



Grupa 1052 - Test rezolvat SGBD an II prof. Andronie

SGBDOracle Oracle DBMS (Academia de Studii Economice din București)



Scan to open on Studocu

grupa 1052C

EXERCITII S7

1. Scrieți și rulați o procedură care să crească cu 10% salariul tuturor angajatilor care nu primesc comision (comision = null) și care au peste 3 ani vechime.

```
SET SERVEROUTPUT ON
CREATE OR REPLACE PROCEDURE creste_salariu AS
BEGIN
    FOR rec IN (SELECT * FROM angajati WHERE comision IS NULL) LOOP
        DECLARE
            vechime INT;
        BEGIN
            vechime := TRUNC(MONTHS_BETWEEN(SYSDATE, rec.data_angajare) /
12);

            IF vechime > 3 THEN
                UPDATE angajati
                SET salariul = rec.salariul * 1.1
                WHERE id_angajat = rec.id_angajat;
            END IF;
        EXCEPTION
            WHEN OTHERS THEN
                DBMS_OUTPUT.PUT_LINE('A aparut o eroare pentru angajatul cu ID: ' ||
rec.id_angajat || '. Eroare: ' || SQLERRM);
        END;
    END LOOP;

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Salariile angajatilor au fost crescute cu 10%.');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('A aparut o eroare generala: ' || SQLERRM);
END;
/

BEGIN
    creste_salariu;
END;
/
```

```
SET SERVEROUTPUT ON
CREATE OR REPLACE PROCEDURE creste_salariu AS
BEGIN
FOR rec IN (SELECT * FROM angajati WHERE comision IS NULL) LOOP
DECLARE
vechime INT;
BEGIN
vechime := TRUNC(MONTHS_BETWEEN(SYSDATE, rec.data_angajare) / 12);
IF vechime > 3 THEN
UPDATE angajati
SET salariul = rec.salariul * 1.1
WHERE id_angajat = rec.id_angajat;
END IF;
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('A apărut o eroare pentru angajatul cu ID: ' || rec.id_angajat || '. Eroarea: ' || SQLERRM);
END;
END LOOP;
COMMIT;
END;
/

```

Script Output | Task completed in 0.815 seconds

Procedure CRESTE_SALARIU compiled

Salariile angajatorilor au fost crescute cu 10%.

PL/SQL procedure successfully completed.

