18.10.2011

Tipuri de date în PL/SQL

Prin tip de date se specifică
- formatul de stocare
- constrângerile care trb să le verif
- domeniul valorilor sale

Var PL/SQL
- specifice PL/SQL
- nespecifice PL/SQL

Nespecifice
- de tip scalar
  - numerice
  - caracter
  - logice/booleene (true, false, null)
  - date calendaristice
  - globalizare (NCHAR, NVARCHAR2)
- compuse
  - record
  - colecții
    - tabele indexate
    - tabele imbricate (nested table)
    - vectori
- LOB
  - interne (CLOB, BLOB, NCLOB) depuse în BD
    cărți, poze, depuse
  - externe (BFILE) - în afara BD
    filme
- tip referință
  - REF cursor
  - REF obiect
- tip obiect

Tipuri nespecifice PL/SQL
- var. de legătură (bind)
- var. gazdă
- var. indicator

Tipuri de date globalizare

PL/SQL suportă 2 seturi de caractere → 1) specifică BD-ului care este utilizat pt definirea identificatorilor și a codului sursă

2) o mulțime de caractere naționale folosită pt reprezentarea datelor cu caracter național (NCS)

Tipurile NCHAR, NVARCHAR permit stocarea în BD a șirurilor de caractere ce folosesc mulțimea NCS. Aceste tipuri de date suportă numai date unicode. Unicode furnizează o valoare cod unică pt fiecare caracter indiferent de program, platformă sau limbă.

<u>LOB</u>

sunt tipuri de date ale căror valori numite locatori specifică localizarea unor obiecte de dimensiuni mari - blocuri de date nestructurate cum ar fi muzică, filme, imagini grafice

<u>DBMS - LOB</u> : pachet - lucrul cu aceste tipuri

SELECT returnează date pt tipurile LONG
LONG RAW

$\updownarrow$ SELECT returnează adresa (locator) pt LOB-uri

Ref cursor și ref obiect reprezintă adresa, locația de memorie a unui element de tip cursor sau obiect în sensul OOP.

Tipul obiect - atribute, metode

<u>Tip nespecifice</u>

a) var de legătură - se declară în mediul gazdă pt transferul valorilor în respectiv din unul sau mai multe programe PL/SQL

<u>Exemplu</u>.
```
VARIABLE x
BEGIN
    SELECT COUNT(*) INTO :x
    FROM -
    WHERE -
END;
PRINT x
```

Variabila gazdă - atribuire transferul valorilor între mediul de programare (precompilatoare) și comenzile SQL ce comunică cu serverul

Variabile indicator - permite comunicarea valorii NULL între programul scris în lb gazdă și sistemul ORACLE

Declararea variabilelor

- atributul %TYPE - var cu acelaşi tip ca o coloană de tabel
  %ROWTYPE -- " ______ ca o linie de tabel

Utilizatorul poate să-şi definească propriile tipuri şi subtipuri în partea declarativă a unui bloc PL/SQL, subprogram sau pachet.

x := null;
- ce tip are x? Răspuns Orice tip, deoarece null apare în toate tipurile.

Tipuri compuse - înregistrări RECORD
- pot fi asignate valori unei înregistrări utilizând → SELECT
  → FETCH (cursoare)
  → :=
  o înregistrare poate fi atribuită altei înregistrări

Se poate ^insera^ o linie într-un tabel utilizând un record
Se poate reactualiza o linie a unui tabel utilizând record (sintaxa SET ROW)
Într-o înregistrare se poate → regăsi şi returna
sau
şterge informaţii din clauza returning a comenzilor update şi delete.

Colecţii.
- tip tablou indexat (index by-table) 1)
  tabel imbricat (nested table) 2)
  vector (varray) 3)

1) poate fi utilizat numai în declaraţii PL/SQL
2) şi 3) pot fi utilizate atât în declaraţii PL/SQL cât şi în declaraţii la nivelul schemei (ca tip a unei coloane a unui tabel relaţional,

Tabloul indexat în PL/SQL are 2 coloane, o col. cu cheia primară şi o coloană care include val. efective

Tablou indexat → tabel relațional
　　prin INSERT　LOOP
tabel relațional → tablou indexat
　　a) prin FETCH (cursoare)
　　b) liste de atribuiri (buclă)
* cum pot șterge liniile unui tablou indexat?
→ prin metoda delete.
→ se asignează null componentelor sale ?!?
se declară un alt tablou indexat, neinițializat și acesta se asignează tabloului de șters

```
set serveroutput on
  DECLARE
    TYPE tablou_numar IS TABLE OF NUMBER
            INDEX BY PLS_INTEGER;
   BEGIN
    FOR i IN 1.. 20 LOOP
          v_tablou(i):=i*i;
          DBMS_OUTPUT.PUT_LINE(v_tablou(i));
     END LOOP;
    FOR i IN v_tablou.FIRST.. v_tablou.LAST LOOP
          v_tablou(i):=NULL;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('tabloul are', || v_tablou.COUNT    etc
```

<u>Vectori</u> - spre deosebire de tablouri indexate, au dim max stabilită la declarare care nu poate fi depășită
= depuși în memorie în afara BD
= sunt structuri dense, nu pot șterge elemente individual
(cum facem? indicii ∈ [1 .. lim_max]

```
DECLARE
   TYPE secventa IS VARRAY(5) OF VARCHAR2(10);
   v_sec secventa := secventa('alb', 'negru', 'rosu', 'verde');

   BEGIN
      v_sec(3) := 'rosu';
      v_sec EXTEND; -- adaugă un elem null
      v_sec(5) := 'albastru'
      -- extinderea la 6 elem va genera eroare
      v_sec.EXTEND;
   END;
```