**HOSPITAL MANAGEMENT SYSTEM**

**AIM**

To create hospital management system using plsql.

**SCHEMA**

Patient(pid,pname,page,pgender,pheight,pweight);

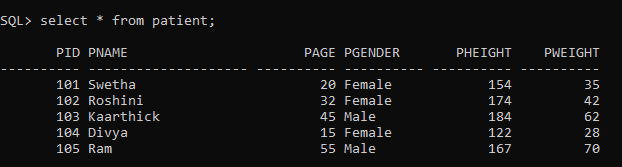
Doctor(did,dname,specialization,salary);

Treatment(did,pid,disease,joindate,discharge,fee);

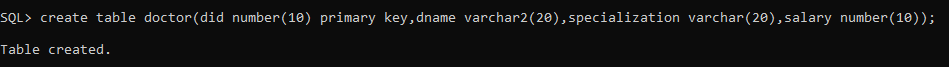
**CREATE TABLE PATIENT**

**C:\Users\Swetha\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot (570).png**

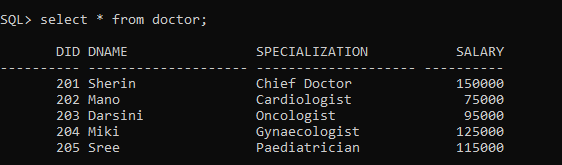
**DISPLAY TABLE PATIENT**

****

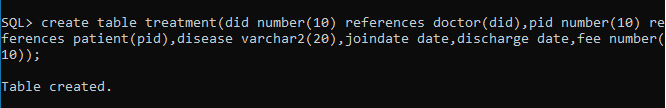
**CREATE TABLE DOCTOR**



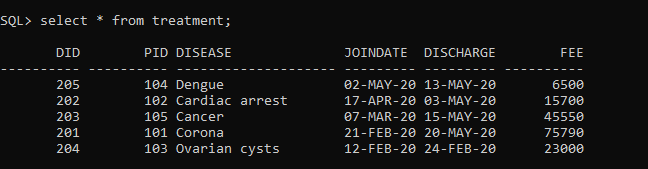
**DISPLAY TABLE DOCTOR**



**CREATE TABLE TREATMENT**

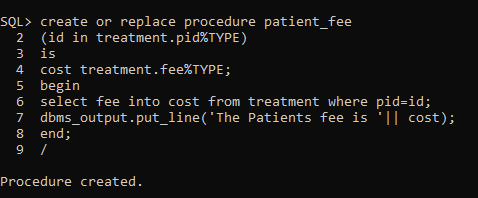
****

**DISPLAY TABLE TREATMENT**

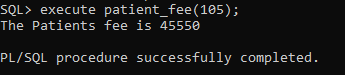


**PROCEDURE**

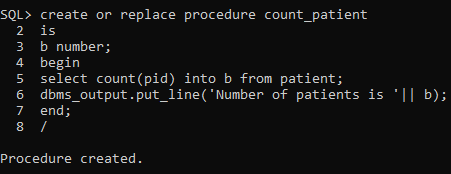
**1. To display the treatment fee of the patient**



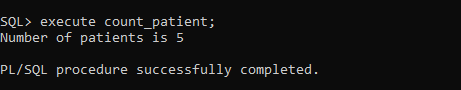
**EXECUTION AND OUTPUT**



**2. To find the number of patients in the hospital**



**EXECUTION AND OUTPUT**



**FUNCTION**

**1. To calculate the BMI of the patient**

create or replace function bmicalc(n1 in number, n2 in number)

return float

is

bmi float(8);

begin

bmi :=(n1/(n2\*n2))\*10000;

return bmi;

end;

/

DECLARE

h number;

W number;

patientbmi float;

BEGIN

select pheight,pweight into h,w

from patient where pid=104;

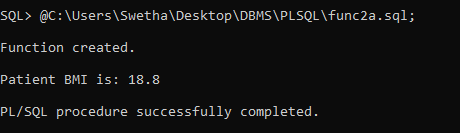
patientbmi:=bmicalc(w,h);

dbms\_output.put\_line('Patient BMI is: '||patientbmi);

END;

/

**EXECUTION AND OUTPUT**



**2. To increment the salary of the doctors by 10 percent**

create or replace function calc(n1 in number, n2 in number)

return float

is

inc float(8);

begin

inc :=n1\*(1+(n2/100));

return inc;

end;

/

DECLARE

sal number;

increment float;

BEGIN

select salary into sal

from doctor where did=201;

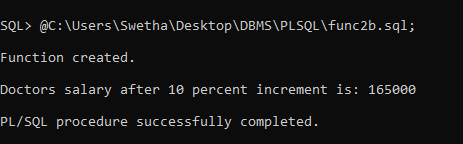
increment:=calc(sal,10);

dbms\_output.put\_line('Doctors salary after 10 percent increment is: '||increment);

