**EXERCIȚII NEREZOLVATE LABORATOARE PLSQL**

--LABORATOR 1

--E2

--a.

select ziua, (select count(\*)

from rental

where extract(month from book\_date) = 10 and

extract (day from book\_date) = ziua) as numar

from

(select level ziua

from dual

connect by level <=31);

--b.

drop table octombrie\_prof;

create table octombrie\_prof

(

id number(4),

data date

);

declare

contor number(3) := 0;

data\_aux date := '01-OCT-2023';

begin

loop

insert into octombrie\_prof

values (contor, data\_aux);

contor := contor+1;

exit when contor = 31;

end loop;

end;

/

select \* from octombrie\_prof;

--E3

DECLARE

nume varchar(255) := '&input';

nrCopii number(8);

aux number(5);

BEGIN

select count(\*)

into aux

from member

where last\_name = nume;

IF aux = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Nimeni cu numele asta');

ELSIF aux > 1 THEN

DBMS\_OUTPUT.PUT\_LINE('Mai multi cu acest nume');

ELSE

SELECT count(nume)

INTO nrCopii

FROM member m ,rental r

WHERE nume = m.last\_name

and r.member\_id = m.member\_id;

DBMS\_OUTPUT.PUT\_LINE('Nr de copii ' || nrCopii);

END IF;

END;

/

--E4

DECLARE

nume varchar(255) := '&input';

nrCopii number(8);

aux number(5);

nrTitluri number(5);

BEGIN

select count(\*)

into nrTitluri

from title;

select count(\*)

into aux

from member

where last\_name = nume;

IF aux = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Nimeni cu numele asta');

ELSIF aux > 1 THEN

DBMS\_OUTPUT.PUT\_LINE('Mai multi cu acest nume');

ELSE

SELECT count(m.member\_id)

INTO nrCopii

FROM member m ,rental r

WHERE nume = m.last\_name

and r.member\_id = m.member\_id;

DBMS\_OUTPUT.PUT\_LINE('Nr de copii '|| nrCopii);

IF nrCopii >= 3 \* nrTitluri / 4 THEN

DBMS\_OUTPUT.PUT\_LINE('Categoria 1');

ELSIF nrCopii >= nrTitluri / 2 THEN

DBMS\_OUTPUT.PUT\_LINE('Categoria 2');

ELSIF nrCopii >= nrTitluri / 4 THEN

DBMS\_OUTPUT.PUT\_LINE('Categoria 3');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Categoria 4');

END IF;

END IF;

END;

/

--E5

create table member\_aho

as select \* from member;

alter table member\_aho

add discount number(3);

declare

v\_id\_membru member\_aho.member\_id%type := &v\_id\_membru;

v\_nr\_filme number(3);

v\_nr\_total\_filme number(3);

v\_discount member\_aho.discount%type;

begin

select 1

into v\_nr\_filme

from member\_aho

where member\_id = v\_id\_membru;

begin

select count(\*)

into v\_nr\_total\_filme

from title;

select count(distinct title\_id)

into v\_nr\_filme

from rental

where member\_id = v\_id\_membru;

dbms\_output.put\_line('Nr filme imprumutate = ' || v\_nr\_filme);

case when v\_nr\_filme > 25/100 \* v\_nr\_total\_filme

then case when v\_nr\_filme > 75/100 \* v\_nr\_total\_filme

then v\_discount := 10;

when v\_nr\_filme > 50/100 \* v\_nr\_total\_filme

then v\_discount := 5;

when v\_nr\_filme > 25/100 \* v\_nr\_total\_filme

then v\_discount := 3;

end case;

update member\_aho

set discount = v\_discount

where member\_id = v\_id\_membru;

dbms\_output.put\_line('Actualizare reusita');

else dbms\_output.put\_line('Membrul nu se incadreaza in nicio categorie pt discount');

end case;

end;

exception when no\_data\_found then dbms\_output.put\_line('Nu exista un membru cu acest id');

end;

/

--LABORATOR 2

--E2

--a

CREATE TYPE tip\_oras AS TABLE OF VARCHAR2(20);

/

DROP TYPE tip\_oras;

DROP TABLE excursie;

select \* from excursie;

CREATE TABLE excursie(

cod\_excursie NUMBER(4),

denumire VARCHAR2(20),

orase tip\_oras,

status VARCHAR2(11),

CHECK( status = 'disponibila' or status = 'anulata')

)

NESTED TABLE orase STORE AS table\_orase;

insert into excursie

values(1,'Excursie1',tip\_oras('Brasov','Sinaia','Campina'),'disponibila');

insert into excursie

values(2,'Excursie2',tip\_oras('Braila','Galati'),'anulata');

insert into excursie

values(3,'Excursie3',tip\_oras('Cluj','Sibiu'),'disponibila');

insert into excursie

values(4,'Excursie4',tip\_oras('Craiova','Targu Jiu'),'anulata');

insert into excursie

values(5,'Excursie5',tip\_oras('Bucuresti'),'disponibila');

