

**GESTIUNEA UNUI MAGAZIN DE INCALTAMINTE SPORT**



Student: Zipisi Vlad-Andrei

Grupa:1096

Profesor coordonator: Conf.univ.dr.Iuliana Botha

Cuprins

[1.OBIECTIVUL PROIECTULUI SI PREZENTAREA TABELELOR 3](#_Toc90930094)

[2.SCHEMA BAZEI DE DATE 4](#_Toc90930095)

[3.CREAREA TABELELOR SI INSERAREA DATELOR 5](#_Toc90930096)

[3.1.Crearea tabelei CUTIE si adaugarea datelor 5](#_Toc90930097)

[3.2.Crearea tabelei MODEL si adaugarea datelor 6](#_Toc90930098)

[3.3.Crearea tabelei DEPARTAMENT si adaugarea datelor 8](#_Toc90930099)

[3.4.Crearea tabelei ANGAJAT si adaugarea datelor 9](#_Toc90930100)

[3.5.Crearea tabelei BON si adaugarea datelor 10](#_Toc90930101)

[4. BLOCURI PL/SQL CONTINAND STRUCTURI DE CONTROL VARIATE 12](#_Toc90930102)

[5. UTILIZAREA CURSORILOR SI A EXCEPTIILOR IN CADRUL BLOCURILOR PL/SQL 16](#_Toc90930103)

[6. FUNCTII SI PROCEDURI 22](#_Toc90930104)

[6.1.Functii 22](#_Toc90930105)

[6.2.Proceduri 26](#_Toc90930106)

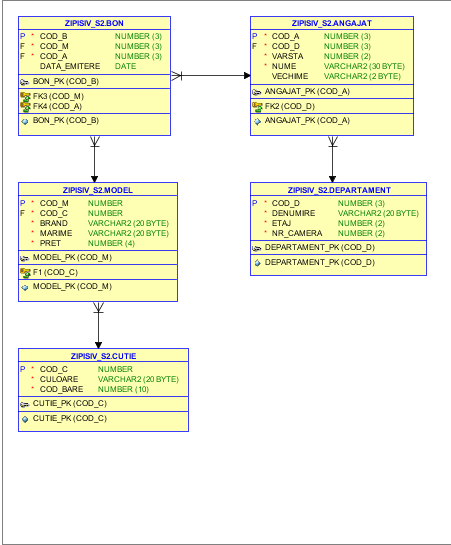
# 1.OBIECTIVUL PROIECTULUI SI PREZENTAREA TABELELOR

Scopul proiectului este acela de a gestiona un magazin de incaltaminte sport, folosind informatii interne. Astfel, in baza noastra de date avem inregistrate informatii despre: departamente, angajati, cutiile fiecarei perechi de incaltaminte, modelul fiecarei perechi de incaltaminte si bonurile fiscale, in total existand 5 tabele de date.

Pentru fiecare departament se cunosc: cod\_d (codul cu care departamentul se identifica), denumirea departamentului, etajul si numarul camerei la care acest departament se poate gasi. Pentru fiecare angajat se cunosc: cod\_a (codul cu care angajatul se identifica), cod\_d (departamentul din care angajatul face parte), varsta angajatului, numele angajatului si vechimea acestuia in firma. Cum fiecare pereche de incaltaminte prezinta o cutie, pentru aceasta se cunosc: cod\_c (codul cu care cutia se identifica), culoarea cutiei si un cod de bare.

Pentru modelul fiecarei perechi se cunosc: cod\_m (codul cu care modelul respectiv se identifica), cod\_c (cutia in care modelul respectiv se gasete), brand-ul, marimea si pretul. Pentru bonul fiscal se cunosc: cod\_b (codul unic al bonului fiscal emis), cod\_m (codul cu care modelul vandut se identifica), cod\_a (codul angajatului care a vandut perechea, respectiv a emis bonul) si data emiterii bonului fiscal.

# 2.SCHEMA BAZEI DE DATE



# 3.CREAREA TABELELOR SI INSERAREA DATELOR

## 3.1.Crearea tabelei CUTIE si adaugarea datelor

CREATE TABLE "CUTIE"

( "COD\_C" NUMBER NOT NULL ENABLE,

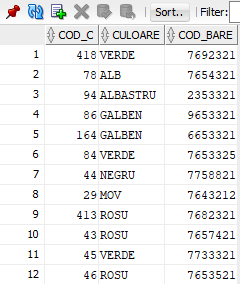
"CULOARE" VARCHAR2(20 BYTE) NOT NULL ENABLE,

"COD\_BARE" NUMBER(10,0) NOT NULL ENABLE,

CONSTRAINT "CUTIE\_PK" PRIMARY KEY ("COD\_C")

USING INDEX ENABLE

)



INSERT INTO CUTIE (COD\_C,CULOARE,COD\_BARE) VALUES (418,'VERDE',07692321);

INSERT INTO CUTIE (COD\_C,CULOARE,COD\_BARE) VALUES (78,'ALB',07654321);

INSERT INTO CUTIE (COD\_C,CULOARE,COD\_BARE) VALUES (94,'ALBASTRU',2353321);

INSERT INTO CUTIE (COD\_C,CULOARE,COD\_BARE) VALUES (86,'GALBEN',09653321);

INSERT INTO CUTIE (COD\_C,CULOARE,COD\_BARE) VALUES (164,'GALBEN',06653321);

INSERT INTO CUTIE (COD\_C,CULOARE,COD\_BARE) VALUES (84,'VERDE',07653325);

INSERT INTO CUTIE (COD\_C,CULOARE,COD\_BARE) VALUES (44,'NEGRU',07758821);

INSERT INTO CUTIE (COD\_C,CULOARE,COD\_BARE) VALUES (29,'MOV',07643212);

INSERT INTO CUTIE (COD\_C,CULOARE,COD\_BARE) VALUES (413,'ROSU',07682321);

INSERT INTO CUTIE (COD\_C,CULOARE,COD\_BARE) VALUES (43,'ROSU',07657421);

INSERT INTO CUTIE (COD\_C,CULOARE,COD\_BARE) VALUES (45,'VERDE',07733321);

INSERT INTO CUTIE (COD\_C,CULOARE,COD\_BARE) VALUES (46,'ROSU',07653521);

## 3.2.Crearea tabelei MODEL si adaugarea datelor

CREATE TABLE "MODEL"

( "COD\_M" NUMBER,

"COD\_C" NUMBER NOT NULL ENABLE,

"BRAND" VARCHAR2(20) NOT NULL ENABLE,

"MARIME" VARCHAR2(20) NOT NULL ENABLE,

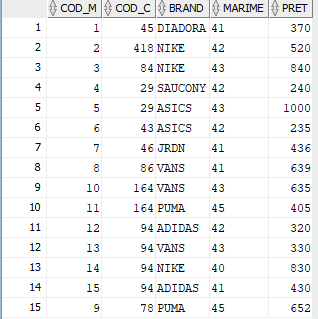
"PRET" NUMBER(4,0) NOT NULL ENABLE,

CONSTRAINT "MODEL\_PK" PRIMARY KEY ("COD\_M")

CONSTRAINT "F1" FOREIGN KEY ("COD\_C")

USING INDEX ENABLE

)



INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (1,45,'DIADORA',41,370);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (2,418,'NIKE',42,520);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (3,84,'NIKE',43,840);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (4,29,'SAUCONY',42,240);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (5,29,'ASICS',43,230);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (6,43,'ASICS',42,235);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (7,46,'JRDN',41,436);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (8,86,'VANS',41,639);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (9,78,'PUMA',45,652);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (10,164,'VANS',43,635);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (11,164,'PUMA',45,405);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (12,94,'ADIDAS',42,320);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (13,94,'VANS',413,330);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (14,94,'NIKE',84,830);

INSERT INTO MODEL (COD\_M,COD\_C,BRAND,MARIME,PRET) VALUES (15,94,'ADIDAS',29,430);

## 3.3.Crearea tabelei DEPARTAMENT si adaugarea datelor

CREATE TABLE "DEPARTAMENT"