Compiler - Log

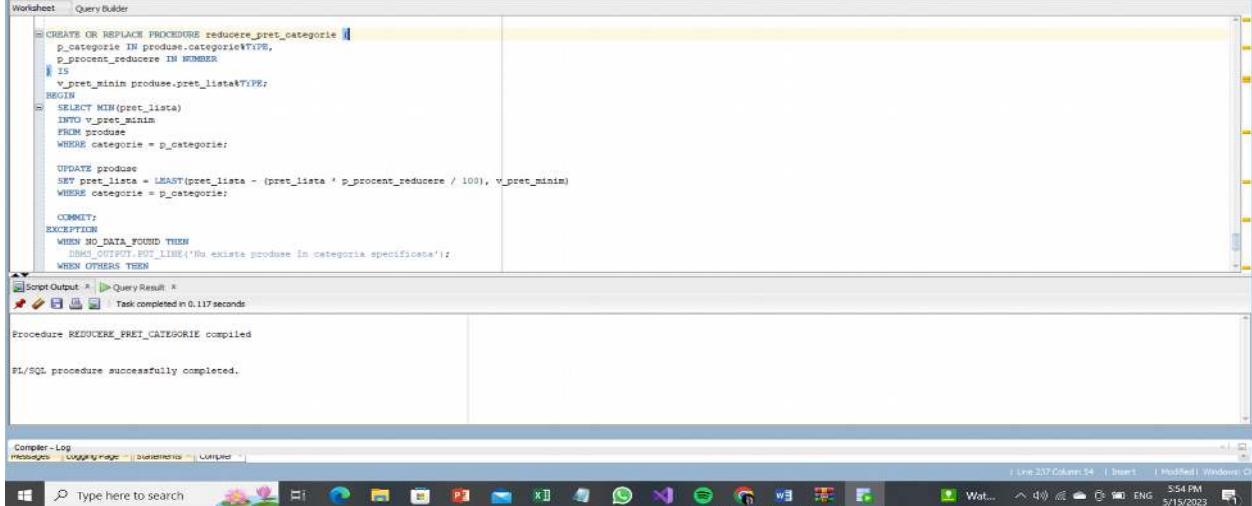
Type here to search

2. Scrieți și rulați o procedura cu doi parametri de tip IN care să reducă cu un anumit procent (primit ca parametru) pretul tuturor produselor dintr-o categorie (primită ca parametru). Pretul nu va fi redus mai mult decât pretul minim.

```
CREATE OR REPLACE PROCEDURE reducere_pret_categorie (
    p_categorie IN produse.categorie%TYPE,
    p_procent_reducere IN NUMBER
) IS
    v_pret_minim produse.pret_lista%TYPE;
BEGIN
    SELECT MIN(pret_lista)
    INTO v_pret_minim
    FROM produse
    WHERE categorie = p_categorie;

    UPDATE produse
    SET pret_lista = LEAST(pret_lista - (pret_lista * p_procent_reducere / 100),
    v_pret_minim)
    WHERE categorie = p_categorie;

    COMMIT;
    EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu există produse în categoria specificată');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('A apărut o eroare: ' || SQLERRM);
        ROLLBACK;
    END;
    /
BEGIN
    reducere_pret_categorie('hardware4', 10);
END;
```



```
CREATE OR REPLACE PROCEDURE reducere_pret_categorie IS
    p_categorie IN produc.catégorie%TYPE,
    p_procent_reducere IN NUMBER;
    v_pret_minim produc.pret_liste%TYPE;
BEGIN
    SELECT MIN(pret_liste)
    INTO v_pret_minim
    FROM produc
    WHERE categorie = p_categorie;

    UPDATE produc
    SET pret_liste = LEAST(pret_liste - (pret_liste * p_procent_reducere / 100), v_pret_minim)
    WHERE categorie = p_categorie;

    COMMIT;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Nu există produse în categoria specificată');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Eroare la execuție');
END;
```

Procedure REDUCERE_PRET_CATEGORIE compiled
PL/SQL procedure successfully completed.

3. Realizati o procedura *afiseaza_angajati* în care sa declarati un cursor pentru a selecta numele, functia si data angajarii salariatilor din tabela Angajati. Parcurgeti fiecare rând al cursorului si, în cazul în care data angajarii depaseste 01-AUG-1999, afisati informatiile preluate. Apelati procedura.

```
CREATE OR REPLACE PROCEDURE afiseaza_angajati AS
CURSOR c_angajati IS
    SELECT nume, id_functie, data_angajare
    FROM angajati;

    v_nume angajati.nume%TYPE;
    v_functie angajati.id_functie%TYPE;
    v_data_angajare angajati.data_angajare%TYPE;
BEGIN
    OPEN c_angajati;

    LOOP
        FETCH c_angajati INTO v_nume, v_functie, v_data_angajare;
        EXIT WHEN c_angajati%NOTFOUND;

        IF v_data_angajare > TO_DATE('01-AUG-1999', 'DD-MON-YYYY') THEN
            DBMS_OUTPUT.PUT_LINE('Nume: ' || v_nume || ', Functie: ' || v_functie || ', Data
angajarii: ' || v_data_angajare);
        END IF;
    END LOOP;

    CLOSE c_angajati;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('A aparut o eroare: ' || SQLERRM);
END;
```

/

```
BEGIN
    afiseaza_angajati;
END;
/
```

```
CREATE OR REPLACE PROCEDURE afiseaza_angajati AS
CURSOR c_angajati IS
    SELECT nume, id_functie, data_angajare
    FROM angajati;

    v_nume angajati.nume%TYPE;
    v_functie angajati.id_functie%TYPE;
    v_data_angajare angajati.data_angajare%TYPE;
BEGIN
    OPEN c_angajati;

    LOOP
        FETCH c_angajati INTO v_nume, v_functie, v_data_angajare;
        EXIT WHEN c_angajati%NOTFOUND;

        IF v_data_angajare > TO_DATE('01-AUG-1999', 'DD-MON-YYYY') THEN
            DBMS_OUTPUT.PUT_LINE('Nume: ' || v_nume || ', Functie: ' || v_functie || ', Data angajarii: ' || v_data_angajare);
        END IF;
    END LOOP;

    CLOSE c_angajati;
EXCEPTION
```

Script Output: Task completed in 0.273 seconds

Procedure AFISEAZA_ANGAJATI compiled

Procedure AFISEAZA_ANGAJATI compiled

PL/SQL procedure successfully completed.

