Proiect SGDB

Ancuta Ioan, Grupa 242

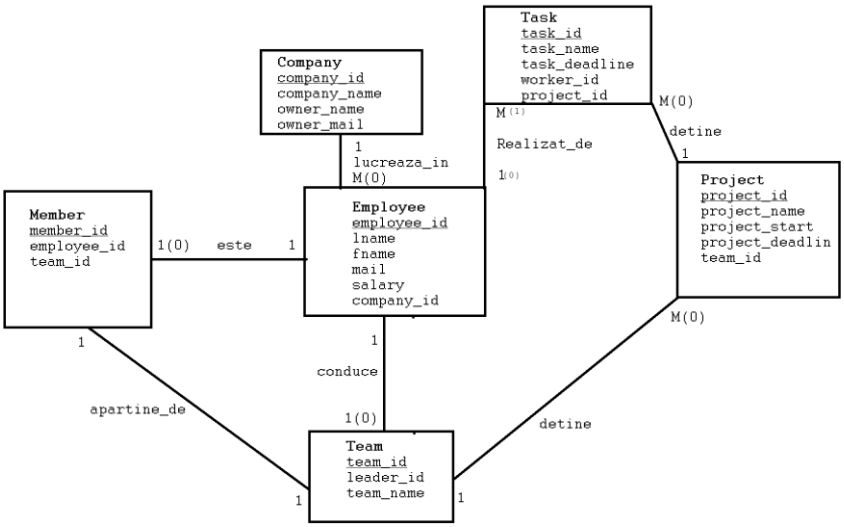
1.Prezentați pe scurt baza de date (utilitatea ei).

Baza de date poate fi folosita pentru persoanele care detin mai multe companii de IT, fiecare cu proiecte separate, intre care pot exista legaturi (exemplu : Angajatul de la compania X participa in cadrul unui proiect de la compania Y pentru a ajuta cu un task). Aceasta baza de date este adresata persoanelor care doresc sa-si usureze afacerea si sa poata obtine un workflow mai bun in viata de zi cu zi. Baza de data este formata din : company,employee,members,teams, projects si task, fiecare dintre acestea oferind detinatorului posibilitatea de a vedea task-urile pe care le are si pe care le-a avut, angajatii din cadrul companiei, dar si angajati care participa in cadrul proiectelor.

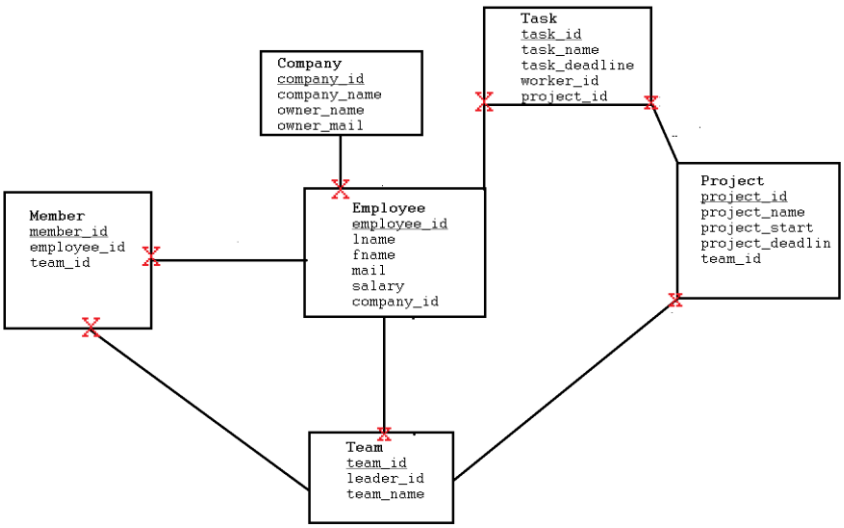
-De ce sa folosesti aceasta baza de date si nu alta ?

Pentru ca aceasta iti ofera posibilitatea sa te mulezi pe nevoile tale, daca ai cateva firme si nu le poti gestiona eficient, aceasta poate sa te ajute prin diferite cai, oferindu-ti posibilitatea de a alege cine unde lucreaza si unde este nevoie de cineva pentru a face mai eficient procesul de creare a diferitelor proiecte.

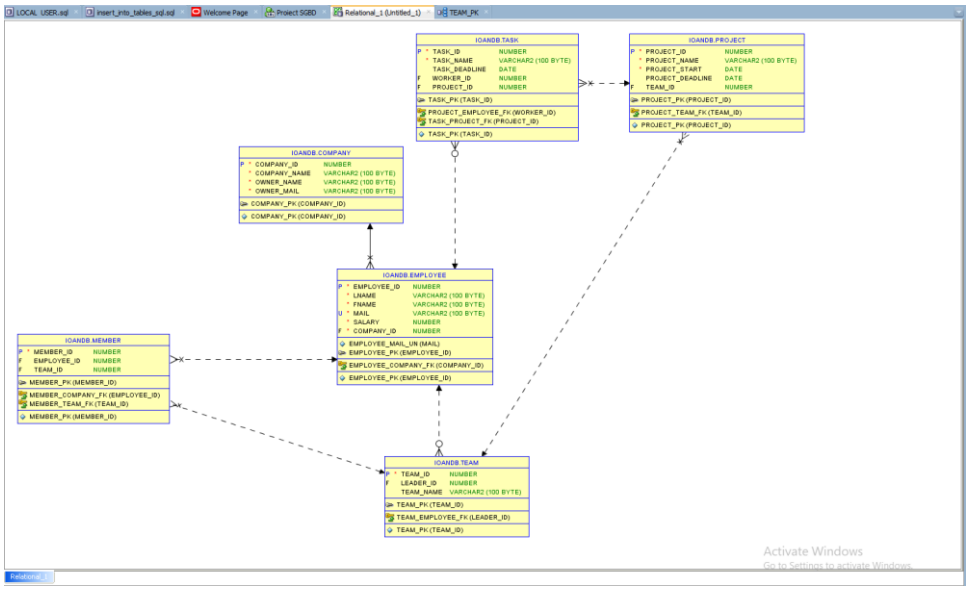
2. Realizați diagrama entitate-relație (ERD).



3. Pornind de la diagrama entitate-relație realizați diagrama conceptuală a modelului propus, integrând toate atributele necesare.



4.Implementați în Oracle diagrama conceptuală realizată: definiți toate tabelele, implementând toate constrângerile de integritate necesare (chei primare, cheile externe etc).