SELECT \* FROM EXCURSIE;

--b

DECLARE

v\_cod\_excursie excursie.cod\_excursie%TYPE := &input\_cod\_excursie;

--v\_oras\_nou VARCHAR2(30) := &input\_oras\_nou; --cand adaugi un oras

v\_orase excursie.orase%TYPE;

v\_orase\_aux excursie.orase%TYPE := tip\_oras();

--v\_oras1 VARCHAR2(20) := &input\_oras\_1; --cand dai swap la 2orase

--v\_oras2 VARCHAR2(30) := &input\_oras\_2; --cand dai swap la 2orase

v\_index1 NUMBER(3); --cand dai swap la 2orase

v\_index2 NUMBER(3); --cand dai swap la 2orase

v\_aux VARCHAR2(30); --cand dai swap la 2orase

--v\_oras\_out VARCHAR2(30) := &input\_oras\_out; --cand stergi un oras

BEGIN

select orase

into v\_orase

from excursie

where cod\_excursie = v\_cod\_excursie;

-- --ULTIMUL ORAS

-- v\_orase.extend();

-- v\_orase(v\_orase.count()) := v\_oras\_nou;

-- v\_orase\_aux := v\_orase\_aux;

-- --AL 2 LEA ORAS

-- IF v\_orase.count() > 0 THEN

-- v\_orase\_aux.extend();

-- v\_orase\_aux(1) := v\_orase(1);

-- v\_orase\_aux.extend();

-- v\_orase\_aux(2) := v\_oras\_nou;

-- FOR i IN 2..v\_orase.count() LOOP

-- v\_orase\_aux.extend();

-- v\_orase\_aux(i+1) := v\_orase(i);

-- END LOOP;

-- ELSE

-- v\_orase\_aux(1) := v\_oras\_nou;

-- DBMS\_OUTPUT.PUT\_LINE('E singuru oras idk,l am pus singur,n are cum sa fie al2lea');

-- END IF;

--SWAP LA 2 ORASE

-- FOR i IN 1..v\_orase.count() LOOP

--

-- IF v\_orase(i) = v\_oras1

-- THEN v\_index1 := i;

-- END IF;

-- IF v\_orase(i) = v\_oras2

-- THEN v\_index2 := i;

-- END IF;

--

-- END LOOP;

--

-- IF v\_index1 is null or v\_index2 is null or v\_orase.count() < 2

-- THEN DBMS\_OUTPUT.PUT\_LINE('O valoare e gresita');

-- ELSE

-- v\_orase\_aux := v\_orase;

-- v\_aux := v\_orase\_aux(v\_index1);

-- v\_orase\_aux(v\_index1) := v\_orase\_aux(v\_index2);

-- v\_orase\_aux(v\_index2) := v\_aux;

-- END IF;

-- --DELETE LA UN ORAS

-- FOR i in 1..v\_orase.count() LOOP

-- IF v\_orase(i) <> v\_oras\_out THEN

-- v\_orase\_aux.extend();

-- v\_orase\_aux(v\_orase\_aux.count()) := v\_orase(i);

-- END IF;

-- END LOOP;

UPDATE excursie

SET orase = v\_orase\_aux

where cod\_excursie = v\_cod\_excursie;

END;

/

select \* from excursie;

--c

DECLARE

v\_cod\_excursie excursie.cod\_excursie%TYPE := &input\_cod\_excursie;

v\_orase excursie.orase%TYPE;

BEGIN

select orase

into v\_orase

from excursie

where cod\_excursie = v\_cod\_excursie;

DBMS\_OUTPUT.PUT\_LINE('Nr excursii: '||v\_orase.count());

FOR i in 1..v\_orase.count() LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_orase(i));

END LOOP;

END;

/

--d

DECLARE

TYPE lista\_cod\_excursii IS TABLE OF excursie.cod\_excursie%TYPE;

v\_orase excursie.orase%TYPE;

v\_excursii lista\_cod\_excursii := lista\_cod\_excursii();

v\_i excursie.cod\_excursie%TYPE;

BEGIN

FOR i in (select cod\_excursie from excursie) loop

v\_excursii.extend();

v\_excursii(v\_excursii.count()) := i.cod\_excursie;

END LOOP;

FOR i in 1..v\_excursii.count() LOOP

v\_orase := tip\_oras();

select orase

into v\_orase

from excursie

where v\_excursii(i) = cod\_excursie;

DBMS\_OUTPUT.PUT\_LINE('Nr excursii: '||v\_orase.count());

FOR i in 1..v\_orase.count() LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_orase(i));

END LOOP;

