Name : Sagar Singh

Prn no : 2020033800104487

ASSIGNMENT- 2

PL/SQL

1. Write PL/SQL block which will give a raise in salary to employees as per the following:

if sal+comm < 5000 increase by 10% of sal

if sal+comm >= 5000 increase by 500 + 12% of sal above 7000

→create table employee(emp_id number(4) constraint pk_id primary key,

sal number(5), comm number(5));

insert into employee values(101,6000,300); insert

into employee values(102,5000,100); insert into

employee values(131,4000,600); insert into

employee values(104,3000,300); insert into

employee values(105,2500,500); insert into

employee values(106,8000,500);

select * from employee;

EMP_ID	SAL	COMM
101	6000	300
102	5000	100
131	4000	600
104	3000	300
105	2500	500
106	8000	500

```
declare   cursor c1 is select * from
employee;

 begin

 for r1 in c1
loop

   if (r1.sal+r1.comm < 5000) then      r1.sal := r1.sal *
1.1;     elsif (r1.sal+r1.comm between 5000 and 7000)
then      r1.sal :=r1.sal+500;     else       r1.sal := r1.sal +
500 + (0.12*(r1.sal-7000));     end if;

   update employee set sal = r1.sal where emp_id = r1.emp_id;

   end loop;
commit;     end;
   /
   select * from employee;
```

EMP_ID	SAL	COMM
101	6500	300
102	5500	100
131	4400	600
104	3300	300
105	2750	500
106	8620	500

2. Suppose you have created the following three tables in your database :

PART_MASTER (P#,PNAME, PRICE, TOT_QTY, REORDER_LEVEL)

PART_TRANS (O#, P#, QTY_REQ), where O# is the order no.

TEMP (P#,QTY)

Write a PL/SQL block which will read records one by one from the transaction table and will check whether after issuing this part, the total quantity of this part is going below the reorder level in the master table or not. If it is then that part is not to be issued, hence give appropriate message and enter the details of this part in the TEMP table. If the total quantity after issuing that part is not going below the reorder level then modify the master table accordingly.

→ create table part_master (P number(2) constraint pk_p primary key,

PNAME varchar2(30),PRICE number(3),

TOT_QTY number(3),REORDER_LEAVEL number(3));

insert into part_master values(1,'tubelight',150,10,2);

insert into part_master values(2,'bulb',50,40,20); insert

into part_master values(3,'tv_remote',50,40,20); insert

into part_master values(4,'charger',500,10,1); insert

into part_master values(5,'speakers',399,5,2);

select * from part_master

P	PNAME	PRICE	TOT_QTY	REORDER_LEAVEL
1	tubelight	150	10	2
2	bulb	50	40	20
3	tv_remote	50	40	20
4	charger	500	10	1
5	speakers	399	5	2

```
create table part_trans(O number(2),P number(2) ,QTY_REQ number(2),
constraint pk_po primary key(P,O), constraint fk_p foreign key (P)
references part_master(P));

 insert into part_trans values(1,2,30);
insert into part_trans values(2,3,40);
insert into part_trans values(3,4,15);
insert into part_trans values(4,2,15);
insert into part_trans values(5,1,7);  insert
into part_trans values(6,5,3); select *
from part_trans;
```

O	P	QTY_REQ
1	2	30
2	3	40
3	4	15
4	2	15
5	1	7
6	5	3

Download CSV

```
create table temp(P number(2),QTY_REQ number(2), constraint
fk_tp foreign key(P) references part_master(P)); declare
        p number(2);
tq number(3);   ro
number(3);      qr
number(2);      n
number(2);
```

```
        cursor c is select a.p,a.tot_qty,a.REORDER_LEAVEL,b.qty_req
from part_master a,part_trans b where a.p=b.p; begin
        open c;
loop
                fetch c into p,tq,ro,qr;
exit when c%notfound;
                n:=tq-ro;
        if n<0 then
                        dbms_output.put_line('WE ARE OUT OF STOCKS');
        insert into temp values(p,qr);
                else
                        if n<ro then
                        dbms_output.put_line('WE ARE OUT OF QUANTITY YOU ARE ASKING');
        insert into temp values(p,qr);
                        end if;
                        if n>ro then
                                update part_master set tot_qty=n where p=part_master.p;
                        end if;
                end if;
        end loop;
close c;
end;
/
```

```
Statement processed.
WE ARE OUT OF STOCKS
WE ARE OUT OF STOCKS
WE ARE OUT OF STOCKS
WE ARE OUT OF QUANTITY YOU ARE ASKING
WE ARE OUT OF QUANTITY YOU ARE ASKING
```

```
select * from part_master;
```

