Sintaxa PL/SQL

Comenzi SQL in PL/SQL

- Cum sunt procesate comenzile SQL:
  - SELECT – cum se salveaza rezultatul returnat
  - UPDATE/DELETE in cadrul blocului PL/SQL
  - LCD (COMMIT, ROLLBACK. SAVEPOINT) in PL/SQL

Sintaxa SELECT

Sintaxa:

```
SELECT  select_list
INTO    {variable_name[, variable_name]...
        | record_name}
FROM    table
[WHERE  condition];
```

Obs: clauza INTO este obligatorie; *cererea trebuie sa intoarca o singura linie*;

```
DECLARE
 v_fname VARCHAR2(25);
BEGIN
 SELECT first_name INTO v_fname
 FROM employees WHERE employee_id=200;
 DBMS_OUTPUT.PUT_LINE(' First Name is : '||v_fname);
END;
```

Sintaxa SELECT

Exemplu1:

```
DECLARE
 v_emp_hiredate   employees.hire_date%TYPE;
 v_emp_salary     employees.salary%TYPE;
BEGIN
  SELECT   hire_date, salary
  INTO     v_emp_hiredate, v_emp_salary
  FROM     employees
  WHERE    employee_id = 100;
  DBMS_OUTPUT.PUT_LINE ('Hire date is :'|| v_emp_hiredate);
  DBMS_OUTPUT.PUT_LINE ('Salary is :'|| v_emp_salary);
END;
```

Sintaxa SELECT

Exemplu2:

```
DECLARE
   v_sum_sal    NUMBER(10,2);
   v_deptno     NUMBER NOT NULL := 60;
BEGIN
  SELECT  SUM(salary)   -- group function
  INTO v_sum_sal  FROM employees
  WHERE     department_id = v_deptno;
  DBMS_OUTPUT.PUT_LINE ('The sum of salary is ' || v_sum_sal);
END;
```

Obs1: ! Curs 2: V_sum_sal := SUM(employees.salary); ???

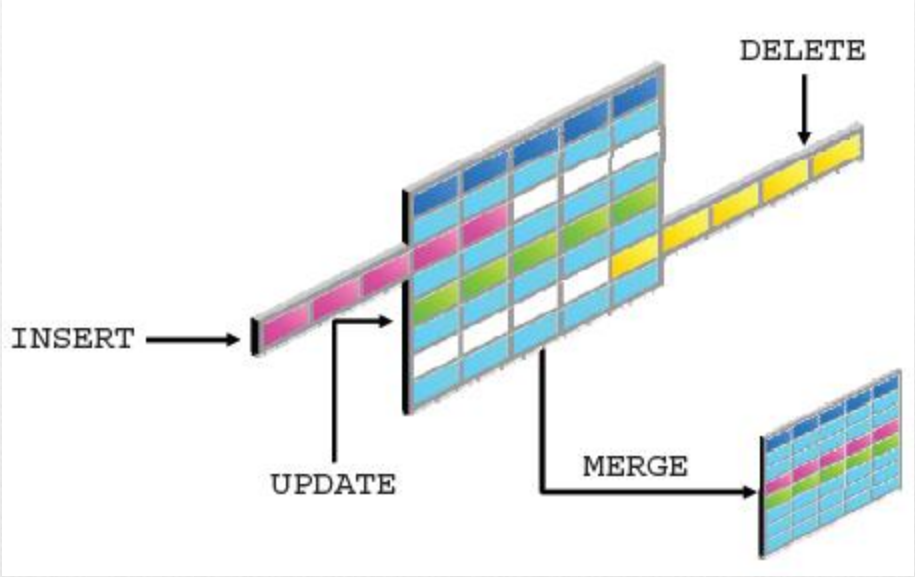
Obs2: ! Curs 2:

DECLARE
hire_date employees.hire_date%TYPE;
employee_id employees.employee_id%TYPE := 176;
BEGIN
SELECT hire_date INTO hire_date
FROM employees
WHERE employee_id = employee_id;
END;

DML in PL/SQL

Modificarile intr-un tabel dintr-o baza de date pot fi implementate cu ajutorul:

- INSERT
- UPDATE
- DELETE
- MERGE



DML in PL/SQL – INSERT/UPDATE

```
BEGIN
 INSERT INTO employees
  (employee_id, first_name, last_name, email,
   hire_date, job_id, salary)
   VALUES(employees_seq.NEXTVAL, 'Ruth', 'Cores',
   'RCORES',CURRENT_DATE, 'AD_ASST', 4000);
END;
```

```
DECLARE
  sal_increase   employees.salary%TYPE := 800;
BEGIN
  UPDATE       employees
  SET          salary = salary + sal_increase
  WHERE        job_id = 'ST_CLERK';
END;
```

