Objektum relációs lehetőségek az Oracle-ben

Katona Endre „Adatbázis alapú rendszerek” diasorozata alapján

Az objektum-relációs adatmodell

- ***Objektum-relációs modell*** (**SQL3)**: A relációs világba (SQL92) átemel objektum-orientált elemeket. A reláció továbbra is alapfogalom, amelyet absztrakt adattípusok definiálásával bővíthetünk.
- ***Objektum-relációs elv:*** Belül minden relációsan működik. Erre egy ráépülő rétegként alakítják ki az objektum-orientált felületet.
- **ORDBMS = Object-Relational DBMS**

A relációs modell bővítése - 1

Alapelv: a reláció (adattábla) továbbra is alapfogalom.

Főbb bővítések:

- ***Kollekciótípusok (komplex attribútumtípusok):*** struktúra, halmaz, multihalmaz, lista, struktúrák halmaza (azaz tábla, lásd később: beágyazott tábla).
- ***Metódusok:*** műveletek a definiált új objektumtípusokhoz.
- ***Hivatkozások:*** kapcsolatok kialakítása

A relációs modell bővítése - 2

Kétféle objektumot szoktak megkülönböztetni:

- ***Sorobjektum:*** a tábla egy sorát tekintjük objektumnak (struktúra). Általában ezt értjük objektumon.
  Példa: Olvasó objektum: (olvasószám, név, lakcím)
- ***Oszlopobjektum:*** a relációséma egy attribútumát tekintjük objektumnak.
  Példa: lakcím objektum: (irszám, helység, utca, házszám)

Objektumazonosító (OID): sorobjektumok azonosítására. Tartalmilag azonos sorokat is megkülönböztet. (Hasonló a ROWID-hez, de nem azonos vele.)

A relációs modell bővítése - 3

Kétféle tábla van:

- ***Relációs tábla:*** mint a relációs modellben.
- ***Objektumtábla:*** sorobjektumok halmaza. Kollekciótípusokat és metódusokat tartalmazhat.

Objektumnézet (object view): relációs alaptáblára definiált objektum-nézettábla.

Összetett attribútumok

Összetett attribútum:

A relációséma egy attribútuma struktúra.

Például $R(a_1, a_2, \mathbf{s(b_1, b_2, b_3)}, a_4)$

Példa:

Könyv (<u>könyvszám</u>, szerző, cím)

Olvasó (<u>olvszám</u>, név, **lakcím(irszám, helység, utca, hsz)**)

Jelölés: lakcím kisbetűvel.

Többértékű attribútumok

Többértékű attribútum:

A relációséma egy attribútuma halmaz.

Például $R(a_1, a_2, \mathbf{A_3}, a_4)$

Példa:

Könyv (<u>könyvszám</u>, **Szerző**, cím)

Olvasó (<u>olvszám</u>, név, lakcím(irszám, helység, utca, hsz))

Jelölés: Szerző nagybetűvel.

Beágyazott táblák

Beágyazott tábla (nested table):

A relációséma egy attribútuma maga is relációséma (azaz struktúrahalmazt reprezentál).

Például $R_1(a_1, a_2, \mathbf{R_2(b_1, b_2, b_3)}, a_4)$

Példa: egy olvasónak több lakcíme lehet:

Könyv (<u>könyvszám</u>, Szerző, cím)

Olvasó (<u>olvszám</u>, név, **Lakcím(irszám, helység, utca, hsz)**)

Jelölés: Lakcím nagybetűvel.

Beágyazott tábla szemléltetése

(egy személynek több lakcíme lehet)

Olvszám	***Név***	***Lakcím***
122	Kiss	***IrSzám*** | ***Helység*** | ***Utca*** | ***Hsz*** 8423 | Győr | Fő u. | 123.
612	Nagy	***IrSzám*** | ***Helység*** | ***Utca*** | ***Hsz*** 6721 | Szeged | Virág u. | 10. 7581 | Pécs | Kő u. | 7.
355	Tóth	***IrSzám*** | ***Helység*** | ***Utca*** | ***Hsz*** 8420 | Győr | Jég u. | 18.

