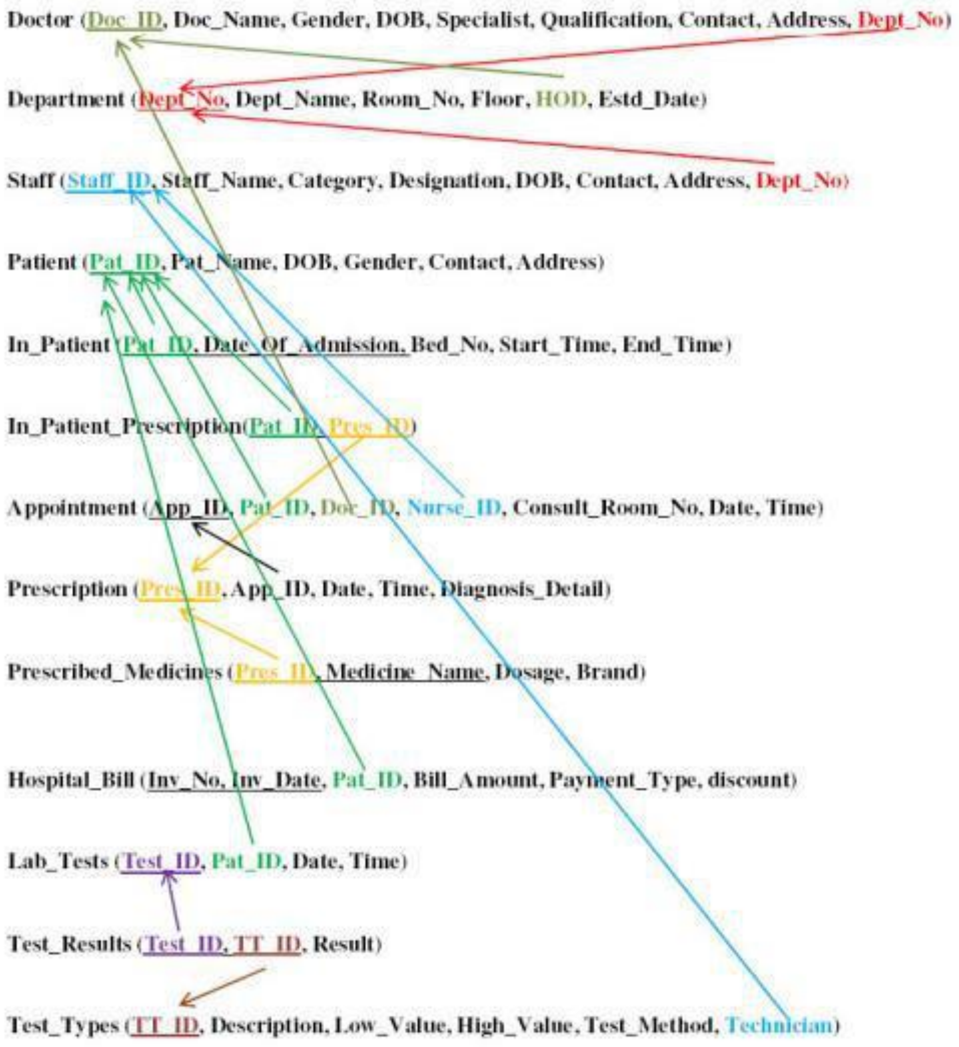
CSE2004- Database Management Systems

Digital Assignment-3

(Cycle Sheet Number- 3)

Name- Omkar Kulkarni
Reg no- 19BIT0196
Slot – L13+L14, D2



Doctor (Doc_ID, Doc_Name, Gender, DOB, Specialist, Qualification, Contact, Address, Dept_No)

Department (Dept_No, Dept_Name, Room_No, Floor, HOD, Estd_Date)

Staff (Staff_ID, Staff_Name, Category, Designation, DOB, Contact, Address, Dept_No)

Patient (Pat_ID, Pat_Name, DOB, Gender, Contact, Address)

In_Patient (Pat_ID, Date_Of_Admission, Bed_No, Start_Time, End_Time)

In_Patient_Prescription(Pat_ID, Pres_ID)

Appointment (App_ID, Pat_ID, Doc_ID, Nurse_ID, Consult_Room_No, Date, Time)

Prescription (Pres_ID, App_ID, Date, Time, Diagnosis_Detail)

Prescribed_Medicines (Pres_ID, Medicine_Name, Dosage, Brand)

Hospital_Bill (Inv_No, Inv_Date, Pat_ID, Bill_Amount, Payment_Type, discount)

Lab_Tests (Test_ID, Pat_ID, Date, Time)

Test_Results (Test_ID, TT_ID, Result)

Test_Types (TT_ID, Description, Low_Value, High_Value, Test_Method, Technician)

Creating Tables:

```
Create table Doctor(
doc_id varchar2(20) constraint docid primary key,
doc_name varchar2(30) not null,
gender varchar2(1) check (gender='M' OR gender='F' OR
gender='T'), dob date not null,
specialist varchar2(25) not null,
qualification varchar2(15) not null,
contact number(10) not null ,
address varchar2(100) not null,
Dept_no varchar2(15)
);
create table Department(
dept_no varchar2(15) constraint deptno primary
key, dept_name varchar2(30) not null, room_no
varchar2(10) not null,
floor number(15) not null,
hod varchar2(20) not null,
constraint hod_fk foreign key(hod) references Doctor(doc_id),
estd_date date not null
);
create table Staff(
staff_id varchar2(15) constraint staff_id_pk primary key,
staff_name varchar2(25) not null, category_
varchar2(20) not null,
designation varchar2(15) not null,
dob date not null,
contact number(20) not null,
address varchar2(100) not null,
dept_no varchar2(15) not null,
constraint dep_no_fk foreign key(dept_no) references Department(dept_no)
```

```
);
create table Patient(
pat_id varchar2(20) constraint pat_id_pk primary
key, pat_name varchar2(30) not null, dob date not
null,
gender varchar2(1) check(gender='M' OR gender='F' OR gender='T'),
contact number(20) not null,
address varchar2(100) not null
);
create table In_Patient(
pat_id varchar2(20) constraint pat_id_ primary key,
doa date not null,
bed_no varchar2(5) not null,
start_time date not null,
end_time date,
constraint pat_id_fk foreign key (pat_id) references Patient (pat_id)
);
create table Appointment(
app_id varchar2(20) constraint app_id_pk primary
key, pat_id varchar2(20) not null,
constraint pat_id_fk1 foreign key(pat_id) references Patient
(pat_id), doc_id varchar2(20) not null,
constraint doc_id_fk foreign key(doc_id) references
Doctor(doc_id), nurse_id varchar2(20) not null,
constraint nurse_id_fk foreign key(nurse_id) references Staff (staff_id),
consult_room_no int not null,
date_ date not null,
Time int not null
);
create table Prescription(
pres_id varchar2(20) constraint pres_id_pk primary key,
app_id varchar2(20) not null,
constraint app_id_fk foreign key(app_id) references Appointment(app_id),
datepres date not null,
diagnosis_detail varchar2(20)
);
create table In_Patient_Prescription (
pat_id varchar2(20) constraint pat_id_pk1 primary key,
pres_id varchar2(20) not null,
constraint pres_id_fk foreign key(pres_id) references
Prescription(pres_id), constraint pat_id_fk2 foreign key(pat_id) references
Patient(pat_id) );
create table Prescribed_Medicines(
pres_id varchar2(25) constraint pres_id_pk1 primary
key, med_name varchar2(40) not null unique, dosage
varchar2(20),
brand varchar2(10),
constraint pres_id_fk1 foreign key (pres_id) references Prescription(pres_id)
);
create table Hospital_Bill(
inv_no number(15) constraint inv_no_pk primary
key, inv_date date not null,
pat_id varchar2(20) not null,
constraint pat_id_fk3 foreign key(pat_id) references Patient(pat_id),
bill_am number(25) not null,
pay_type varchar2(25) not null,
discount number(20) check (discount>=0 AND discount <100)
);
create table Lab_Tests(
test_id varchar2(20) constraint test_id_pk primary
key, pat_id varchar2(20) not null,
constraint pat_id_fk5 foreign key(pat_id) references Patient(pat_id)
date_lab date not null,
time_lab int not null
);
create table Test_Types(
tt_id varchar2(20) constraint tt_id_pk primary key,
des varchar2(30) not null, low_val number(25) not
null,
high_val number(25) not null,
test_met varchar2(25) not null,
technician varchar2(20) not null,
constraint technician_fk foreign key(technician) references Staff(staff_id)
);
create table Test_Results(
test_id varchar2(20) constraint test_id_pk1 primary key,
tt_id varchar2(20) not null,
results varchar2(10) not null,
constraint tt_id_fk foreign key(tt_id) references Test_Types(tt_id),
constraint test_id_fk foreign key(test_id) references
Lab_Tests(test_id) );

date_lab date not null,
```

```
time_lab int not null
);
create table Test_Types(
tt_id varchar2(20) constraint tt_id_pk primary key,
des varchar2(30) not null, low_val number(25) not
null,
high_val number(25) not null,
test_met varchar2(25) not null,
technician varchar2(20) not null,
constraint technician_fk foreign key(technician) references Staff(staff_id)
);
create table Test_Results(
test_id varchar2(20) constraint test_id_pk1 primary key,

tt_id varchar2(20) not null,

results varchar2(10) not null,

constraint tt_id_fk foreign key(tt_id) references Test_Types(tt_id),

constraint test_id_fk foreign key(test_id) references

Lab_Tests(test_id) );
```

