**Rezolvarea laboratoarelor**

**Grupa 232 – 2**

**SGBD , 2023 – 2024**

**Forma răspunsurilor**

Nume student – Nr exercițiu

Rezolvare

**Laborator 2**

--4

select category, count(distinct title), count(\*), count(r.copy\_id)

from rental r, title t

where r.title\_id = t.title\_id

group by category

having count(\*) = (select max(count(\*))

from rental r, title t

where r.title\_id = t.title\_id

group by category);

-- Hodivoianu Anamaria - ex 5

select t.title\_id, t.title, count(tc.copy\_id)

from title\_copy tc, title t

where t.title\_id = tc.title\_id

and tc.status = 'AVAILABLE'

group by (t.title\_id, t.title)

order by t.title\_id;

--6

select t.title\_id,t.copy\_id as nr\_exemplar, t.status,

(case

when r.act\_ret\_date is NULL then 'AVAILABLE'

else 'RENTED'

end) as status\_corect

from title\_copy t

join rental r on t.title\_id = r.title\_id;

-- MINCU ADRIAN – ex 6

select t.TITLE, r.ACT\_RET\_DATE, tc.STATUS, *decode*(r.ACT\_RET\_DATE, null, 'RENTED', 'AVAILABLE') as "STATUS CORECT"  
from rental r, TITLE\_COPY tc, title t  
where r.TITLE\_ID = tc.TITLE\_ID and t.TITLE\_ID = r.TITLE\_ID;

Select a.title\_id, title, copy\_id, status status\_setat,

case when (a.title\_id, copy\_id) not in (select title\_id, copy\_id

From rental

Where act\_ret\_date is null)

then 'AVAILABLE'

else 'RENTED'

end status\_corect

From title\_copy a, title b

Where a.title\_id = b.title\_id

order by 1,2;

--7. a

Select count(\*)

From ( Select a.title\_id, title, copy\_id, status status\_setat,

case when (a.title\_id, copy\_id) not in (select title\_id, copy\_id

From rental

Where act\_ret\_date is null)

then 'AVAILABLE'

else 'RENTED'

end status\_corect

From title\_copy a, title b

Where a.title\_id = b.title\_id)

Where status\_setat <> status\_corect;

--7.b

Create table title\_copy\_prof

As select \* from title\_copy;

Update title\_copy\_prof p

Set status = case when (p.title\_id, copy\_id) not in (select title\_id, copy\_id

From rental

Where act\_ret\_date is null)

then 'AVAILABLE'

else 'RENTED'

End

Where status <> case when (p.title\_id, copy\_id) not in (select title\_id, copy\_id

From rental

Where act\_ret\_date is null)

then 'AVAILABLE'

else 'RENTED'

End;

-- Mincu Adrian – ex 8

select *decode*(*count*(rez),0,'DA','NU') as raspuns  
from (  
 select BOOK\_DATE as rez  
 from RENTAL  
 MINUS  
 select RES\_DATE  
 from RESERVATION  
 );

-- Hodivoianu Anamaria - ex 8

select (case when exists (select 1

from reservation rs, rental rn

where rs.title\_id = rn.title\_id

and rs.member\_id = rn.member\_id

and rs.res\_date <> rn.book\_date)

then 'Nu' else 'Da' end)

from dual;

-- Mincu Adrian – ex 9

select m.FIRST\_NAME || ' ' || m.LAST\_NAME as nume, t.TITLE, *count*(t.title)  
from MEMBER m, rental r, title t  
where m.MEMBER\_ID = r.MEMBER\_ID and r.TITLE\_ID = t.TITLE\_ID  
group by m.FIRST\_NAME || ' ' || m.LAST\_NAME, t.TITLE ,t.title;

--Ilie George ex 9

SELECT

m.last\_name,

m.first\_name,

t.title,

COUNT(r.title\_id)

FROM member m

right JOIN rental r ON m.member\_id = r.member\_id

right JOIN title t ON r.title\_id = t.title\_id

GROUP BY m.last\_name, m.first\_name, t.title

ORDER BY m.last\_name, m.first\_name, t.title;

select m.member\_id, first\_name, last\_name,

t.title\_id, t.title, count(r.title\_id)

from member m, title t, rental r

where t.title\_id = r.title\_id (+)

and m.member\_id(+) = r.member\_id

group by m.member\_id, first\_name, last\_name, t.title\_id, title

order by 1,4;

-- Mincu Adrian – ex 10

select m.FIRST\_NAME || ' ' || m.LAST\_NAME as nume, t.TITLE ,*count*(tc.TITLE\_ID) as nr\_exmemplare  
from MEMBER m  
right JOIN rental r ON m.member\_id = r.member\_id  
right JOIN title t ON r.title\_id = t.title\_id  
join TITLE\_COPY tc on tc.TITLE\_ID = t.TITLE\_ID  
group by m.FIRST\_NAME || ' ' || m.LAST\_NAME, t.TITLE ,tc.TITLE\_ID;