Hivatkozások

Hivatkozás típusú attribútum: egy másik objektumtábla adott sorára (objektumazonosítójára) hivatkozik.

Jelölések:

- a: hagyományos attribútum
- a(*T): a T tábla egy sorára hivatkozó attribútum.
- a({*T}): T-beli sorok halmazára hivatkozó attribútum.

A hivatkozás ***kapcsolatok*** kezelésére alkalmas:

- Hasonló a külső kulcshoz, de nem egyenértékű vele!
- Ha hivatkozott sort töröljük, majd azonos tartalommal újra felvesszük, akkor új objektumazonosítót kap.

Példák hivatkozásra - 1

Példa 1:1 kapcsolatra

Könyv (könyvszám, Szerző, cím, kivétel, **olvasója(*Olvasó)**)

Olvasó (olvasószám, név, lakcím(irsz, helység, utca, hsz), **könyve(*Könyv)**)

Példa 1:N kapcsolatra

Könyv (könyvszám, Szerző, cím, kivétel, **olvasója(*Olvasó)**)

Olvasó (olvasószám, név, lakcím(irsz, helység, utca, hsz), **könyvei({*Könyv})**)

Példák hivatkozásra - 2

Példa N:M kapcsolatra

Könyv (könyvszám, Szerző, cím, **olvasói({*Olvasó})**)

Olvasó (olvasószám, név, lakcím(irsz, helység, utca, hsz), **könyvei({*Könyv})**)

Probléma: kivétel és visszahozás dátuma nem tartható nyilván.

Megoldás:

Könyv (könyvszám, Szerző, cím, **Kölcsön(olvasója(*Olvasó), kivétel, visszahozás)**)

Olvasó (olvasószám, név, lakcím(irsz, helység, utca, hsz), **könyvei({*Könyv})**)

Objektum-relációs lehetőségek Oracle-ben

- ***ADT = Abstract Data Type = UDT = User Defined Type:*** felhasználó által definiált új adattípus.
- **Deklarálása:**

```
CREATE TYPE típusnév AS OBJECT
( attribútumok
metódusok
);
```

- **A típusokhoz metódusokat is létrehozhatunk (itt nem tárgyaljuk).**

Példa új adattípus használatára - 1

- Olvasó (olvszám, név, lakcím(irszám, helység, utca, hsz))

```
CREATE TYPE CímTípus AS OBJECT
( irszám  NUMBER,
  helység CHAR(20),
  utca    CHAR(20),
  házszám NUMBER,
);
CREATE TABLE Olvasó
( olvasószám NUMBER,
  név CHAR(30),
  lakcím CímTípus
);
```

- Itt a címTípust ***oszlopobjektumként*** használjuk.

Példa új adattípus használatára - 2

- ***Konstruktor metódus:***
- implicite definiálódik minden objektumtípushoz
- neve megegyezik az objektumtípuséval
- paraméterei megegyeznek az attribútumokkal
- visszaadott értéke az új objektum

- ***Beszúrás konstruktor metódussal:***

```
INSERT INTO Olvasó VALUES
(123, 'Nagy Péter',
CímTípus(6720,'Szeged','Kő u.',3)
);
```

Példa új adattípus használatára - 3

- *Lekérdezés (alias név kötelező):*

```
SELECT Olv.név
FROM Olvasó Olv
WHERE Olv.lakcím.helység = 'Pécs';
```

Egymásba ágyazott típusok

- Olvasó (olvasószám,
  személy(név, lakcím(irszám, helység, utca, hsz)))

```
CREATE TYPE CímT AS OBJECT
( irszám  NUMBER,
  helység CHAR(20),
  utca    CHAR(20),
  házszám NUMBER
);
CREATE TYPE SzemélyT AS OBJECT
( név CHAR(30), lakcím CímT );

CREATE TABLE Olvasó
( olvasószám NUMBER,
  személy SzemélyT
);
```

Egymásba ágyazott típusok használata

- ***Beszúrás konstruktor függvénnyel:***

```
INSERT INTO Olvasó VALUES
   (123, SzemélyT('Nagy Péter',
    CímT(6720,'Szeged','Kő u.',3))
   );
```

- ***Lekérdezés:***

```
SELECT Olv.olvasószám
FROM Olvasó Olv
WHERE Olv.személy.lakcím.helység =
  'Pécs';
```

Objektumtáblák

- ***Objektumtábla: sorobjektumok (ADT-k) halmaza.***
- ***Létrehozása:***

CREATE TYPE típus AS OBJECT (attribútumok);

CREATE TABLE tábla OF típus;

- ***Az objektumtáblát kétféleképp kezelhetjük:***
- egyoszlopos táblaként,
- többoszlopos táblaként (mintha relációs tábla lenne).

