

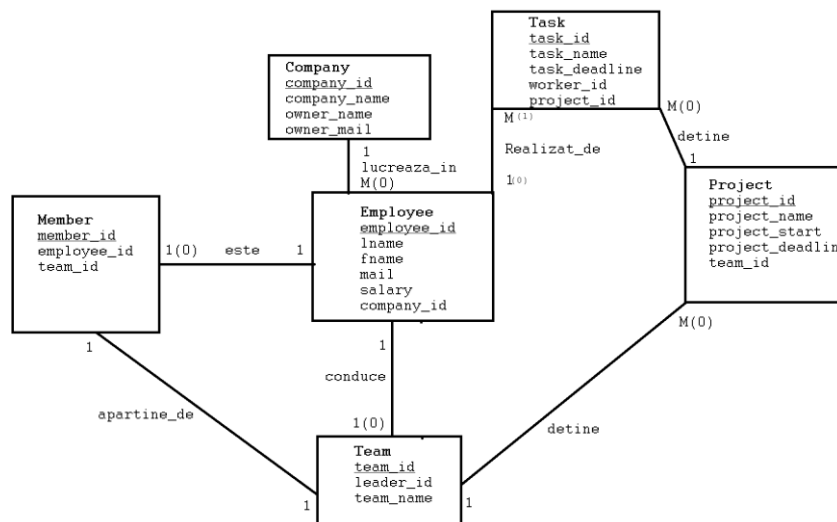
1. Prezențați pe scurt baza de date (utilitatea ei).

Baza de date poate fi folosită pentru persoanele care dețin mai multe companii de IT, fiecare cu proiecte separate, între care pot exista legături (exemplu: Angajatul de la compania X participă în cadrul unui proiect de la compania Y pentru a ajuta cu un task). Această bază de date este adresată persoanelor care doresc să-și ușureze afacerea și să poată obține un workflow mai bun în viața de zi cu zi. Baza de date este formată din: company, employee, members, teams, projects și task, fiecare dintre acestea oferind detinatorului posibilitatea de a vedea task-urile pe care le are și pe care le-a avut, angajații din cadrul companiei, dar și angajați care participă în cadrul proiectelor.

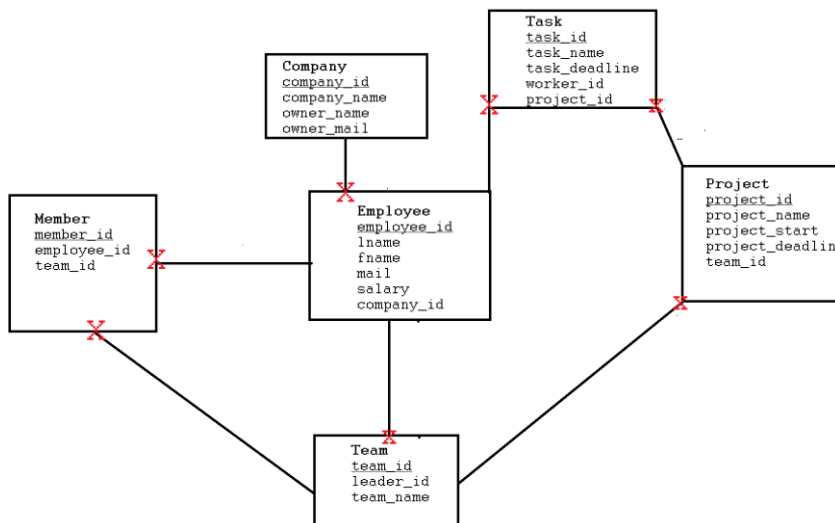
-De ce să folosești această bază de date și nu alta?

Pentru că aceasta îți oferă posibilitatea să te mulezi pe nevoile tale, dacă ai câteva firme și nu le poți gestiona eficient, aceasta poate să te ajute prin diferite cai, oferindu-ți posibilitatea de a alege cine unde lucrează și unde este nevoie de cineva pentru a face mai eficient procesul de creare a diferitelor proiecte.