Data Insertion:

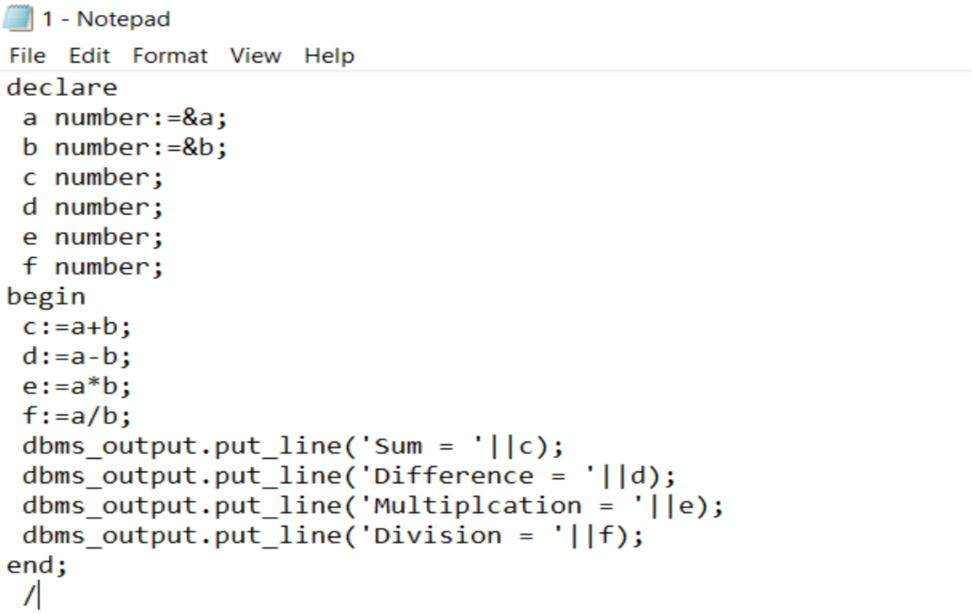
```
insert into Doctor values('D0001','Raghavan','M','13-September-
1975','Cardiology','MBBS',9882881929,'232,B1,Jodhpur','DP0001');

insert into Doctor values('D0002','Manish','M','24-NOVEMBER-
1986','Cardiology','BAMS',9421895758,'A1,M2,Jodhpur','DP0001');

insert into Doctor values('D0003','Shruti','F','25-February-
1990','Neurology','MD',9781545451,'1035,Ajmer','DP0002');

insert into Doctor values('D0004','Varun','M','01-April-
1989','Oncology','MD',9464154512,'9512,Jaipur','DP0004');

insert into Doctor values('D0005','Praveen','M','21-November-1993','General
Medicine','MD',9624412484,'1028,Bhilwara','DP0005');



insert into Department values('DP0001','Cardiology','B-131',1,'D0002','09-May-2018');

insert into Department values('DP0002','Neurology','C-226',2,'D0002','17-August-2015');

insert into Department values('DP0003','Opthalmology','A-512',19,'D0003','24-JULY-2016');

insert into Department values('DP0004','Oncology','M-918',9,'D0004','24-July-2019');

insert into Department values('DP0005','General Medicine','G-512',15,'D0005','18-July-2020');



insert into Staff values('S0001','Tanvi','Nurse','Staff nurse','31-July-
1996',9467984778,'12,A2,Jodhpur','DP0001');

insert into Staff values('S0002','Neetu','Nurse','Staff Nurse','15-January-
2000',9427438928,'16,DP2,Jodhpur','DP0002');

insert into Staff values('S0003','Gauri','Nurse','Staff Nurse','18-June-
1996',9421862561,'Palghar','DP0003');

insert into Staff values('S0004','Namira','Nurse','Staff Nurse','15-October-
1998',978945121,'Beawar','DP0004');
```

```
insert into Staff values('S0005','Divij','Ward Boy','Ward Boy','18-December-
2000',978951124,'Ajmer','DP0005');


insert into Patient values('P101','Mani','15-December-
1998','M',8674829478,'M2,K9 Street,Jodhpur');

insert into Patient values('P220','Steve','20-January-2000','M',9476835991,'23,U45,Pali');

insert into Patient values('P303','Gayle','21-MARCH-2001','M',944548412,'1026,Pali'); insert

into Patient values('P004','Karthik','21-June-1995','M',944587122,'651,Udaipur'); insert into

Patient values('P055','Karthik','11-December-2010','M',9451212145,'152,Pali');


insert into In_Patient values('P101','14-Mar-2017','B101','14-Mar-2017','15-Mar-2017');

insert into In_Patient values('P220','09-Apr-2017','B101','20-July-2020','30-July 2020'); insert

into In_Patient values('P303','20-July-2020','B102','21-July-2020','25-July-2020'); insert into

In_Patient values('P004','30-July-2009','B104','30-July-2020','3-August-2020'); insert into

In_Patient values('P055','1-August-2020','B111','1-August-2020','1-AUGUST 2019');


insert into Appointment values('A0001','P101','D0001','S0001',111,'11-July-2017',14);

insert into Appointment values('A0002','P220','D0002','S0002',112,'12-January-2020',14);

insert into Appointment values('A0003','P303','D0003','S0003',122,'12-February-2020',15);

insert into Appointment values('A0004','P004','D0004','S0004',111,'15-January-2020',19);

insert into Appointment values('A0005','P055','D0005','S0005',131,'22-January-2020',7);
insert into Prescription values('PR0001','A0001','16-July-2017','Artery Blockage');

insert into Prescription values('PR0002','A0002','20-July-2019','Anxiety Attack');

insert into Prescription values('PR0003','A0003','21-July-2020','Cardiac Arrest');

insert into Prescription values('PR0004','A0004','29-July-2019','Accident');

insert into Prescription values('PR0005','A0005','1-August-2020','Common Cold');



insert  into  In_Patient_Prescription  values('P101','PR0001');

insert  into  In_Patient_Prescription  values('P220','PR0002');

insert  into  In_Patient_Prescription  values('P303','PR0003');

insert  into  In_Patient_Prescription  values('P004','PR0004');

insert into In_Patient_Prescription values('P055','PR0005');

insert into Prescribed_Medicines values('PR0001','REMDESIVIR','Thrice a day','Ravonil');

insert into Prescribed_Medicines values('PR0002','Cyclopham','Once a day','Medirev');

insert into Prescribed_Medicines values('PR0003','Lidocaine','Twice a day','Revital');

insert into Prescribed_Medicines values('PR0004','Respira','Twice a day','Revital'); insert

into Prescribed_Medicines values('PR0005','Combiflam','After every meal','Cipla');



insert into Lab_Tests values('T0001','P101','18-July-2020',18);

insert into Lab_Tests values('T0002','P220','30-July-2020',10);

insert into Lab_Tests values('T0003','P303','25-July-2020',22);

insert into Lab_Tests values('T0004','P004','5-August-2019',5);

insert into Lab_Tests values('T0005','P055','1-August-2020',18);



insert into Test_Types values('TT0001','URINE TEST',12,2.2,'LAB','S0001');

insert into Test_Types values('TT0002','CAT Scan',9.6,13.6,'Lab','S0002');

insert into Test_Types values('TT0003','Blood Test',90,130,'LAB','S0003');

insert into Test_Types values('TT0004','blood glucose level',35.2,37.5,'LAB','S0001');
insert into Test_Types values('TT0005','Normal Checkup',97.3,97.4,'Cabin','S0005');
insert into Test_Results values('T0001','TT0001','Positive');
insert into Test_Results values('T0002','TT0002','Negative');
insert into Test_Results values('T0003','TT0003','Positive');
insert into Test_Results values('T0004','TT0004','Negative');
insert into Test_Results values('T0005','TT0005','Negative');



insert into Hospital_Bill values(9,'26-July-2020','P101',6000,'Cash',10); insert
into Hospital_Bill values(1,'04-August-2020','P220',40000,'Cash',25);

insert into Hospital_Bill values(5,'1-August-2020','P303',500000,'Card',0);
insert into Hospital_Bill values(6,'1-August-2020','P004',5000,'Card',0);
insert into Hospital_Bill values(7,'1-August-2021','P055',500,'Card',0);
```