Példa objektumtáblára

```
CREATE TYPE OlvasóT AS OBJECT
( olvasószám NUMBER,
  név CHAR(30),
  lakcím CHAR(50)
);

CREATE TABLE Olvasó OF OlvasóT;
```

- Itt az OlvasóT típust ***sorobjektumként*** használjuk.

Objektumtábla használata

- ***Beszúrás egyoszlopos táblaként, konstruktorral:***

```
INSERT INTO Olvasó VALUES
( OlvasóT(112,'Kiss','8423 Győr Fő u.123') );
```

- ***Beszúrás többoszlopos táblaként:***

```
INSERT INTO Olvasó VALUES
(112,'Kiss','8423 Győr Fő u.123');
```

- ***Lekérdezés egyoszlopos táblaként:***

```
SELECT VALUE(Olv) FROM Olvasó Olv
  WHERE Olv.olvasószám=112;
```

- ***Lekérdezés többoszlopos táblaként:***

```
SELECT név, lakcím FROM Olvasó
  WHERE olvasószám=112;
```

Beágyazott táblák (nested tables)

- ***Objektum típus definiálása:***
- CREATE TYPE típus AS OBJECT (attribútumok);

- ***Tábla típus definiálása:***
- CREATE TYPE táblatípus AS TABLE OF típus;

Példa beágyazott táblára

- Olvasó (olvszám, név, Lakcím(irszám, helység, utca, hsz))

```
CREATE TYPE CímT AS OBJECT
( irszám  NUMBER,
  helység CHAR(20),
  utca    CHAR(20),
  házszám NUMBER
);

CREATE TYPE CímekT AS TABLE OF CímT;

CREATE TABLE Olvasó
( olvasószám NUMBER,
  név CHAR(30),
  lakcím CímekT
) NESTED TABLE lakcím STORE AS lakcímTábla;
```

Beágyazott tábla használata

- ***Beszúrás konstruktor-függvényekkel:***

```
INSERT INTO Olvasó VALUES
(612, 'Nagy Éva', CímekT(
  CímT(6721,'Szeged','Virág u.',10),
  CímT(7581,'Pécs','Kő u.',7)
));
```

Beágyazott tábla helyett külső kulcs

Olvasó (olvasószám, név)

112	Kiss
612	Nagy
355	Tóth

Lakcímek (*olvasószám*, lakcím)

112	8423 Győr Fő u. 123
612	6721 Szeged Virág u. 10
612	7581 Pécs Kő u. 7
355	8420 Győr Jég u. 18

Altípusok, öröklés

```
CREATE TYPE helyiség AS OBJECT
( épület CHAR(10),
  ajtószám NUMBER(4),
  név VARCHAR2(20),
  terület NUMBER
) NOT FINAL;
CREATE TYPE tanterem UNDER helyiség
( férőhely NUMBER(4),
  tábla VARCHAR2(20),
  vetítő VARCHAR2(20)
);
```

- A *tanterem* örökli a *helyiség* attribútumait.

Helyettesíthetőség

- **`CREATE TABLE Helyiségek OF helyiség;`**

- A *Helyiségek* objektumtábla vegyesen tartalmazhat fő- és altípusokat:

```
INSERT INTO Helyiségek VALUES
  (helyiség('Irinyi', 123, 'Raktár', 25));
INSERT INTO Helyiségek VALUES
  (tanterem('Bolyai', 205, 'Riesz', 51,
     92, 'normál tábla', 'vetítővászon'));
```

- ***Többszörös öröklés nincs* (a 10g verzióban).**

Dinamikus tömbök

- ***Dinamikus tömb:*** absztrakt adattípusként hozható létre a VARRAY (variable length array) kulcsszóval.
- Mérete tetszőleges, de maximális mérete megadandó.