Firca liviu Nicolae – ex 7 b

update title\_copy\_fln t

set t.status = nvl((select decode(act\_ret\_date,null,'AVAILABLE','RENTED')

from rental r

where (r.copy\_id= t.copy\_id and r.title\_id=t.title\_id

and r.book\_date=(select max(k.book\_date)

from rental k

where k.copy\_id=r.copy\_id and r.title\_id=k.title\_id))

),status);

**Laborator 3**

--8

select \* from reservation;

select \* from rental where (member\_id, title\_id) in

(select member\_id, title\_id from reservation);

select case when count(\*) = 0 then 'Da'

else 'Nu'

end raspuns

from

(select res\_date, member\_id, title\_id

from reservation

minus

select book\_date, member\_id, title\_id

from rental);

--9

select \* from title where title\_id not in (select title\_id from rental);

select m.member\_id, first\_name, last\_name,

t.title\_id, t.title, count(r.title\_id)

from member m, title t, rental r

where t.title\_id = r.title\_id (+)

and m.member\_id(+) = r.member\_id

group by m.member\_id, first\_name, last\_name, t.title\_id, title

order by 1,4;

--11

select t.title\_id, title, c.copy\_id, status

from title t, title\_copy c, rental r

where t.title\_id = c.title\_id

and c.copy\_id = r.copy\_id

and c.title\_id = r.title\_id

order by 1,3;

--12

--a

select (select count(\*)

from rental

where to\_char(sysdate,'MM-YYYY') =

to\_char(book\_date,'MM-YYYY')

and extract (day from book\_date) = 1) zi\_1,

(select count(\*)

from rental

where to\_char(sysdate,'MM-YYYY') =

to\_char(book\_date,'MM-YYYY')

and extract (day from book\_date) = 2) zi\_2

from dual;

--b

select ziua, (select count(\*) from rental where to\_char(book\_date,'dd.mm.yyyy') = to\_char(ziua,'dd.mm.yyyy')) as nr

from (select distinct book\_date ziua

from rental

where to\_char(book\_date,'mm-yyyy') = to\_char(sysdate,'mm-yyyy'));

--c

select ziua, (select count(\*) from rental where to\_char(book\_date,'dd.mm.yyyy') = to\_char(ziua,'dd.mm.yyyy')) as nr

from (select distinct book\_date ziua

from rental

where to\_char(book\_date,'mm-yyyy') = to\_char(add\_months(sysdate,-1),'mm-yyyy'));

**--PLSQL1**

Dogarel Andrei-exercitiul 6

DECLARE

v\_dep departments.department\_name%TYPE;

v\_cnt NUMBER(5);

BEGIN

SELECT department\_name, count(e.employee\_id)

INTO v\_dep, v\_cnt

FROM employees e, departments d

WHERE e.department\_id=d.department\_id

GROUP BY department\_name

HAVING COUNT(\*) = (SELECT MAX(COUNT(\*))

FROM employees

GROUP BY department\_id);

DBMS\_OUTPUT.PUT\_LINE('Departamentul '|| v\_dep || ' Numar de angajati: ' || v\_cnt);

END;

/

Tîrîlă Patric-Gabriel ex 6

variable rezultat VARCHAR2(35)

variable nr\_ang NUMBER

begin

select department\_name, count(\*)

into :rezultat, :nr\_ang

from employees e, departments d

where e.department\_id = d.department\_id

group by department\_name

having count(\*) = (

select max(count(\*))

from employees

group by department\_id

);

end;

/

print nr\_ang

**--PL/SQL2**

**--1 Dogaru Mihail Danut**

**DECLARE**

**TYPE tablou\_imbricat IS TABLE OF employees.employee\_id%TYPE;**

**t tablou\_imbricat := tablou\_imbricat();**

**cnt NUMBER(5) := 1;**

**salariu employees.salary%TYPE;**

**salariuNou employees.salary%TYPE;**

**BEGIN**

**FOR i in (select \* from**

**(select employee\_id,salary**

**from employees**

**where commission\_pct is null**

**order by salary )**

**where rownum < 6)**

**LOOP**

**t.extend;**

**salariu := i.salary;**

**t(cnt) := i.employee\_id;**

**salariuNou := salariu + salariu\*0.05;**

**--UPDATE employees**

**--SET salary = salary + salary\*0.05**

**--WHERE employee\_id = t(cnt);**

**DBMS\_OUTPUT.PUT\_LINE(t(cnt)||' salariu vechi: '||salariu||' salariu nou: '|| salariuNou);**

**cnt := cnt + 1;**

**END LOOP;**

**END;**

**/**

Gavrila Florin-Alexandru – exercitiul 1

declare

type colectie is table of number index by pls\_integer;