5.Adăugați informații coerente în tabelele create (minim 3-5 înregistrări pentru fiecare entitate independentă; minim 10 înregistrări pentru tabela asociativă).

```
INSERT INTO company( company_id, company_name, owner_name, owner_mail) VALUES (1,'ALCATRA', 'Adrian Faptura', 'adrianfaketaxes@gmail.com');

INSERT INTO company( company_id, company_name, owner_name, owner_mail) VALUES (2,'GAZURO', 'Mugurel Murel', '123mugurelu@gmail.com');

INSERT INTO company( company_id, company_name, owner_name, owner_mail) VALUES (3,'BMWsmalld', 'Mircea Mistuitoru', 'gariantu@gmail.com');

INSERT INTO company( company_id, company_name, owner_name, owner_mail) VALUES (4,'LAGAR', 'Gogea Cliptukluu', 'larabatie@gmail.com');

INSERT INTO company( company_id, company_name, owner_name, owner_mail) VALUES (5,'COM', 'Andreea Ancuta', 'lbae@gmail.com');
```

```
INSERT INTO employee( employee_id, lName, fName, mail,salary, company_id)

VALUES (1,'Marius','Ovidiu','marius@gmail',1200,1);

INSERT INTO employee( employee_id, lName, fName, mail,salary, company_id)

VALUES (2,'Inglesia','Ovidiu','mipu@gmail',2400,1);

INSERT INTO employee( employee_id, lName, fName, mail,salary, company_id)

VALUES (3,'Jumi','Daniel','dancila@gmail',13000,1);

INSERT INTO employee( employee_id, lName, fName, mail,salary, company_id)

VALUES (4,'Samuel','Samule','sam@gmail',7300,2);

INSERT INTO employee( employee_id, lName, fName, mail,salary, company_id)

VALUES (5,'Sami','Ovidiu','movid@gmail',2345,3);

 INSERT INTO employee( employee_id, lName, fName, mail,salary, company_id)

VALUES (6,'Gustavo','Olips','gusti@gmail',24344,2);

 INSERT INTO employee( employee_id, lName, fName, mail,salary, company_id)

VALUES (7,'Gabrie','Oles','olesie@gmail',2333,5);

INSERT INTO employee( employee_id, lName, fName, mail,salary, company_id)

VALUES (8,'Maghere','Ovidiu','mag@gmail',5894,5);

INSERT INTO employee( employee_id, lName, fName, mail,salary, company_id)

VALUES (9,'Lapita','Olimpia','fatatati@gmail',1230,4);
```

```
INSERT INTO employee( employee_id, lName, fName, mail,salary, company_id)
VALUES (10,'Lapita','Garcea','baiatutati@gmail',980,3);
INSERT INTO employee( employee_id, lName, fName, mail,salary, company_id)
VALUES (11,'Crinu','Olita','crini@gmail',5000,4);


INSERT INTO team(team_id,leader_id,team_name)
VALUES (101,11,'Exotica');
INSERT INTO team(team_id,leader_id,team_name)
VALUES (102,2,'Echias');
INSERT INTO team(team_id,leader_id,team_name)
VALUES (103,2,'Perenia');
INSERT INTO team(team_id,leader_id,team_name)
VALUES (104,8,'lagar');
INSERT INTO team(team_id,leader_id,team_name)
VALUES (111,5,'Extensia');
INSERT INTO team(team_id,leader_id,team_name)
VALUES (158,1,'Sublagar');
select * from member;


INSERT INTO member(member_id, employee_id, team_id) VALUES(1,11,101);
INSERT INTO member(member_id, employee_id, team_id) VALUES(2,2,102);
INSERT INTO member(member_id, employee_id, team_id) VALUES(16,2,103);
INSERT INTO member(member_id, employee_id, team_id) VALUES(17,8,104);
INSERT INTO member(member_id, employee_id, team_id) VALUES(3,8,104);
INSERT INTO member(member_id, employee_id, team_id) VALUES(4,5,111);
INSERT INTO member(member_id, employee_id, team_id) VALUES(5,1,158);
INSERT INTO member(member_id, employee_id, team_id) VALUES(6,2,101);
INSERT INTO member(member_id, employee_id, team_id) VALUES(7,11,102);
INSERT INTO member(member_id, employee_id, team_id) VALUES(8,1,104);
INSERT INTO member(member_id, employee_id, team_id) VALUES(9,7,111);
INSERT INTO member(member_id, employee_id, team_id) VALUES(10,5,158);
INSERT INTO member(member_id, employee_id, team_id) VALUES(11,4,101);
INSERT INTO member(member_id, employee_id, team_id) VALUES(12,10,102);
```

```
INSERT INTO member(member_id, employee_id, team_id) VALUES(13,3,104);
INSERT INTO member(member_id, employee_id, team_id) VALUES(14,6,111);
INSERT INTO member(member_id, employee_id, team_id) VALUES(15,10,158);

INSERT INTO project(project_id, project_name,project_start,project_deadline,team_id)
VALUES(1001, 'Cerberos',TO_DATE('2015-Jan-01','yy-mm-dd'),TO_DATE('2015-Apr-01','yy-mm-dd'),103);
INSERT INTO project(project_id, project_name,project_start,project_deadline,team_id)
VALUES(1023, 'Ignatius',TO_DATE('2016-Sep-01','yy-mm-dd'),TO_DATE('2023-Feb-01','yy-mm-dd'),101);
INSERT INTO project(project_id, project_name,project_start,project_deadline,team_id)
VALUES(1041, 'Xerxes',TO_DATE('2010-Dec-01','yy-mm-dd'),TO_DATE('2020-Jul-01','yy-mm-dd'),101);
INSERT INTO project(project_id, project_name,project_start,project_deadline,team_id)
VALUES(1301, 'Biggie',TO_DATE('2011-Nov-01','yy-mm-dd'),TO_DATE('2025-May-01','yy-mm-dd'),103);
INSERT INTO project(project_id, project_name,project_start,project_deadline,team_id)
VALUES(2001, 'Pikachu',TO_DATE('2013-Jun-01','yy-mm-dd'),TO_DATE('2015-Apr-03','yy-mm-dd'),102);

INSERT INTO task(task_id, task_name, task_deadline,worker_id,project_id)
VALUES(1322,'LOGIN',TO_DATE('2015-Mar-04','yy-mm-dd'),NULL,1001);
INSERT INTO task(task_id, task_name, task_deadline,worker_id,project_id)
VALUES(2232,'BackEndBugSolver',TO_DATE('2019-Mar-04','yy-mm-dd'),2,1041);
INSERT INTO task(task_id, task_name, task_deadline,worker_id,project_id)
VALUES(1002,'CurrencyProb',TO_DATE('2018-Apr-04','yy-mm-dd'),11,1041);
INSERT INTO task(task_id, task_name, task_deadline,worker_id,project_id)
VALUES(7952,'Mobile',TO_DATE('2015-Mar-04','yy-mm-dd'),4,1023);
INSERT INTO task(task_id, task_name, task_deadline,worker_id,project_id)
VALUES(8232,'DataLeak',TO_DATE('2015-Mar-04','yy-mm-dd'),2,1023);
INSERT INTO task(task_id, task_name, task_deadline,worker_id,project_id)
VALUES(9322,'shopingcartproblem',TO_DATE('2017-Mar-04','yy-mm-dd'),4,1023);
INSERT INTO task(task_id, task_name, task_deadline,worker_id,project_id)
VALUES(3322,'backdoorhack',TO_DATE('2015-Feb-04','yy-mm-dd'),null,1001);
INSERT INTO task(task_id, task_name, task_deadline,worker_id,project_id)
VALUES(9999,'login issues',TO_DATE('2014-Mar-04','yy-mm-dd'),10,2001);
INSERT INTO task(task_id, task_name, task_deadline,worker_id,project_id)
VALUES(1000,'edit',TO_DATE('2012-Mar-04','yy-mm-dd'),NULL,1301);
```

```
select * from company;
```

Query Result

SQL | All Rows Fetched: 6 in 0.796 seconds

	COMPANY_ID	COMPANY_NAME	OWNER_NAME	OWNER_MAIL
1	1	ALCATRA	Adrian Faptura	adrianfaketaxes@gmail.com
2	2	GAZURO	Mugurel Murel	123mugurelu@gmail.com
3	3	BMWsmalld	Mircea Mistuitoru	gariantu@gmail.com
4	4	LAGAR	Gogea Cliptukluu	larabatie@gmail.com
5	5	COM	Andreea Ancuta	lbae@gmail.com
6	6	test	Andreea Ancuta	lbae@gmail.com

```
select * from employee;
```

Query Result

SQL | All Rows Fetched: 11 in 0.083 seconds

	EMPLOYEE_ID	LNAME	FNAME	MAIL	SALARY	COMPANY_ID
1	1	Marius	Ovidiu	marius@gmail	1200	1
2	2	Inglesia	Ovidiu	mipu@gmail	2400	1
3	3	Jumi	Daniel	dancila@gmail	13000	1
4	4	Samuel	Samule	sam@gmail	7300	2
5	5	Sami	Ovidiu	movid@gmail	2345	3
6	6	Gustavo	Olips	gusti@gmail	24344	2
7	7	Gabrie	Oles	olesie@gmail	2333	5
8	8	Maghere	Ovidiu	mag@gmail	5894	5
9	9	Lapita	Olimpia	fatatati@gmail	1230	4
10	10	Lapita	Garcea	baiatutati@gmail	980	3
11	11	Crinu	Olita	crini@gmail	5000	4

```
select * from member;
```