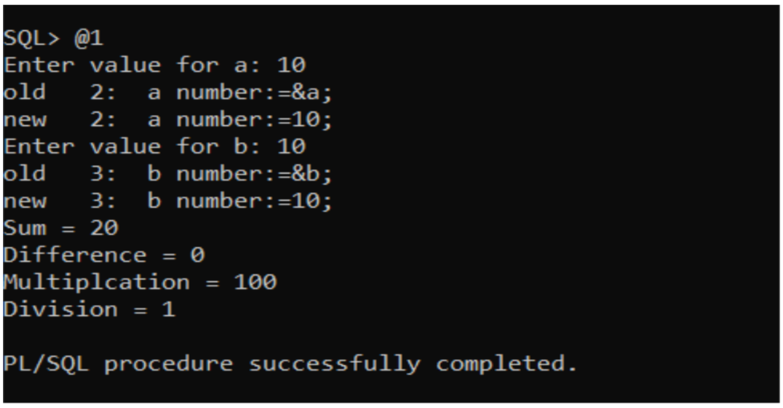
1. Write a PL/SQL program to implement a simple calculator.

```
declare
 a number:=&a;
 b number:=&b;
 c number;
 d number;
 e number;
 f num
ber;
begin
c:=a+b
; d:=a-
b;
e:=a*b
```

```
;

f:=a/b;

dbms_output.put_line('Sum = '||c);

dbms_output.put_line('Difference =

'||d);

dbms_output.put_line('Multiplcation =

'||e); dbms_output.put_line('Division =

'||f); end;
```



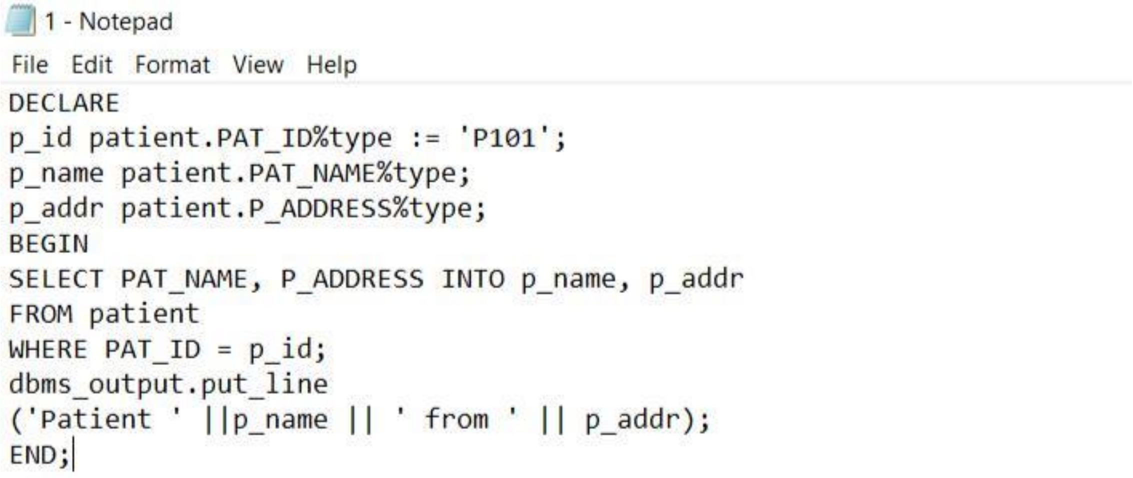
```
declare
 a number:=&a;
 b number:=&b;
 c number;
 d number;
 e number;
 f number;
begin
 c:=a+b;
 d:=a-b;
 e:=a*b;
 f:=a/b;
 dbms_output.put_line('Sum = '||c);
 dbms_output.put_line('Difference = '||d);
 dbms_output.put_line('Multiplcation = '||e);
 dbms_output.put_line('Division = '||f);
end;
 /
```



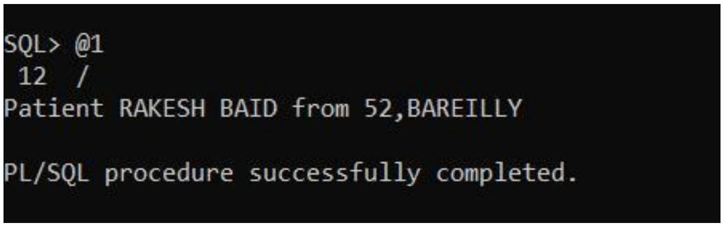
```
SQL> @1
Enter value for a: 10
old   2:  a number:=&a;
new   2:  a number:=10;
Enter value for b: 10
old   3:  b number:=&b;
new   3:  b number:=10;
Sum = 20
Difference = 0
Multiplcation = 100
Division = 1

PL/SQL procedure successfully completed.
```

