Sistemi baza podataka

Dr Ivan Luković

Dr Slavica Kordić

Vladimir Dimitrieski

PL/SQL - OSNOVI

Uloga jezika PL/SQL i struktura PL/SQL programa

- PL/SQL - jezik III generacije
- PL/SQL - predstavlja proceduralno proširenje SQL-a
- PL/SQL se može koristiti iz različitih okruženja
  - SQL*Plus
  - Oracle Developer Suite (Forms, Reports, Oracle Portal, Oracle Discoverer)
  - SQL Developer

Osobine jezika PL/SQL

- Strukturirano programiranje i organizacija programa po blokovima
- Proceduralna podrška osnovnih struktura: sekvenca, selekcija i iteracija
- Podrška neproceduralnog jezika SQL
- Mogućnost deklarisanja promenljivih i konstanti i upotreba osnovnih i složenih tipova podataka
- Upotreba kursora - proceduralna obrada rezultata SQL SELECT naredbe
- Mogućnost obrade grešaka i izuzetaka, indikovanih od strane DBMS ORACLE

Osnovna struktura PL/SQL bloka

- **Tipovi PL/SQL blokova**
  - anonimni (netipizovani)
  - tipizovani (procedura, funkcija)

Struktura anonimnog PL/SQL bloka

[DECLARE
    Deklarativni (neobavezni) deo programa:
        * deklaracija i inicijalizacija promenljivih
        * deklaracija i inicijalizacija konstanti
        * deklaracija tipova podataka
        * deklaracija kursora
        * deklaracija izuzetaka
        * deklaracija procedura i funkcija
]
BEGIN
    Izvršni (obavezni) deo programa:
        * Proceduralne naredbe
        * SQL naredbe
[EXCEPTION
    Deo za obradu izuzetaka (neobavezni):
        * WHEN <izuzetak> THEN <blok izvršnih naredbi>
]
END;

Primer jednog PL/SQL bloka

```
-- Ovo je oznaka za jednolinijski komentar
/*
        Ovo je način za definisanje višelinijskog komentara
*/
DECLARE         -- Deklarativni deo bloka
        Br_torki NUMBER(6) := 0;                -- Deklarisana i inicijalizovana lokalna promenljiva
        L_OznDeo Deo.OznDeo%TYPE;               -- Deklaracija saglasno tipu kolone iz tabele Deo
BEGIN                                           -- Izvršni deo bloka
        SELECT COUNT(*)
        INTO Br_torki
        FROM Deo_koji_se_dobavlja
        WHERE OznDeo = :p_OznDeo;               -- Referenca na promenljivu iz pozivajućeg okruženja

        IF Br_torki = 0 THEN
                SELECT COUNT(*)
                INTO Br_torki
                FROM Deo_iz_proizvodnje
                WHERE OznDeo = :p_OznDeo;  -- Referenca na promenljivu iz pozivajućeg okruženja

                IF Br_torki = 0 THEN
                        RAISE NO_DATA_FOUND;
                END IF;
        END IF;

-- Deo za obradu izuzetaka
EXCEPTION       -- Povratak na izvršni deo programa NIJE MOGUĆ!
        /*
                NO_DATA_FOUND je predefinisani IZUZETAK
        */
        WHEN NO_DATA_FOUND THEN
                Raise_application_error (-20000, 'Deo mora biti sadržan u najmanje jednoj potklasi');
END;
```

Osnovni leksički elementi

- **Skup simbola**
  - A-Z, a-z, 0-9, (,), &, <, >, =, !, ;, ., ', @, %, ^, {, }, [,], _, ", #, ?, +, -, *, /
- **Delimiteri**
  - (,), %, <>, !=, *, **, :=, itd.
- **Literali**
  - numerički (114, 12.5, -1.E3)
  - karakter ('O''vo je string')
  - logički (TRUE, FALSE, NULL)
- **Komentari**:
  - jednolinijski (--)
  - višelinijski (/* */)
- **Identifikatori**
  - do 30 znakova, prvi znak mora biti slovo.
  - dozvoljeni karakteri: slova, brojevi, _, #, $.

