

# Objektum-relációs adatbázisok – 2. rész

Felhasználói típusok (User-Defined Types)

Objektum ID-k

Beágyazott táblák (Nested Tables)

# Mező elérés (*field access*) problematikája az ORACLE-ben

- Egy objektum  $F$  mezőjét  $A.F$  kifejezéssel elérhetjük, amelynek ez az értéke
- Azonban egy aliaszt kell használni, pl.  $rr, R$  relációra, annak  $A$  attribútumára mint pl.  $rr.A.F$

## Példa: Oracle-ben mezők elérése

- Rossz:

```
SELECT kedvencTea.név  
FROM Vendégek;
```

- Rossz :

```
SELECT Vendégek.kedvencTea.név  
FROM Vendégek;
```

- Jó:

```
SELECT dd.kedvencTea.név  
FROM Vendégek dd;
```

# A REF-k (hivatkozások) követése: SQL-99 stílus

- $A \rightarrow B$  csak akkor értelmes ha:
  1. Ha  $A$  egy REF  $T$  típusú.
  2. A  $T$  típusú objektum mezője (komponense)  $B$ .
- Az  $A$  által hivatkozott/mutatott objektum  $B$  ***mezőjének*** értékét jelöli

```
CREATE TYPE FelszoigaltTípus AS (
  teázó      REF TeázóTípus,
  tea        REF TeaTípus,
  ár         FLOAT
);
```

## Példa: REF-k (hivatkozások) követése

- Emlékezzünk rá, hogy az **Felszoigál** egy olyan reláció egy **FelszoigaltTípus(teázó, tea, ár)** sortípussal, ahol teázó és tea REF-ek, hivatkozások a **TeázóTípus** és **TeaTípus** típusú objektumokra.

- Keresd meg a Joe által felszoigált teákat:

SELECT ff.tea()->név

FROM Felszoigál ff

WHERE ff.teázó()->név = 'Joe's Teahouse';

A nyilat követve  
kapjuk meg  
a hivatkozott  
„teázó”-t és „tea”-t

Először használjuk a generátor metódust, hogy hozzáférjünk a *teázó* és *tea* komponenshez

# Oracle stílusban REF (hivatkozás) követése

- REF követése implicit a pontban.
- A REF-t nyomon követni: egy „elem” után egy pont, majd a megjelölt objektum mezőjének követésével, amire hivatkozik, kapjuk meg az értéket
- Példa:

```
SELECT ff.tea.név  
       FROM Felszolgál ff  
       WHERE ff.teázó.név = 'Joe''s Teahouse';
```

# Oracle Deref művelete - motiváció

- Ha a Joe által értékesített teákra mint tea objektumok halmazára van szükségünk, megpróbálhatjuk az alábbi:

```
SELECT ff.tea  
FROM felszolgál ff  
WHERE ff.teázó.név = 'Joe''s Tenhouse';
```

- Legális SQL, de *ff.tea* maga egy hivatkozás, ezért egy zagyvaság.

# DEREF használata

- Ahhoz, hogy a **TeaTípus** objektumait láthassuk:

```
SELECT Deref(ff.tea)
FROM Felszolgal ff
WHERE ff.teázó.név = 'Joe''s Teahouse';
```

- Egy ilyen értéket állít elő:

**TeaTípus('Brisk', 'Lipton')**



# Ismétlés – még egy példakód

- Az alábbiak voltak már:

```
CREATE TYPE TeázóTípus AS OBJECT (  
    név      CHAR(20),  
    cím      CHAR(20)  
);
```

```
CREATE TYPE TeaTípus AS OBJECT (  
    név      CHAR(20),  
    gyártó CHAR(20)  
);
```

```
CREATE TYPE FelszolgáltTípus AS OBJECT (  
    teázó REF TeázóTípus,  
    tea   REF TeaTípus,  
    ár    FLOAT  
);
```

```
CREATE TABLE Teázók OF TeázóTípus;
```

```
CREATE TABLE Teák OF TeaTípus;
```

```
CREATE TABLE Felszolgál OF FelszolgáltTípus;
```

- Szúrjunk be a táblákba sorokat:

```
INSERT INTO Teázók VALUES (TeázóTípus('Joe''s Teahouse', 'Maple str 1'));
```

```
INSERT INTO Teák VALUES (TeaTípus('Pyramid', 'Lipton'));
```

```
INSERT INTO Felszolgál
```

```
    SELECT FelszolgáltTípus(REF(ba), REF(be), 3.5)
```

```
    FROM Teázók ba, Teák be
```

```
    WHERE ba.név = 'Joe''s Teahouse' AND be.név = 'Pyramid';
```