2) Write a PL/SQL program to practice reading the record from a table into local variables using different data types and %TYPE and display the same using locally declared variables.

```
DECLARE

p_id patient.PAT_ID%type := 'P101';

p_name patient.PAT_NAME%type;

p_addr patient.P_ADDRESS%type;

BEGIN

SELECT PAT_NAME, P_ADDRESS INTO p_name, p_addr

FROM patient

WHERE PAT_ID = p_id;

dbms_output.put_line

('Patient ' ||p_name || ' from ' || p_addr);

END;
```



```
DECLARE
p_id patient.PAT_ID%type := 'P101';
p_name patient.PAT_NAME%type;
p_addr patient.P_ADDRESS%type;
BEGIN
SELECT PAT_NAME, P_ADDRESS INTO p_name, p_addr
FROM patient
WHERE PAT_ID = p_id;
dbms_output.put_line
('Patient ' ||p_name || ' from ' || p_addr);
END;
```



```
SQL> @1
 12  /
Patient RAKESH BAID from 52,BAREILLY

PL/SQL procedure successfully completed.
```

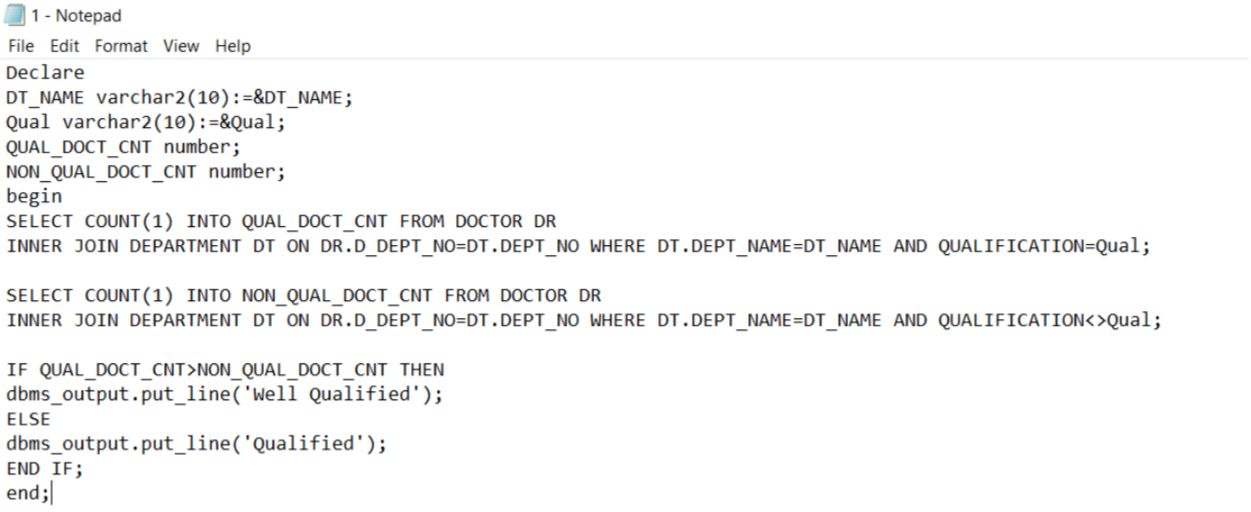
3) Write a PL/SQL program to find the number of doctors in a given department with a given qualification (read values for department and qualification from user during runtime). If number is more than the number of doctors in that department with other qualifications then display ‘Well qualified’ else ‘Qualified’.

```
Declare

DT_NAME varchar2(10):=&DT_NAME;

Qual varchar2(10):=&Qual;

QUAL_DOCT_CNT number;

NON_QUAL_DOCT_CNT number;

begin

SELECT COUNT(1) INTO QUAL_DOCT_CNT FROM DOCTOR DR

INNER JOIN DEPARTMENT DT ON DR.D_DEPT_NO=DT.DEPT_NO WHERE
DT.DEPT_NAME=DT_NAME AND QUALIFICATION=Qual;


SELECT COUNT(1) INTO NON_QUAL_DOCT_CNT FROM DOCTOR DR

INNER JOIN DEPARTMENT DT ON DR.D_DEPT_NO=DT.DEPT_NO WHERE
DT.DEPT_NAME=DT_NAME AND QUALIFICATION<>Qual;


IF QUAL_DOCT_CNT>NON_QUAL_DOCT_CNT

THEN dbms_output.put_line('Well Qualified'); ELSE
```

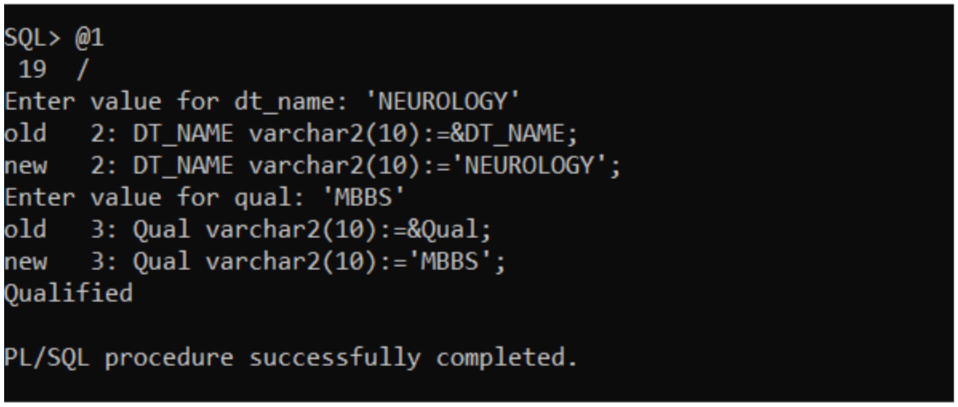
dbms_output.put_line('Qualified');

END IF;

end;



```
Declare
DT_NAME varchar2(10):=&DT_NAME;
Qual varchar2(10):=&Qual;
QUAL_DOCT_CNT number;
NON_QUAL_DOCT_CNT number;
begin
SELECT COUNT(1) INTO QUAL_DOCT_CNT FROM DOCTOR DR
INNER JOIN DEPARTMENT DT ON DR.D_DEPT_NO=DT.DEPT_NO WHERE DT.DEPT_NAME=DT_NAME AND QUALIFICATION=Qual;

SELECT COUNT(1) INTO NON_QUAL_DOCT_CNT FROM DOCTOR DR
INNER JOIN DEPARTMENT DT ON DR.D_DEPT_NO=DT.DEPT_NO WHERE DT.DEPT_NAME=DT_NAME AND QUALIFICATION<>Qual;

IF QUAL_DOCT_CNT>NON_QUAL_DOCT_CNT THEN
dbms_output.put_line('Well Qualified');
ELSE
dbms_output.put_line('Qualified');
END IF;
end;
```