Query Result

SQL | All Rows Fetched: 17 in 0.641

	MEMBER_ID	EMPLOYEE_ID	TEAM_ID
1	1	11	101
2	2	2	102
3	3	8	104
4	4	5	111
5	5	1	158
6	6	2	101
7	7	11	102
8	8	1	104
9	9	7	111
10	10	5	158
11	11	4	101
12	12	10	102
13	13	3	104
14	14	6	111
15	15	10	158
16	16	2	103
17	17	8	104

```
select * from team;
```

Query Result

SQL | All Rows Fetched: 6 in 0.31

	TEAM_ID	LEADER_ID	TEAM_NAME
1	101	11	Exotica
2	103	2	Perenia
3	104	8	lagar
4	111	5	Extensia
5	158	1	Sublagar
6	102	2	Echias

```
select * from project;
```

Query Result

SQL | All Rows Fetched: 5 in 0.138 seconds

	PROJECT_ID	PROJECT_NAME	PROJECT_START	PROJECT_DEADLINE	TEAM_ID
1	1001	Cerberos	01-JAN-15	01-APR-15	103
2	1023	Ignatius	01-SEP-16	01-FEB-23	101
3	1041	Xerxes	01-DEC-10	01-JUL-20	101
4	1301	Biggie	01-NOV-11	01-MAY-25	103
5	2001	Pikachu	01-JUN-13	03-APR-15	102

```
select * from task;
```

Query Result

SQL | All Rows Fetched: 9 in 0.132 seconds

	TASK_ID	TASK_NAME	TASK_DEADLINE	WORKER_ID	PROJECT_ID
1	1322	LOGIN	04-MAR-15	(null)	1001
2	2232	BackEndBugSolver	04-MAR-19	2	1041
3	1002	CurrencyProb	04-APR-18	11	1041
4	7952	Mobile	04-MAR-15	4	1023
5	8232	DataLeak	04-MAR-15	2	1023
6	9322	shopingcartproblem	04-MAR-17	4	1023
7	3322	backdoorhack	04-FEB-15	(null)	1001
8	9999	login issues	04-MAR-14	10	2001
9	1000	edit	04-MAR-12	(null)	1301

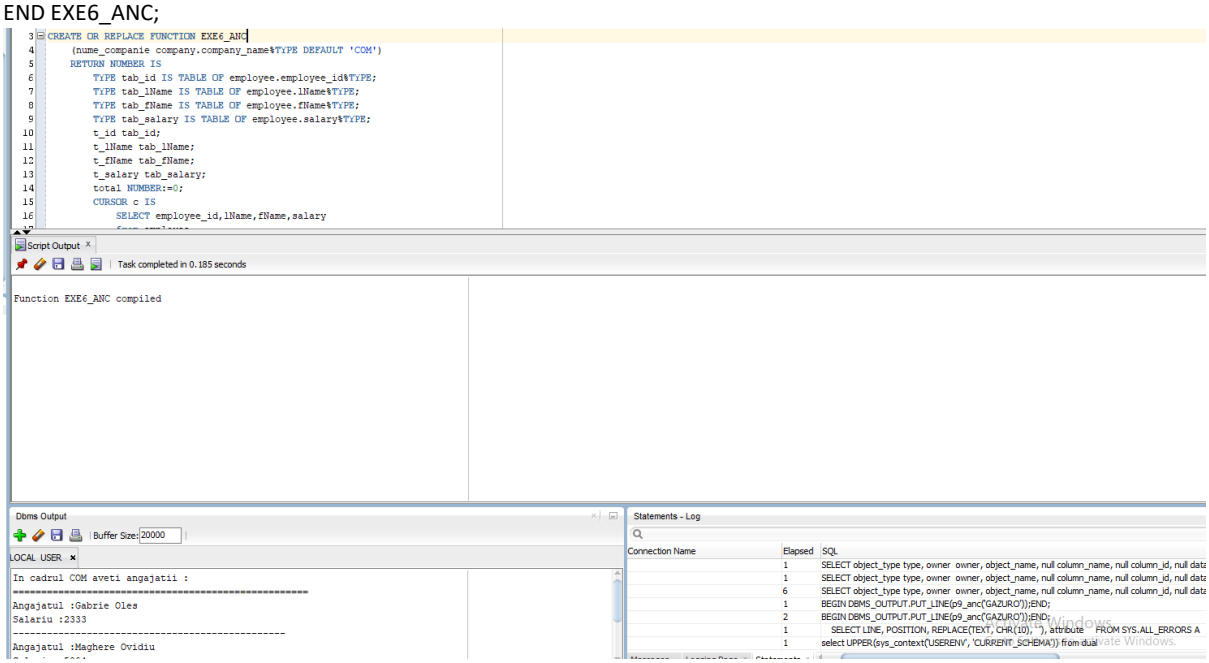
6.Definiți un subprogram stocat care să utilizeze un tip de colecție studiat. Apelați subprogramul.

```
CREATE OR REPLACE FUNCTION EXE6_ANC
(nume_companie company.company_name%TYPE DEFAULT 'COM')
   RETURN NUMBER IS
TYPE tab_id IS TABLE OF employee.employee_id%TYPE;
TYPE tab_lName IS TABLE OF employee.lName%TYPE;
TYPE tab_fName IS TABLE OF employee.fName%TYPE;
TYPE tab_salary IS TABLE OF employee.salary%TYPE;
t_id tab_id;
t_lName tab_lName;
t_fName tab_fName;
t_salary tab_salary;
total NUMBER:=0;
CURSOR c IS
SELECT employee_id,lName,fName,salary
from employee
where company_id IN (select company_id
from company
where company_name=nume_companie);
BEGIN
OPEN c;
FETCH c  BULK COLLECT INTO t_id, t_lName, t_fName,t_salary;
CLOSE c;
DBMS_OUTPUT.PUT_LINE('In cadrul ' || nume_companie || ' aveti angajatii : ');
DBMS_OUTPUT.PUT_LINE('=====================================================');
FOR i in t_id.FIRST..t_id.LAST LOOP
DBMS_OUTPUT.PUT_LINE('Angajatul :' || t_lName(i) || ' ' || t_fName(i));
      DBMS_OUTPUT.PUT_LINE('Salariu :' || t_salary(i));
      IF (i<t_id.LAST) THEN DBMS_OUTPUT.PUT_LINE('-----------------------------------------------');
      END IF;
total:=total+t_salary(i);
END LOOP;
   DBMS_OUTPUT.PUT_LINE('=====================================================');
```

DBMS_OUTPUT.PUT_LINE('Total de plata pe luna : ' || total);