P	PNAME	PRICE	TOT_QTY	REORDER_LEAVEL
1	tubelight	150	2	2
2	bulb	50	2	20
3	tv_remote	50	2	20
4	charger	500	2	1
5	speakers	399	2	2

select * from part_trans;

O	P	QTY_REQ
1	2	30
2	3	40
3	4	15
4	2	15
5	1	7
6	5	3

select * from temp;

P	QTY_REQ
2	30
3	40
2	15
1	7
5	3

3. Given a table ISSUE(ROLLNO,BOOKNO,ISSUE_DATE,RETURN_DATE),check which students have to pay fine and the amount of fine to be paid. A student can keep the book for fifteen days. If the number of days has exceeded 15 then fine is calculated as follows:

Upto 7 days **50 paise per day**

8 - 15 **Re. 1 per day from 8th day onwards**

more than 15 days **Rs. 1.5 per day from 16th day onwards**

After calculating the fine store the required information in the FINE table so that a report can later on printed out.

→ create table issue(rollno number(3), bookno

number(3) constraint pk_bno primary key,

issue_date   date, return_date date);

insert into issue
values(570,181,to_date('05/05/2020','dd/mm/yy'),to_date('02/06/2020','dd/mm/yy'));

insert into issue
values(512,199,to_date('14/08/2020','dd/mm/yy'),to_date('24/09/2020','dd/mm/yy'));

insert into issue
values(516,100,to_date('03/07/2020','dd/mm/yy'),to_date('02/09/2020','dd/mm/yy'));

select * from issue;

ROLLNO	BOOKNO	ISSUE_DATE	RETURN_DATE
570	181	05-MAY-20	02-JUN-20
512	199	14-AUG-20	24-SEP-20
516	100	03-JUL-20	02-SEP-20

create table fine(rollno number(3) ,bookno number(3) ,fine_amount number(4,2),constraint fk_bno foreign key(bookno) references issue(bookno));

declare rollno number(3);

bookno number(3); issue_date

date; return_date date; fine

number(4,2); days_up

number(2); cursor c1 is select

rollno,bookno,

issue_date,return_date from

issue;

```
begin  open c1;  loop   fetch c1 into
rollno,bookno,issue_date,return_date;   exit when
c1%notfound;
 fine:=0;   days_up:= round(return_date-
issue_date);

 if days_up>15 and days_up<22 then
  fine:=days_up*0.5;   elsif days_up>15
and days_up<30 then    fine:=days_up;
elsif days_up>15 and days_up>30 then
fine:=days_up*1.5;
 else     fine:=0;   end if;  insert into fine
values(rollno,bookno,fine);

 end loop;
close c1; end;
/ select * from
fine;
```

ROLLNO	BOOKNO	FINE_AMOUNT
570	181	28
512	199	61.5
516	100	91.5

ASSIGNMENT- 3

PL/SQL

1. Write a PL/SQL block for the following table as per the requirement :
CANDIDATE (SEAT_NO, FORM_NO, NAME, CATEGORY, PERCENTAGE)

Except the seat numbers, all other information regarding the candidates appearing in the MCA entrance test is available in the CANDIDATE table. Seat numbers are to be generated as per the following:

Seat numbers are 8 characters in length. The first three characters are '111'. The fourth character represents the category to which the candidate belongs. The rest of the four characters are used to denote the rank of the candidate as per his percentage, category-wise. For e.g., a candidate belonging to the GENERAL category and having secured the highest percentage amongst all other candidates in the same category will have a seat number as '11120001', where '2' is the code for the GENERAL category. In this way generate the seat numbers for all the candidates in the CANDIDATE table. Please mention any assumptions that you make. The codes for the different categories are:

GENERAL	**:**	**2**
SC	**:**	**3**
ST	**:**	**4**
SEBC	**:**	**5**
PH	**:**	**6**

→create table candidate(seatno varchar2(8),

formno number(5) constraint pk_f primary key,

cname varchar2(20), cate varchar2(10), percentage

number(3));

insert into candidate values (null,1,'vrajesh','gen',80);

insert into candidate values (null,2,'vansh','gen',76); insert

into candidate values (null,3,'darshit','sc',70); insert into

candidate values (null,4,'virat','st',66); insert into

candidate values (null,5,'rohit','sebc',83); insert into

candidate values (null,6,'ravi','sebc',89); insert into

candidate values (null,7,'ram','ph',84); insert into

candidate values (null,8,'darsh','sc',77); insert into

candidate values (null,9,'rahul','st',56);

select * from candidate;