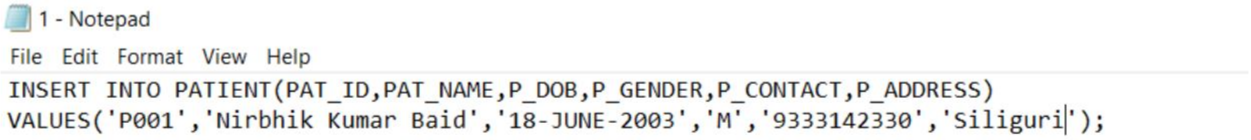


```
SQL> @1
 19  /
Enter value for dt_name: 'NEUROLOGY'
old   2: DT_NAME varchar2(10):=&DT_NAME;
new   2: DT_NAME varchar2(10):='NEUROLOGY';
Enter value for qual: 'MBBS'
old   3: Qual varchar2(10):=&Qual;
new   3: Qual varchar2(10):='MBBS';
Qualified

PL/SQL procedure successfully completed.
```

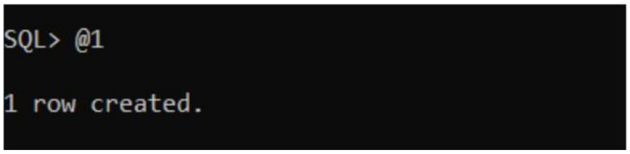
4) Write a PL/SQL program to insert records into any of the tables in your database.

INSERT INTO PATIENT(PAT_ID,PAT_NAME,P_DOB,P_GENDER,P_CONTACT,P_ADDRESS)

VALUES('P001','Nirbhik Kumar Baid','18-JUNE-2003','M','9333142330','Siliguri');



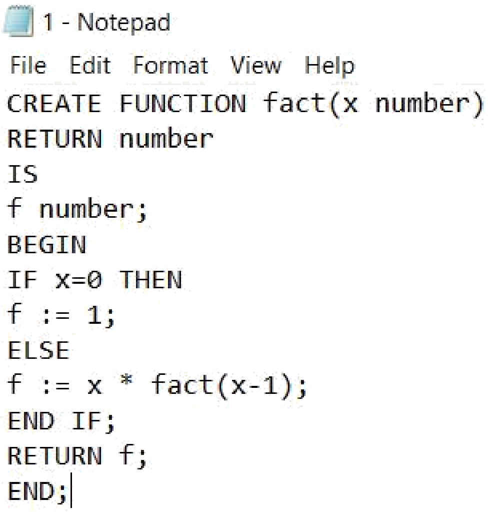
```
INSERT INTO PATIENT(PAT_ID,PAT_NAME,P_DOB,P_GENDER,P_CONTACT,P_ADDRESS)
VALUES('P001','Nirbhik Kumar Baid','18-JUNE-2003','M','9333142330','Siliguri');
```



```
SQL> @1

1 row created.
```

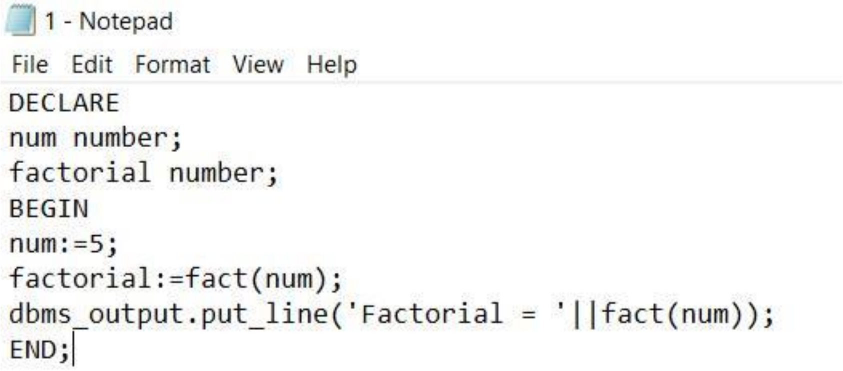
5) Create a function to find the factorial of a given number.

```
CREATE FUNCTION fact(x number)
RETURN number
IS
f number;
BEGIN
IF x=0
THEN f := 1;
ELSE
f := x * fact(x-1);
END IF;
RETURN
f; END;
```

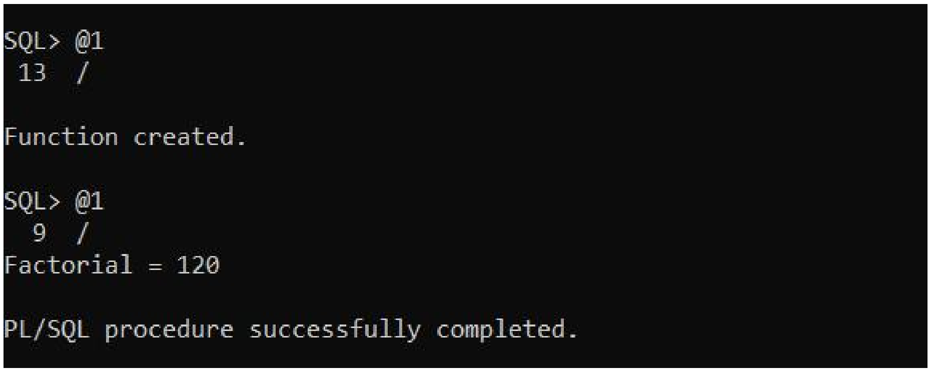


```
CREATE FUNCTION fact(x number)
RETURN number
IS
f number;
BEGIN
IF x=0 THEN
f := 1;
ELSE
f := x * fact(x-1);
END IF;
RETURN f;
END;
```

```
DECLARE
num number
factorial number;
BEGIN
num:=5;
factorial:=fact(num);
dbms_output.put_line('Factorial = '||fact(num));
END;
```



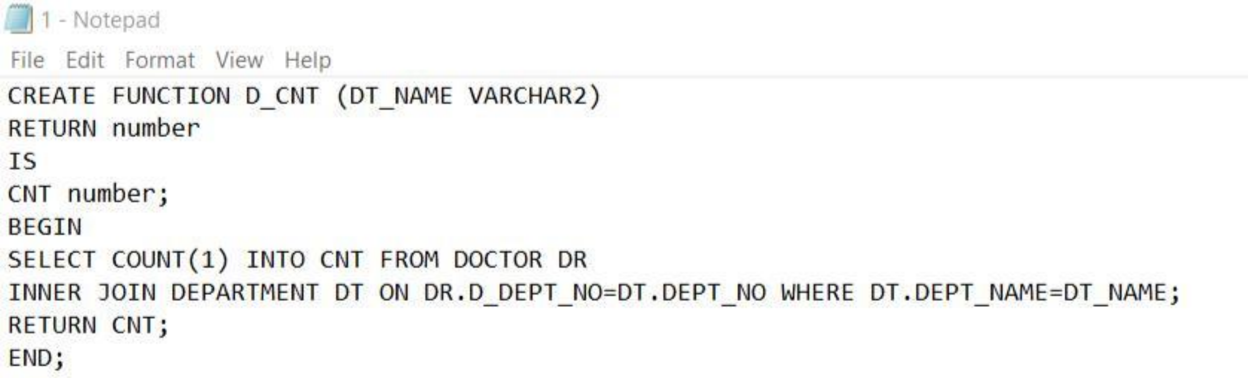
```
DECLARE
num number;
factorial number;
BEGIN
num:=5;
factorial:=fact(num);
dbms_output.put_line('Factorial = '||fact(num));
END;
```



```
SQL> @1
 13  /

Function created.

SQL> @1
  9  /
Factorial = 120

PL/SQL procedure successfully completed.
```