END LOOP;

END;

/

--e

DECLARE

TYPE lista\_cod\_excursii IS TABLE OF excursie.cod\_excursie%TYPE;

v\_excursii lista\_cod\_excursii := lista\_cod\_excursii();

v\_orase excursie.orase%TYPE;

v\_min number(3) := 999;

BEGIN

FOR i in (select cod\_excursie from excursie) loop

v\_excursii.extend();

v\_excursii(v\_excursii.count()) := i.cod\_excursie;

END LOOP;

FOR i in 1..v\_excursii.count() LOOP

v\_orase := tip\_oras();

select orase

into v\_orase

from excursie

where v\_excursii(i) = cod\_excursie;

IF v\_orase.count() < v\_min

THEN v\_min := v\_orase.count();

END IF;

END LOOP;

FOR i in 1..v\_excursii.count() LOOP

select orase

into v\_orase

from excursie

where v\_excursii(i) = cod\_excursie;

IF v\_orase.count() = v\_min

THEN

DELETE

FROM excursie

WHERE cod\_excursie = v\_excursii(i);

END IF;

END LOOP;

END;

/

--E3

create or replace type tip\_orase\_ami is varray(10) of varchar2(20);  
  
create table excursie\_ami (  
 cod\_excursie number(4) primary key ,  
 denumire varchar2(20),  
 orase tip\_orase\_ami,  
 status varchar2(20) default 'disponibila' check (*lower*(status) in ('disponibila','anulata'))  
);  
  
insert all  
 into excursie\_ami values (1,'paradis',tip\_orase\_ami('napoli','capri','positano'),'disponibila')  
 into excursie\_ami values (2,'vis',tip\_orase\_ami('paris','lyon'),'anulata')  
 into excursie\_ami values (3,'munte',tip\_orase\_ami('sinaia','busteni','predeal','brasov'),'disponibila')  
 into excursie\_ami values (4,'mare',tip\_orase\_ami('constinesti'),'disponibila')  
 into excursie\_ami values (5,'excelent',tip\_orase\_ami('ploiesti','bucuresti'),'anulata')  
select 'worked' from dual;  
  
declare  
 v\_excursie\_id excursie\_ami.cod\_excursie%type := 1;  
 v\_orase excursie\_ami.orase%type;  
 v\_nume\_oras\_1 varchar2(20) := 'napoli';  
 v\_nume\_oras\_2 varchar2(20) := 'positano';  
 i number(4);  
 index1 number(4);  
 index2 number(4);  
 tempSwap varchar2(20);  
begin  
 select orase  
 into v\_orase  
 from excursie\_ami  
 where cod\_excursie = v\_excursie\_id;  
  
 v\_orase.*extend*();  
 v\_orase(v\_orase.last) := 'roma';  
  
 for i in v\_orase.first..v\_orase.last loop  
 DBMS\_OUTPUT.*PUT*(v\_orase(i) || ' ');  
 end loop;  
  
 DBMS\_OUTPUT.*NEW\_LINE*();  
  
 v\_orase.*extend*();  
 i := v\_orase.last;  
 while i > 2 loop  
 v\_orase(i) := v\_orase(i-1);  
 i := i-1;  
 end loop;  
 v\_orase(2) := 'salermo';  
  
 for i in v\_orase.first..v\_orase.last loop  
 DBMS\_OUTPUT.*PUT*(v\_orase(i) || ' ');  
 end loop;  
  
 DBMS\_OUTPUT.*NEW\_LINE*();  
  
 for i in v\_orase.first..v\_orase.last loop  
 if v\_orase(i) = v\_nume\_oras\_1 then index1 := i;  
 end if;  
 if v\_orase(i) = v\_nume\_oras\_2 then index2 := i;  
 end if;  
 end loop;  
  
 tempSwap := v\_orase(index1);  
 v\_orase(index1) := v\_orase(index2);  
 v\_orase(index2) := tempSwap;  
  
 for i in v\_orase.first..v\_orase.last loop  
 DBMS\_OUTPUT.*PUT*(v\_orase(i) || ' ');  
 end loop;  
  
 DBMS\_OUTPUT.*NEW\_LINE*();  
  
 DBMS\_OUTPUT.*PUT*('Excursie ' || v\_excursie\_id || ', Nr. Orase ' || v\_orase.COUNT || ': ');  
 for i in v\_orase.first..v\_orase.last loop  
 DBMS\_OUTPUT.*PUT*(v\_orase(i) || ' ');  
 end loop;  
  