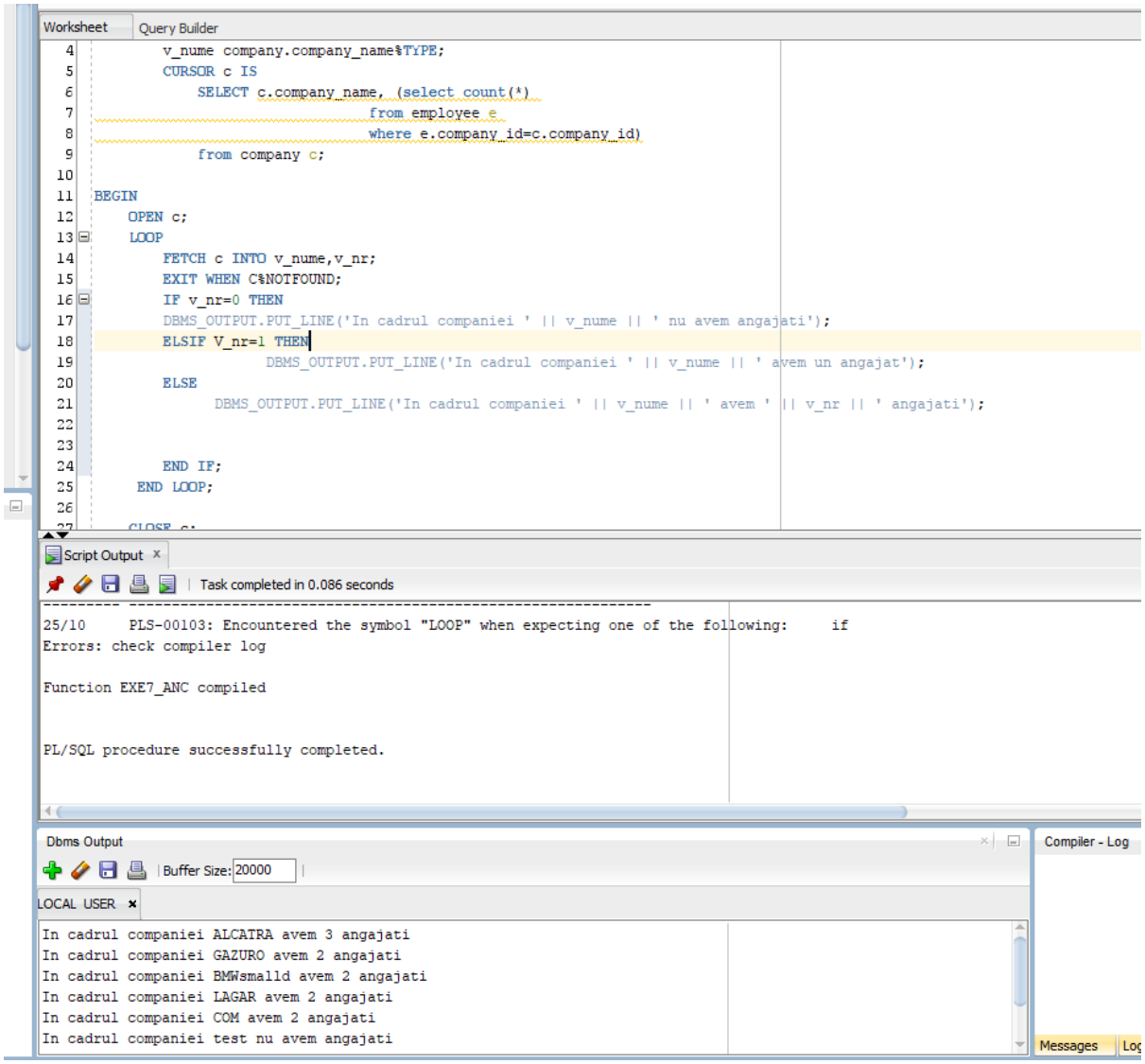
RETURN 0;

END EXE6_ANC;



7.Definiți un subprogram stocat care să utilizeze un tip de cursor studiat. Apelați subprogramul.

```
CREATE OR REPLACE FUNCTION EXE7_ANC
    RETURN NUMBER IS
        v_nr number(4);
        v_nume company.company_name%TYPE;
        CURSOR c IS
            SELECT c.company_name, (select count(*)
                                    from employee e
                                    where e.company_id=c.company_id)
            from company c;

BEGIN
```

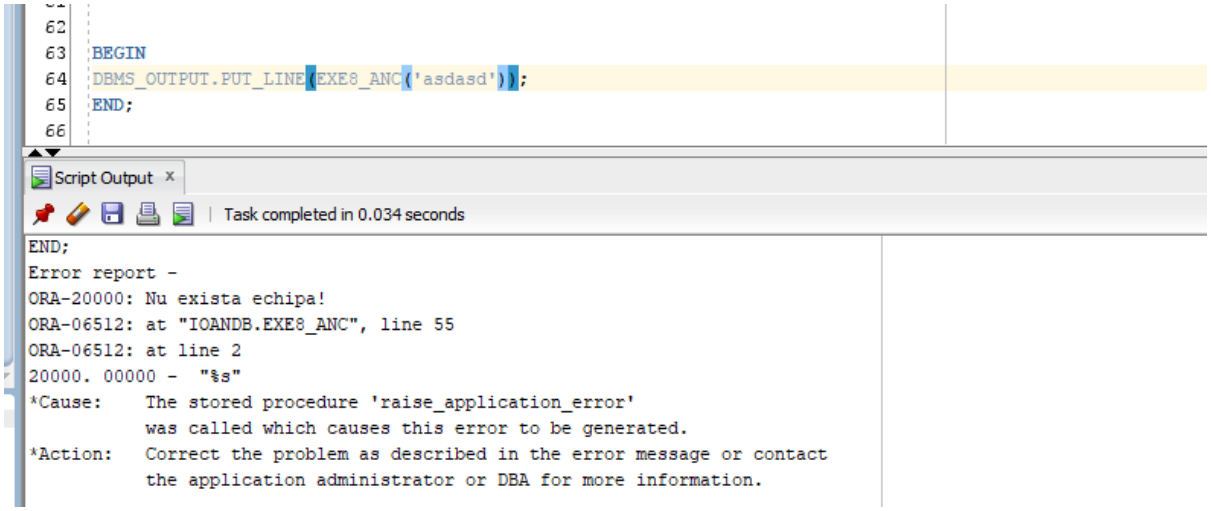



```
CREATE OR REPLACE FUNCTION EXE7_ANC
   RETURN NUMBER IS
v_nr number(4);
      v_nume company.company_name%TYPE;
CURSOR c IS
         SELECT c.company_name, (select count(*)
                  from employee e
                  where e.company_id=c.company_id)
         from company c;

BEGIN
OPEN c;
   LOOP
      FETCH c INTO v_nume,v_nr;
      EXIT WHEN C%NOTFOUND;
```

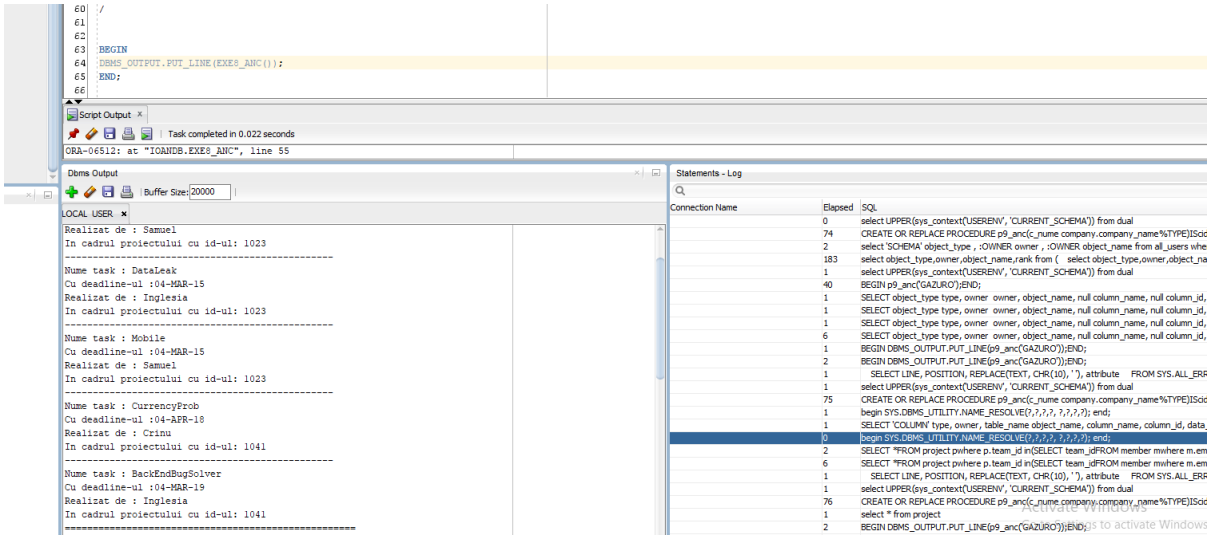
```
    IF v_nr=0 THEN
    DBMS_OUTPUT.PUT_LINE('In cadrul companiei ' || v_nume || ' nu avem angajati');
    ELSIF V_nr=1 THEN
            DBMS_OUTPUT.PUT_LINE('In cadrul companiei ' || v_nume || ' avem un angajat');
    ELSE
        DBMS_OUTPUT.PUT_LINE('In cadrul companiei ' || v_nume || ' avem ' || v_nr || ' angajati');



    END IF;
  END LOOP;


CLOSE c;
RETURN 0;
END EXE7_ANC;
/
BEGIN
DBMS_OUTPUT.PUT_LINE(EXE7_ANC);
END;
```

8.Definiți un subprogram stocat de tip funcție care să utilizeze 3 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogsamulastfelîncât să evidențiați toate cazurile tratate.