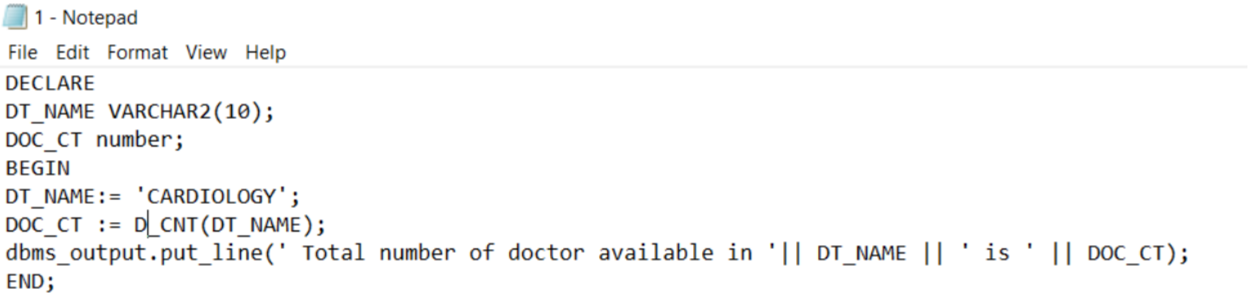
6) Create a function DOC_COUNT to find the number of doctors in the given department. Use the department name as the input parameter for the function.

```
CREATE FUNCTION D_CNT (DT_NAME VARCHAR2)

RETURN number

IS

CNT number;

BEGIN

SELECT COUNT(1) INTO CNT FROM DOCTOR DR

INNER JOIN DEPARTMENT DT ON DR.D_DEPT_NO=DT.DEPT_NO
WHERE DT.DEPT_NAME=DT_NAME;

RETURN CNT;

END;
```

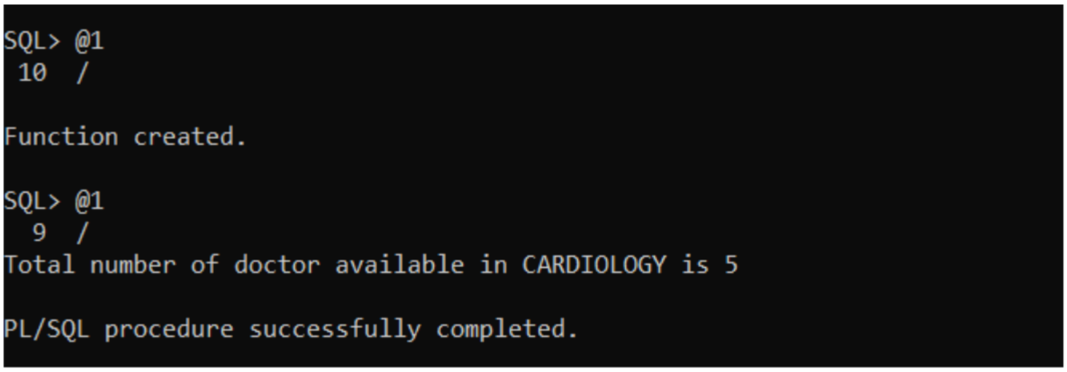


```
CREATE FUNCTION D_CNT (DT_NAME VARCHAR2)
RETURN number
IS
CNT number;
BEGIN
SELECT COUNT(1) INTO CNT FROM DOCTOR DR
INNER JOIN DEPARTMENT DT ON DR.D_DEPT_NO=DT.DEPT_NO WHERE DT.DEPT_NAME=DT_NAME;
RETURN CNT;
END;
```

```
DECLARE
DT_NAME VARCHAR2(10);
DOC_CT number;
BEGIN
DT_NAME:= 'CARDIOLOGY';
DOC_CT := D_CNT(DT_NAME);
dbms_output.put_line(' Total number of doctor available in '|| DT_NAME || ' is ' ||
DOC_CT); END;
```



```
DECLARE
DT_NAME VARCHAR2(10);
DOC_CT number;
BEGIN
DT_NAME:= 'CARDIOLOGY';
DOC_CT := D_CNT(DT_NAME);
dbms_output.put_line(' Total number of doctor available in '|| DT_NAME || ' is ' || DOC_CT);
END;
```



```
SQL> @1
 10  /

Function created.

SQL> @1
  9  /
Total number of doctor available in CARDIOLOGY is 5

PL/SQL procedure successfully completed.
```

CURSORS:

1.--- C**URSOR to give 5% additional discount to all senior citizen patients.**

```
SET SERVEROUTPUT ON;

DECLARE

   v_name VARCHAR2(20);

   v_amt NUMBER(20);

   CURSOR cur_x IS

   SELECT h.Pat_id,h.discount FROM Hospital_Bill as h

   WHERE h.pat_id=(select pa.pat_id from Patient as pa where pa.dob<'20-Oct-1960');

BEGIN

   UPDATE Hospital_Bill

   SET discount = discount+0.05

   WHERE Hospital_Bill.Pat_ID=(select pa.pat_id from Patient as pa where pa.dob<'20-Oct-1960');

 OPEN cur_x;

 LOOP

  FETCH cur_x INTO v_name,v_amt;

  dbms.output.put_line(v_name||' '||v_amt);

  EXIT WHEN cur_x%NOTFOUND;

 END LOOP;

 CLOSE cur_x;

END;
```

2.--- CURSOR to change the department number from 1 as 5 for all doctors with a qualification ‘MD’

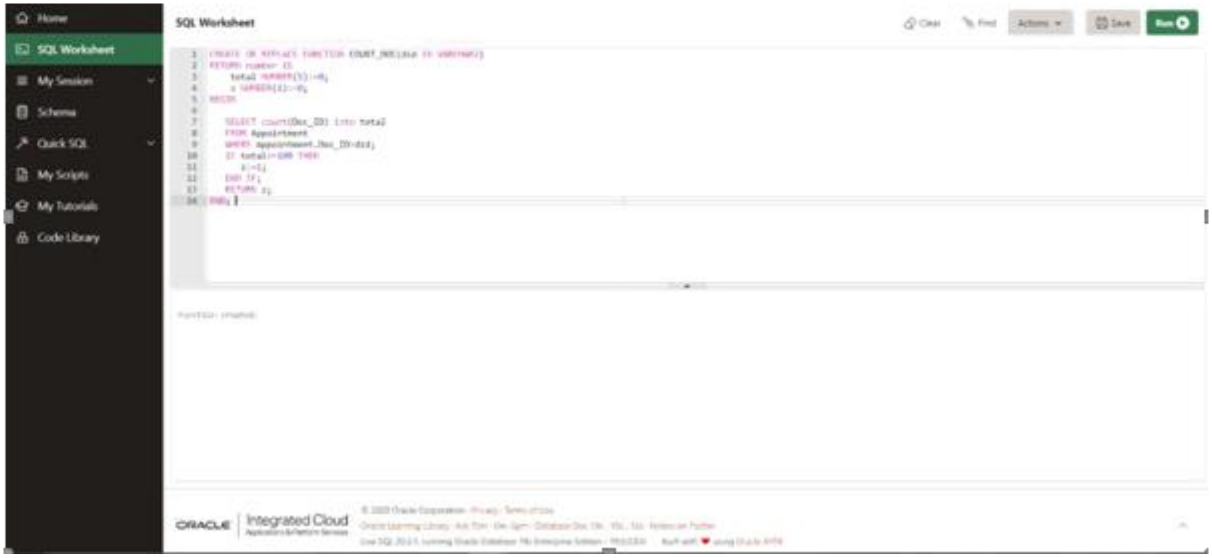
```
SET SERVEROUTPUT ON;

DECLARE

   v_name VARCHAR2(20);

   v_dept_no NUMBER(5);

   CURSOR cur_x IS

   SELECT Doc_Name,Dept_No FROM Doctor

   WHERE Doctor.Dept_No=Department.Dept_No AND Doctor.Qualification='MD';

BEGIN

   UPDATE Department,Doctor

   SET Dept_No=5

   WHERE Doctor.Dept_No=Department.Dept_No AND Doctor.Qualification='MD';

 OPEN cur_x;

 LOOP

  FETCH cur_x INTO v_name,v_dept_no;

  dbms.output.put_line(v_name||' '||v_dept_no);

  EXIT WHEN cur_x%NOTFOUND;

 END LOOP;

 CLOSE cur_x;

END;
```

