az előzőleg létrehozott T_LAKCIM osztályt.

```
DROP TYPE t_lakcim FORCE;
```

Hozzon létre egy emberek táblát, melyben szerepelnek a név, születési év és lakcím mezők. Ez utóbbi adattípusa legyen a T_LAKCIM osztály.

```
CREATE TABLE dolgozok (  
    kod NUMBER(3) PRIMARY KEY,  
    nev CHAR(20) NOT NULL,  
    szulev NUMBER(4) CHECK (szulev > 1900),  
    lakcim t_lakcim  
);
```

Vigyen fel egy új rekordot a dolgozok táblába.

```
INSERT INTO DOLGOZOK  
VALUES (1, 'ANNA', 1995,  
    T_LAKCIM(3527, 'MISKOLC', 'NAGY UT', 12));
```

Listázza ki a dolgozók nevét és a városukat, város szerinti sorrendben.

```
SELECT d.nev, d.lakcim.varos  
FROM dolgozok d  
ORDER BY 2;
```

Kérdezze le a dogozók létszámát városenkénti bontásban.

```
SELECT count(d.nev), d.lakcim.varos  
FROM dolgozok d  
GROUP BY d.lakcim.varos;
```

Hozzon létre egy t_ember osztályt, melyben adattagként szerepel a név, születési év, igazolványszám és lakcím.

```
CREATE OR REPLACE TYPE t_ember AS OBJECT (  
    nev CHAR(20),  
    sz_ev NUMBER(4),  
    igazolvany_sz CHAR(10),  
    cim T_LAKCIM  
);
```

Hozzon létre egy táblát, amely t_emberek típusú objektumokat tartalmaz.

```
CREATE TABLE emberek OF t_ember
```

Vigyen fel egy új embert a emberek táblába.

```
INSERT INTO emberek
VALUES (T_EMBER('GABI',1994,'IG1',
      T_LAKCIM(2333,'Dorog','Petofi',23)))
```

Kérdezze le az ember objektumok OID-jét és az objektumban tárolt ember nevét.

```
SELECT REF(e), e.nev
FROM emberek e
```

Módosítsa az IG1 igazolványszámú ember lakcímét egy új értékre.

```
UPDATE emberek
SET cim = t_lakcim(3111,'Ozd','Patak' ,45)
WHERE igazolvany_sz = 'IG1'
```

Hozzon létre egy t_ember osztályt, melyben adattagként szerepel a név, születési év, igazolványszám és lakcim. A létrehozott osztályhoz később leszármazott osztályokat kívánunk majd létrehozni.

```
CREATE OR REPLACE TYPE t_ember AS OBJECT (
  nev CHAR(20),
  sz_ev NUMBER(4),
  igazolvany_sz CHAR(10),
  cim T_LAKCIM
) NOT FINAL
```

Hozzon létre egy t_ugyfel osztályt, amely a t_ember osztályból származik, és tartalmaz ügyfélkód és ügyféltípus valamint belepési dátum adattagokat

```
CREATE OR REPLACE TYPE t_ugyfel UNDER t_ember (
  ukod NUMBER(4),
  belepes_datum DATE,
  u_tipus CHAR(3)
)
```

Hozza létre az ügyfelek objektumainak tábláját.

```
CREATE TABLE ugyfelek OF t_ugyfel
```

Vigyen fel egy új ügyfél objektumot az ügyfelek táblába.

```
INSERT INTO ugyfelek
VALUES (t_ugyfel('Zoltan',1987,'IG2',
      t_lakcim(4555,'Szolnok','Hatar',45),1,SYSDATE,'LAK'))
```

Kérdezze le az ügyfelek létszámát városonként.

```
SELECT u.cim.varos, COUNT(*)
FROM ugyfelek u
GROUP BY u.cim.varos
```

Hozzon létre olyan t_ugyfelek osztályt, amely ügyfelek halmazát tartalmazza.

```
CREATE TYPE t_ugyfelek AS TABLE OF t_ugyfel;
```