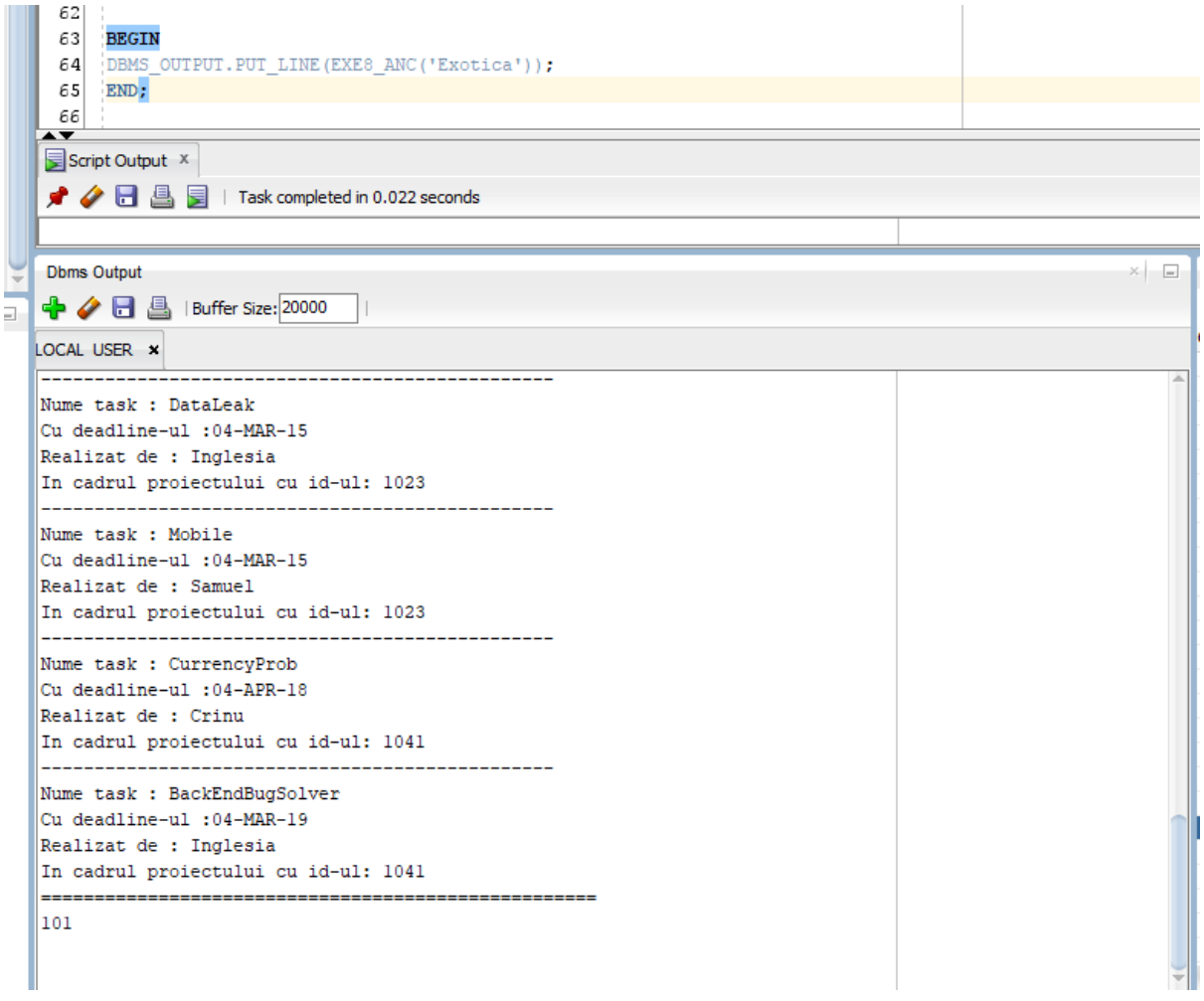
1.First RUN= rulam cu numele unei echipe care nu exista.



2.Second RUN=rulam cu echipa default



3.Third Run=rulam cu o echipa data.



CREATE OR REPLACE FUNCTION EXE8_ANC

(teamname team.team_name%TYPE DEFAULT 'Exotica')

RETURN NUMBER IS

TYPE tab_id IS TABLE OF task.task_id%TYPE;

```
TYPE tab_task_name IS TABLE OF task.task_name%TYPE;
TYPE tab_task_deadline IS TABLE OF task.task_deadline%TYPE;
TYPE tab_worker IS TABLE OF task.worker_id%TYPE;
TYPE tab_project_id IS TABLE OF task.project_id%TYPE;
    tNamet VARCHAR(200);
    teid NUMBER;
    is_found_rec boolean := false;

t_id tab_id;
t_name tab_task_name;
t_dl tab_task_deadline;
t_worker tab_worker;
t_pid tab_project_id;

total NUMBER:=0;
tid employee.employee_id%TYPE;
CURSOR c IS
SELECT *
FROM task t
where t.project_id in (
select project_id
from project p
where p.team_id in(select team_id
from team te
        where te.team_name=teamname));
BEGIN
    SELECT team_id into teid
    from team
    where team_name=teamname;
OPEN c;
FETCH c  BULK COLLECT INTO t_id, t_name , t_dl ,t_worker,t_pid;
CLOSE c;

DBMS_OUTPUT.PUT_LINE('In cadrul echipe ' || teamname || ' aveti task-urile urmatoare : ');
DBMS_OUTPUT.PUT_LINE('==================================================');
```

```
FOR i in t_id.FIRST..t_id.LAST LOOP
        is_found_rec := true;

DBMS_OUTPUT.PUT_LINE('Nume task : ' || t_name(i));
DBMS_OUTPUT.PUT_LINE('Cu deadline-ul :' || t_dl(i));
       select lName into tNamet
          from employee
          where employee_id=t_worker(i);
DBMS_OUTPUT.PUT_LINE('Realizat de : ' || tNamet);

DBMS_OUTPUT.PUT_LINE('In cadrul proiectului cu id-ul: ' || t_pid(i));
       IF (i<t_id.LAST) THEN DBMS_OUTPUT.PUT_LINE('-----------------------------------------------');
       END IF;
END LOOP;

   DBMS_OUTPUT.PUT_LINE('=================================================');
    /* if not is_found_rec then
      RAISE exception_name1;  */
    end if

end if;
RETURN teid;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      RAISE_APPLICATION_ERROR(-20000,'Nu exista echipa!');
/*    WHEN exception_name1 THEN
      RAISE_APPLICATION_ERROR(-20000,'Compania nu are nimic la activ!'); */
   WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Alta eroare!');

END EXE8_ANC;
/
BEGIN
DBMS_OUTPUT.PUT_LINE(EXE8_ANC);
END;
```