( "COD\_D" NUMBER(3,0) NOT NULL ENABLE,

"DENUMIRE" VARCHAR2(20) NOT NULL ENABLE,

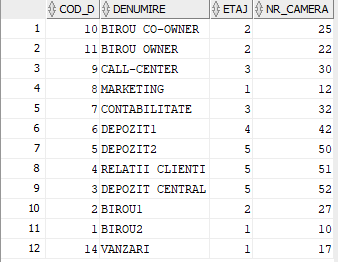
"ETAJ" NUMBER(2,0) NOT NULL ENABLE,

"NR\_CAMERA" NUMBER(2,0) NOT NULL ENABLE,

CONSTRAINT "DEPARTAMENT\_PK" PRIMARY KEY ("COD\_D")

USING INDEX ENABLE

)



INSERT INTO DEPARTAMENT (COD\_D,DENUMIRE,ETAJ,NR\_CAMERA) VALUES (010,'BIROU CO-OWNER',2,21);

INSERT INTO DEPARTAMENT (COD\_D,DENUMIRE,ETAJ,NR\_CAMERA) VALUES (011,'BIROU OWNER',2,22);

INSERT INTO DEPARTAMENT (COD\_D,DENUMIRE,ETAJ,NR\_CAMERA) VALUES (009,'CALL-CENTER',3,30);

INSERT INTO DEPARTAMENT (COD\_D,DENUMIRE,ETAJ,NR\_CAMERA) VALUES (008,'MARKETING',1,12);

INSERT INTO DEPARTAMENT (COD\_D,DENUMIRE,ETAJ,NR\_CAMERA) VALUES (007,'CONTABILITATE',3,32);

INSERT INTO DEPARTAMENT (COD\_D,DENUMIRE,ETAJ,NR\_CAMERA) VALUES (006,'DEPOZIT1',4,42);

INSERT INTO DEPARTAMENT (COD\_D,DENUMIRE,ETAJ,NR\_CAMERA) VALUES (005,'DEPOZIT2',5,50);

INSERT INTO DEPARTAMENT (COD\_D,DENUMIRE,ETAJ,NR\_CAMERA) VALUES (004,'RELATII CLIENTI',5,51);

INSERT INTO DEPARTAMENT (COD\_D,DENUMIRE,ETAJ,NR\_CAMERA) VALUES (003,'DEPOZIT CENTRAL',5,52);

INSERT INTO DEPARTAMENT (COD\_D,DENUMIRE,ETAJ,NR\_CAMERA) VALUES (002,'BIROU1',2,27);

INSERT INTO DEPARTAMENT (COD\_D,DENUMIRE,ETAJ,NR\_CAMERA) VALUES (001,'BIROU2',1,10);

INSERT INTO DEPARTAMENT (COD\_D,DENUMIRE,ETAJ,NR\_CAMERA) VALUES (014,'VANZARI',1,17);

## 3.4.Crearea tabelei ANGAJAT si adaugarea datelor

CREATE TABLE "ANGAJAT"

( "COD\_A" NUMBER(3,0) NOT NULL ENABLE,

"COD\_D" NUMBER(3,0) NOT NULL ENABLE,

"VARSTA" NUMBER(2,0) NOT NULL ENABLE,

"NUME" VARCHAR2(30) NOT NULL ENABLE,

"VECHIME" VARCHAR2(2),

CONSTRAINT "ANGAJAT\_PK" PRIMARY KEY ("COD\_A")

CONSTRAINT "FK2" FOREIGN KEY ("COD\_D")

USING INDEX ENABLE

)



INSERT INTO ANGAJAT (COD\_A,COD\_D,VARSTA,NUME,VECHIME) VALUES (020,010,21,'CIPRIAN ENACHE',2);

INSERT INTO ANGAJAT (COD\_A,COD\_D,VARSTA,NUME,VECHIME) VALUES (021,002,21,'ANDREEA TITU',2);

INSERT INTO ANGAJAT (COD\_A,COD\_D,VARSTA,NUME,VECHIME) VALUES (022,002,21,'MARIUS IONESCU',3);

INSERT INTO ANGAJAT (COD\_A,COD\_D,VARSTA,NUME,VECHIME) VALUES (023,003,21,'ANDREI POPESCU',4);

INSERT INTO ANGAJAT (COD\_A,COD\_D,VARSTA,NUME,VECHIME) VALUES (024,004,21,'ALEX IONESCU',1);

INSERT INTO ANGAJAT (COD\_A,COD\_D,VARSTA,NUME,VECHIME) VALUES (025,005,21,'TUDOR PETRESCU',4);

INSERT INTO ANGAJAT (COD\_A,COD\_D,VARSTA,NUME,VECHIME) VALUES (026,011,21,'COSTIN PLESA',2);

INSERT INTO ANGAJAT (COD\_A,COD\_D,VARSTA,NUME,VECHIME) VALUES (027,005,21,'ANDREEA PAVEL',1);

INSERT INTO ANGAJAT (COD\_A,COD\_D,VARSTA,NUME,VECHIME) VALUES (028,001,21,'MIRUNA AXINTE',1);

INSERT INTO ANGAJAT (COD\_A,COD\_D,VARSTA,NUME,VECHIME) VALUES (029,009,21,'CRISTINA BARBU',2);

INSERT INTO ANGAJAT (COD\_A,COD\_D,VARSTA,NUME,VECHIME) VALUES (030,014,21,'NICOLETA BRAGA',1);

INSERT INTO ANGAJAT (COD\_A,COD\_D,VARSTA,NUME,VECHIME) VALUES (031,014,21,'ALEXANDRU IOAN',1);

## 3.5.Crearea tabelei BON si adaugarea datelor

CREATE TABLE "BON"

( "COD\_B" NUMBER(3,0) NOT NULL ENABLE,

"COD\_M" NUMBER(3,0) NOT NULL ENABLE,

"COD\_A" NUMBER(3,0) NOT NULL ENABLE,

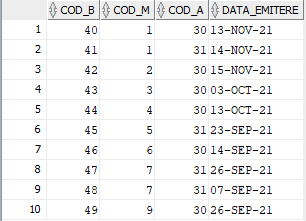
"DATA\_EMITERE" DATE,

CONSTRAINT "BON\_PK" PRIMARY KEY ("COD\_B")

CONSTRAINT "FK3" FOREIGN KEY ("COD\_M")

CONSTRAINT "FK4" FOREIGN KEY ("COD\_A")

USING INDEX ENABLE)



INSERT INTO BON (COD\_B,COD\_M,COD\_A,DATA\_EMITERE) VALUES (040,001,032,'13-NOV-21');

INSERT INTO BON (COD\_B,COD\_M,COD\_A,DATA\_EMITERE) VALUES (041,001,033,'14-NOV-21');

INSERT INTO BON (COD\_B,COD\_M,COD\_A,DATA\_EMITERE) VALUES (042,002,032,'15-NOV-21');

INSERT INTO BON (COD\_B,COD\_M,COD\_A,DATA\_EMITERE) VALUES (043,003,032,'03-OCT-21');

INSERT INTO BON (COD\_B,COD\_M,COD\_A,DATA\_EMITERE) VALUES (044,004,033,'13-OCT-21');

INSERT INTO BON (COD\_B,COD\_M,COD\_A,DATA\_EMITERE) VALUES (045,005,033,'23-SEP-21');

INSERT INTO BON (COD\_B,COD\_M,COD\_A,DATA\_EMITERE) VALUES (046,006,033,'14-SEP-21');

INSERT INTO BON (COD\_B,COD\_M,COD\_A,DATA\_EMITERE) VALUES (047,007,032,'26-SEP-21');

INSERT INTO BON (COD\_B,COD\_M,COD\_A,DATA\_EMITERE) VALUES (048,007,032,'07-SEP-21');

INSERT INTO BON (COD\_B,COD\_M,COD\_A,DATA\_EMITERE) VALUES (049,009,033,'21-SEP-21');

# 4. BLOCURI PL/SQL CONTINAND STRUCTURI DE CONTROL VARIATE

1.Sa se prezinte angajatii ce au id-ul (cod\_a) cuprins intre 20 si 25 :

SET SERVEROUTPUT ON

DECLARE

v\_nume VARCHAR2(20);

v\_varsta NUMBER NOT NULL:=0;