SEATNO	FORMNO	CNAME	CATE	PERCENTAGE
-	1	vrajesh	gen	80
-	2	vansh	gen	76
-	3	darshit	sc	70
-	4	virat	st	66
-	5	rohit	sebc	83
-	6	ravi	sebc	89
-	7	ram	ph	84
-	8	darsh	sc	77
-	9	rahul	st	56

declare cursor c1 is select * from candidate order by

cate,percentage desc; seat varchar2(8); cnt number(2); cat

varchar2(10); crank varchar2(10);

```
begin
   cnt :=0;
cat := 'gen';

   for  r1  in  c1
loop      seat :=
'111';

   if(cat = r1.cate) then
 if r1.cate='gen' then
     cnt   :=   cnt   +   1;                    crank   :=
lpad(to_char(cnt),4,0);       seat:='1112' || crank;
           elsif r1.cate='sc' then
     cnt   :=   cnt   +   1;                   crank   :=
lpad(to_char(cnt),4,0);     seat :='1113' || crank;
            elsif r1.cate='st' then
                      cnt := cnt + 1;
                     crank := lpad(to_char(cnt),4,0);
           seat :='1114' || crank;
           elsif r1.cate='sebc' then
                     cnt := cnt + 1;
crank := lpad(to_char(cnt),4,0);
seat :='1115' || crank;               else
 cnt := cnt + 1;
                     crank :=
lpad(to_char(cnt),4,0);                        seat
:='1116' || crank;          end if;
```

```
   else     cnt :=0;     cat :=
r1.cate;              if
r1.cate='gen' then
                        cnt := cnt + 1;
crank := lpad(to_char(cnt),4,0);
seat:='1112' || crank;
             elsif r1.cate='sc' then
                        cnt := cnt + 1;
crank := lpad(to_char(cnt),4,0);
seat :='1113' || crank;
              elsif r1.cate='st' then
                         cnt := cnt + 1;
 crank := lpad(to_char(cnt),4,0);
seat :='1114' || crank;
              elsif r1.cate='sebc' then
                         cnt := cnt + 1;
 crank := lpad(to_char(cnt),4,0);
seat :='1115' || crank;               else
 cnt := cnt + 1;                          crank :=
lpad(to_char(cnt),4,0);                            seat
:='1116' || crank;          end if;


   end if;


    update candidate set seatno=seat where r1.formno= formno;
end loop;     end;
/
```

select * from candidate;

SEATNO	FORMNO	CNAME	CATE	PERCENTAGE
11120001	1	vrajesh	gen	80
11120002	2	vansh	gen	76
11130002	3	darshit	sc	70
11140001	4	virat	st	66
11150002	5	rohit	sebc	83
11150001	6	ravi	sebc	89
11160001	7	ram	ph	84
11130001	8	darsh	sc	77
11140002	9	rahul	st	56

2. For the table STUDENT(Rollno, Name, Mrks1, Mrks2, Mrks3, Percentage), write a PL/SQL code which calculates the percentage for each student assuming passing marks as 40 and total marks as 100 for each subject. Also display the details about the students who have scored the highest in each subject.

→create table student (rollno number(2) constraint pk_rollno primary key,

sname varchar2(20), marks1 number(3), marks2 number(3), marks3

number(3), percentage number(4,2));

insert into student values (1,'aman',56,65,76,null);

insert into student values (2,'akash',66,65,88,null);

insert into student values (3,'abhi',41,55,66,null);

insert into student values (4,'atul',59,70,48,null); insert

into student values (5,'akash',67,69,86,null); insert

into student values (6,'akii',79,85,44,null);

select * from student;