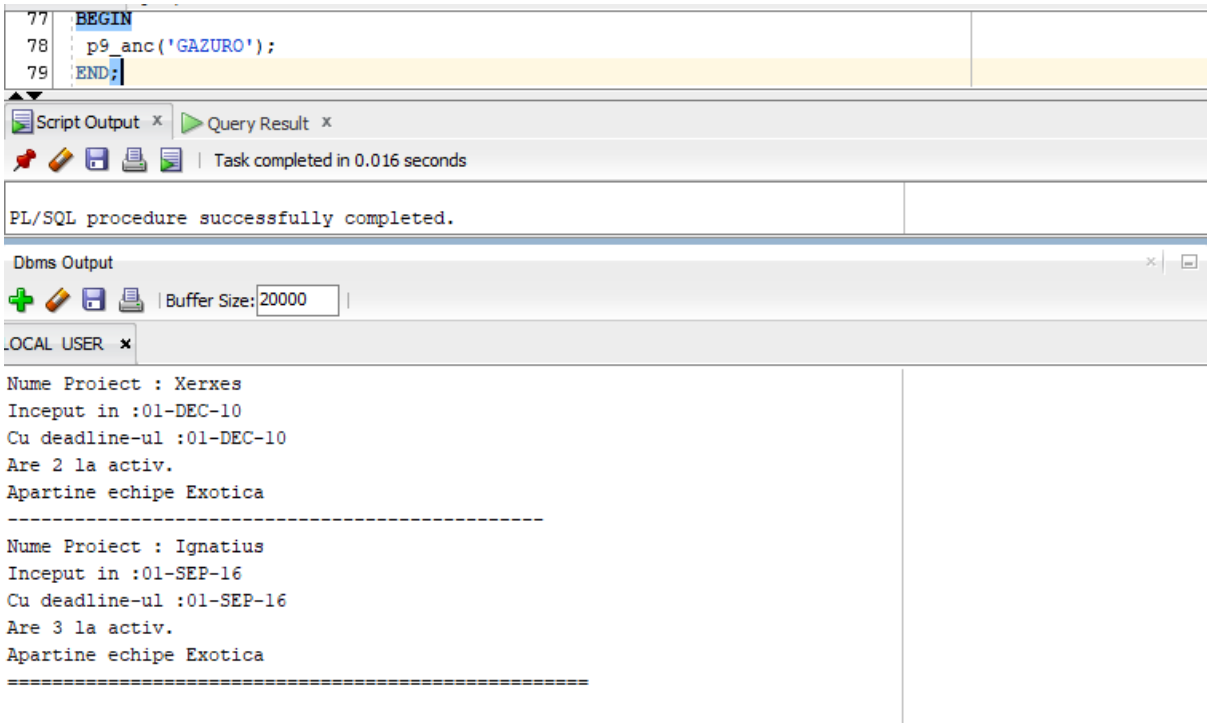
9.Definiți un subprogram stocat de tip procedură care să utilizeze 5 dintre tabelele definite. Tratați toate excepțiile care pot apărea. Apelați subprogramulastfel încât să evidențiați toate cazurile tratate.

```
CREATE OR REPLACE PROCEDURE p9_anc
(c_nume company.company_name%TYPE)
IS
cid company.company_id%TYPE;
--                      select company_id into cid
--          from company
--          where company_name=c_nume;



TYPE tab_id IS TABLE OF project.project_id%TYPE;
TYPE tab_project_name IS TABLE OF project.project_name%TYPE;
TYPE tab_project_start IS TABLE OF project.project_start%TYPE;
TYPE tab_project_deadline IS TABLE OF project.project_deadline%TYPE;
TYPE tab_teamid IS TABLE OF project.team_id%TYPE;
      is_found_rec boolean := false;
      exception_name1 EXCEPTION;

t_id tab_id;
t_name tab_project_name;
t_s tab_project_start;
t_dl tab_project_deadline;
t_tid tab_teamid;

nrtask NUMBER;
teamname team.team_name%TYPE;



CURSOR c IS
SELECT *
FROM project p
where p.team_id in(
SELECT team_id
FROM member m
```

```
where m.employee_id in ( SELECT employee_id
FROM employee e
where e.company_id=(select company_id
                                        from company
                                        where company_name=c_nume)));
BEGIN
OPEN c;
FETCH c  BULK COLLECT INTO t_id,t_name,t_s ,t_dl,t_tid;
CLOSE c;

FOR i in t_id.FIRST..t_id.LAST LOOP
            is_found_rec := true;

            DBMS_OUTPUT.PUT_LINE('Nume Proiect : ' || t_name(i));
            DBMS_OUTPUT.PUT_LINE('Inceput in :' || t_s(i));
            DBMS_OUTPUT.PUT_LINE('Cu deadline-ul :' || t_s(i));

            select count(*) into nrtask
               from task t
               where t.project_id=t_id(i);
            SELECT te.team_name into teamname
            from team te
            where te.team_id=t_tid(i);
            DBMS_OUTPUT.PUT_LINE('Are ' || nrtask || ' la activ.');
            DBMS_OUTPUT.PUT_LINE('Apartine echipe ' || teamname);

            IF (i<t_id.LAST) THEN DBMS_OUTPUT.PUT_LINE('--------------------------------------------------');
            END IF;
      END LOOP;
       if not is_found_rec then
      RAISE exception_name1;
      end if;
   DBMS_OUTPUT.PUT_LINE('==================================================');
   EXCEPTION
    WHEN exception_name1 THEN
```

```
      RAISE_APPLICATION_ERROR(-20000,'Compania nu are nimic la activ!');
   WHEN OTHERS THEN
         RAISE_APPLICATION_ERROR(-20001,'Nu exista compania!');
END p9_anc;
/


select * from project;


BEGIN
p9_anc('COM');
END;
```

1.Cand rulam fara erori



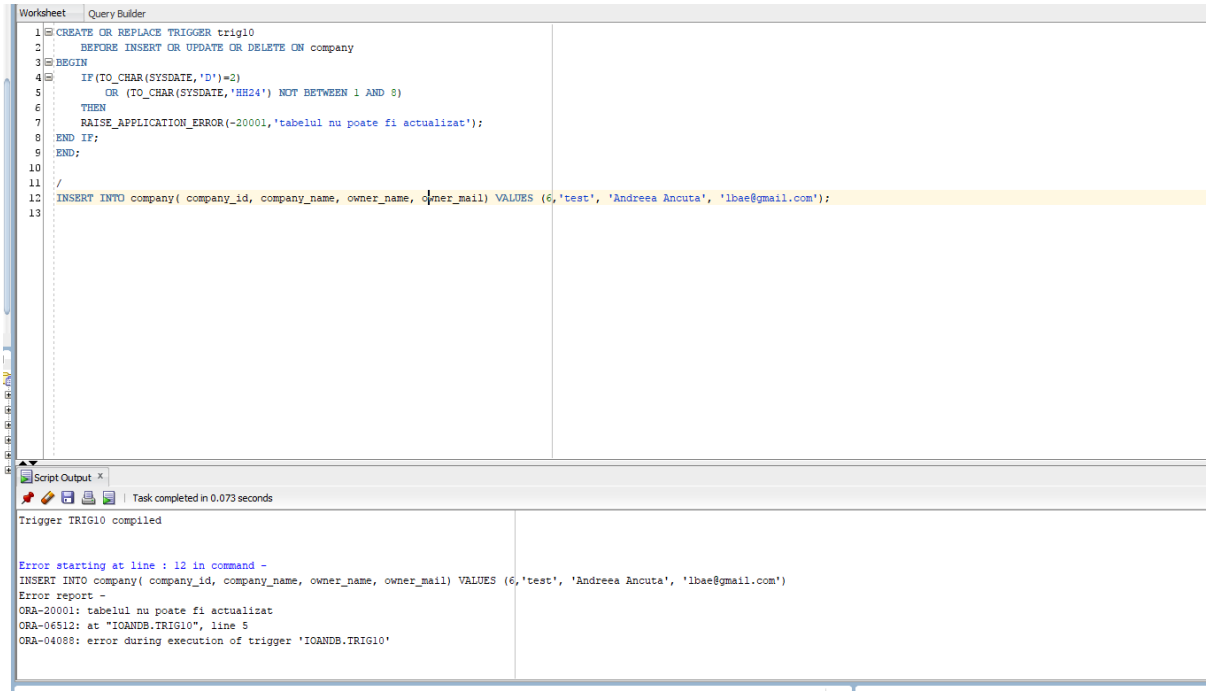
2.Cand rulam cu o companie care nu exista

```
 p9_anc('asd');
END;
Error report -
ORA-20001: Nu exista compania!
ORA-06512: at "IOANDB.P9_ANC", line 73
ORA-06512: at line 2
```