BEGIN

FOR i IN 20..25 LOOP

SELECT nume, varsta

INTO v\_nume, v\_varsta

FROM angajat

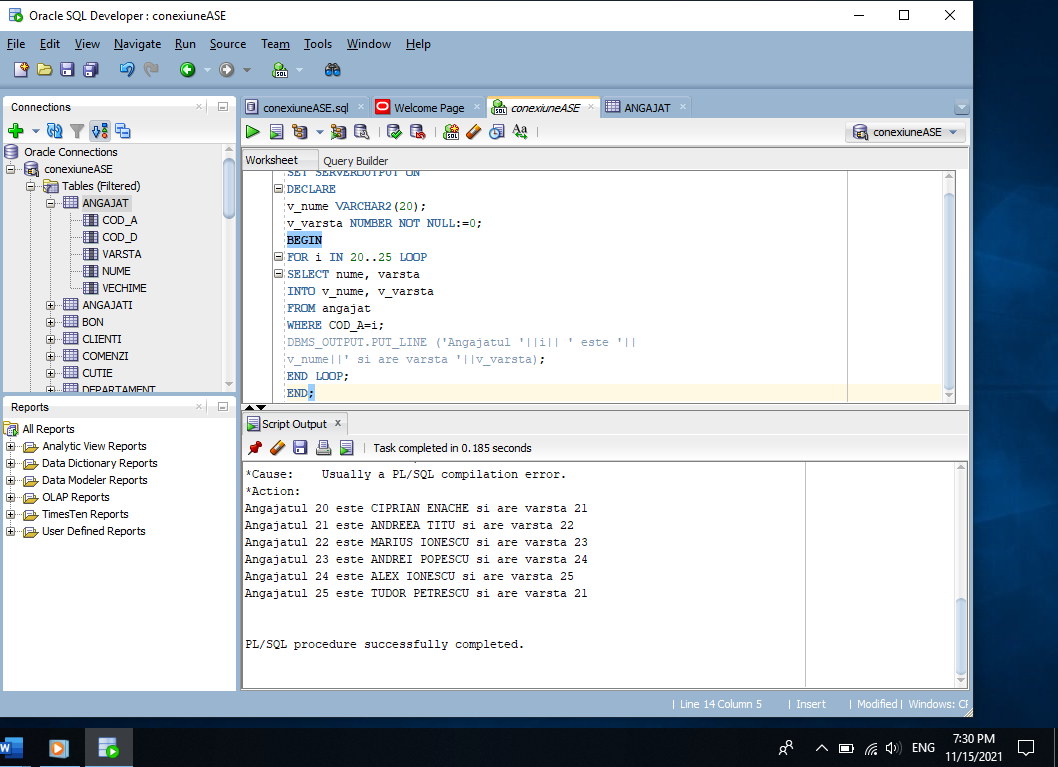
WHERE COD\_A=i;

DBMS\_OUTPUT.PUT\_LINE ('Angajatul '||i|| ' este '||

v\_nume||' si are varsta '||v\_varsta);

END LOOP;

END;



2.Sa se afiseze numele, implicit si vechimea angajatilor ce au vechimea mai mare decat vechimea medie si id-ul in intervalul 20-26 :

Sa se afiseze pizza\_id

DECLARE

v\_vechime\_med NUMBER;

v\_nume ANGAJAT.NUME%TYPE;

v\_vechime NUMBER;

v\_cod\_a NUMBER := 20;

BEGIN

SELECT AVG(vechime) INTO v\_vechime\_med FROM ANGAJAT;

DBMS\_OUTPUT.PUT\_LINE('Vechimea medie este '||v\_vechime\_med);

LOOP

SELECT NUME, VECHIME INTO v\_nume, v\_vechime FROM ANGAJAT WHERE cod\_a = v\_cod\_a;

IF v\_vechime > v\_vechime\_med THEN

DBMS\_OUTPUT.PUT\_LINE('Angajatul '||v\_nume||' are vechimea '||v\_vechime);

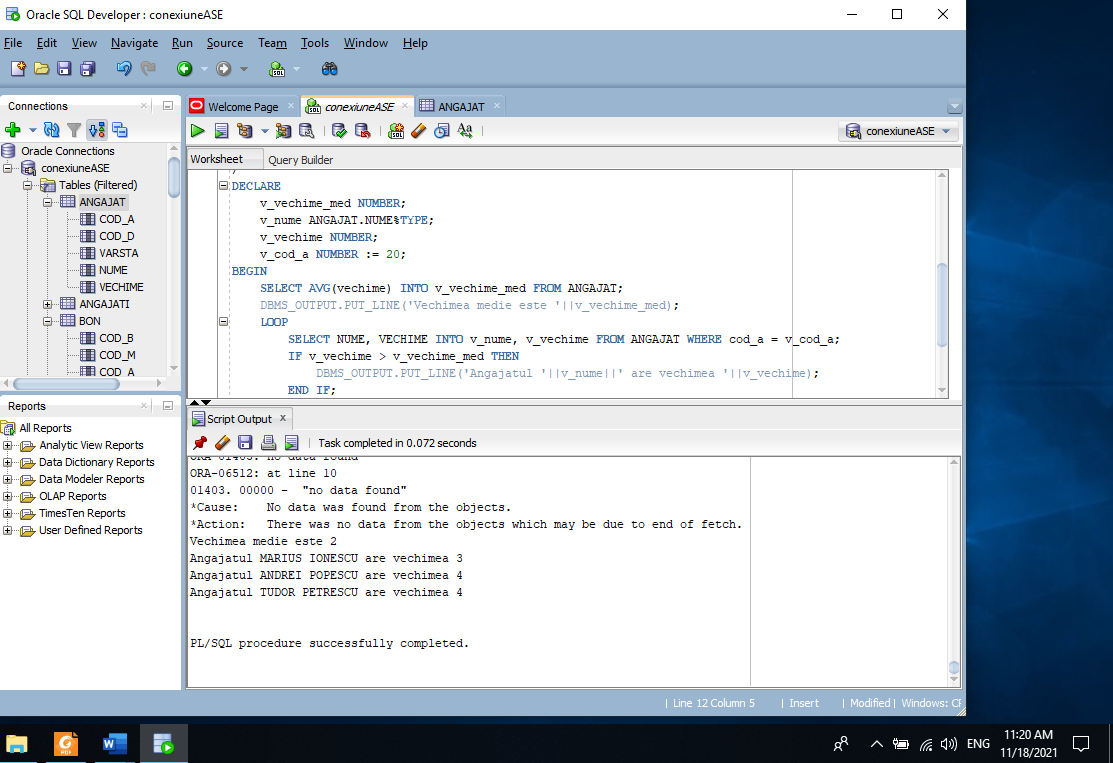
END IF;

v\_cod\_a := v\_cod\_a + 1;

EXIT WHEN v\_cod\_a >26 ;

END LOOP;

END;



3.Folosind structura loop, sa se afiseze departamentele cu codul in intervalul 1-6 si etajul la care acestea se gasesc :

SET SERVEROUTPUT ON

DECLARE

v\_denumire DEPARTAMENT.DENUMIRE%type;

v\_etaj DEPARTAMENT.ETAJ%type;

i number(4):=1;

BEGIN

loop

select DENUMIRE, ETAJ into v\_denumire, v\_etaj from DEPARTAMENT where cod\_d=i;