Hozzon létre egy ügyek táblát, melyben az ügy paraméterei között szerepel az érintett ügyfelek listája.

```
CREATE TABLE ugyek (  
    kod NUMBER(2) PRIMARY KEY,  
    leiras CHAR(20),  
    résztvevok t_ugyfelek  
)  
NESTED TABLE résztvevok STORE AS s_tabla
```

Vigyen fel egy új rekordot az ügyek táblába.

```
INSERT INTO ugyek  
VALUES (1, 'Macskavita', t_ugyfelek(  
    t_ugyfel('Zoltan', 1987, 'IG2',  
    t_lakcim(4555, 'Szolnok', 'Hatar', 45), 1, SYSDATE, 'LAK'),  
    t_ugyfel('Feri', 1989, 'IG5',  
    t_lakcim(7623, 'Szeged', 'Hatar', 44), 2, SYSDATE, 'LAK'),  
    t_ugyfel('Sanyi', 1991, 'IG6',  
    t_lakcim(2161, 'Csolnok', 'Domb', 75), 3, SYSDATE, 'LAK'))  
)
```

Listázza ki a 'Macskavita' leírású ügy ügyfeleinek nevét és városát.

```
SELECT u.nev, u.cim.varos  
FROM TABLE  
(SELECT p.resztvevok  
    FROM ugyek p  
    WHERE p.leiras LIKE 'Macskavita%') u
```

Vegye ki az IG3-as kódú személyt az 1-es kódú ügyek ügyfelei közül (ügyfél kitörlése az ügyfelek listájából)

```
DELETE FROM TABLE (  
    SELECT p.resztvevok  
    FROM ugyek p  
    WHERE p.kod = 1)  
u WHERE u.ukod = 1
```

Vigyen fel egy új ügyfelet az 1-es kódú ügghöz.

```
INSERT INTO TABLE (  
    SELECT p.resztvevok  
    FROM ugyek p  
    WHERE p.leiras LIKE 'Macskavita%')  
VALUES (t_ugyfel('Zoltan', 1987, 'IG2',  
    t_lakcim(4555, 'Szolnok', 'Hatar', 45), 1, SYSDATE, 'LAK'))
```

Hozzon létre egy olyan panaszok táblát, ahol a panasz benyújtó ügyfélre hivatkozás tárolódik.

```
CREATE TABLE panaszok
```

```
(
    kod NUMBER(3) PRIMARY KEY,
    leiras CHAR(20),
    ugyfel REF t_ugyfel
)
```

Vigyen fel egy új rekordot a panaszok táblába, melynek ügyfele a Zoltan nevű ügyfél.

```
INSERT INTO panaszok VALUES (1, 'csendháborítás',
    (SELECT REF(u) FROM ugyfelek u WHERE nev LIKE 'Zoltan%'))
```

Kérdezzük le a panaszok leírását és az ügyfél nevét.

```
SELECT p.leiras, p.ugyfel.nev
FROM panaszok p
```

Kérdezze le a ügyek leírását a résztvevő nevével egy eredmény táblázatba.

```
SELECT u.leiras, r.nev
FROM ugyek u,
TABLE (
    SELECT résztvevok
    FROM ugyek v
    WHERE v.kod = u.kod) r
```

Definiálon egy olyan osztályt, amely ügyfelekre történő hivatkozások listáját tartalmazza.

```
CREATE OR REPLACE TYPE t_ugyfelek_ref AS
TABLE OF REF t_ugyfel
```

Hozzon létre egy olyan panaszok táblát, ahol több hivatkozott ügyfél is lehet.

```
CREATE TABLE panaszok (
    kod NUMBER(3) PRIMARY KEY,
    leiras CHAR(20),
    ugyfelek t_ugyfelek_ref)
NESTED TABLE ugyfelek STORE AS seged2
```

