**PL/SQL QUERIES AND OUTPUT**

**TOPIC 1: RECORDS**

**Q1 .display the test details of the urinalysis test performed on 16 feb 2020.**

Ans. declare

type txt is record

(

tid int,

trdate date,

tamt int

);

t txt;

begin

select T\_id,TR\_date,T\_amt into t from test where T\_name='urinalysis' and T\_date='16-feb-2020';

dbms\_output.put\_line('test id is '||t.tid||'test result date is'||t.trdate||'test amount = '||t.tamt );

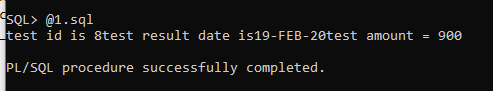
exception

when too\_many\_rows then

dbms\_output.put\_line('more than one row present');

end;

/



**Q2 .display patient details with patient name Nola**

Ans.declare

type pat is record

(

pno int,

ptype varchar(20),

paddress varchar(255),

page varchar(50)

);

p pat;

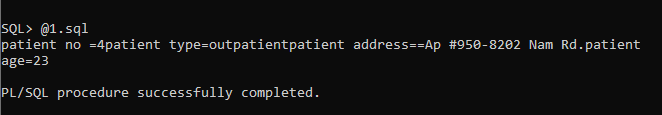
begin

select p\_id,p\_type,p\_address,p\_age into p from patient where P\_Fname='Nola';

dbms\_output.put\_line('patient no ='||p.pno||'patient type='||p.ptype || 'patient address==' || p.paddress || 'patient age=' || p.page);

end;

/



**Q3. display job details of the job 'Surgeon'.**

Ans. declare

type jbt is record

(

jno varchar(20),

jname varchar2(33)

);

j jbt;

begin

select j\_id,j\_name into j from jobtype where j\_name='Surgeon';

dbms\_output.put\_line('job no='||j.jno||' and job name='||j.jname);

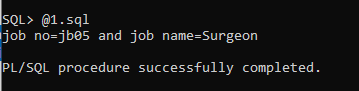
exception

when too\_many\_rows then

dbms\_output.put\_line('more than one row returned');

end;

/



**Q5. display room details of the room with room id 15.**

Ans.declare

type rm is record

(

roomno int,

rphoneno varchar2(20),

rtype varchar2(20),

rfloorno varchar2(20),

rbeds varchar2(20),

rstaffid varchar2(20)

);

r rm;

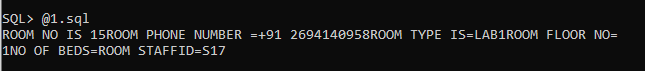
begin

select R\_NO,R\_PH\_NO,R\_TYPE,R\_FloorNO,No\_Of\_Beds,S\_id into r from room where R\_NO=15 ;

dbms\_output.put\_line('ROOM NO IS '||r.roomno ||'ROOM PHONE NUMBER ='|| r.rphoneno || 'ROOM TYPE IS='||r.rtype || 'ROOM FLOOR NO= '||r.rfloorno || 'NO OF BEDS=' || r.rbeds || 'ROOM STAFFID=' || r.rstaffid);

end;

/



**TOPIC 2: CURSORS**

**Q1. display test details of all the X-ray tests performed.**

Ans. declare

cursor cur is select T\_id,T\_name from test where T\_name='X-ray';

n test.T\_id%type;

l test.T\_name%type;

begin

open cur;

loop

fetch cur into n,l;

exit when cur%notfound;

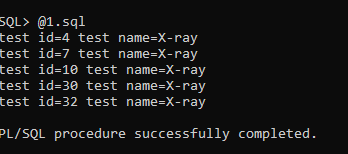
dbms\_output.put\_line('test id='||n||' test name='||l);

end loop;

close cur;

end;

/



**Q2. display lowest 5 salaries of the staff**

Ans. declare

s staff.S\_salary%type;

cursor sal\_cur is select S\_salary from staff order by S\_salary;

begin

open sal\_cur;

for i in 1..5 loop

fetch sal\_cur into s;

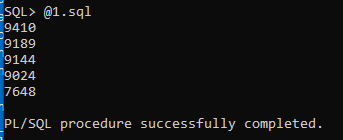
dbms\_output.put\_line(s);

end loop;

close sal\_cur;

end;

/



**Q3.display highest 5 salaries of the staff.**

Ans. declare

s staff.S\_salary%type;

cursor sal\_cur is select S\_salary from staff order by S\_salary desc;

begin

open sal\_cur;

for i in 1..5 loop

fetch sal\_cur into s;