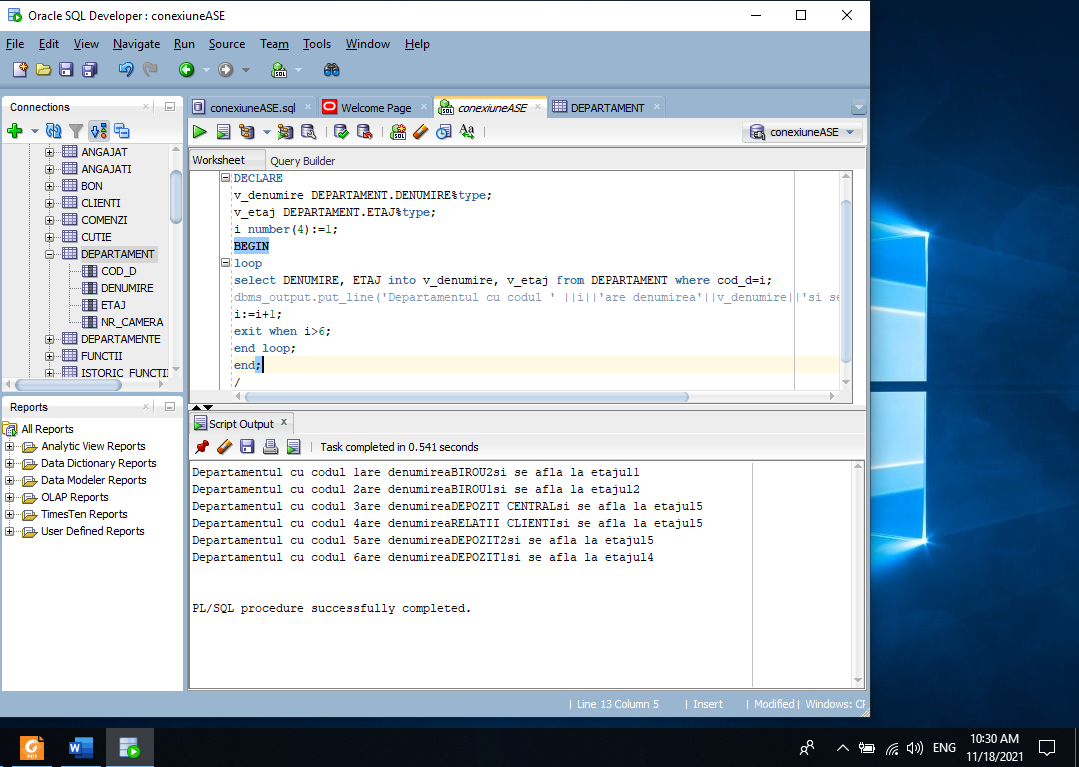
dbms\_output.put\_line('Departamentul cu codul ' ||i||'are denumirea'||v\_denumire||'si se afla la etajul'||v\_etaj);

i:=i+1;

exit when i>6;

end loop;

end;



4.Folosind structura WHILE sa se afiseze pretul fiecarui model de incaltaminte ce are codul in intervalul 4-12 si taxa pe valoare adaugata aferenta.(19%) :

DECLARE

v\_pret MODEL.PRET%type;

i number(4):=4;

BEGIN

WHILE i<=12 loop

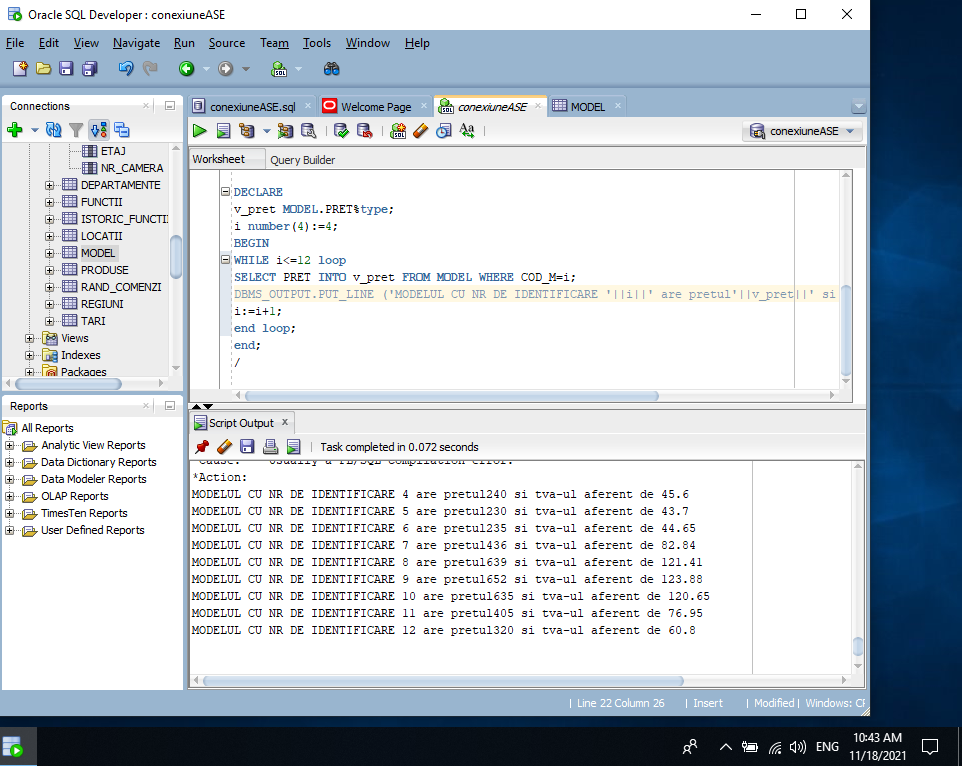
SELECT PRET INTO v\_pret FROM MODEL WHERE COD\_M=i;

DBMS\_OUTPUT.PUT\_LINE ('MODELUL CU NR DE IDENTIFICARE '||i||' are pretul'||v\_pret||' si tva-ul aferent de '||v\_pret\*0.19);

i:=i+1;

end loop;

end;



# 5. UTILIZAREA CURSORILOR SI A EXCEPTIILOR IN CADRUL BLOCURILOR PL/SQL

5.Sa se afiseze culoarea cutiei si codul de bare pt cutia cu numarul 14. Daca aceasta cutie cu numarul specificat nu exista, sa se trateze exceptia :

set serveroutput on

declare

v\_cod\_b NUMBER(9);

v\_culoare VARCHAR2(30);

begin

select COD\_BARE, CULOARE into v\_cod\_b, v\_culoare

from CUTIE where cod\_c=14;

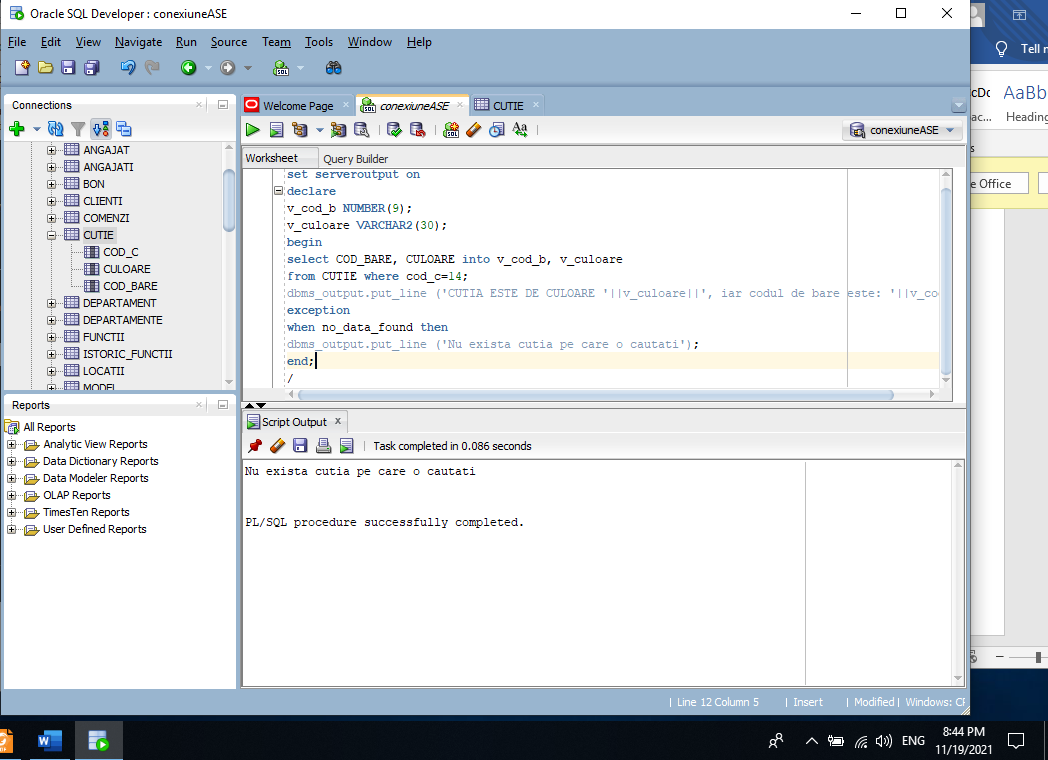
dbms\_output.put\_line ('CUTIA ESTE DE CULOARE '||v\_culoare||', iar codul de bare este: '||v\_cod\_b);

exception

when no\_data\_found then

dbms\_output.put\_line ('Nu exista cutia pe care o cautati');

end;