v\_col colectie;

v\_salariu colectie;

v\_salariu\_nou number;

begin

select employee\_id

bulk collect into v\_col

from (

select \*

from employees

where commission\_pct is NULL

order by 8)

where rownum <= 5;

select salary

bulk collect into v\_salariu

from (

select \*

from employees

where commission\_pct is NULL

order by 8)

where rownum <= 5;

for i in v\_col.first..v\_col.last loop

dbms\_output.put(nvl(v\_col(i), 0) || ' ');

update employees

set salary = 1.05 \* salary

where employee\_id = v\_col(i)

returning salary

into v\_salariu\_nou;

dbms\_output.put('Salariu vechi: ' || v\_salariu(i) || ' Salariu nou: ' || v\_salariu\_nou);

dbms\_output.new\_line;

end loop;

dbms\_output.new\_line;

end;

/

**Laborator 5 – PLSQL 3**

--E1

select j.job\_id, j.job\_title, count(e.employee\_id)

from jobs j, employees e

where j.job\_id = e.job\_id

group by j.job\_id, j.job\_title

order by count(employee\_id);

--a. cursor explicit

DECLARE

CURSOR c1 IS

SELECT job\_title, job\_id

FROM jobs;

CURSOR c2 IS

SELECT last\_name, salary, job\_id

FROM employees;

v\_titlu jobs.job\_title %TYPE;

v1\_id jobs.job\_id %TYPE;

v\_nume employees.last\_name %TYPE;

v\_salary employees.salary %TYPE;

v2\_id employees.job\_id %TYPE;

BEGIN

OPEN c1;

LOOP

FETCH c1 into v\_titlu, v1\_id;

EXIT WHEN c1%NOTFOUND;

DBMS\_OUTPUT.put\_line('Id-ul jobului : '

|| v1\_id

|| ' , '

|| v\_titlu);

OPEN c2;

LOOP

FETCH c2 into v\_nume, v\_salary, v2\_id;

EXIT WHEN c2%NOTFOUND;

IF v1\_id = v2\_id THEN

DBMS\_OUTPUT.put\_line('Angajatul cu numele '

|| v\_nume || ' avand salariul: '

|| v\_salary);

END IF;

END LOOP;

close c2;

END LOOP;

close c1;

END;

/

--b. ciclu cursor

--c. ciclu cursor cu subcereri

-- MINCU ADRIAN

begin  
 for v\_job in (select JOB\_TITLE, JOB\_ID  
 from JOBS)  
 loop  
 DBMS\_OUTPUT.*PUT\_LINE*('JOB: ' || v\_job.JOB\_TITLE);  
 DBMS\_OUTPUT.*NEW\_LINE*();  
 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);  
 end loop;  
 end loop;  
end;

declare  
 v\_existsEmployees number(1) := 0;  
begin  
 for v\_job in (select JOB\_TITLE, JOB\_ID  
 from JOBS)  
 loop  
 DBMS\_OUTPUT.*PUT\_LINE*('JOB: ' || v\_job.JOB\_TITLE);  
 DBMS\_OUTPUT.*NEW\_LINE*();  
 v\_existsEmployees := 0;  
 for v\_emp in (select LAST\_NAME, SALARY  
 from EMPLOYEES  
 where JOB\_ID = v\_job.JOB\_ID)  
 loop  
 v\_existsEmployees := 1;  
 DBMS\_OUTPUT.*PUT\_LINE*(' ' || v\_emp.LAST\_NAME || ' ' || v\_emp.SALARY);  
 end loop;  
 if v\_existsEmployees = 0 then DBMS\_OUTPUT.*PUT\_LINE*(' NU are employees acest JOB');  
 end if;  
 end loop;  
end;

--d. expresii cursor

-- Hodivoianu Anamaria

declare

type refcursor is ref cursor;

cursor c\_job is

select job\_title, cursor (select e.last\_name, e.salary

from employees e

where e.job\_id = j.job\_id)

from jobs j;

v\_cursor refcursor;

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

v\_name employees.last\_name%type;

v\_salary employees.salary%type;

begin

open c\_job;

loop

fetch c\_job into v\_job\_title, v\_cursor;

exit when c\_job%notfound;

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

loop

fetch v\_cursor into v\_name, v\_salary;

exit when v\_cursor%notfound;

dbms\_output.put\_line(v\_name || ' ' || v\_salary);

end loop;

end loop;

close c\_job;

end;

/