END;

/

**EXECUTION AND OUTPUT**



**CURSORS**

**1. To create an explicit cursor to display the product names with their cost in descending order.**

**Explicit cursor code**

DECLARE

name patient.pname%type;

h patient.pheight%type;

w patient.pweight%type;

bmi float(10);

CURSOR pat is

SELECT pname,pheight,pweight FROM patient;

BEGIN

OPEN pat;

LOOP

FETCH pat into name,h,w;

EXIT WHEN pat%notfound;

bmi:=(w/(h\*h))\*10000;

IF bmi<18 THEN

dbms\_output.put\_line(name||':Underweight BMI:'||bmi);

ELSIF bmi<25 THEN

dbms\_output.put\_line(name||': Normal BMI:'||bmi);

ELSE

dbms\_output.put\_line(name||': Overweight BMI:'||bmi);

END IF;

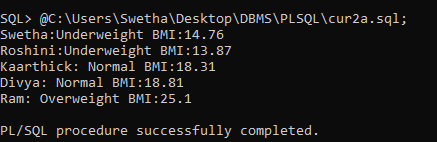
END LOOP;

CLOSE pat;

END;

/

**EXECUTION AND OUTPUT**



**2. To create an implicit cursor to update all the product stock by 10.**

**Implicit Cursor Code**

DECLARE

total\_rows number(2);

BEGIN

UPDATE treatment

SET fee=fee+500;

IF sql%notfound THEN

dbms\_output.put\_line('No records updated');

ELSIF sql%found THEN

total\_rows:= sql%rowcount;

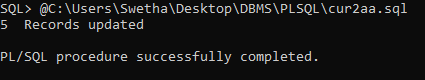
dbms\_output.put\_line(total\_rows || ' Records updated');

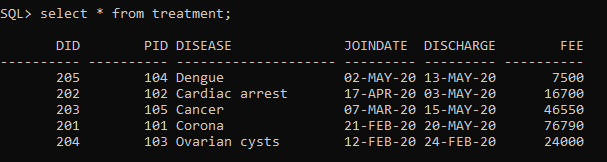
END IF;

END;

/

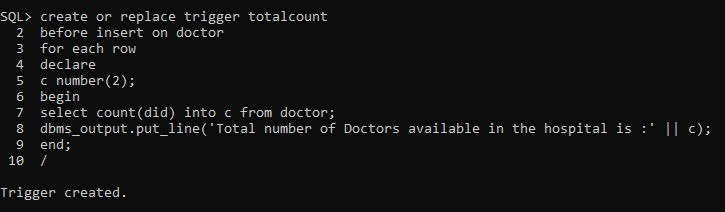
**EXECUTION AND OUTPUT**



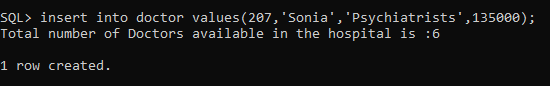


**TRIGGERS**

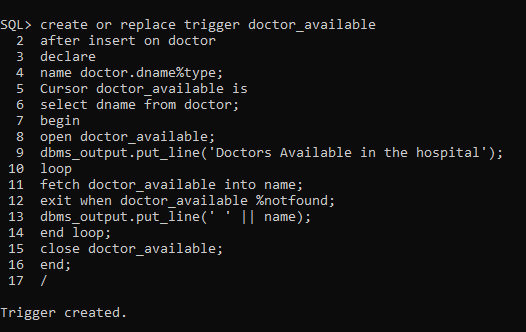
**1. To create a trigger that displays the total number of doctors available in the hospital before inserting a doctor**



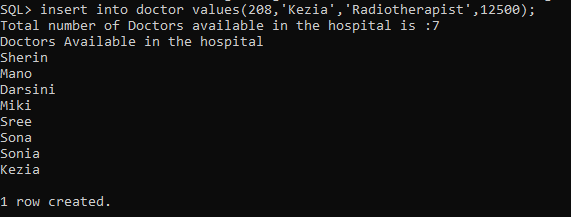
**EXECUTION AND OUTPUT**



**2. To create a trigger that displays names of doctors available in the hospital after inserting a doctor**



**EXECUTION AND OUTPUT**



**PACKAGE**

**1. To create a package with two procedures to fid count of patients and doctors in the hospital.**

create or replace package hospi

as

procedure patient\_count;

procedure doctor\_count;

end hospi;

/

create or replace package body hospi as

procedure patient\_count

as

b number;

begin

select count(pid) into b from patient;

dbms\_output.put\_line('Total number of patients in the hospital is ' || b);

end;

procedure doctor\_count

as

c number;

begin

select count(did) into c from doctor;

dbms\_output.put\_line('Total number of doctors in the hospital is ' || c);

end;

end hospi;

/

**EXECUTION AND OUTPUT**