DML in PL/SQL – DELETE/MERGE

Exemplu:

```
DECLARE
  deptno   employees.department_id%TYPE := 10;
BEGIN
  DELETE FROM   employees
  WHERE  department_id = deptno;
END;
```

```
BEGIN
MERGE INTO copy_emp c
     USING employees e
     ON (e.employee_id = c.empno)
   WHEN MATCHED THEN
     UPDATE SET
       c.first_name      = e.first_name,
       c.last_name       = e.last_name,
       c.email           = e.email,
       . . .
   WHEN NOT MATCHED THEN
     INSERT VALUES(e.employee_id, e.first_name, e.last_name,
          . . .,e.department_id);
END;
```

Cursoare predefinite

- Un cursor este un pointer catre o zona privata de memorie (definit de SGBD)
- Este utilizat pentru a controla/manipula rezultatul unei cereri
- Sunt 2 tipuri de variabile cursor:
  - Implicite (definite automat)
  - Explicite (definite de utilizator)

Atributele unui cursor

- Când se procesează o comandă LMD, motorul SQL deschide un cursor implicit.
- Atributele scalare ale cursorului implicit
  - SQL%ROWCOUNT
  - SQL%FOUND
  - SQL%NOTFOUND
  - SQL%ISOPEN

furnizează informații referitoare la ultima comandă INSERT, UPDATE, DELETE sau SELECT INTO executată.

- Înainte ca Oracle să deschidă cursorul SQL implicit, atributele acestuia au valoarea null.

Atributele unui cursor

- Exemplu:

```
DECLARE
v_rows_deleted VARCHAR2(30);
v_empno employees.employee_id%TYPE := 176;
BEGIN
DELETE FROM employees
WHERE employee_id = v_empno;
v_rows_deleted := (SQL%ROWCOUNT ||' row deleted.');
DBMS_OUTPUT.PUT_LINE (v_rows_deleted);
END;
```

Atributele unui cursor

- Exemplu:

```
DECLARE
  TYPE alfa IS TABLE OF NUMBER;  beta alfa;
BEGIN
  SELECT cod_artist BULK COLLECT INTO beta FROM artist;
  FORALL j IN 1..beta.COUNT
    INSERT INTO tab_art SELECT cod_artist,cod_opera
      FROM   opera
      WHERE  cod_artist = beta(j);
  FOR j IN 1..beta.COUNT LOOP
    DBMS_OUTPUT.PUT_LINE ('Pentru artistul ' || beta(j) || ' au
fost inserate ' ||  SQL%BULK_ROWCOUNT(j) || 'inregistrari');
  END LOOP;
  DBMS_OUTPUT.PUT_LINE ('Numarul total este '||SQL%ROWCOUNT);
END;
```

Comenzi de control a executiei

- IF
- CASE
- LOOP, WHILE, FOR
- Generalitati

Clauza IF

- Un program PL/SQL poate executa diferite porțiuni de cod, în funcție de rezultatul unui test (predicat). Instrucțiunile care realizează acest lucru sunt cele condiționale (IF, CASE).
- Structura instrucțiunii IF în PL/SQL este similară instrucțiunii IF din alte limbaje procedurale, permițând efectuarea unor acțiuni în mod selectiv, în funcție de anumite condiții. Instrucțiunea IF-THEN-ELSIF are următoarea formă sintactică:

```
IF condiţie1 THEN
   secvenţa_de_comenzi_1
[ELSIF condiţie2 THEN
   secvenţa_de_comenzi_2]
…
[ELSE
   secvenţa_de_comenzi_n]
END IF;
```

Clauza IF - exemplu

```
DECLARE
v_myage number:=&m;
BEGIN
    IF v_myage < 11 THEN
        DBMS_OUTPUT.PUT_LINE(' I am a child ');
    ELSIF v_myage < 20 THEN
        DBMS_OUTPUT.PUT_LINE(' I am young ');
    ELSIF v_myage < 30 THEN
        DBMS_OUTPUT.PUT_LINE(' I am in my twenties');
    ELSIF v_myage < 40 THEN
        DBMS_OUTPUT.PUT_LINE(' I am in my thirties');
    ELSE
        DBMS_OUTPUT.PUT_LINE(' I am always young ');
    END IF;
END;
```