ROLLNO	SNAME	MARKS1	MARKS2	MARKS3	PERCENTAGE
1	aman	56	65	76	-
2	akash	66	65	88	-
3	abhi	41	55	66	-
4	atul	59	70	48	-
5	akash	67	69	86	-
6	akii	79	85	44	-

```
declare   cursor c1 is select * from

student;  tot

student.marks1%type;  per

student.percentage%type;  m1

student.marks1%type :=0;  m2

student.marks1%type :=0;  m3

student.marks1%type :=0;


begin

   for r1 in c1
   loop


    if (r1.marks1>40 and r1.marks2>40 and r1.marks3>40 ) then

tot := r1.marks1+r1.marks2+r1.marks3;

    per := tot/3;

else     per :=

null;
```

```
    end if;

    update student set percentage = per where rollno = r1.rollno;

    if m1<r1.marks1 then
m1:=r1.marks1;
    end if;
    if m2<r1.marks2 then
m2:=r1.marks2;
    end if;
    if m3<r1.marks3 then
m3:=r1.marks3;
    end if;

   end   loop;
for  r1  in  c1
loop

   if(r1.marks1=m1) then
   dbms_output.put_line('Highest marks scored in subject-1');
dbms_output.put_line(r1.sname ||' '|| r1.marks1);
   end      if;                              if(r1.marks2=m2)      then
dbms_output.put_line('Highest  marks  scored  in  subject-2');
dbms_output.put_line(r1.sname|| ' '|| r1.marks2);
   end      if;                              if(r1.marks1=m3)      then
dbms_output.put_line('Highest  marks  scored  in  subject-3');
dbms_output.put_line(r1.sname|| ' '|| r1.marks3);
```

```
    end if;

    end loop;

end;
/

select * from student;
```

```
Statement processed.
Highest marks scored in subject-3
akash 88
Highest marks scored in subject-1
akii 79
Highest marks scored in subject-2
akii 85
```

ROLLNO	SNAME	MARKS1	MARKS2	MARKS3	PERCENTAGE
1	aman	56	65	76	65.67
2	akash	66	65	88	73
3	abhi	41	55	66	54
4	atul	59	70	48	59
5	akash	67	69	86	74
6	akii	79	85	44	69.33

For the following tables,

CANDIDATE (<u>CID</u>, CNAME, CADDRESS, CBIRTH_DT)

TEST (<u>TID</u>, TNAME, TOT_MRKS, PASS_MKS)

TEST_CENTRE (<u>TCID</u>, LOCATION, MGR, CAPACITY)

TEST_TAKEN (<u>CID, TID, TCID</u>, TEST_DT, SCORE)

3. Write a PL/SQL block which will accept the test id and test centre id from the user and display details about all those candidates who have scored more than average

for that test. Also display details about the dates on which that test centre had full attendance.

→create table candidate(cid number(3) constraint pk_cid primary key,

cname varchar2(20) constraint nn_cname not null, caddress

varchar2(20), cbirth_dt date);

insert into candidate values(1,'peter','chennai','15/JAN/1985');

insert into candidate values(2,'rohan','delhi','07/AUG/1988'); insert

into candidate values(3,'prakash','Surat','22/OCT/1988'); select

*from candidate;

CID	CNAME	CADDRESS	CBIRTH_DT
1	peter	chennai	15-JAN-85
2	rohan	delhi	07-AUG-88
3	prakash	Surat	22-OCT-88

create table ctest(tid number(3) constraint pk_tid primary key, tname

varchar2(20) constraint nn_tname not null, tot_mrks number(3),

pass_mrks number(3) constraint ck_pass_mrks check (pass_mrks>=35));

insert into ctest values(1,'oracle fundamentals',100,35);

insert into ctest values(2,'c & c++',100,35); insert into

ctest values(3,'java',100,35); select *from ctest;

TID	TNAME	TOT_MRKS	PASS_MRKS
1	oracle fundamentals	100	35
2	c & c++	100	35
3	java	100	35

create table test_centre(tcid number(3) constraint pk_tcid1 primary key,

loc varchar2(10), mgr varchar2(10), cap number(3) constraint

ck_capacity1 check(cap<=30));

insert into test_centre values(1,'surat','Examinor',1); insert

into test_centre values(2,'bangalore','Examinor',30); insert

into test_centre values(3,'mumbai','Examinor',23); insert

into test_centre values(4,'baroda','Examinor',25); select *

from test_centre;