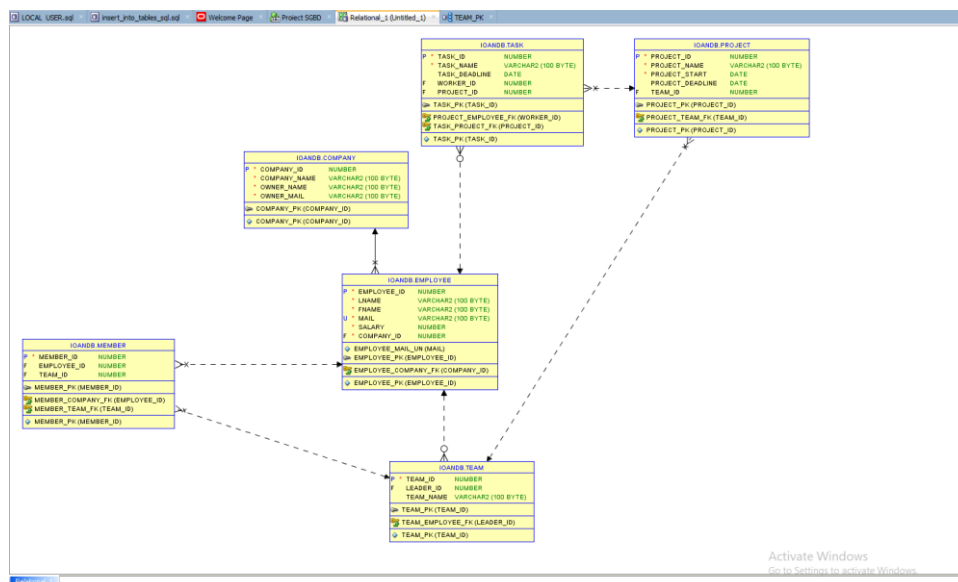
2. Realizați diagrama entitate-relație (ERD).



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,'Iagar');
```

```
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,'SubIagar');
```

```
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,'shoppingcartproblem',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 x				
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 BMWsmallld	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 x						
SQL All Rows Fetched: 11 in 0.083 seconds						
EMPLOYEE_ID	LNNAME	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 x

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 x

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 x

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 x

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;
```

```
3 CREATE OR REPLACE FUNCTION EXE6_ANC
4 (p_nume_companie company.company_name%TYPE DEFAULT 'COM')
5 RETURN NUMBER IS
6     TYPE tab_id IS TABLE OF employee.employee_id%TYPE;
7     TYPE tab_lName IS TABLE OF employee.lName%TYPE;
8     TYPE tab_fName IS TABLE OF employee.fName%TYPE;
9     TYPE tab_salary IS TABLE OF employee.salary%TYPE;
10    t_id tab_id;
11    t_lName tab_lName;
12    t_fName tab_fName;
13    t_salary tab_salary;
14    total NUMBER:=0;
15    CURSOR c IS
16        SELECT employee_id,lName,fName,salary
17        FROM employee;
18
19 BEGIN
20     FOR i IN c LOOP
21         t_id:=i.employee_id;
22         t_lName:=i.lName;
23         t_fName:=i.fName;
24         t_salary:=i.salary;
25         total:=total+t_salary;
26     END LOOP;
27     DBMS_OUTPUT.PUT_LINE('Total de plata pe luna : ' || total);
28     RETURN 0;
29 END EXE6_ANC;
```

Script Output X

Task completed in 0.185 seconds

Function EXE6_ANC compiled

DBMS Output

LOCAL USER X

In cadrul COM aveti angajatii :

Angajatul :Gabrie Oles

Salariu :2333

Angajatul :Maghere Ovidiu

Salariu :2333

Statements - Log

Connection Name	Elapsed	SQL
1		SELECT object_type,owner,owner,object_name,null column_name,null column_id,null data
1		SELECT object_type,owner,owner,object_name,null column_name,null column_id,null data
6		SELECT object_type,owner,owner,object_name,null column_name,null column_id,null data
1		BEGIN DBMS_OUTPUT.PUT_LINE(p9_anc('GAZUROC'))END;
2		BEGIN DBMS_OUTPUT.PUT_LINE(p9_anc('GAZUROC'))END;
1		SELECT LINE, POSITION, REPLACE(TEXT, CHR(10), ' '), attribute FROM SYS.ALL_ERRORS A
1		select UPPER(sys_context('USERENV', 'CURRENT_SCHEMA')) from dual

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

```
1 CREATE OR REPLACE FUNCTION EXE7_ANC
2 RETURN NUMBER IS
3     v_nr number(4);
4     v_nume company.company_name%TYPE;
5     CURSOR c IS
6         SELECT c.company_name, (select count(*)
7                                 from employee e
8                                 where e.company_id=c.company_id)
9         from company c;
10
11 BEGIN
```

Worksheet Query Builder