4. Realizati o functie vechime_angajat (p_cod angajati.id_angajat%type) care sa returneze vechimea angajatului (calculata drept diferența între data actuala si cea a angajarii) care are codul primit ca parametru. Tratati exceptiile aparute. Apelati functia dintr-un bloc PL/SQL si utilizati un cursor pentru a parcurge toti angajatii.

```
CREATE OR REPLACE FUNCTION vechime_angajat (p_cod angajati.id_angajat%type)
RETURN NUMBER IS
    v_vechime NUMBER;
BEGIN
    SELECT EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
data_angajare)
    INTO v_vechime
    FROM angajati
    WHERE id_angajat = p_cod;

    RETURN v_vechime;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN NULL;
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('A aparut o eroare: ' || SQLERRM);
        RETURN NULL;
END;
/

DECLARE
    v_cod_angajat angajati.id_angajat%type;
    v_vechime NUMBER;
```

```
CURSOR c_angajati IS
  SELECT id_angajat
    FROM angajati;
BEGIN
  OPEN c_angajati;
  LOOP
    FETCH c_angajati INTO v_cod_angajat;
    EXIT WHEN c_angajati%NOTFOUND;

    v_vechime := vechime_angajat(v_cod_angajat);
    DBMS_OUTPUT.PUT_LINE('Vechimea angajatului cu codul ' || v_cod_angajat || ' este:
' || v_vechime || ' ani');
  END LOOP;

  CLOSE c_angajati;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('A aparut o eroare: ' || SQLERRM);
    CLOSE c_angajati;
END;
/
```



```
CREATE OR REPLACE FUNCTION vechime_angajat (p_cod_angajat angajati.id_angajat%type) RETURN NUMBER IS
  v_vechime NUMBER;
BEGIN
  SELECT EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM data_angajare)
  INTO v_vechime
  FROM angajati
  WHERE id_angajat = p_cod;

  RETURN v_vechime;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RETURN NULL;
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('A aparut o eroare: ' || SQLERRM);
    RETURN NULL;
END;
/

DECLARE
  v_cod_angajat angajati.id_angajat%type;
```

5. Realizati o procedura vechime_angajat_proc (p_cod IN angajati.id_angajat %type, p_vechime OUT number) care sa calculeze vechimea angajatului care are codul primit ca parametru. Tratati exceptiile aparute. Apelati procedura dintr-un bloc PL/SQL si utilizati un cursor pentru a parurge toti angajatii.

```
CREATE OR REPLACE PROCEDURE vechime_angajat_proc (p_cod IN
angajati.id_angajat%type, p_vechime OUT NUMBER) IS
BEGIN
  SELECT EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM
data_angajare)
  INTO p_vechime
  FROM angajati
  WHERE id_angajat = p_cod;
```

```
EXCEPTION
WHEN NO_DATA_FOUND THEN
    p_vechime := NULL;
WHEN OTHERS THEN
    p_vechime := NULL;
END;
/

DECLARE
    v_cod_angajat angajati.id_angajat%type;
    v_vechime NUMBER;

CURSOR c_angajati IS
    SELECT id_angajat
    FROM angajati;
BEGIN
    OPEN c_angajati;
    LOOP
        FETCH c_angajati INTO v_cod_angajat;
        EXIT WHEN c_angajati%NOTFOUND;

        vechime_angajat_proc(v_cod_angajat, v_vechime);
        DBMS_OUTPUT.PUT_LINE('Vechimea angajatului cu codul ' || v_cod_angajat || ' este:
' || v_vechime || ' ani');
    END LOOP;

    CLOSE c_angajati;
EXCEPTION
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('A aparut o eroare: ' || SQLERRM);
    CLOSE c_angajati;
END;
/
```

The screenshot shows the Oracle SQL Developer interface with the following details:

- Query Editor:** Displays the PL/SQL code for the `VECHIME_ANGAJAT_PROC` procedure.
- Script Output:** Shows the successful compilation message:

```
PL/SQL procedure successfully completed.
```
- Messages - Log:** Displays the log messages for the procedure compilation.
- System Bar:** Shows the Windows taskbar with various application icons and system status.