TCID	LOC	MGR	CAP
1	surat	Examinor	1
2	bangalore	Examinor	30
3	mumbai	Examinor	23
4	baroda	Examinor	25

create table test_taken(cid number(3) constraint fk_cid references candidate on delete cascade,

tid number(3) constraint fk_tid references ctest on delete cascade, tcid number(3) constraint

fk_tcid references test_centre on delete cascade, test_date date, score number(3), constraint pk_cid_tid_tcid primary key(cid,tid,tcid));

insert into test_taken values(1,1,2,'04/APRIL/2004',65);

insert into test_taken values(2,2,2,'08/AUG/2004',32); insert into test_taken values(3,1,2,'07/JUN/2005',85); insert into test_taken values(2,1,3,'01/FEB/2007',79); insert into test_taken values(2,3,1,'09/MAY/2006',85); insert into test_taken values(3,2,3,'03/MARCH/2004',72); insert into test_taken values(1,3,1,'10/JAN/2006',61); insert into test_taken values(3,2,2,'11/NOV/2004',77); insert into test_taken values(2,1,1,'02/FEB/2004',44); select *from test_taken;

CID	TID	TCID	TEST_DATE	SCORE
1	1	2	04-APR-04	65
2	2	2	08-AUG-04	32
3	1	2	07-JUN-05	85
2	1	3	01-FEB-07	79
2	3	1	09-MAY-06	85
3	2	3	03-MAR-04	72
1	3	1	10-JAN-06	61
3	2	2	11-NOV-04	77
2	1	1	02-FEB-04	44

declare t_id ctest.tid%type;

t_cid test_centre.tcid%type;

```
 cursor  c1(t_id  ctest.tid%type,
t_cid  test_centre.tcid%type)  is
select  avg(score)  as  average
from  test_taken  where
tid=t_id and tcid=t_cid;  cursor
c2 (t_id ctest.tid%type,  t_cid
test_centre.tcid%type)  is
select  *  from  test_taken
where tcid=t_cid and tid = t_id;

 cursor c3 (t_cid test_centre.tcid%type) is
select test_date from test_taken  where
tcid=t_cid group by test_date  having
count(*) = (select cap  from test_centre
where tcid=t_cid );

begin
  t_id :=1;
t_cid :=1;

  for r1 in c1(t_id,t_cid)  loop  for r2 in c2(t_id,t_cid)  loop
if(r2.score >= r1.average) then  for r in (select * from candidate
where cid = r2.cid )  loop  dbms_output.put_line (r.cid || ' ' ||
r.cname || ' ' || r.cbirth_dt);  end loop;
  end if;
end loop;
end loop;
```

```
   dbms_output.put_line ( 'Dates on which  ' || t_cid || ' remained full are ');

for r3 in c3 (t_cid)    loop    dbms_output.put_line(r3.test_date);    end

loop; end;

/
```

```
Statement processed.
2 rohan 07-AUG-88
Dates on which  1 remained full are
09-MAY-06
02-FEB-04
10-JAN-06
```

4. Write a PL/SQL block, which will accept the candidate id from the user and for this user display details about all the tests that he has appeared for as well as the details about his scores and the maximum scores of those tests.

```
→declare   c_id

candidate.cid%type;


 cursor c1 (c_id candidate.cid%type) is

select   *   from   candidate      where

cid=c_id;


 cursor c2 (c_id candidate.cid%type) is  select

tid,score from test_taken  where cid = c_id;


 cursor c3 (t_id test_taken.tid%type) is

select   tname   from   ctest      where

tid=t_id;
```

```
 mscore test_taken.score%type :=0;


begin
c_id :=1;


 for r1 in c1(c_id)  loop  if c1%notfound then   if
c1%rowcount=0 then
dbms_output.put_line('candidate does not exist');
  end if;    exit;    else     for r2 in c2(c_id)     loop     if
c2%notfound then      if c2%rowcount=0 then
dbms_output.put_line('candidate has not given any exam');
    end if;
exit;        else
    if c2%rowcount=1 then
dbms_output.put_line('candidate details');
dbms_output.put_line('candidate name : '|| r1.cname);
dbms_output.put_line('Test names | Score');
    end if;     for r3 in c3(r2.tid)
loop        if (mscore<r2.score)
then        mscore:=r2.score;
       end if;
    dbms_output.put_line(r3.tname || ' | '|| r2.score);
end loop;     end if;
   dbms_output.put_line('candidate max score = '|| mscore);
end loop;    end if;   end loop;  end;
/
```

```
Statement processed.
candidate details
candidate name : peter
Test names | Score
oracle fundamentals | 65
candidate max score = 65
java | 61
candidate max score = 65
```