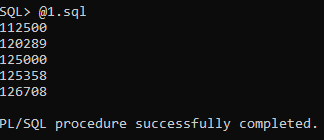
dbms\_output.put\_line(s);

end loop;

close sal\_cur;

end;

/



**Q4 .display only the patient admit and discharge date of all the patients.**

Ans. declare

prid patient\_record.rec\_id%type;

adate patient\_record.admit\_date%type;

ddate patient\_record.discharge\_date%type;

cursor patientrecord is

select rec\_id, admit\_date, discharge\_date FROM patient\_record;

begin

open patientrecord;

loop

fetch patientrecord into prid,adate,ddate;

exit when patientrecord%notfound;

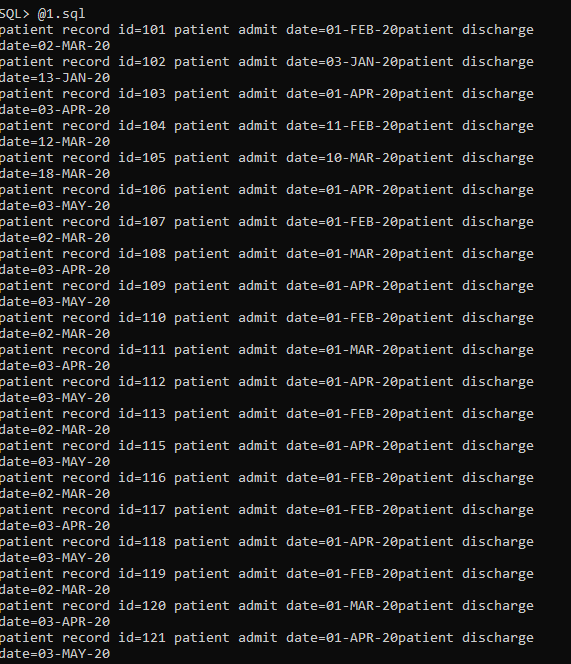
dbms\_output.put\_line('patient record id=' ||prid || ' patient admit date='|| adate || 'patient discharge date=' || ddate);

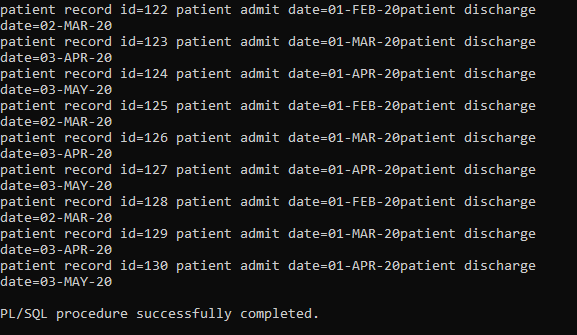
end loop;

close patientrecord;

end;

/





**TOPIC 3:PROCEDURES**

**Greetings**

CREATE OR REPLACE PROCEDURE greetings

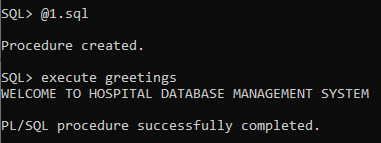
AS

BEGIN

dbms\_output.put\_line('WELCOME TO HOSPITAL DATABASE MANAGEMENT SYSTEM');

END;

/



**Q1. retrieve the shift of the staff when staff id is given.**

Ans. create or replace procedure p1(sid in varchar2)

as

staff\_shift varchar2(50);

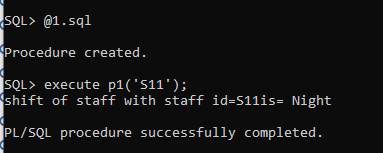
begin

select shift into staff\_shift from staff where sid=S\_id;

dbms\_output.put\_line('shift of staff with staff id='||sid||'is= '||staff\_shift);

end;

/



**Q2. retrieve the room no of the patient when patient record id is given.**

Ans. create or replace procedure p2(recid in varchar2)

as

roomno varchar2(50);

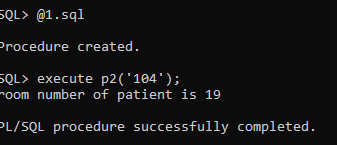
begin

select R\_NO into roomno from patient\_record where recid=rec\_id;

dbms\_output.put\_line('room number of patient is '||roomno);

end;

