Tema 8

create sequence seq\_tema;

create or replace package pachet\_a is

-- functie care obtine id ul departamentului pe baza numelui

function get\_department\_id(p\_department\_name in departments.department\_name%type)

return departments.department\_id%type;

-- functie care obtine id ul jobului pe baza numelui

function get\_job\_id(p\_job\_title in jobs.job\_title%type)

return jobs.job\_id%type;

-- functie care obtine id ul managerului pe baza numelui si prenumelui

function get\_manager\_id(p\_first\_name in employees.first\_name%type, p\_last\_name in employees.last\_name%type)

return employees.employee\_id%type;

-- functie care obtine cel mai mic salariu dintr un departament pentru un anumit job

function get\_min\_salary(p\_department\_id in departments.department\_id%type, p\_job\_id in jobs.job\_id%type)

return employees.salary%type;

-- procedura pentru adaugarea unui angajat

procedure add\_employee(p\_first\_name in employees.first\_name%type,

p\_last\_name in employees.last\_name%type,

p\_phone\_number in employees.phone\_number%type,

p\_email in employees.email%type,

p\_department\_name in departments.department\_name%type,

p\_job\_title in jobs.job\_title%type,

p\_manager\_first\_name in employees.first\_name%type,

p\_manager\_last\_name in employees.last\_name%type);

end pachet\_a;

/

create or replace package body pachet\_a is

function get\_department\_id(p\_department\_name in departments.department\_name%type)

return departments.department\_id%type is

v\_department\_id departments.department\_id%type;

begin

select department\_id

into v\_department\_id

from departments

where department\_name = p\_department\_name;

return v\_department\_id;

exception

when no\_data\_found then

raise\_application\_error(-20001, 'departamentul nu exista');

end get\_department\_id;

function get\_job\_id(p\_job\_title in jobs.job\_title%type)

return jobs.job\_id%type is

v\_job\_id jobs.job\_id%type;

begin

select job\_id

into v\_job\_id

from jobs

where job\_title = p\_job\_title;

return v\_job\_id;

exception

when no\_data\_found then

raise\_application\_error(-20002, 'jobul nu exista');

end get\_job\_id;

function get\_manager\_id(p\_first\_name in employees.first\_name%type, p\_last\_name in employees.last\_name%type)

return employees.employee\_id%type is

v\_manager\_id employees.employee\_id%type;

begin

select employee\_id

into v\_manager\_id

from employees

where first\_name = p\_first\_name and last\_name = p\_last\_name;

return v\_manager\_id;

exception

when no\_data\_found then

raise\_application\_error(-20003, 'managerul nu exista');

end get\_manager\_id;

function get\_min\_salary(p\_department\_id in departments.department\_id%type, p\_job\_id in jobs.job\_id%type)

return employees.salary%type is

v\_min\_salary employees.salary%type;

begin

select min(salary)

into v\_min\_salary

from employees

where department\_id = p\_department\_id and job\_id = p\_job\_id;

return v\_min\_salary;

exception

when no\_data\_found then

raise\_application\_error(-20004, 'nu exista salariu pentru departament si job');

end get\_min\_salary;

procedure add\_employee(p\_first\_name in employees.first\_name%type,

p\_last\_name in employees.last\_name%type,

p\_phone\_number in employees.phone\_number%type,

p\_email in employees.email%type,

p\_department\_name in departments.department\_name%type,

p\_job\_title in jobs.job\_title%type,

p\_manager\_first\_name in employees.first\_name%type,

p\_manager\_last\_name in employees.last\_name%type) is

v\_department\_id departments.department\_id%type;

v\_job\_id jobs.job\_id%type;

v\_manager\_id employees.employee\_id%type;

v\_min\_salary employees.salary%type;

v\_new\_employee\_id employees.employee\_id%type;

begin

v\_department\_id := get\_department\_id(p\_department\_name);

v\_job\_id := get\_job\_id(p\_job\_title);

v\_manager\_id := get\_manager\_id(p\_manager\_first\_name, p\_manager\_last\_name);

v\_min\_salary := get\_min\_salary(v\_department\_id, v\_job\_id);

select seq\_tema.nextval into v\_new\_employee\_id from dual;

insert into employees (employee\_id, first\_name, last\_name, phone\_number, email,

hire\_date, job\_id, salary, manager\_id, department\_id, commission\_pct)

values (v\_new\_employee\_id, p\_first\_name, p\_last\_name, p\_phone\_number, p\_email,

sysdate, v\_job\_id, v\_min\_salary, v\_manager\_id, v\_department\_id, null);

dbms\_output.put\_line('Angajatul a fost adaugat cu succes. id: ' || v\_new\_employee\_id);

exception

when others then

raise\_application\_error(-20005, 'eroare la adaugarea angajatului');

end add\_employee;

end pachet\_a;

/