6.Sa se afiseze angajatul ce are vechimea de 2 ani. Sa se trateze exceptiile ce pot aparea (too many rows) :

DECLARE

v\_nume ANGAJAT.NUME%type;

BEGIN

SELECT NUME into v\_nume FROM ANGAJAT WHERE VECHIME=2;

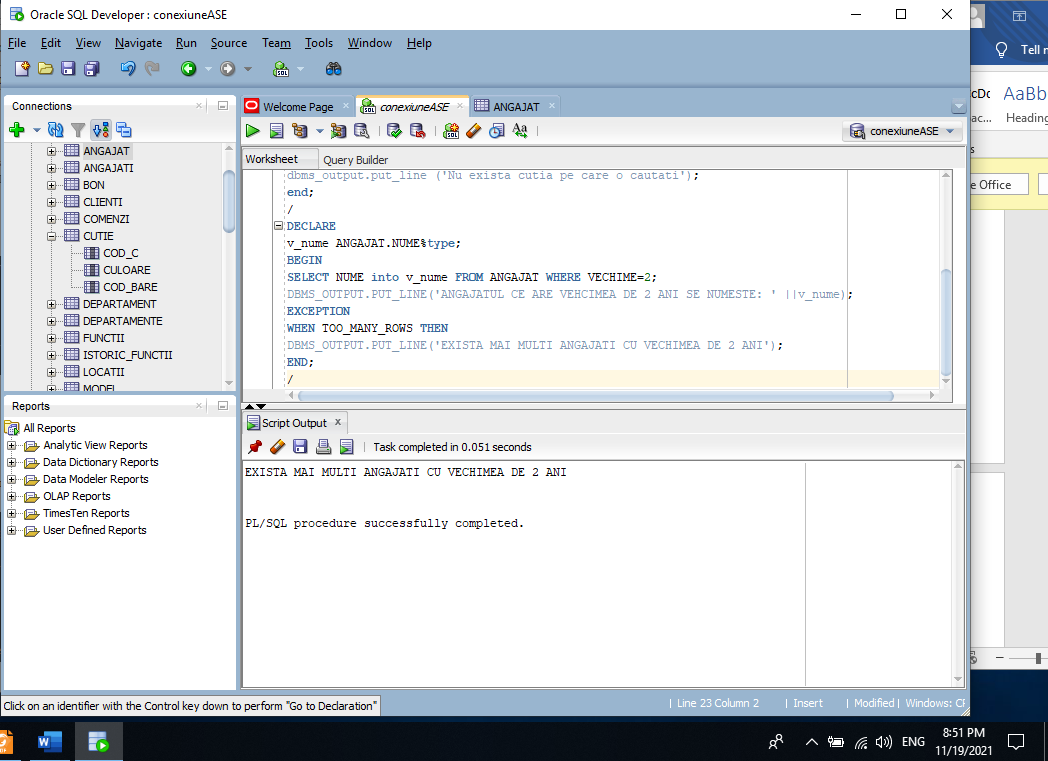
DBMS\_OUTPUT.PUT\_LINE('ANGAJATUL CE ARE VEHCIMEA DE 2 ANI SE NUMESTE: ' ||v\_nume);

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('EXISTA MAI MULTI ANGAJATI CU VECHIMEA DE 2 ANI');

END;



7.Sa se majorize cu 19% (tva-ul) pretul perechilor ce apartin brand-ului Nike :

DECLARE

excep2 EXCEPTION;

BEGIN

UPDATE MODEL SET PRET=PRET\*1.19 WHERE BRAND='NIKE';

IF SQL%NOTFOUND THEN

RAISE excep2;

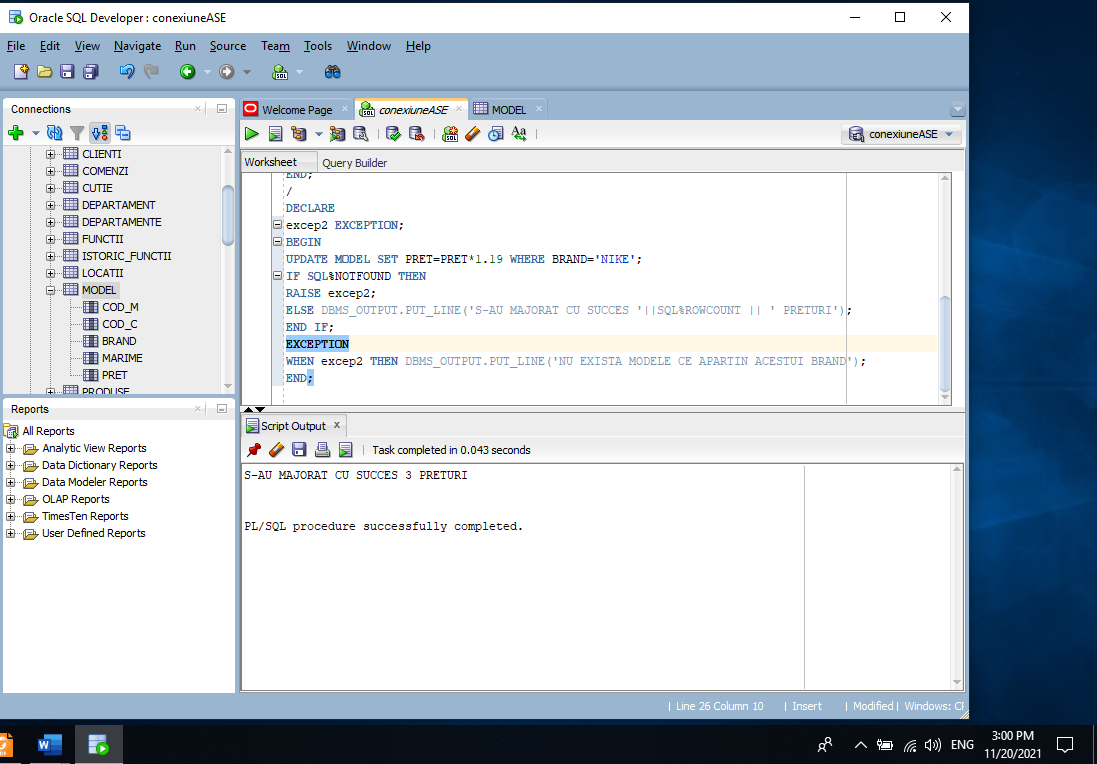
ELSE DBMS\_OUTPUT.PUT\_LINE('S-AU MAJORAT CU SUCCES '||SQL%ROWCOUNT || ' PRETURI');

END IF;

EXCEPTION

WHEN excep2 THEN DBMS\_OUTPUT.PUT\_LINE('NU EXISTA MODELE CE APARTIN ACESTUI BRAND');

END;



8.Sa se adauge la modelele cu pretul de sub 300 de lei, un adaos in valoare de 50 de lei folosind un cursor implicit. Sa se afiseze daca exista astfel de modele ce vor beneficia de acest adaos :