FUNCTIONS AND PROCEDURES:

1.-- PL/SQL stored function COUNT_DOC to count the number of doctors who have treated at least 100 patients if given a doctor id as input parameter.

```
CREATE OR REPLACE FUNCTION COUNT_DOC(did IN VARCHAR2)

RETURN number IS

    total NUMBER(5):=0;

    a NUMBER(1):=0;

BEGIN
```

```
  SELECT count(Doc_ID) into total
  FROM Appointment
  WHERE Appointment.Doc_ID=did; IF
  total>=100 THEN
    z:=1;
  END IF;
  RETURN z;
END;
```

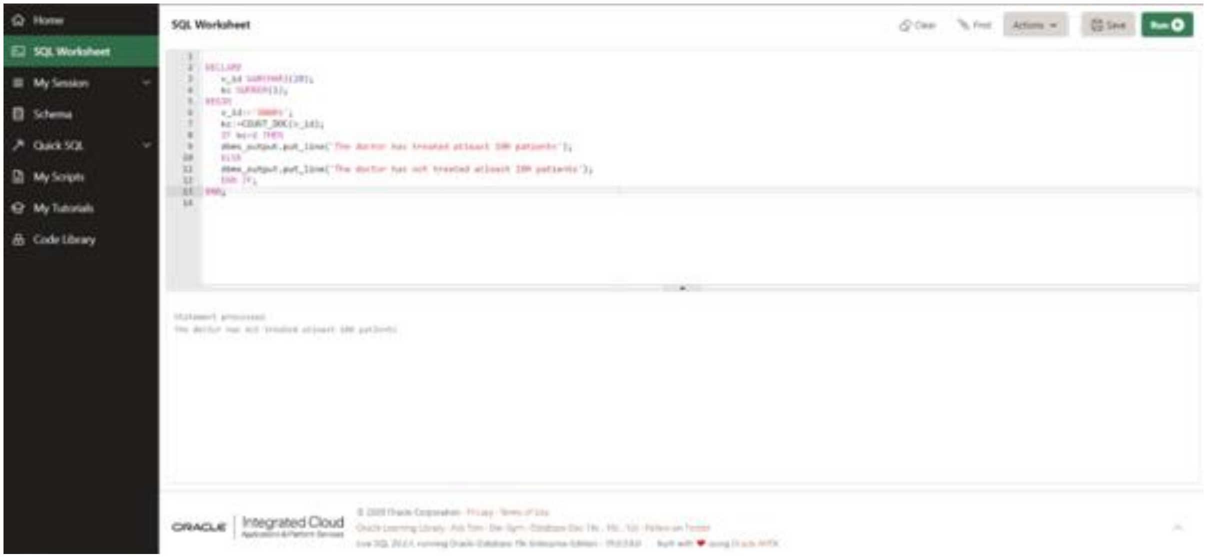


```
CREATE OR REPLACE FUNCTION COUNT_DOC(did IN VARCHAR2)
RETURN number IS
    total NUMBER(5):=0;
    z NUMBER(1):=0;
BEGIN

   SELECT count(Doc_ID) into total
   FROM Appointment
   WHERE Appointment.Doc_ID=did;
   IF total>=100 THEN
     z:=1;
   END IF;
   RETURN z;
END;
```

```
Function created.
```

```
DECLARE
  v_id VARCHAR2(20);
  kc NUMBER(1);
BEGIN
  v_id:='D0001';
  kc:=COUNT_DOC(v_id);
  IF kc=1 THEN
  dbms_output.put_line('The doctor has treated atleast 100 patients');
  ELSE
  dbms_output.put_line('The doctor has not treated atleast 100 patients');
  END IF;
```

```
END;
```

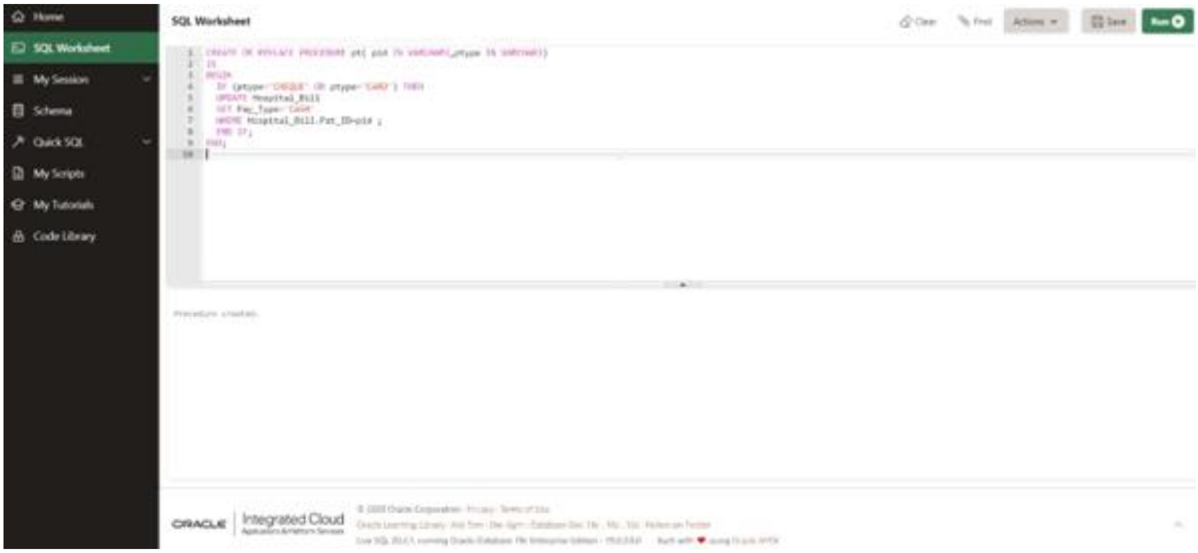


```
DECLARE
    v_id VARCHAR2(20);
    kc NUMBER(3);
BEGIN
    v_id:='[illegible]';
    kc:=COUNT_DOC(v_id);
    IF kc>=[illegible] THEN
    dbms_output.put_line('The doctor has treated atleast 100 patients');
    ELSE
    dbms_output.put_line('The doctor has not treated atleast 100 patients');
    END IF;
END;
```

```
Statement processed.
The doctor has not treated atleast 100 patients
```