```

4      v_nume company.company_name%TYPE;
5      CURSOR c IS
6          SELECT c.company_name, (select count(*)
7              from employee e
8              where e.company_id=c.company_id)
9          from company c;
10
11 BEGIN
12     OPEN c;
13     LOOP
14         FETCH c INTO v_nume,v_nr;
15         EXIT WHEN C%NOTFOUND;
16         IF v_nr=0 THEN
17             DBMS_OUTPUT.PUT_LINE('In cadrul companiei ' || v_nume || ' nu avem angajati');
18         ELSIF v_nr=1 THEN
19             DBMS_OUTPUT.PUT_LINE('In cadrul companiei ' || v_nume || ' avem un angajat');
20         ELSE
21             DBMS_OUTPUT.PUT_LINE('In cadrul companiei ' || v_nume || ' avem ' || v_nr || ' angajati');
22         END IF;
23     END LOOP;
24     CLOSE c;
25
26
27

```

Script Output x

Task completed in 0.086 seconds

25/10 PLS-00103: Encountered the symbol "LOOP" when expecting one of the following: if
Errors: check compiler log

Function EXE7_ANC compiled

PL/SQL procedure successfully completed.

Dbms Output x

Buffer Size: 20000

LOCAL USER x

```

In cadrul companiei ALCATRA avem 3 angajati
In cadrul companiei GAZURO avem 2 angajati
In cadrul companiei BMWsmallld avem 2 angajati
In cadrul companiei LAGAR avem 2 angajati
In cadrul companiei COM avem 2 angajati
In cadrul companiei test nu avem angajati

```

Compiler - Log

Messages Log

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 subprogramul astfel încât să evidențiați toate cazurile tratate.

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

```

62
63 BEGIN
64 DBMS_OUTPUT.PUT_LINE(EXE8_ANC('asdasd'));
65 END;
66

```

Script Output x

Task completed in 0.034 seconds