BEGIN

UPDATE MODEL

SET PRET = PRET + 50

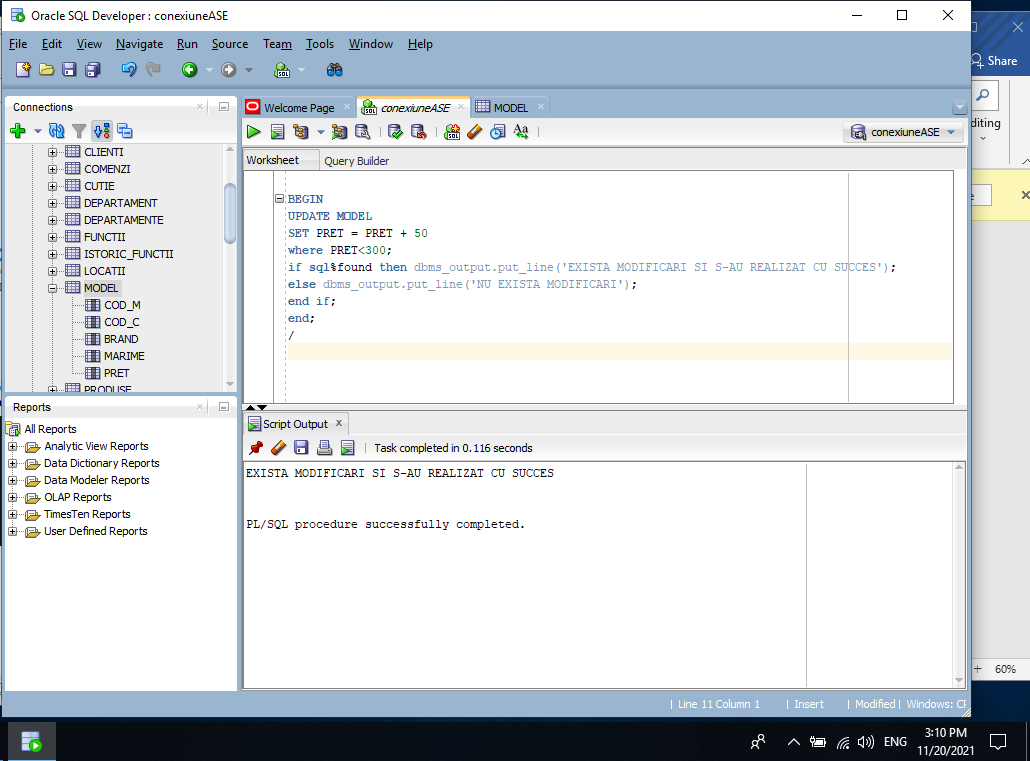
where PRET<300;

if sql%found then dbms\_output.put\_line('EXISTA MODIFICARI SI S-AU REALIZAT CU SUCCES');

else dbms\_output.put\_line('NU EXISTA MODIFICARI');

end if;

end;



9.Sa se afiseze codul bonurilor ce au fost emise inaintea datei de 15 septembrie 2021 :

DECLARE

CURSOR D IS

SELECT COD\_B

FROM BON

WHERE DATA\_EMITERE<TO\_DATE('15-SEP-2021','DD-MM-YYYY');

V\_COD BON.COD\_B%TYPE;

BEGIN

OPEN D;

LOOP

FETCH D INTO V\_COD;

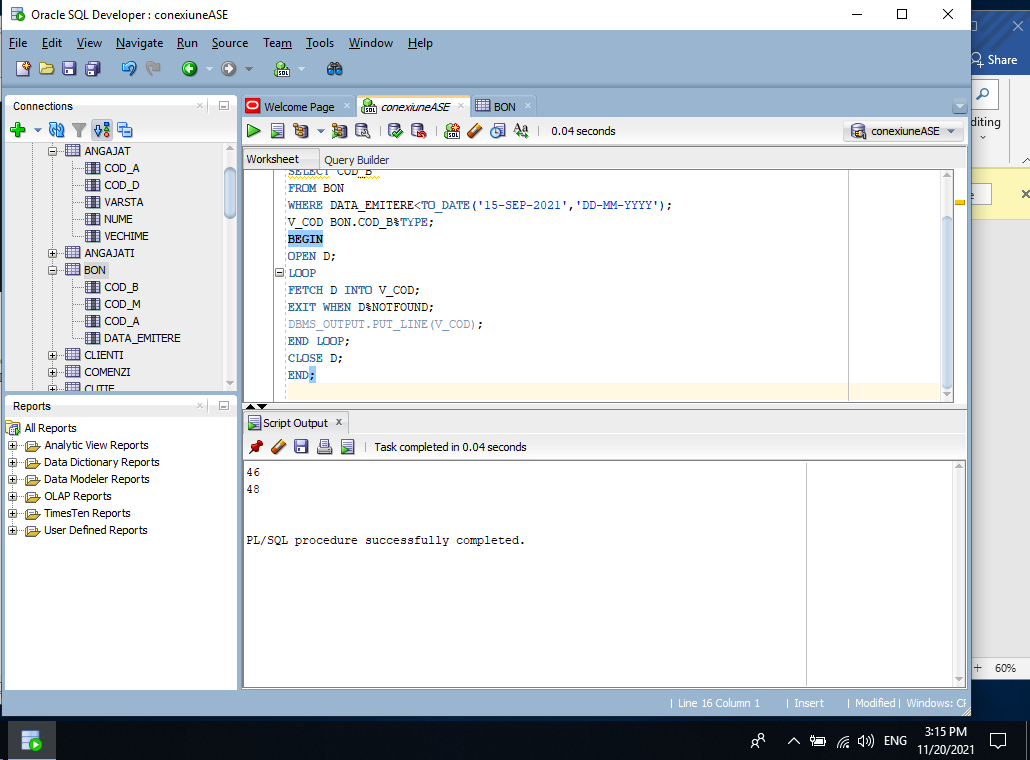
EXIT WHEN D%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(V\_COD);

END LOOP;

CLOSE D;

END;



10.Sa se afiseze toate departamentele ce se gasesc intre etajele 1 si 3 folosind un cursor explicit. Se va afisa codul departamentului, denumirea ,etajul si numarul camerei :

DECLARE

CURSOR C IS SELECT cod\_d ,denumire, etaj, nr\_camera

FROM departament

WHERE etaj BETWEEN 1 AND 3;

cod\_d DEPARTAMENT.COD\_D%TYPE;

denumire DEPARTAMENT.DENUMIRE%TYPE;

etaj DEPARTAMENT.ETAJ%TYPE;

nr\_camera DEPARTAMENT.NR\_CAMERA%TYPE;

BEGIN

OPEN C;

LOOP

FETCH C INTO COD\_D,DENUMIRE,ETAJ, NR\_CAMERA;

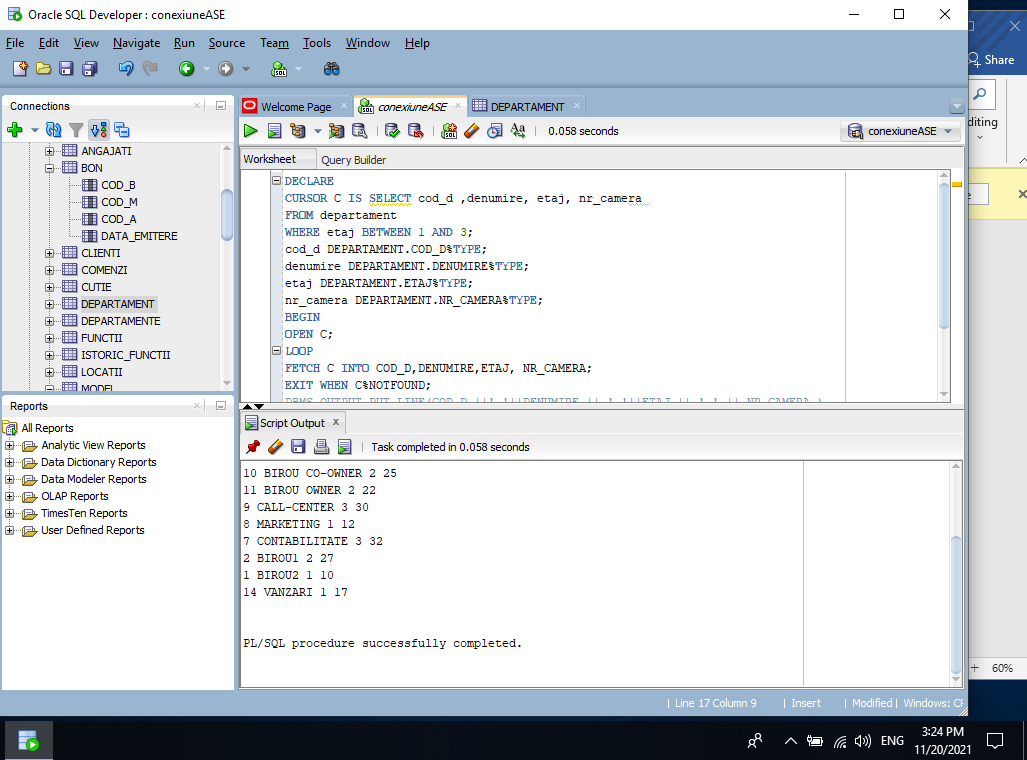
EXIT WHEN C%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE(COD\_D ||' '||DENUMIRE || ' '||ETAJ || ' ' || NR\_CAMERA );