# Ismétlés – még egy példakód

- Kérdezzük le a Felszolgál tábla tartalmát az alábbi lekérdezéssel:

```
SELECT Deref(se.teázó) AS TEÁZÓ, Deref(se.tea) AS TEA, ár FROM Felszolgál se;
```

TEÁZÓ	TEA	ÁR
TeázókTípus('Joe''s Teahouse', 'Maple str 1')	TeaTípus('Pyramid', 'Lipton')	3.5

- Vagy egy másik lekérdezés:

```
SELECT se.teázó.név AS TEÁZÓ_NÉV, se.tea.név AS TEA_NÉV, ár FROM Felszolgál se;
```

TEÁZÓ_NÉV	TEA_NÉV	ÁR
Joe's Teahouse	Pyramid	3.5

# Metódusok – ORACLE szintaxis

- Az osztályok többek mint adatszerkezetek; lehetnek metódusaik.
- Tanulmányozni fogjuk az Oracle szintaxisát

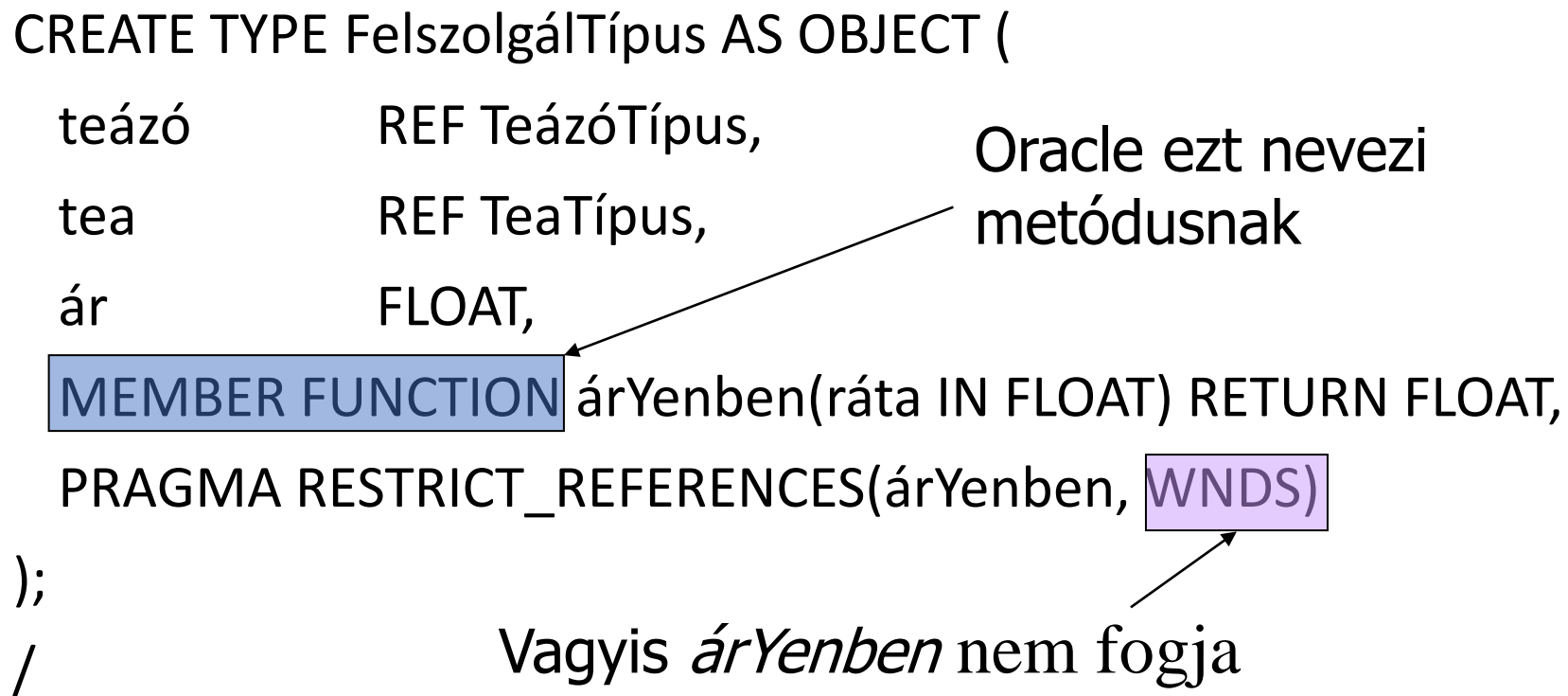
# Metódus definíció (Oracle)

- A metódusok deklarálhatjuk a CREATE TYPE-ban
- Definiálhatjuk a CREATE TYPE BODY utasításban
  - Használva a PL/SQL szintaxisát a metódusokra
  - SELF változó arra az objektumra vonatkozik, amelyre a metódust alkalmazni kívánjuk.