/



**Q3. retrieve the prescribed medicine given the prescription id.**

Ans. create or replace procedure p3(prid in varchar2)

as

mname varchar2(50);

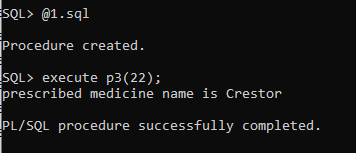
begin

select med\_name into mname from prescription where prid=pr\_id;

dbms\_output.put\_line('prescribed medicine name is '||mname);

end;

/



**TOPIC 4:FUNCTIONS**

**Q1.display the total number of patients.**

Ans. create or replace function totalpatients

return number IS

total number(2) := 0;

begin

select count(\*) into total

from patient;

return total;

end;

/

declare

c number(2);

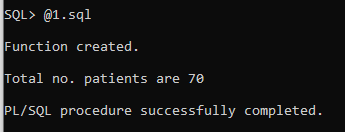
begin

c := totalpatients();

dbms\_output.put\_line('Total no. patients are ' || c);

end;

/



**Q2. retrieve the room floor given the room number.**

create or replace function floor(rn in number)

return string

as

rfloor string(100);

begin

select R\_FloorNo into rfloor from room where rn=R\_NO;

return rfloor;

end;

/

SQL> @1.sql

Function created.

SQL> select floor(30) from dual;

FLOOR(30)

----------

2

**TOPIC 5:TRIGGERS**

**1.Update the amount required for test.**

create table new1

( tid2 int,

tamount2 int

);

commit;

Commit complete.

create or replace trigger amt\_change1

after update of T\_amt on test

for each row

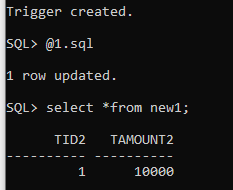
begin

insert into new1 values(:new.T\_id,:new.T\_amt);

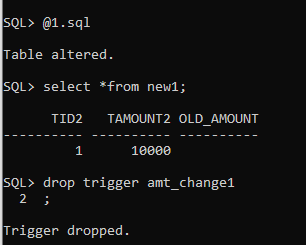
end;

/

SQL>update test set T\_amt=10000 where T\_id=1;



alter table new1 add old\_amount number;



drop trigger amt\_change1;

Trigger dropped.

**2.To insert or update a patient record**

create or replace trigger t

before insert or delete or update on patient\_record

begin

case

when inserting then

dbms\_output.put\_line('inserted');

when updating.then

dbms\_output.put\_line('updated');

end case;

end;

/

SQL> @1.sql

Trigger created.

SQL> set serveroutput on

3. To update patient records.

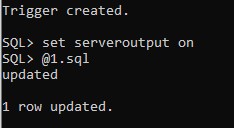
SQL> update patient\_record

2 set bed\_no='9'

3 where rec\_id=111;

updated

1 row updated.



**QUERIES GIVEN DURING REVIEW**

1. **Query that returns the name of all the radiologists available in the hospital.**

**Ans.** declare

cursor cur is select s\_fname from staff where j\_id='jb08';

n staff.s\_fname%type;

begin

open cur;

loop

fetch cur into n;

exit when cur%notfound;

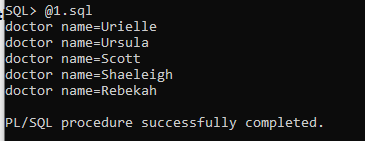
dbms\_output.put\_line('doctor name=' ||n );

end loop;

close cur;

end;

/



1. **Print the details of all the patients treated during a certain period of time.(btween jan and feb month).**

Ans. declare

pid patient.P\_id%type;

pfname patient.p\_fname%type;

pphno patient.p\_ph\_no%type;

paddress patient.p\_address%type;

adate patient\_record.admit\_date%type;

prid patient\_record.pr\_id%type;

cursor use\_cur is

select P\_id,p\_fname,p\_ph\_no,p\_address,admit\_date,pr\_id from patient,patient\_record

where P\_id= pr\_id and (admit\_date>'01-jan-2020' and admit\_date <'03-feb-20');

begin

open use\_cur;

loop

fetch use\_cur into pid,pfname,pphno,paddress,adate,prid;

exit when use\_cur%notfound;

dbms\_output.put\_line('patient id=' || pid || ' patient name =' || pfname || ' patient contact no=' ||pphno || 'patient address=' || paddress );

end loop;

close use\_cur;

end;

/

