

Limbajul PL/SQL- Interactiunea cu Oracle

- Functii SQL:

PL/SQL permite utilizarea functiilor SQL, inclusiv AVG,MIN, MAX,COUNT, SUM, STDDEV, VARIANCE

- nume_functie ([ALL|DISTINCT] expresie)
- functiile de grupare ignora NULL,cu exceptia lui COUNT(*)

- Pseudocoloane SQL:

CURRVAL, LEVEL, NEXTVAL, ROWID, ROWNUM

- Se folosesc in instr.SQL, nu in cele procedurale.
 - CURRVAL si NEXTVAL:
 - O secventa este o schema ce genereaza numere secventiale.

Limbajul PL/SQL- Interactiunea cu Oracle

- `CREATE SEQUENCE nume_secv INCREMENT BY pas
MINVALUE val1 MAXVALUE val2 NOCYCLE NOCACHE
ORDER`
 - Secventa este initializata prin `NEXTVAL`.
 - Exista tabela `USER_SEQUENCES` cu `LAST_VAL`
- `LEVEL`
- `ROWID` returneaza adresa binara a unei linii.
 - Tipul de data este `ROWID`

`DECLARE`

`mid1 ROWID;`

`mname emp.name%TYPE;`

`mjob emp.job%TYPE;`

`CURSOR C1 IS SELECT ename, job, ROWID FROM emp;`

Limbajul PL/SQL- Interactiunea cu Oracle

```
BEGIN
```

```
OPEN C1;
```

```
LOOP
```

```
    FETCH C1 INTO mname, mjob, mid1;
```

```
    EXIT WHEN C1%NOTFOUND;
```

```
    UPDATE emp SET sal=sal*1.1 WHERE
```

```
        ROWID=mid1;
```

```
    ...
```

```
    COMMIT;
```

```
END LOOP;
```

```
CLOSE C1;
```

```
END;
```

Limbajul PL/SQL- Interactiunea cu Oracle

- ROWNUM asigneaza un numar unic fiecarei linii.
CURSOR C1 IS SELECT empno, sal FROM emp
WHERE sal>2000 AND ROWNUM <15;
- Operatori SQL:
ALL, ANY/SOME, BETWEEN, EXISTS, IN, IS NULL,
LIKE, AND, NOT ,OR.
- Operatori de tip multime:
INTERSECT, MINUS, UNION, UNION ALL
- Operatori de tip linie:
ALL, DISTINCT, PRIOR, CONNECT BY

Limbajul PL/SQL- Interactiunea cu Oracle

```
CREATE PROCEDURE creare AS
```

```
BEGIN
```

```
    CREATE TABLE dept1(cimp1 NUMBER(2),...) –  
    incorect
```

```
END;
```

```
CREATE PROCEDURE sterge (nume_tabela IN  
    VARCHAR2) AS
```

```
BEGIN
```

```
    DROP TABLE nume_tabela; -- incorect
```

```
END;
```

Limbajul PL/SQL- Interactiunea cu Oracle

- Cursori impliciti si expliciti.

Declararea:

```
CURSOR nume_cursor [(p1,p2,...)]
```

```
[RETURN tip_returnat] IS SELECT...;
```

tip_returnat specifica o inregistrare sau o linie intr-o tabela.

p1,p2,.. au forma:

```
nume_parametru [IN] tip [{:= | DEFAULT} expresie]
```

```
OPEN nume_cursor(v1,v2,...); -- notatie pozitionala,
```

```
OPEN nume_cursor(p1=>v1,...); --notatie cu nume,
```

```
OPEN nume_cursor(v1,p2=>v2,...); -- mixta
```

Comanda pe un cursor inchis produce exceptia:

```
INVALID_CURSOR.
```

Limbajul PL/SQL- Interactiunea cu Oracle

- Cursori in pachete.

```
CREATE PACKAGE pachet1 AS
```

```
    CURSOR c1 RETURN emp%ROWTYPE;
```

```
    /* Declaratie specificare cursor*/
```

```
....
```

```
END pachet1;
```

```
CREATE PACKAGE BODY pachet1 AS
```

```
    CURSOR c1 RETURN emp%ROWTYPE IS
```

```
        SELECT * FROM emp WHERE sal >4000;
```

```
    /*Definirea corpului cursorului */
```

```
...
```

```
END pachet1;
```

Limbajul PL/SQL- Interactiunea cu Oracle

- Folosirea cursorului in Ciclu FOR

```
DECLARE
```

```
    result NUMBER(5);
```

```
    CURSOR c1 IS
```

```
        SELECT cimp1,cimp2 FROM T1 WHERE marca=10;
```

```
BEGIN
```

```
    FOR c1_rec IN c1 LOOP
```

```
        result:=c1_rec.cimp1+c1_rec.cimp2;
```

```
        INSERT INTO temp VALUES(result,NULL,NULL);
```

```
    END LOOP;
```

```
    COMMIT;
```

```
END;
```