# Példa: metódus deklaráció

Adjuk hozzá *árYenben-t*, *FelszolgáTípus*-hoz.

```
CREATE TYPE FelszolgáTípus AS OBJECT (  
    teázó      REF TeázóTípus,  
    tea        REF TeaTípus,  
    ár         FLOAT,  
    MEMBER FUNCTION árYenben(ráta IN FLOAT) RETURN FLOAT,  
    PRAGMA RESTRICT_REFERENCES(árYenben, WNDS)  
);  
/
```



Vagyis *árYenben* nem fogja  
módosítani az adatbázis állapotát

# Metódus definíció – Oracle stílusban

□ A *create-body* utasítás formája:

```
CREATE TYPE BODY <típusnév> AS
```

```
    <metódus definíciók = PL/SQL eljárás definíciók,  
      a “MEMBER FUNCTION” használva a  
      “PROCEDURE” helyén>
```

```
END;
```

```
/
```

# Példa: Metódus definíció

Az (IN) nincs a „body”-ban,  
csak a deklarációban

```
CREATE TYPE BODY FelszolgáLTípus AS  
  MEMBER FUNCTION  
    árYenben(ráta FLOAT) RETURN FLOAT IS  
    BEGIN  
      RETURN ráta * SELF.ár;  
    END;  
END;  
/
```

*Csak akkor használjunk  
zárójelet, ha legalább  
egy argumentum van*

# Metódus használata

- Az objektum neve után legyen egy pont, majd a metódus neve, és végül az argumentumok, ha egyáltalán vannak.

- Példa:

```
SELECT ff.tea.név,  
       ff.árYenben(114.0)  
FROM Felszolgal ff  
WHERE ff.teázó.név = 'Joe''s Teahouse';
```



# Rendező metódusok (*order methods*): SQL-99

- Mindegyik *T* UDT két metódust definiálhat **EQUAL** és **LESSTHAN**.
  - Mindegyik metódus egy *T* típus argumentumot kap bemenetként és egy másik *T* típusú objektumra alkalmazza.
  - TRUE értéket ad vissza akkor és csak akkor ha a cél objektum = (vagy <) mint az az argumentumban szereplő objektum.
- Lehetővé teszi, hogy *T* típusú objektumokat hasonlítsunk össze =, <, >=, stb. segítségével a WHERE záradékban és a rendezésben (ORDER BY).

# Rendező metódusok: Oracle

- Bármilyen UDT típusra bármelyik metódust *rendező metódusnak* deklarálhatjuk.
- A rendező metódusok visszatérő értéke  $<0$ ,  $=0$ , vagy  $>0$  lehet, ahogy a SELF objektumhoz viszonyítva az argumentum értéke  $<$ ,  $=$ , vagy  $>$

# Példa: Rendező metódusok deklarálás

□ Rendezd a TeázóTípus objektumokat név szerint:

```
CREATE TYPE TeázóTípus AS OBJECT (  
    név CHAR(20),  
    cím CHAR(20),  
    ORDER MEMBER FUNCTION before(  
        teázó2 IN TeázóTípus) RETURN INT,  
    PRAGMA RESTRICT_REFERENCES(before,  
        WNDS, RNDS, WNPS, RNPS)  
);  
/
```

Nem kérdezi le az adatbázistáblákat,  
nincs adatbázis/csomag-állapotváltozás.  
Egy „csomag” eljárások és változók gyűjteménye.

# Példa: Rendező metódusok definiálás

```
CREATE TYPE BODY TeázóTípus AS
  ORDER MEMBER FUNCTION
    before(teázó2 TeázóTípus) RETURN INT IS
  BEGIN
    IF SELF.név < teázó2.név THEN RETURN -1;
    ELSIF SELF.név = teázó2.név THEN RETURN 0;
    ELSE RETURN 1;
    END IF;
  END;
END;
/
```