3.Cand rulam cu o compania care nu are nimic la activ

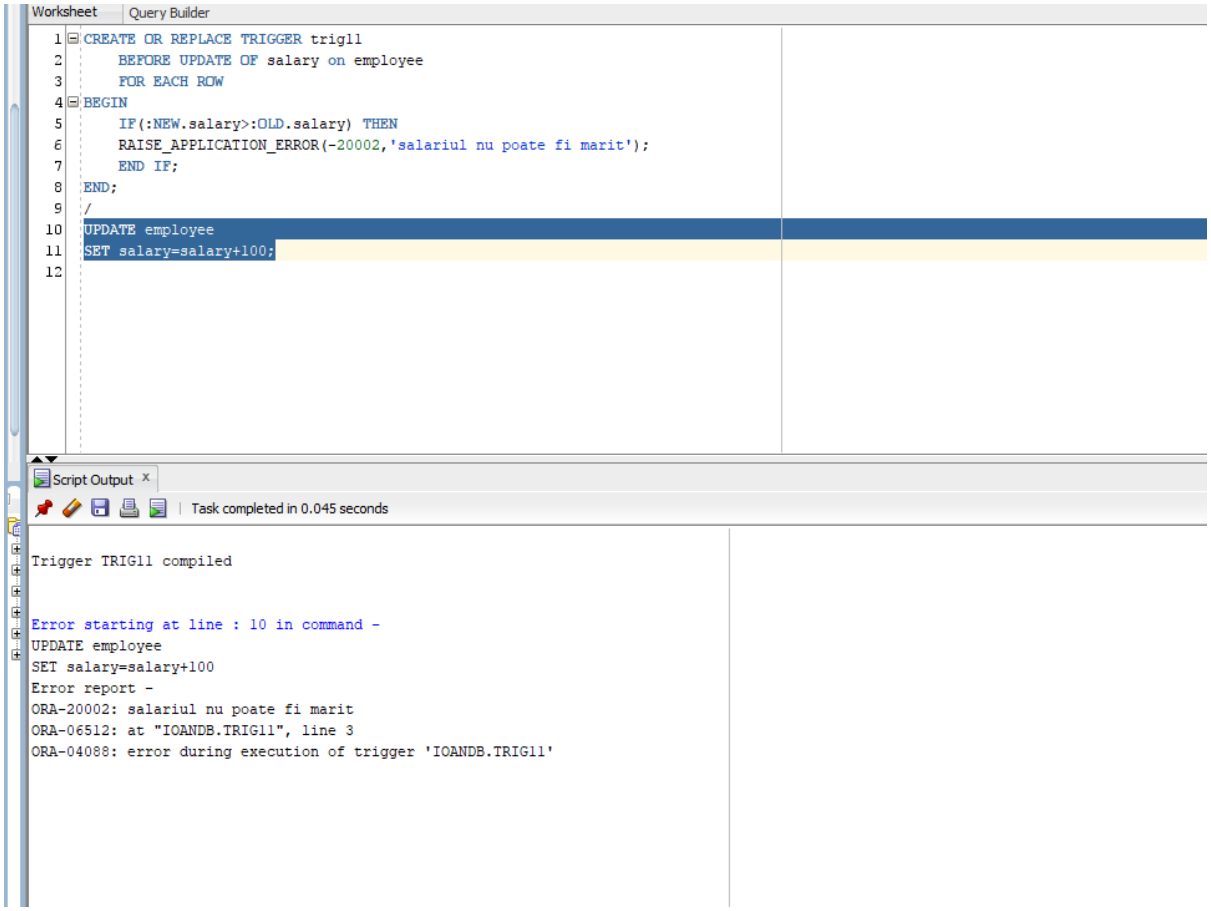
```
 p9_anc('com');
END;
Error report -
ORA-20001: Compania nu are nimic la activ!
ORA-06512: at "IOANDB.P9_ANC", line 73
ORA-06512: at line 2
```

10.Definiți un triggerde tip LMD la nivel de comandă. Declanșați trigger-ul.




```
CREATE OR REPLACE TRIGGER trig10
   BEFORE INSERT OR UPDATE OR DELETE ON company
BEGIN
   IF(TO_CHAR(SYSDATE,'D')=2)
      OR (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 1 AND 8)
   THEN
   RAISE_APPLICATION_ERROR(-20001,'tabelul nu poate fi actualizat');
END IF;
END;

/
INSERT INTO company( company_id, company_name, owner_name, owner_mail) VALUES (6,'test', 'Andreea Ancuta',
'lbae@gmail.com');
```

11.Definiți un trigger tip LMD la nivel de linie.Declanșați trigger-ul.

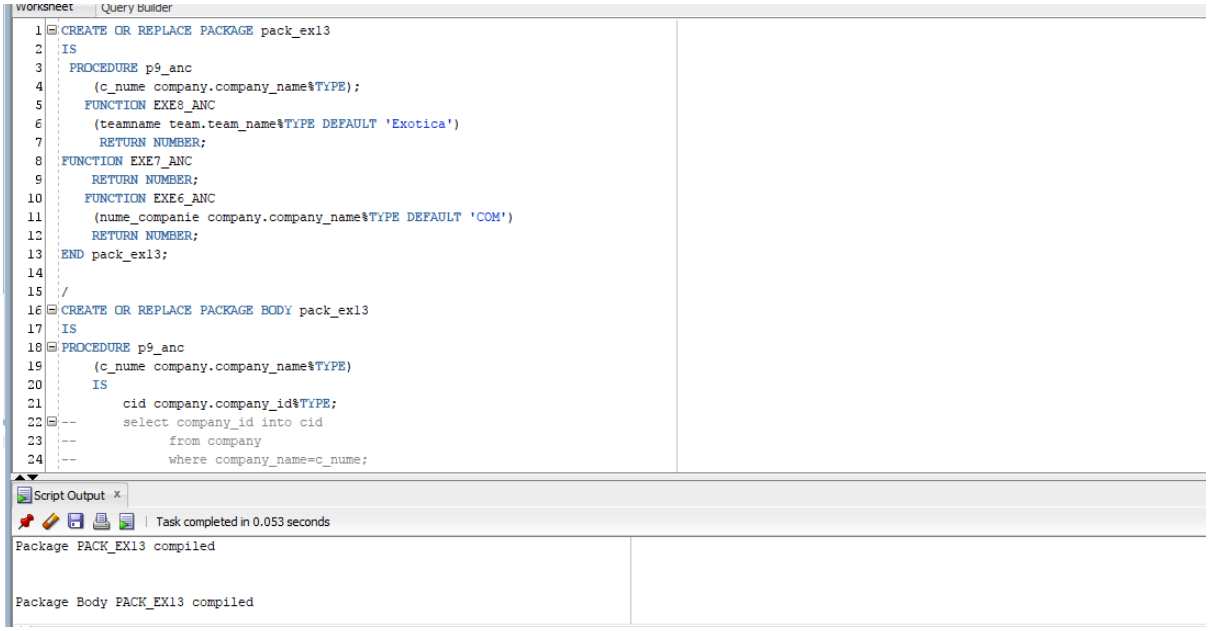



```
CREATE OR REPLACE TRIGGER trig11
   BEFORE UPDATE OF salary on employee
   FOR EACH ROW
BEGIN
   IF(:NEW.salary>:OLD.salary) THEN
   RAISE_APPLICATION_ERROR(-20002,'salariul nu poate fi marit');
   END IF;
END;
/
UPDATE employee
SET salary=salary+100;
```

1**2. Definiți un triggerde tip LDD. Declanșați trigger-ul.**

```
CREATE TABLE audit_ianc
(utilizator VARCHAR2(30),
nume_bd VARCHAR2(50),
eveniment VARCHAR(20),
nume_obiect VARCHAR(30),
data DATE);
/
CREATE OR REPLACE TRIGGER trigg12_ianc
   AFTER CREATE OR DROP OR ALTER ON SCHEMA
BEGIN
   INSERT INTO audit_ianc
   VALUES  (SYS.LOGIN_USER, SYS.DATABSE_NAME, SYS.SYSEVENT, SYS.DICTIONARY_OBJ_NAME,
SYSDATE);
END;

INSERT INTO project(project_id, project_name,project_start,project_deadline,team_id)
VALUES(1001, 'Cerberos',TO_DATE('2015-Jan-01','yy-mm-dd'),TO_DATE('2015-Apr-01','yy-mm-
dd'),103);
```

13.Definiți un pachet care să conțină toate obiectele definite în cadrul proiectului.