Vigyen fel egy új panasz rekordot.

```
INSERT INTO panaszok
VALUES (1, 'Dobverés', t_ugyfelek_ref(
    (SELECT REF(u) FROM ugyfelek u WHERE u.nev = 'Zoltan'),
    (SELECT REF(u) FROM ugyfelek u WHERE u.nev = 'laci')))
)
```

Vigyen fel olyan új panasz rekordot, amelyhez nem tartozik ügyfél.

```
INSERT INTO panaszok
VALUES (3, 'labdazás', t_ugyfelek_ref())
```

Addjon egy új ügyfelet egy panaszhoz.

```
INSERT INTO
TABLE(SELECT ugyfelek FROM panaszok WHERE kod = 3)
```

```
VALUES
  ((SELECT REF(u) FROM ugyfelek u WHERE u.nev LIKE 'dani%' ))
```

Listázza ki a panaszok leírását és az ügyfeleinek létszámát.

```
SELECT a.leiras,
  (SELECT count(*)
   FROM TABLE
    (select ugyfelek FROM panaszok WHERE kod = a.kod))
FROM panaszok a
```

Adja meg a panaszok leírását és az ügyfelek nevét egy táblázatban.

```
SELECT p.leiras, Deref(COLUMN_VALUE).nev
FROM panaszok p, TABLE(p.ugyfelek) r
```

Listázza ki az ügyfelekénti panaszok darabszámát.

```
SELECT t2.C, COUNT(t1.A)
FROM
  (SELECT p.leiras A , Deref(COLUMN_VALUE).nev B
   FROM panaszok p, TABLE(p.ugyfelek) r) t1
RIGHT OUTER JOIN
  (SELECT nev C FROM ugyfelek) t2
ON t2.C = t1.B
GROUP BY t2.C
```

Hozzon létre olyan t_ember leszármaztatható osztályt, melyhez tartozik két lekérdező metódus. Az egyik metódus az életkort, másik a lakcímet adja vissza.

```
CREATE OR REPLACE TYPE t_ember2 AS OBJECT (
  nev CHAR(20),
  szuletesi_ev NUMBER(4),
  igazolvany_szam CHAR(10),
  cim t_kalcim,
  MEMBER FUNCTION eletkor RETURN NUMBER,
  MEMBER FUNCTION lakcim RETURN CHAR
) NOT FINAL;

CREATE OR REPLACE TYPE BODY T_EMBER2 AS
MEMBER FUNCTION ELETKOR RETURN NUMBER AS
BEGIN
  RETURN TO_NUMBER(TO_CHAR(SYSDATE, 'YYYY')) - SELF.SZULETESI_EV;
END;

MEMBER FUNCTION LAKCIM RETURN CHAR AS
BEGIN
  RETURN TRIM(SELF.CIM.VAROS) || ' ' || TRIM(SELF.CIM.UTCA) ||
  ' ' || TRIM(TO_END);
END;
```

Hozza létre az emberek táblát és vigyen fel rekordokat.

```
CREATE TABLE emberek2 OF t_ember2;
INSERT INTO EMBEREK2
VALUES
  (t_ember2('Zoli',1987,'IG3',t_lakcim(5565,'Baja','Otto',23)))
```

Listázza ki az emberek nevét, életkorát és lakcímét.

```
SELECT E.NEV, E.ELETKOR(), E.LAKCIM()  
FROM EMBEREK2 E
```

Készítsen t_ugyfel osztályt az alábbi metódusokkal: pénz_felvétel, pénz_behelyzés és hitelbíráló.

```
CREATE OR REPLACE TYPE T_UGYFEL UNDER T_EMBER (  
    UKOD NUMBER(4),  
    BELEPES_DATUM DATE,  
    MEMBER FUNCTION PENZ_FELVESZ (ERTEK IN NUMBER) RETURN NUMBER,  
    MEMBER FUNCTION PENZ_BETESZ (ERTEK IN NUMBER) RETURN NUMBER,  
    MEMBER FUNCTION HITEL_BIRALAT(ERTEK IN NUMBER) RETURN NUMBER  
);
```