Példa: Könyv (könyvszám, Szerző, cím)

```
CREATE TYPE SzerzőT AS VARRAY(5)
   OF VARCHAR2(20);

CREATE TABLE Könyv
( könyvszám NUMBER,
   szerző SzerzőT,
   cím CHAR(50)
);
```

Dinamikus tömb használata

Beszúrás:

```
INSERT INTO Könyv VALUES
   (1234, SzerzőT('Sályi','Szelezsán'),
   'Adatbázisok');
```

Lekérdezés:

```
SELECT szerző, cím FROM Könyv;
```

A szerzők felsorolva jelennek meg.

- ***Megjegyzés:*** dinamikus tömb nem indexelhető.

Objektumazonosító (OID)

- ***OID: egyedi (bináris) belső azonosító minden sorhoz.*** (Relációs táblában ez nincs, de van helyette ROWID.)

- ***REF függvény:*** sorobjektum ➔ OID

- ***DEREF függvény:*** OID ➔ sorobjektum

Példa OID lekérdezésére

Típus definiálása:

```
CREATE TYPE OlvasóT AS OBJECT
( olvasószám NUMBER,
  név CHAR(30),
  lakcím CHAR(50)
);
```

Objektumtábla definiálása:

```
CREATE TABLE Olvasó OF OlvasóT;
```

OID lekérdezése:

```
SELECT REF(Olv) FROM Olvasó Olv
  WHERE Olv.olvasószám=112;
```

Kapcsolat megadása OID-vel - 1

Olvasó (olvasószám, név, lakcím)

Könyv (könyvszám, szerző, cím, olvasója(*Olvasó))

```
CREATE TYPE OlvasóT AS OBJECT
( olvasószám NUMBER,
  név CHAR(30),
  lakcím CHAR(50)
);
CREATE TABLE Olvasó OF OlvasóT;
CREATE TABLE Könyv
( könyvszám NUMBER,
  szerző CHAR(20),
  cím CHAR(30),
  olvasója REF OlvasóT
);
```

Kapcsolat megadása OID-vel - 2

Adott könyv olvasójának adatait kérjük le:

```
SELECT cím, DEREF(olvasója) FROM Könyv
  WHERE könyvszám=1234;
```

A lekérdezés eredményeként megkapjuk:

cím, olvasószám, név, lakcím

Hagyományos relációs megoldás (külső kulcs):

Olvasó (olvasószám, név, lakcím)

Könyv (könyvszám, szerző, cím, *olvasója*)

```
SELECT cím, olvasószám, név, lakcím
  FROM Könyv, Olvasó
  WHERE Könyv.olvasója=Olvasó.olvasószám
    AND könyvszám=1234;
```

Rekurzív kapcsolat megadása OID-vel

```
CREATE TYPE Személy AS OBJECT
( név     VARCHAR2(30),
  főnöke REF Személy
);
CREATE TABLE Dolgozó OF Személy;
```

Adott dolgozó főnökének, mint objektumnak a lekérdezése:

```
SELECT DEREF(D1.főnöke) FROM Dolgozó D1
  WHERE D1.név='Tóth Pál';
```

Adott dolgozó főnöke nevének lekérdezése:

```
SELECT D1.főnöke.név FROM Dolgozó D1
  WHERE D1.név='Tóth Pál';
```

Megj.: ez rövidített írásmód **DEREF(D1.főnöke).név** helyett.

OID-kapcsolat vagy külső kulcs kapcsolat?

Példa: kitörlünk egy hivatkozott rekordot (pl. Olvasó), majd azonos tartalommal újra felvesszük. Eredmény:

Külső kulcs kapcsolatnál integritás helyreáll.

OID kapcsolatnál integritás elromlik, mert az újonnan felvitt rekord már más OID-t kap!

Hogyan lehetne a két megközelítést összehozni?

Megoldás: objektumnézetek REF-ekkel (nem tárgyaljuk).