```

END;
Error report -
ORA-20000: Nu exista echipa!
ORA-06512: at "IOANDB.EXE8_ANC", line 55
ORA-06512: at line 2
20000. 00000 - "%s"
*Cause:      The stored procedure 'raise_application_error'
              was called which causes this error to be generated.
*Action:     Correct the problem as described in the error message or contact
              the application administrator or DBA for more information.

```

2. Second RUN=rulam cu echipa default

60 /

61

62

63 BEGIN

64 DBMS_OUTPUT.PUT_LINE(EXE8 Anc());

65 END;

66

Script Output x

Task completed in 0.022 seconds

ORA-06512: at "IOANDB.EXE8 Anc", line 55

Dbms Output

Buffer Size: 20000

LOCAL USER x

Realizat de : Samuel

In cadrul proiectului cu id-ul: 1023

Nume task : DataLeak

Cu deadline-ul :04-MAR-15

Realizat de : Inglesia

In cadrul proiectului cu id-ul: 1023

Nume task : Mobile

Cu deadline-ul :04-MAR-15

Realizat de : Samuel

In cadrul proiectului cu id-ul: 1023

Nume task : CurrencyProb

Cu deadline-ul :04-APR-18

Realizat de : Crinu

In cadrul proiectului cu id-ul: 1041

Nume task : BackEndBugSolver

Cu deadline-ul :04-MAR-19

Realizat de : Inglesia

In cadrul proiectului cu id-ul: 1041

=====

101

Statements - Log

Q	Connection Name	Elapsed	SQL
0			select UPPER(sys_context('USERENV', 'CURRENT_SCHEMA')) from dual
74			CREATE OR REPLACE PROCEDURE p9_anc(c_name company.company_name%TYPE)IS
2			select SCHEMA object_type, OWNER owner, OWNER object_name from all_users where
183			select object_type,owner object_name,rank from (select object_type,owner,object_na
1			select UPPER(sys_context('USERENV', 'CURRENT_SCHEMA')) from dual
40			BEGIN p9_anc('GAZUROT');END;
1			SELECT object_type, owner owner, object_name, null column_name, null column_id,
1			SELECT object_type, owner owner, object_name, null column_name, null column_id,
1			SELECT object_type, owner owner, object_name, null column_name, null column_id,
6			SELECT object_type, owner owner, object_name, null column_name, null column_id,
1			BEGIN DBMS_OUTPUT.PUT_LINE(p9_anc('GAZUROT'));END;
2			BEGIN DBMS_OUTPUT.PUT_LINE(p9_anc('GAZUROT'));END;
1			SELECT LINE, POSITION, REPLACE(TEXT, CHR(10), ' '), attribute FROM SYS.ALL_ERR
1			SELECT LINE, POSITION, REPLACE(TEXT, CHR(10), ' '), attribute FROM SYS.ALL_ERR
75			CREATE OR REPLACE PROCEDURE p9_anc(c_name company.company_name%TYPE)IS
1			begin SYS.DBMS_UTILITY.NAME_RESOLVE(?,?,?,?,?,?); end;
1			SELECT 'COLUMN' type, owner, table_name object_name, column_name, column_id, data
2			SELECT * FROM project p WHERE p.team_id IN (SELECT team_id FROM member m WHERE m.en
6			SELECT * FROM project p WHERE p.team_id IN (SELECT team_id FROM member m WHERE m.en
1			SELECT LINE, POSITION, REPLACE(TEXT, CHR(10), ' '), attribute FROM SYS.ALL_ERR
1			select UPPER(sys_context('USERENV', 'CURRENT_SCHEMA')) from dual
76			CREATE OR REPLACE PROCEDURE p9_anc(c_name company.company_name%TYPE)IS
1			select * from project
2			BEGIN DBMS_OUTPUT.PUT_LINE(p9_anc('GAZUROT'));END;

3.Third Run=rumam cu o echipa data.

62

63 BEGIN

64 DBMS_OUTPUT.PUT_LINE(EXE8 Anc('Exotica'));

65 END;

66

Script Output x

Task completed in 0.022 seconds

Dbms Output

Buffer Size: 20000

LOCAL USER x

Nume task : DataLeak

Cu deadline-ul :04-MAR-15

Realizat de : Inglesia

In cadrul proiectului cu id-ul: 1023

Nume task : Mobile

Cu deadline-ul :04-MAR-15

Realizat de : Samuel

In cadrul proiectului cu id-ul: 1023

Nume task : CurrencyProb

Cu deadline-ul :04-APR-18

Realizat de : Crinu

In cadrul proiectului cu id-ul: 1041

Nume task : BackEndBugSolver

Cu deadline-ul :04-MAR-19

Realizat de : Inglesia

In cadrul proiectului cu id-ul: 1041

=====

101

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-uri 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 subprogramul astfel 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;
```

```

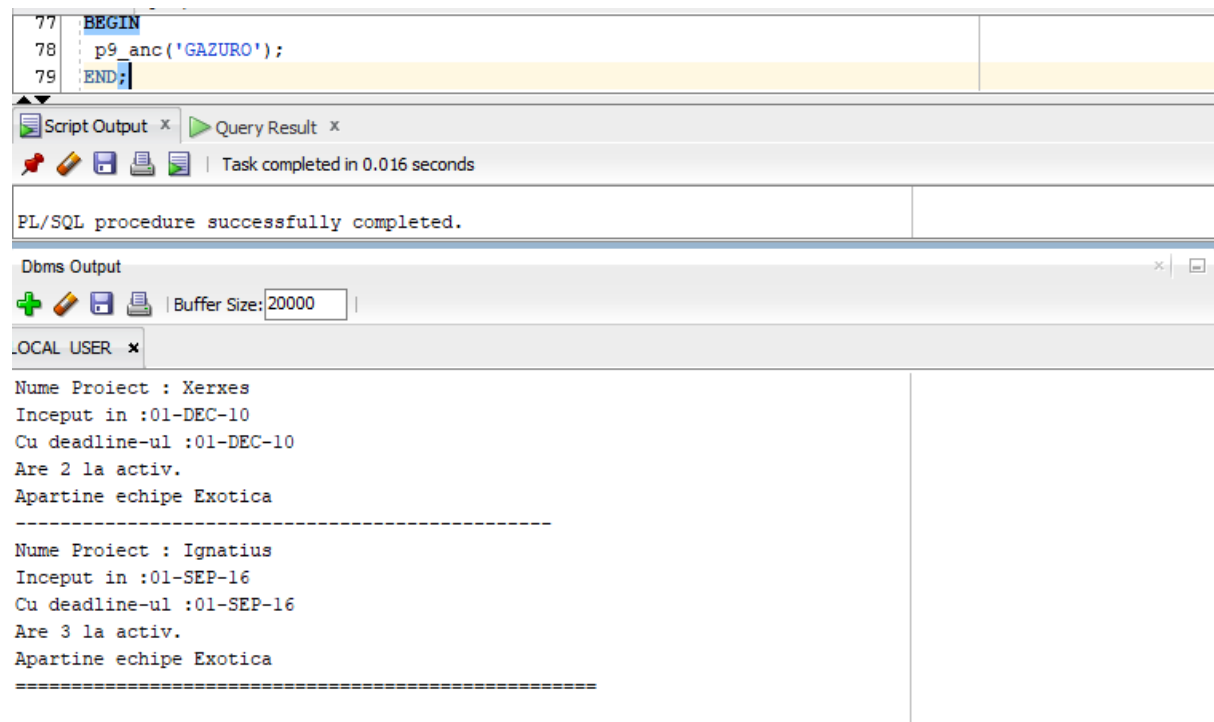
BEGIN

p9_anc('COM');

END;

```

1.Cand rulam fara erori



The screenshot shows the SQL Developer interface. The script editor at the top contains the following code:

```

77 BEGIN
78   p9_anc('GAZURO');
79 END;

```

Below the script editor, the 'Script Output' tab is active, displaying the message: 'PL/SQL procedure successfully completed.' The status bar indicates 'Task completed in 0.016 seconds'.