Ugrađivanje blokova i tok izvršenja programa

DECLARE
Deklarativni deo programa:
* GLOBALNE DEKLARACIJE
* deklaracije procedura i funkcija

PROCEDURE | FUNCTION
lokalne deklaracije
BEGIN

EXCEPTION

END;

BEGIN
Izvršni deo programa:

DECLARE
lokalne deklaracije
BEGIN

EXCEPTION

END;

EXCEPTION
Deo za obradu izuzetaka:
* WHEN <izuzetak> THEN
DECLARE
lokalne deklaracije
BEGIN

EXCEPTION

END;

END;

Ugrađivanje blokova i tok izvršenja programa

- Važe uobičajeni mehanizmi toka izvršenja programa
  - Nakon završetka ugrađenog bloka, kontrola izvođenja programa se predaje pozivajućem ("okružujućem") bloku
- Koncept lokalnosti i globalnosti deklaracija važi na uobičajen način
  - Lokalne deklaracije nisu vidljive u pozivajućem bloku. Globalne deklaracije su vidljive u pozvanom bloku.

Tipovi podataka

- Skalarni (osnovni)
  - Specifični Oracle tipovi i ANSI SQL standardni tipovi
  - karakter
    - VARCHAR2 (do 32767 bajtova)
    - CHAR (do 32767, default 1)
    - LONG (do 32760)
      - slično sa *varchar2* samo malo kraći
    - NVARCHAR2, NCHAR
  - numerički
    - NUMBER(p,s) (decimalni tip, od 1E-130 do 10E125)
    - BINARY_INTEGER (4B integer, u rasponu od –(2E31-1) do 2E31-1)
    - PLS_INTEGER (4B "pakovani" integer, u rasponu od –(2E31-1) do 2E31-1) – zauzima manje prostora
      - u novijim aplikacijama se često koristi umesto *binary_integer*-a zbog boljih performansi

Tipovi podataka

- Skalarni (osnovni)
  - datumski
    - DATE
    - TIMESTAMP
    - TIMESTAMP WITH TIME ZONE
    - TIMESTAMP WITH LOCAL TIME ZONE
    - INTERVAL DAY TO SECOND
    - INTERVAL YEAR TO MONTH
  - logički
    - BOOLEAN

Tipovi podataka

- Složeni (Composite)
  - RECORD
  - TABLE
  - VARRAY
- Pokazivački (Reference)
  - REF CURSOR, REF objektni_tip
  - ROWID, RAW
- LOB
  - BFILE (fajl do 4GB)
  - BLOB (do 4GB)
  - CLOB (do 4GB), NCLOB (do 4GB)
  - RAW (do 32760), LONG RAW (do 32760)

Promenljive i konstante

- PL/SQL promenljive i konstante
  - Skalarne (osnovne)
  - Složene (Composite)
  - Pokazivačke (Reference)
  - LOB

Deklarisanje PL/SQL promenljivih i konstanti

identifier [CONSTANT] datatype
[NOT NULL] [:= | DEFAULT expr]

identifier [CONSTANT] {variable%TYPE |
    table.column%TYPE}
[NOT NULL] [:= | DEFAULT expr]

Primeri deklaracija promenljivih i konstanti

```
DECLARE
   V_prom1 NUMBER(2);
   V_prom2 CHAR;
   V_prom3 VARCHAR2(40) := ' ';
   V_prom4 VARCHAR2(40) NOT NULL := ' ';
   V_prom5 VARCHAR2(40) NOT NULL DEFAULT ' ';
   V_prom6 DATE NOT NULL := SYSDATE + 2;
   C_prom7 CONSTANT DATE:= SYSDATE;
   V_prom8 V_Prom6%TYPE := TO_DATE('01.01.2001',
   'DD.MM.YYYY');
   V_prom9 Radnik.Mbr%TYPE := 100;
BEGIN
   NULL;
END;
```