Clauza CASE

- Permite implementarea unor condiţii multiple; are următoarea formă sintactică:

```
[<<eticheta>>]
CASE test_var
    WHEN valoare_1 THEN secvenţa_de_comenzi_1;
    WHEN valoare_2 THEN secvenţa_de_comenzi_2;
    …
    WHEN valoare_k THEN secvenţa_de_comenzi_k;
    [ELSE altă_secvenţă;]
END CASE [eticheta];
```

- Se va executa secvența_de_comenzi_p, dacă valoarea selectorului test_var are valoare_p. Selectorul test_var poate fi o variabilă sau o expresie complexă care poate conține chiar şi apeluri de funcții.
- Clauza ELSE este opțională

Clauza CASE - exemplu

```
DECLARE
  v_zi  CHAR(2) := UPPER('&p_zi');
BEGIN
  CASE
    WHEN v_zi = 'L'  THEN
              DBMS_OUTPUT.PUT_LINE('Luni');
    WHEN v_zi = 'M'  THEN
              DBMS_OUTPUT.PUT_LINE('Marti');
    …
    WHEN v_zi = 'D'  THEN
              DBMS_OUTPUT.PUT_LINE('Duminica');
    ELSE DBMS_OUTPUT.PUT_LINE('Este o eroare!');
  END CASE;
END;
```

Clauze iterative

- LOOP
- WHILE
- FOR

```
LOOP
  statement1;
  . . .
  EXIT [WHEN condition];
END LOOP;
```

```
WHILE condition LOOP
  statement1;
  statement2;
  . . .
END LOOP;
```

```
FOR counter IN [REVERSE]
    lower_bound..upper_bound LOOP
  statement1;
  statement2;
  . . .
END LOOP;
```

Clauze iterative exemple

```
DECLARE
  v_countryid    locations.country_id%TYPE := 'CA';
  v_loc_id       locations.location_id%TYPE;
  v_counter      NUMBER(2) := 1;
  v_new_city     locations.city%TYPE := 'Montreal';
BEGIN
  SELECT MAX(location_id) INTO v_loc_id FROM locations
  WHERE country_id = v_countryid;
  LOOP
    INSERT INTO locations(location_id, city, country_id)
    VALUES((v_loc_id + v_counter), v_new_city, v_countryid);
    v_counter := v_counter + 1;
    EXIT WHEN v_counter > 3;
  END LOOP;
END;
```

Clauze iterative exemple

```
DECLARE
  v_countryid   locations.country_id%TYPE := 'CA';
  v_loc_id      locations.location_id%TYPE;
  v_new_city    locations.city%TYPE := 'Montreal';
  v_counter     NUMBER := 1;
BEGIN
  SELECT MAX(location_id) INTO v_loc_id FROM locations
  WHERE country_id = v_countryid;
  WHILE v_counter <= 3 LOOP
    INSERT INTO locations(location_id, city, country_id)
    VALUES((v_loc_id + v_counter), v_new_city, v_countryid);
    v_counter := v_counter + 1;
  END LOOP;
END;
```

Clauze iterative exemple

```
DECLARE
  v_countryid   locations.country_id%TYPE := 'CA';
  v_loc_id      locations.location_id%TYPE;
  v_new_city    locations.city%TYPE := 'Montreal';
BEGIN
  SELECT MAX(location_id) INTO v_loc_id
    FROM locations
    WHERE country_id = v_countryid;
  FOR i IN 1..3 LOOP
    INSERT INTO locations(location_id, city, country_id)
    VALUES((v_loc_id + i), v_new_city, v_countryid );
  END LOOP;
END;
```

Clauza CONTINUE

- Care este ultima valoare a lui **v_total** afisata?

```
DECLARE
v_total SIMPLE_INTEGER := 0;
BEGIN
FOR i IN 1..10 LOOP
    v_total := v_total + i;
dbms_output.put_line('Total is: '|| v_total);
    CONTINUE WHEN i > 5;
    v_total := v_total + i;
dbms_output.put_line('Out of Loop Total is:'|| v_total);
END LOOP;
END;
```

Exercitii

- Se considera *emp* drept o copie a tabelului *employees* in care a fost adaugata coloana stars in care se trece cate o '*' pentru fiecare 1000 din salariu?
- Comentati rezolvarea urmatoare:

```
DECLARE
v_empno emp.employee_id%TYPE := 176;
v_asterisk emp.stars%TYPE := NULL;
v_sal emp.salary%TYPE;
BEGIN
   SELECT NVL(ROUND(salary/1000), 0) INTO v_sal
   FROM emp WHERE employee_id = v_empno;
FOR i IN 1..v_sal LOOP
   v_asterisk := v_asterisk ||'*';
END LOOP;
   UPDATE emp SET stars = v_asterisk
   WHERE employee_id = v_empno; COMMIT;
END;
```