The 'Dbms Output' tab is also visible, showing the output of the procedure:

```

Nume Proiect : Xerxes
Inceput in :01-DEC-10
Cu deadline-ul :01-DEC-10
Are 2 la activ.
Apartine echipe Exotica
-----
Nume Proiect : Ignatius
Inceput in :01-SEP-16
Cu deadline-ul :01-SEP-16
Are 3 la activ.
Apartine echipe Exotica
=====

```

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 companie 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 trigger de tip LMD la nivel de comandă. Declanșați trigger-ul.

The screenshot shows the SQL Developer interface. The top pane displays the SQL script for creating a trigger and inserting data. The bottom pane shows the script output, indicating that the trigger was compiled successfully but an error occurred during the execution of the INSERT statement.

```

Worksheet Query Builder
1 CREATE OR REPLACE TRIGGER trig10
2 BEFORE INSERT OR UPDATE OR DELETE ON company
3 BEGIN
4 IF (TO_CHAR(SYSDATE,'D')=2)
5 OR (TO_CHAR(SYSDATE,'HH24') NOT BETWEEN 1 AND 8)
6 THEN
7 RAISE_APPLICATION_ERROR(-20001,'tabelul nu poate fi actualizat');
8 END IF;
9 END;
10
11 /
12 INSERT INTO company( company_id, company_name, owner_name, owner_mail) VALUES (6,'test', 'Andreea Ancuta', 'lbae@gmail.com');
13
Script Output x
Task completed in 0.073 seconds
Trigger TRIG10 compiled
Error starting at line : 12 in command -
INSERT INTO company( company_id, company_name, owner_name, owner_mail) VALUES (6,'test', 'Andreea Ancuta', 'lbae@gmail.com')
Error report -
ORA-20001: tabelul nu poate fi actualizat
ORA-06512: at "IOANDB.TRIG10", line 5
ORA-04088: error during execution of trigger 'IOANDB.TRIG10'

```

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.

The screenshot shows the Oracle SQL Developer interface. The top pane, titled 'Query Builder', contains the following SQL code:

```
1 CREATE OR REPLACE TRIGGER trig11
2   BEFORE UPDATE OF salary on employee
3   FOR EACH ROW
4 BEGIN
5   IF (:NEW.salary > :OLD.salary) THEN
6     RAISE_APPLICATION_ERROR(-20002, 'salariul nu poate fi marit');
7   END IF;
8 END;
9 /
10 UPDATE employee
11 SET salary=salary+100;
```

The bottom pane, titled 'Script Output', shows the execution results:

```
Trigger TRIG11 compiled

Error starting at line : 10 in command -
UPDATE employee
SET salary=salary+100
Error report -
ORA-20002: salariul nu poate fi marit
ORA-06512: at "IOANDB.TRIG11", line 3
ORA-04088: error during execution of trigger 'IOANDB.TRIG11'
```

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;

12. Definiți un trigger de 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.DATABASE_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.

```
worksheet Query Builder
1 CREATE OR REPLACE PACKAGE pack_ex13
2 IS
3 PROCEDURE p9_anc
4 (c_nume company.company_name%TYPE);
5 FUNCTION EXE8_ANC
6 (teamname team.team_name%TYPE DEFAULT 'Exotica')
7 RETURN NUMBER;
8 FUNCTION EXE7_ANC
9 RETURN NUMBER;
10 FUNCTION EXE6_ANC
11 (nume_comanie company.company_name%TYPE DEFAULT 'COM')
12 RETURN NUMBER;
13 END pack_ex13;
14
15 /
16 CREATE OR REPLACE PACKAGE BODY pack_ex13
17 IS
18 PROCEDURE p9_anc
19 (c_nume company.company_name%TYPE)
20 IS
21 cid company.company_id%TYPE;
22 -- select company_id into cid
23 -- from company
24 -- where company_name=c_nume;
```

Script Output x

Task completed in 0.053 seconds

Package PACK_EX13 compiled

Package Body PACK_EX13 compiled

```
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_comanie 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_num;

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-uri 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;

/

```