Deklarisanje PL/SQL promenljivih i konstanti

- Pravila deklarisanja:
  - Konstante moraju biti inicijalizovane.
  - NOT NULL promenljive moraju biti inicijalizovane.
  - Jedna deklaracija dozvoljava deklarisanje tačno jednog identifikatora.
  - Uvesti i poštovati konvencije imenovanja promenljivih i konstanti.
  - **Ne nazivati promenljive i konstante istim imenima, kao što su nazivi kolona tabela**, **ili nazivi samih tabela**.
    - obično *v_imeVarijable*

PL/SQL izrazi

- Klasifikacija PL/SQL izraza:
  - Numerički izrazi
  - Karakter izrazi
  - Logički izrazi
  - Datumski izrazi
  - Selekcioni izrazi (izrazi IF-tipa)

PL/SQL izrazi

- Izrazi se formiraju na uobičajen način, korišćenjem odgovarajućih operatora.
- U izrazima je dozvoljena upotreba najvećeg broja predefinisanih ***jednosložnih*** ORACLE SQL funkcija.
- U izrazima je dozvoljena upotreba jednosložnih, korisnički definisanih funkcija.
- Dozvoljena je upotreba predefinisanih operatora: [NOT] IN, [NOT] LIKE, [NOT] BETWEEN AND i IS [NOT] NULL.
- **LOGIČKI TIP PODATAKA**: BOOLEAN. Moguće vrednosti: TRUE, FALSE, NULL. Vrednost NULL se u logičkim izrazima tretira kao **FALSE**!

PL/SQL izrazi

- Konverzija podataka različitih tipova
  - Pri izračunavanju izraza, vrši se implicitna konverzija podataka, kada je to moguće.
  - Preporučljivo je koristiti uvek funkcije za eksplicitnu konverziju podataka TO_CHAR, TO_DATE i TO_NUMBER.

Selekcioni izrazi (Izrazi IF tipa)

```
CASE [expr] WHEN comparison_expr1 THEN return_expr1
        [ WHEN comparison_expr2 THEN return_expr2
          WHEN comparison_exprn THEN return_exprn
        ]
        [ ELSE else_expr
        ]
END;
```

Primer selekcionog PL/SQL izraza

```
CASE Status
    WHEN 'A' THEN 'Odlican'
    WHEN 'B' THEN 'Zadovoljava'
    ELSE 'Ne zadovoljava‘
END;

CASE
    WHEN Status = 'A' THEN 'Odlican'
    WHEN Status = 'B' THEN 'Zadovoljava'
    ELSE 'Ne zadovoljava‘
END;
```

Osnovne PL/SQL naredbe

- **"Prazna" naredba**
  - **NULL**
  - svaka BEGIN – END sekcija mora da ima makar jednu naredbu

- Primer upotrebe prazne naredbe

```
BEGIN
  NULL;
END;
```

Osnovne PL/SQL naredbe

- Naredba dodele vrednosti
  - Variable := expression

- Primeri upotrebe naredbe za dodelu vrednosti

```
DECLARE
    v_a BOOLEAN := TRUE;
    v_b NUMBER NOT NULL := 0;
BEGIN
    v_a := 5 > 3;
    v_b := v_b + 1;
END;
```

Prikaz vrednosti izraza

- PL/SQL na nivou DBMS-a i SQL*Plus-a – kombinacija:
  - SET SERVEROUTPUT ON i
  - DBMS_OUTPUT.PUT_LINE (message)

View -> DBMS OUTPUT

Primeri predaje vrednosti izraza

```
DECLARE
        V_A NUMBER;
        S_A NUMBER := '10';
BEGIN
    V_A := S_A * 6 / 1.5;
    DBMS_OUTPUT.PUT_LINE('Stampa vrednosti za V_A');
    DBMS_OUTPUT.PUT_LINE('Vrednost za V_A je: ' || V_A);
    DBMS_OUTPUT.PUT_LINE('Vrednost za V_A je: ' ||
    TO_CHAR(V_A));
END;
```

Zadatak

- U konzolu ispisati trenutno vreme, datum rođenja, dan u sedmici kada ste rođeni kao i broj dana koje ste proživeli do sada. Datum rođenja uneti kroz interaktivni prompt.