2.--- PL/SQL stored procedure to adjust the payment type of hospital bills to CASH if the patient id and amount details given as input.

```
CREATE OR REPLACE PROCEDURE pt( pid IN VARCHAR2,ptype IN VARCHAR2)

IS

BEGIN

  IF (ptype='CHEQUE' OR ptype='CARD') THEN

  UPDATE Hospital_Bill

  SET Pay_Type='CASH'

  WHERE Hospital_Bill.Pat_ID=pid ;

  END IF;

END;
```



```
CREATE OR REPLACE PROCEDURE pt( pid IN VARCHAR2,ptype IN VARCHAR2))
IS
BEGIN
  IF (ptype='CHEQUE' OR ptype='CARD') THEN
  UPDATE Hospital_Bill
  SET Pay_Type='CASH'
  WHERE Hospital_Bill.Pat_ID=pid ;
  END IF;
END;
```

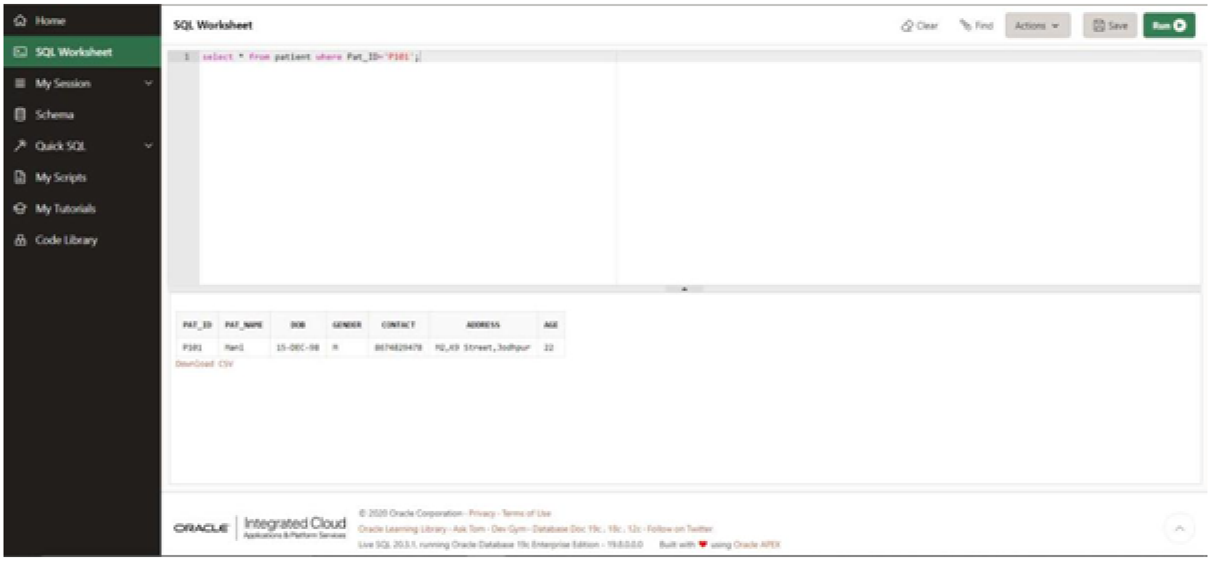
```
Procedure created.
```

TRIGGERS:

1.--- Add an attribute with patients table to store the age of the patients. Then answer the following question; 1. Write a Trigger to find and fill the age of a patient whenever a patient record is inserted into patients table.

```
ALTER TABLE Patient

ADD Age NUMBER(2);


CREATE OR REPLACE PROCEDURE uage(pid VARCHAR2)

IS

nage number;

ndob date;

BEGIN
select dob into ndob from patient where pat_ID=pid; select
   (sysdate-to_date(ndob))/365 into nage from dual;


  Update Patient

  set Patient.age=nage

where Pat_ID=pid;
```

```
END;

CREATE OR REPLACE TRIGGER p_age

BEFORE INSERT ON Patient

FOR EACH ROW

ENABLE

BEGIN

  uage('P101');

END;

INSERT INTO Patient VALUES('101','Ravi','19-Oct-2019','M','9856487235','Vasant Kunj,Delhi',60);
select * from patient where Pat_ID='P101';
```



SQL Worksheet

```
select * from patient where Pat_ID='P101';
```

PAT_ID	PAT_NAME	DOB	GENDER	CONTACT	ADDRESS	AGE
P101	Mani	15-DEC-98	M	8674820478	M2,X9 Street,Jodhpur	22

2.--- Create a table EMP_SALARY with attributes ID, Basic, DA, HRA, Deduction,

Net_Salary. Here, ID refers the Staff_ID of staff table. Treat ‘Net_Salary’ as a derived attribute and don’t insert a value through insert operation. The value for Net Salary can be calculated as follows; Net_Salary = Basic + DA + HRA – Deduction

2. Write a Trigger to perform the following; whenever new staff is recruited and a designation is assigned, insert an appropriate record into EMP_SALARY table. Refer the following table for salary details.

Designation	**Basic**	**DA**	**HRA**	**Deduction**
Staff nurse	**6000**	**2000**	**2000**	**2% of basic**
Head nurse	**8000**	**2500**	**3000**	**2% of basic**

Technician	**6000**	**2000**	**2000**	**2% of basic**
Senior technician	**9000**	**2500**	**3500**	**2.5% of basic**
Junior attender	**5000**	**1500**	**2000**	**2% of basic**
Senior attender	**6500**	**2000**	**2000**	**2% of basic**

```
CREATE TABLE EMP_SALARY(ID NUMBER(10),Basic Number(10),DA Number(10),HRA
Number(10),Deduction Number(10));

ALTER TABLE EMP_SALARY

ADD Designation VARCHAR2(10);

ALTER TABLE EMP_SALARY

Net_Salary Number(10) AS (Basic+DA+HRA-Deduction);

CREATE OR REPLACE FUNCTION Desret(eid IN VARCHAR2)

return number IS

   des varchar(20);

BEGIN

  SELECT Designation into des

  FROM EMP_SALARY

  WHERE 'ID'= eid;

  IF EMP_SALARY.Designation='Staff Nurse' THEN

    update EMP_SALARY

    set Basic=6000,DA=2000,HRA=2000,Deduction=120

    where Designation='Staff Nurse';

  END IF;

  IF EMP_SALARY.Designation='Head Nurse' THEN

    update EMP_SALARY

    set Basic=8000,DA=2500,HRA=3000,Deduction=160

    where Designation='Head Nurse';

   END IF;

   IF EMP_SALARY.Designation='Technician' THEN
```

```
    update EMP_SALARY

    set Basic=6000,DA=2000,HRA=2000,Deduction=120

    where Designation='Technician';

    END IF;

END;

CREATE OR REPLACE TRIGGER emp_rec

BEFORE INSERT ON EMP_SALARY

FOR EACH ROW

ENABLE

BEGIN

  Desret('110');

END;

INSERT INTO EMP_SALARY VALUES('110',6000,2000,2000,120,'Staff Nurse');

INSERT INTO EMP_SALARY VALUES('111',8000,2500,3000,160,'Head Nurse');

INSERT INTO EMP_SALARY VALUES('112',6000,2000,2000,120,'Technician');

INSERT INTO EMP_SALARY VALUES('113',9000,2500,3500,225,'Senior Technician');

INSERT INTO EMP_SALARY VALUES('114',5000,1500,2000,100,'Junior Attender');

INSERT INTO EMP_SALARY VALUES('115',6500,2000,2000,130,'Senior Attender');
```