END LOOP;

CLOSE C;

END;



# 6. FUNCTII SI PROCEDURI

## 6.1.Functii

11.Sa se creeze o functie care primeste ca parametru codul unui angajat si returneaza codul departamentului in care lucreaza :

create or replace function departament\_a

(id\_a number)

return angajat.cod\_d%type

is

cod angajat.cod\_d%type;

begin

select cod\_d into cod from angajat where cod\_a=id\_a;

return cod;

end;

/

accept id\_a prompt 'Introduceti id-ul angajatului:';

declare

cod\_d\_a angajat.cod\_d%type;

h\_id\_a number(4):=&id\_a;

begin

cod\_d\_a:=departament\_a(h\_id\_a);

dbms\_output.put\_line(cod\_d\_a);

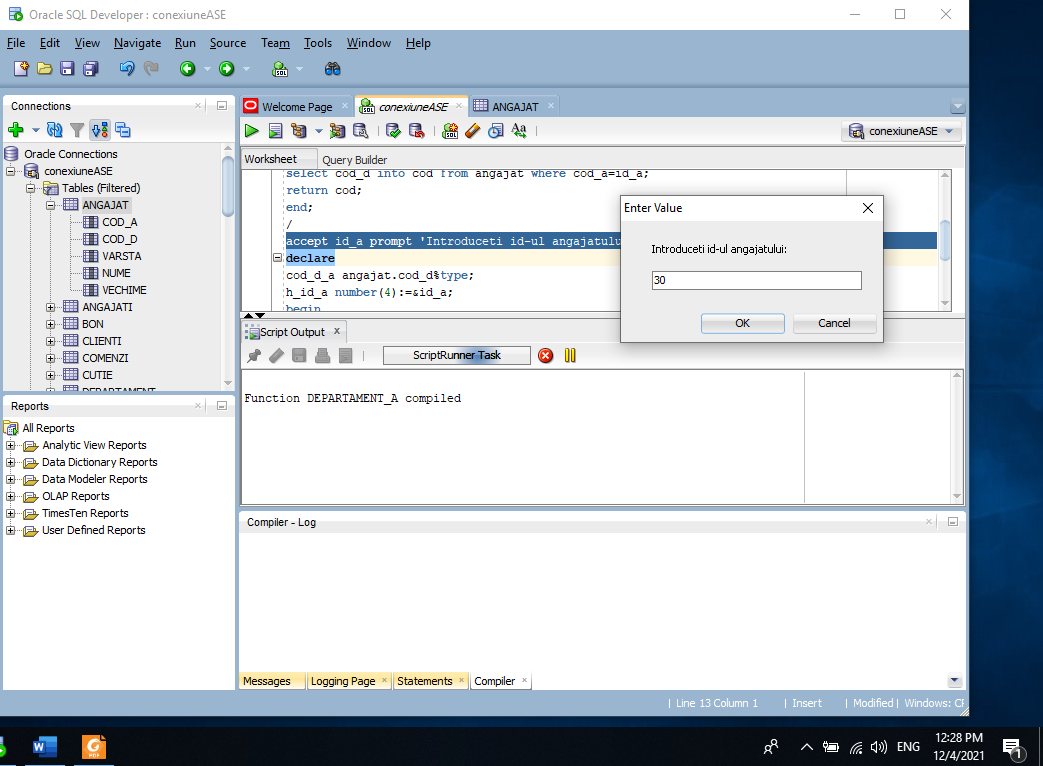
exception

when no\_data\_found then

dbms\_output.put\_line('Nu exista angajatul cu acel ID');

end;

/



12.Sa se verifice care e valoarea medie a perechilor din deposit, din tabela model folosind o functie :

create or replace function val\_medie (valoare out

model.pret%type)

return number is

begin

select AVG(pret) into valoare from model;

return valoare;

end;

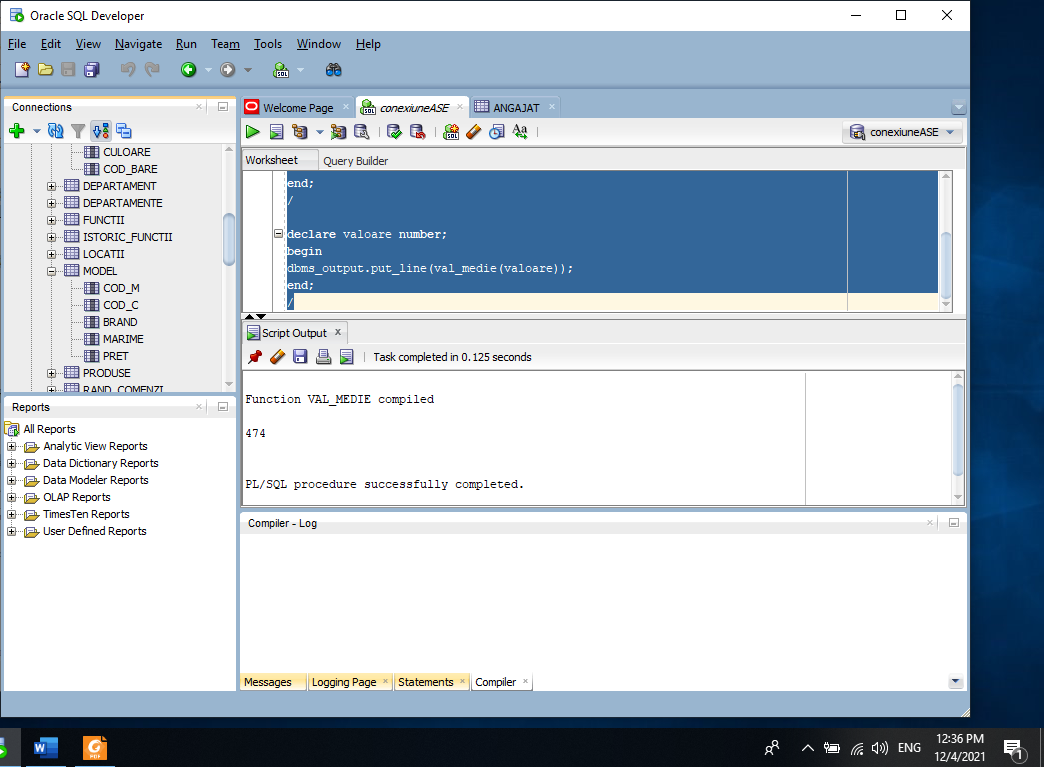
/

declare valoare number;

begin

dbms\_output.put\_line(val\_medie(valoare));

end;/



13.Sa se creeze functia care verifica daca exista un model de incaltaminte in depozit, din tabela model,cu un id dat :

create or replace function verificare(m\_id model.cod\_m%type)

return boolean is

nr number;

begin

select count (cod\_m) into nr from model where cod\_m=m\_id;

if nr=0 then

return false;

else

return true;

end if;

exception

when no\_data\_found then

return null;

end;