CREATE OR REPLACE PACKAGE pack_ex13

IS

PROCEDURE p9_anc

(c_nume company.company_name%TYPE);

FUNCTION EXE8_ANC

(teamname team.team_name%TYPE DEFAULT 'Exotica')

RETURN NUMBER;

FUNCTION EXE7_ANC

RETURN NUMBER;

FUNCTION EXE6_ANC

(nume_companie company.company_name%TYPE DEFAULT 'COM')

RETURN NUMBER;

END pack_ex13;

/

CREATE OR REPLACE PACKAGE BODY pack_ex13

IS

PROCEDURE p9_anc

(c_nume company.company_name%TYPE)

IS

cid company.company_id%TYPE;

-- select company_id into cid

-- from company

```
--          where company_name=c_nume;



TYPE tab_id IS TABLE OF project.project_id%TYPE;
TYPE tab_project_name IS TABLE OF project.project_name%TYPE;
TYPE tab_project_start IS TABLE OF project.project_start%TYPE;
TYPE tab_project_deadline IS TABLE OF project.project_deadline%TYPE;
TYPE tab_teamid IS TABLE OF project.team_id%TYPE;
      is_found_rec boolean := false;
      exception_name1 EXCEPTION;

t_id tab_id;
t_name tab_project_name;
t_s tab_project_start;
t_dl tab_project_deadline;
t_tid tab_teamid;

nrtask NUMBER;
teamname team.team_name%TYPE;



CURSOR c IS
SELECT *
FROM project p
where p.team_id in(
SELECT team_id
FROM member m
where m.employee_id in ( SELECT employee_id
FROM employee e
where e.company_id=(select company_id
                                    from company
                                    where company_name='COM')));
BEGIN
OPEN c;
FETCH c  BULK COLLECT INTO t_id,t_name,t_s ,t_dl,t_tid;
```

```
CLOSE c;


FOR i in t_id.FIRST..t_id.LAST LOOP
            is_found_rec := true;


            DBMS_OUTPUT.PUT_LINE('Nume Proiect : ' || t_name(i));
            DBMS_OUTPUT.PUT_LINE('Inceput in :' || t_s(i));
            DBMS_OUTPUT.PUT_LINE('Cu deadline-ul :' || t_s(i));


            select count(*) into nrtask
               from task t
               where t.project_id=t_id(i);
            SELECT te.team_name into teamname
            from team te
            where te.team_id=t_tid(i);
            DBMS_OUTPUT.PUT_LINE('Are ' || nrtask || ' la activ.');
            DBMS_OUTPUT.PUT_LINE('Apartine echipe ' || teamname);


            IF (i<t_id.LAST) THEN DBMS_OUTPUT.PUT_LINE('--------------------------------------------------');
            END IF;
      END LOOP;
       if not is_found_rec then
      RAISE exception_name1;
      end if;
   DBMS_OUTPUT.PUT_LINE('==================================================');
   EXCEPTION
    WHEN exception_name1 THEN
      RAISE_APPLICATION_ERROR(-20000,'Compania nu are nimic la activ!');
   WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Nu exista compania!');
END p9_anc;


FUNCTION EXE8_ANC
(teamname team.team_name%TYPE DEFAULT 'Exotica')
   RETURN NUMBER IS
```

```
TYPE tab_id IS TABLE OF task.task_id%TYPE;
TYPE tab_task_name IS TABLE OF task.task_name%TYPE;
TYPE tab_task_deadline IS TABLE OF task.task_deadline%TYPE;
TYPE tab_worker IS TABLE OF task.worker_id%TYPE;
TYPE tab_project_id IS TABLE OF task.project_id%TYPE;
    tNamet VARCHAR(200);
    teid NUMBER;
    is_found_rec boolean := false;

t_id tab_id;
t_name tab_task_name;
t_dl tab_task_deadline;
t_worker tab_worker;
t_pid tab_project_id;

total NUMBER:=0;
tid employee.employee_id%TYPE;
CURSOR c IS
SELECT *
FROM task t
where t.project_id in (
select project_id
from project p
where p.team_id in(select team_id
from team te
where te.team_name=teamname));
BEGIN
    SELECT team_id into teid
    from team
    where team_name=teamname;
OPEN c;
FETCH c  BULK COLLECT INTO t_id, t_name , t_dl ,t_worker,t_pid;
CLOSE c;

DBMS_OUTPUT.PUT_LINE('In cadrul echipe ' || teamname || ' aveti task-urile urmatoare : ');
```

```
DBMS_OUTPUT.PUT_LINE('==================================================');
FOR i in t_id.FIRST..t_id.LAST LOOP
        is_found_rec := true;

DBMS_OUTPUT.PUT_LINE('Nume task : ' || t_name(i));
DBMS_OUTPUT.PUT_LINE('Cu deadline-ul :' || t_dl(i));
       select lName into tNamet
          from employee
          where employee_id=t_worker(i);
DBMS_OUTPUT.PUT_LINE('Realizat de : ' || tNamet);

DBMS_OUTPUT.PUT_LINE('In cadrul proiectului cu id-ul: ' || t_pid(i));
       IF (i<t_id.LAST) THEN DBMS_OUTPUT.PUT_LINE('--------------------------------------------------');
       END IF;
END LOOP;

   DBMS_OUTPUT.PUT_LINE('==================================================');
    /* if not is_found_rec then
       RAISE exception_name1;  */
    end if

end if;
RETURN teid;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      RAISE_APPLICATION_ERROR(-20000,'Nu exista echipa!');
/*     WHEN exception_name1 THEN
      RAISE_APPLICATION_ERROR(-20000,'Compania nu are nimic la activ!'); */
   WHEN OTHERS THEN
            RAISE_APPLICATION_ERROR(-20001,'Alta eroare!');

END EXE8_ANC;

FUNCTION EXE7_ANC
()
```

```
   RETURN NUMBER IS
v_nr number(4);
      v_nume company.company_name%TYPE;
CURSOR c IS
         SELECT c.company_name, (select count(*)
                        from employee e
                        where e.company_id=c.company_id)
         from company c;

BEGIN
OPEN c;
   LOOP
      FETCH c in v_nume,v_nr;
      EXIT WHEN C%NOTFOUND;
      IF v_nr=0 THEN
      DBMS_OUTPUT.PUT_LINE('In cadrul companiei ' || v_nume || ' nu avem angajati');
      ELSE IF V_nr=1 THEN
               DBMS_OUTPUT.PUT_LINE('In cadrul companiei ' || v_nume || ' avem un angajat');
      ELSE
           DBMS_OUTPUT.PUT_LINE('In cadrul companiei ' || v_nume || ' avem ' || v_nr || ' angajati');



      END IF;
    END LOOP;

CLOSE c;
RETURN 0;
END EXE7_ANC;



FUNCTION EXE6_ANC
(nume_companie company.company_name%TYPE DEFAULT 'COM')
   RETURN NUMBER IS
TYPE tab_id IS TABLE OF employee.employee_id%TYPE;
TYPE tab_lName IS TABLE OF employee.lName%TYPE;
```

```
TYPE tab_fName IS TABLE OF employee.fName%TYPE;
TYPE tab_salary IS TABLE OF employee.salary%TYPE;
t_id tab_id;
t_lName tab_lName;
t_fName tab_fName;
t_salary tab_salary;
total NUMBER:=0;
CURSOR c IS
SELECT employee_id,lName,fName,salary
from employee
where company_id IN (select company_id
from company
where company_name=nume_companie);
BEGIN
OPEN c;
FETCH c  BULK COLLECT INTO t_id, t_lName, t_fName,t_salary;
CLOSE c;
DBMS_OUTPUT.PUT_LINE('In cadrul ' || nume_companie || ' aveti angajatii : ');
DBMS_OUTPUT.PUT_LINE('==================================================');
FOR i in t_id.FIRST..t_id.LAST LOOP
DBMS_OUTPUT.PUT_LINE('Angajatul :' || t_lName(i) || ' ' || t_fName(i));
      DBMS_OUTPUT.PUT_LINE('Salariu :' || t_salary(i));
      IF (i<t_id.LAST) THEN DBMS_OUTPUT.PUT_LINE('--------------------------------------------------');
      END IF;
total:=total+t_salary(i);
END LOOP;
   DBMS_OUTPUT.PUT_LINE('==================================================');
DBMS_OUTPUT.PUT_LINE('Total de plata pe luna : ' || total);
RETURN 0;
END EXE6_ANC;



END pack_ex13;
/
```