Rešenje

```
BEGIN
  DBMS_OUTPUT.PUT_LINE('Danas je: ' || SYSDATE);
  DBMS_OUTPUT.PUT_LINE('Rodjen sam: ' || TO_DATE('&Dat_rodj',
        'DD.MM.YYYY'));
  DBMS_OUTPUT.PUT_LINE('Bio je: ' || TO_CHAR(TO_DATE(
        '&Dat_rodj', 'DD.MM.YYYY'), 'DAY'));
  DBMS_OUTPUT.PUT_LINE('Sada sam stariji ' || TO_CHAR(
        ROUND(SYSDATE - TO_DATE('&Dat_rodj',
        'DD.MM.YYYY'),0)) || ' dana');
END;
```

Unos kroz podrazumevani prompt

- Dve varijante:
  - *&promenjiva*
    - za svaku promenjivu zahteva se unos vrednosti
  - *&&promenjiva*
    - unos se zahteva samo jednom za vreme trenutne sesija
    - ako je potrebno obrisati vrednost potrebno je izvršiti naredbu
      - UNDEFINE *promenjiva*

Rešenje 2

```
UNDEFINE Dat_rodj;

BEGIN
  DBMS_OUTPUT.PUT_LINE('Danas je: ' || SYSDATE);
  DBMS_OUTPUT.PUT_LINE('Rodjen sam: ' || TO_DATE('&&Dat_rodj',
          'DD.MM.YYYY'));
  DBMS_OUTPUT.PUT_LINE('Bio je: ' || TO_CHAR(TO_DATE(
          '&&Dat_rodj', 'DD.MM.YYYY'), 'DAY'));
  DBMS_OUTPUT.PUT_LINE('Sada sam stariji ' || TO_CHAR(
          ROUND(SYSDATE - TO_DATE('&&Dat_rodj',
          'DD.MM.YYYY'),0)) || ' dana');
END;
```

Ugrađivanje blokova i opseg delovanja promenljivih

```
DECLARE
-- opseg delovanja x – do kraja spoljnjeg bloka
  x  BINARY_INTEGER;
BEGIN
  DECLARE
    -- opseg delovanja y – do kraja unutrašnjeg bloka
    y  PLS_INTEGER;
  BEGIN
      y := x;
  END;
END;
```

Ugrađivanje blokova i opseg delovanja promenljivih

- **NAPOMENA:** Naziv lokalno deklarisane konstrukcije ima prioritet, u odnosu na naziv globalno deklarisane konstrukcije

Ugrađivanje blokova i opseg delovanja promenljivih

```
DECLARE
  x  BINARY_INTEGER;     -- vidljivost: u spoljnjem bloku
BEGIN
  DECLARE
    x  VARCHAR2(20);
          -- vidljivost: samo u unutrašnjem bloku
    y  PLS_INTEGER;
          -- vidljivost: samo u unutrašnjem bloku
  BEGIN
     y := TO_NUMBER(x, '$99,990.00');
  END;
END;
```

Upotreba SQL naredbi u PL/SQL-u

- Dva načina upotrebe:
  - direktni
  - posredni, putem PL/SQL kursora

Direktni način upotrebe SELECT naredbe

```
SELECT select_list
INTO {variable[, variable]...
                    | record_variable}
FROM table
[WHERE condition]
...
```

Direktni način upotrebe SELECT naredbe

- SELECT naredba mora da vrati **JEDAN I SAMO JEDAN** red
- U protivnom, dolazi do pokretanja odgovarajućih izuzetaka
- Klauzula INTO obezbeđuje memorisanje vrednosti preuzete (selektovane) torke
- U izrazima, upotrebljenim u okviru naredbe SELECT, moguće je referenciranje na PL/SQL i bind (host) promenljive
- NAPOMENA: važno je poštovati konvencije imenovanja promenljivih, kolona tabela i samih tabela

Zadatak

- Prebrojati projekte. Rezultat smestiti u definisanu varijablu i prikazati ga u konzoli.

Zadatak

```
DECLARE
    v_Count NUMBER(3);
BEGIN
    SELECT COUNT(*)
    INTO   v_Count
    FROM   Projekat;
    DBMS_OUTPUT.PUT_LINE(v_Count);
END;
```