Limbajul PL/SQL- Interactiunea cu Oracle

- Folosirea subinterogarii.

```
DECLARE
```

```
bonus REAL;
```

```
BEGIN
```

```
FOR emp_rec IN( SELECT empno, sal,comm FROM  
emp) LOOP
```

```
bonus:=(emp_rec.sal*0.1)+(emp_rec.comm*0.05);
```

```
INSERT INTO bonuses VALUES  
(emp_rec.empno,bonus);
```

```
END LOOP;
```

```
COMMIT;
```

```
END;
```

Limbajul PL/SQL- Interactiunea cu Oracle

- Cursor cu parametri.

```
CURSOR nume_c (p1,p2,...) IS SELECT...
```

```
FOR v1 IN nume_c(v1,v2,...) LOOP
```

```
....
```

```
END LOOP;
```

- Variabile cursor.

- Are tipul REF CURSOR

- Definirea: TYPE ref_type_name IS REF CURSOR
RETURN return_type;

Ex. TYPE tipc1 IS REF CURSOR RETURN
emp%ROWTYPE;

TYPE tipc2 IS REF CURSOR;

- Declararea: var_cursor tip_cursor;

Limbajul PL/SQL- Interactiunea cu Oracle

- Variabile cursor – parametri formali.

```
TYPE tip_c IS REF CURSOR RETURN  
emp%ROWTYPE;
```

```
PROCEDURE proc1 (par1 IN OUT tip_c) IS ...
```

- Controlul variabilelor cursor: OPEN-FOR, FETCH, CLOSE.

```
- OPEN nume_var_cursor |:nume_var_cursor_host}  
FOR SELECT...;
```

```
attribute: %FOUND, %NOTFOUND, %ISOPEN,  
%ROWCOUNT.
```

```
IF NOT nume_var_cursor%ISOPEN THEN
```

```
OPEN nume_var_cursor FOR SELECT * FROM emp;
```

```
END IF;
```

Limbajul PL/SQL- Interactiunea cu Oracle

- OPEN-FOR poate deschide aceeasi variabila cursor pentru diferite interogari.

```
Ex. CREATE PACKAGE p1 AS
```

```
...
```

```
TYPE tipemp IS REF CURSOR RETURN emp%ROWTYPE;
```

```
PROCEDURE open_emp (emp_c IN OUT tipemp);
```

```
END p1;
```

```
CREATE PACKAGE BODY p1 AS
```

```
...
```

```
PROCEDURE open_emp (emp_c IN OUT tipemp) IS
```

```
  BEGIN
```

```
    OPEN emp_c FOR SELECT * FROM emp;
```

```
  END open_mp;
```

```
END p1;
```

Limbajul PL/SQL- Interactiunea cu Oracle

Ex. CREATE PACKAGE p2 AS

TYPE generic1 IS REF CURSOR;

TYPE emptip IS REF CURSOR RETURN

emp%ROWTYPE;

TYPE depttip IS REF CURSOR RETURN

dept%ROWTYPE;

END p2;

CREATE PROCEDURE open_emp(emp_cv IN OUT
p2.emptip) AS

BEGIN

OPEN emp_cv FOR SELECT * FROM emp WHERE

...;

END open mp;

Limbajul PL/SQL- Interactiunea cu Oracle

```
CREATE PACKAGE p3 AS
```

```
    TYPE generic2 IS REF CURSOR;
```

```
    TYPE emptip IS REF CURSOR RETURN emp%ROWTYPE;
```

```
    PROCEDURE open_emp(emp_c IN OUT emptip, I IN NUMBER);
```

```
END p3;
```

```
CREATE PACKAGE BODY p3 AS
```

```
    PROCEDURE open_emp(emp_c IN OUT emptip, I IN NUMBER) IS
```

```
        BEGIN
```

```
            IF I=1 THEN
```

```
                OPEN emp_c FOR SELECT * FROM emp WHERE comm =0;
```

```
            IF I=2 THEN
```

```
                OPEN emp_c FOR SELECT * FROM emp WHERE sal>1000;
```

```
            END IF;
```

```
        END open_emp;
```

```
END p3;
```

Limbajul PL/SQL- Interactiunea cu Oracle

```
CREATE PACKAGE BODY p3 AS
```

```
  PROCEDURE open_emp(generic_cv IN OUT generic2, I IN  
  NUMBER) IS
```

```
  BEGIN
```

```
    IF I=1 THEN
```

```
      OPEN generic_cv FOR SELECT * FROM emp;
```

```
    ELSIF I=2 THEN
```

```
      OPEN generic_cv FOR SELECT * FROM dept;
```

```
    ELSIF
```

```
      OPEN generic_cv FOR SELECT * FROM salgrade;
```

```
    END IF;
```

```
  END open_emp;
```

```
END p3;
```