 DBMS\_OUTPUT.*NEW\_LINE*();  
end;  
/

--LABORATOR 3

--E2

declare

v\_nr\_ang number(3) := 0;

v\_val\_lunara number(30, 2) := 0;

v\_val\_medie employees.salary%type := 0;

v\_nr\_ang2 number(3) := 0;

v\_val\_lunara2 number(30, 2) := 0;

v\_val\_medie2 employees.salary%type := 0;

begin

for v\_job in (select job\_id, job\_title

from jobs)

loop

dbms\_output.put\_line('Job-ul ' || v\_job.job\_title || ': ');

v\_nr\_ang := 0;

v\_val\_lunara := 0;

v\_val\_medie := 0;

for v\_emp in (select last\_name, salary

from employees

where job\_id = v\_job.job\_id)

loop

dbms\_output.put\_line(v\_emp.last\_name || ' ' || v\_emp.salary);

v\_nr\_ang := v\_nr\_ang + 1;

v\_val\_lunara := v\_val\_lunara + v\_emp.salary;

v\_nr\_ang2 := v\_nr\_ang2 + 1;

v\_val\_lunara2 := v\_val\_lunara2 + v\_emp.salary;

end loop;

v\_val\_medie := v\_val\_lunara / v\_nr\_ang;

dbms\_output.put\_line('Numar angajati = ' || v\_nr\_ang);

dbms\_output.put\_line('Valoare lunara venituri = ' || v\_val\_lunara);

dbms\_output.put\_line('Valoare medie venituri = ' || v\_val\_medie);

dbms\_output.new\_line;

end loop;

v\_val\_medie2 := v\_val\_lunara2 / v\_nr\_ang2;

dbms\_output.put\_line('Numar total angajati = ' || v\_nr\_ang2);

dbms\_output.put\_line('Valoare totala lunara venituri = ' || v\_val\_lunara2);

dbms\_output.put\_line('Valoare medie venituri = ' || v\_val\_medie2);

end;

/

--E3

declare

v\_suma\_totala number(30, 2) := 0;

v\_suma number (10, 2) := 0;

v\_procent number(5, 2);

begin

for v\_venituri in (select employee\_id, salary, commission\_pct

from employees

group by employee\_id, salary, commission\_pct)

loop

v\_suma\_totala := v\_suma\_totala + v\_venituri.salary + v\_venituri.salary \* nvl(v\_venituri.commission\_pct, 0);

end loop;

dbms\_output.put\_line('Suma totala = ' || v\_suma\_totala);

dbms\_output.new\_line;

for v\_job in (select job\_id, job\_title

from jobs)

loop

dbms\_output.put\_line('Job-ul ' || v\_job.job\_title || ': ');

for v\_emp in (select last\_name, salary, commission\_pct

from employees

where job\_id = v\_job.job\_id)

loop

v\_suma := v\_emp.salary + v\_emp.salary \* nvl(v\_emp.commission\_pct, 0);

v\_procent := v\_suma \* 100 / v\_suma\_totala;

dbms\_output.put\_line(v\_emp.last\_name || ': salariu + comision = ' || v\_suma || ' = ' || v\_procent || '%');

end loop;

dbms\_output.new\_line;

end loop;

end;

/

--E4

declare

v\_nr\_ang number(3);

begin

for v\_job in (select job\_id, job\_title

from jobs)

loop

dbms\_output.put\_line('Job-ul ' || v\_job.job\_title || ': ');

v\_nr\_ang := 0;

for v\_emp in (select \*

from (select last\_name, salary

from employees

where job\_id = v\_job.job\_id

order by salary desc)

where rownum < 6)

loop

dbms\_output.put\_line(v\_emp.last\_name || ' ' || v\_emp.salary);

v\_nr\_ang := v\_nr\_ang + 1;

end loop;

if v\_nr\_ang < 5 then dbms\_output.put\_line('Sunt mai putin de 5 angajati');

end if;

dbms\_output.new\_line;

end loop;

end;

/

--E5

declare

v\_nr\_ang number(3);

v\_poz number(1);

v\_salariu employees.salary%type;

begin

for v\_job in (select job\_id, job\_title

from jobs)

loop

dbms\_output.put\_line('Job-ul ' || v\_job.job\_title || ': ');

v\_nr\_ang := 0;

v\_poz := 0;

v\_salariu := 0;

for v\_emp in (select last\_name, salary

from employees

where job\_id = v\_job.job\_id

order by salary desc)

loop

if v\_emp.salary <> v\_salariu then

v\_poz := v\_poz + 1;

v\_salariu := v\_emp.salary;

end if;

if v\_poz < 6 then

dbms\_output.put\_line(v\_emp.last\_name || ' salariu ' || v\_emp.salary || ' pozitia ' || v\_poz);

end if;

v\_nr\_ang := v\_nr\_ang + 1;

exit when v\_poz = 6;

end loop;

if v\_nr\_ang < 5 then dbms\_output.put\_line('Sunt mai putin de 5 angajati');

end if;

dbms\_output.new\_line;

end loop;

end;

/

--LABORATOR 4