/

declare

begin

if(verificare(99)) then

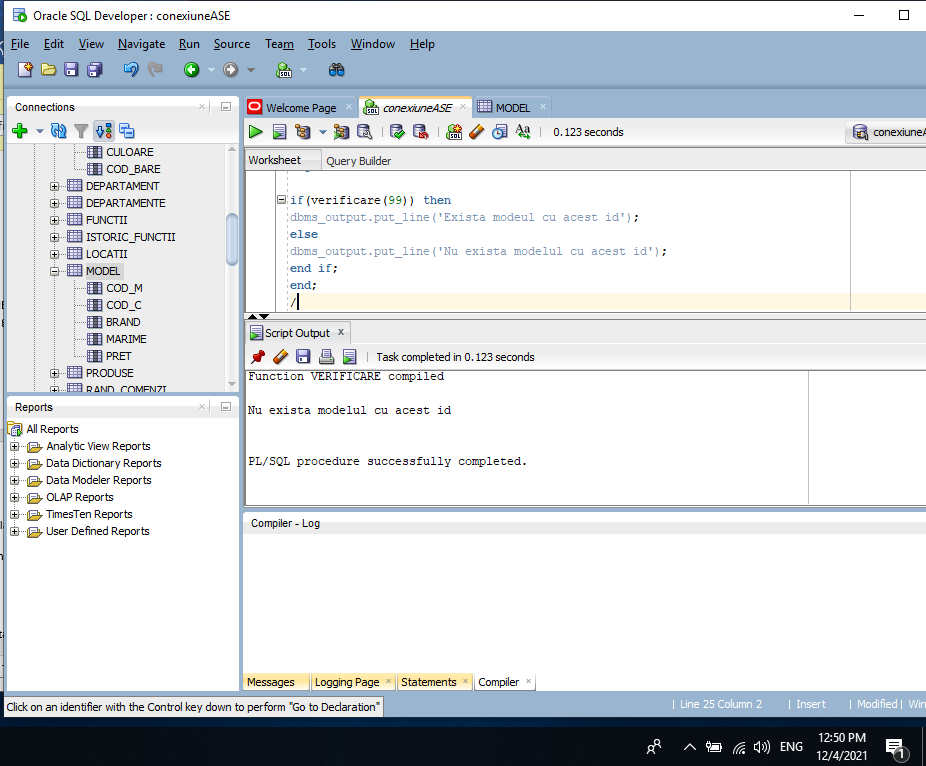
dbms\_output.put\_line('Exista modeul cu acest id');

else

dbms\_output.put\_line('Nu exista modelul cu acest id');

end if;

end;



## 6.2.Proceduri

14.Sa se creeze o procedura care modifica pretul unei perechi de incaltaminte citite de la tastatura :

CREATE OR REPLACE PROCEDURE MODIFICA(C\_M IN NUMBER, PR IN NUMBER) IS

BEGIN

UPDATE model SET pret = PR WHERE COD\_M = C\_M;

IF SQL%FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Modificare efectuata!');

ELSE

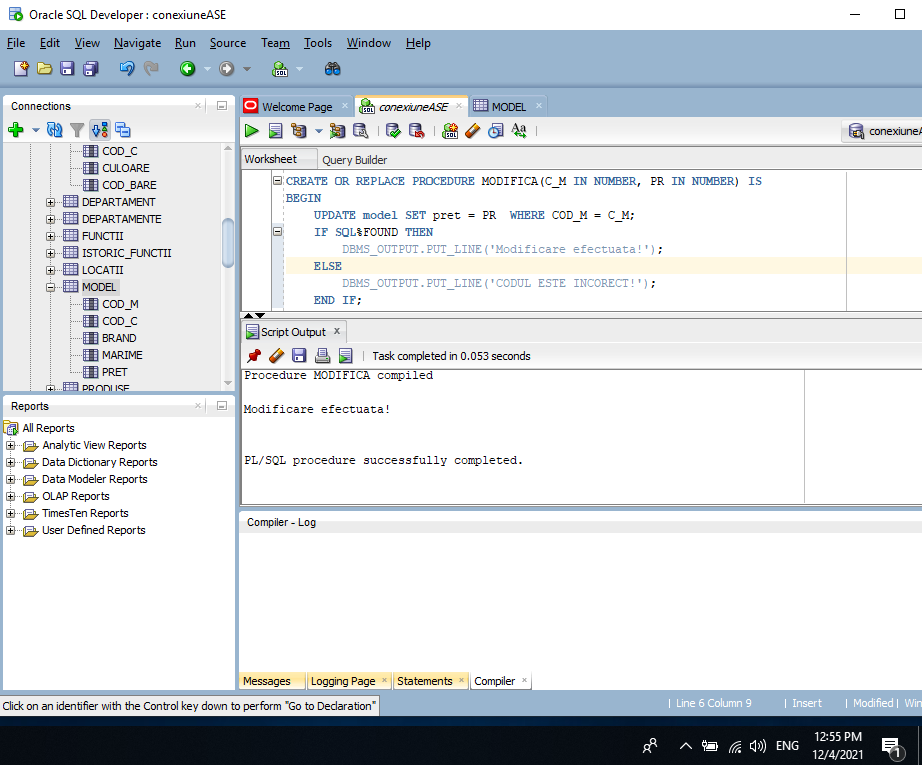
DBMS\_OUTPUT.PUT\_LINE('CODUL ESTE INCORECT!');

END IF;

END;

/

EXECUTE MODIFICA(5, 1000);



15.Sa se creeze o procedura ce afiseaza doar angajatii care nu au emis pana acum un bon :

CREATE OR REPLACE PROCEDURE ANG\_EMITERE IS

CURSOR C IS SELECT NUME FROM ANGAJAT WHERE cod\_a NOT IN (SELECT cod\_a FROM BON);

BEGIN

FOR R IN C LOOP

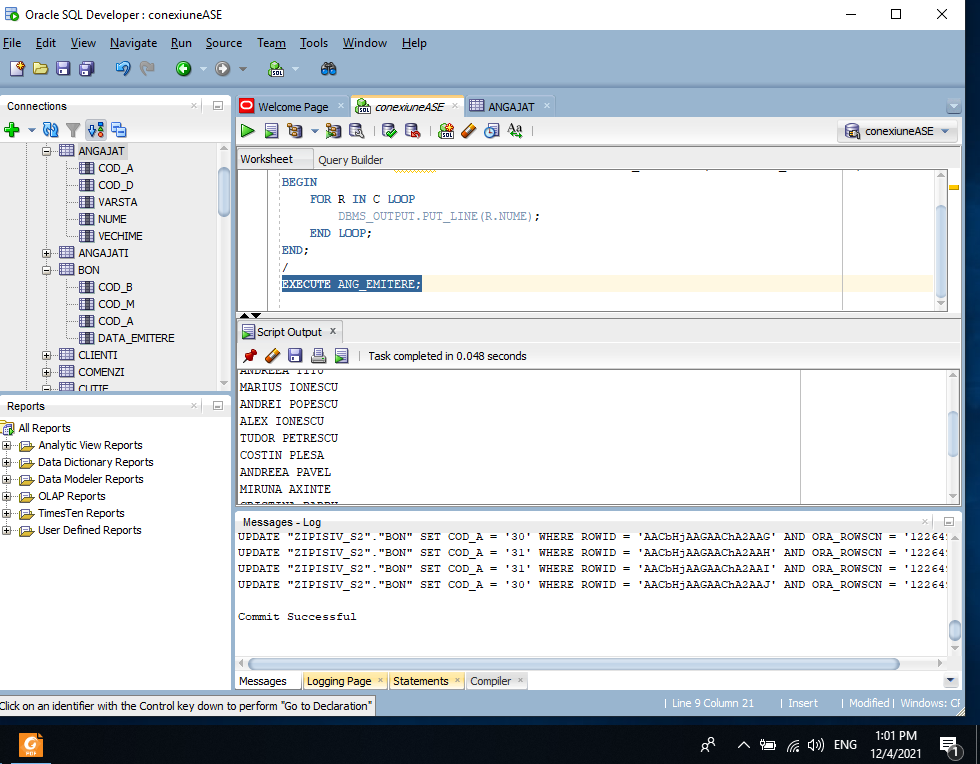
DBMS\_OUTPUT.PUT\_LINE(R.NUME);

END LOOP;

END;

/

EXECUTE ANG\_EMITERE;



16.Sa se creeze o procedura prin care sa afisati informatiile despre primele 4 bonuri emise :

create or replace procedure INFO\_bonuri

as

cursor c is select \* from BON order by data\_emitere;

begin

for var\_bon in c loop

exit when c%rowcount>4;

dbms\_output.put\_line('Bonul '||var\_bon.cod\_b||' a fost emis la data de:

'||var\_bon.data\_emitere||' de angajatul cu acest cod '||var\_bon.cod\_a);

end loop;

end;

declare

begin

INFO\_bonuri();

end;

