**Expt No.: 3** **Name: Swaranjana Nayak**
**Date: 17/10/2020** **Reg. No.: 19BCE0977**

# CYCLE SHEET 3

**Aim:**
To solve the given problems by implementing in SQL.

**Hospital Database**

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(nurse, lab technician, cashier, security), 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 (cash/credit card/debit card), 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)

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)

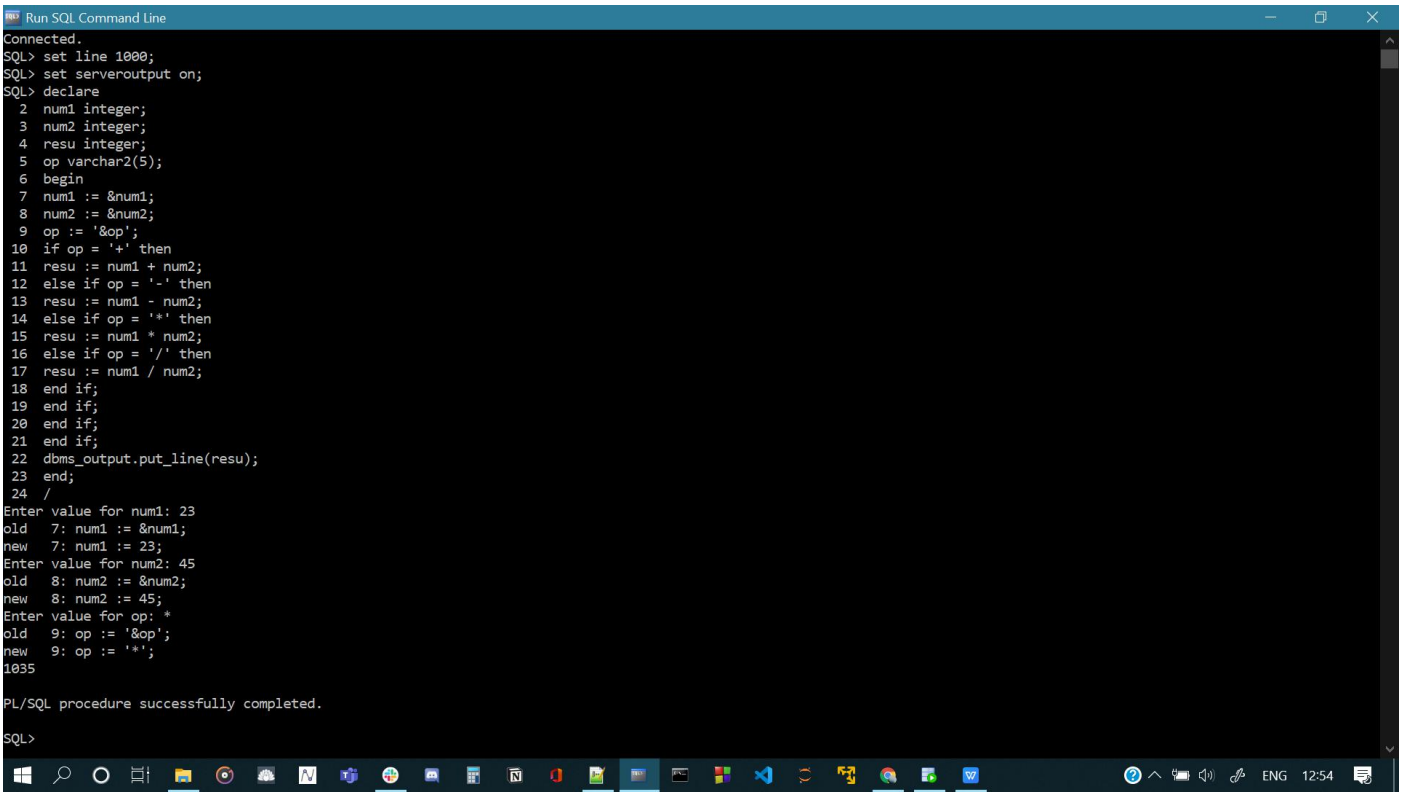
Figure 1: Primary key and foreign keys

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

### Code:

```
declare
num1 integer;
num2 integer;
resu integer;
op varchar2(5);
begin
num1 := &num1;
num2 := &num2;
op := '&op';
if op = '+' then
resu := num1 + num2;
else if op = '-' then
resu := num1 - num2;
else if op = '*' then
resu := num1 * num2;
else if op = '/' then
resu := num1 / num2;
end if;
end if;
end if;
end if;
dbms_output.put_line(resu);
end;
```

### Output:



**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.**

## Code:

```
declare
did doctor.doc_id%type;
dna doctor.doc_name%type;
gen doctor.gender%type;
dob doctor.dob%type;
spe doctor.specialist%type;
con doctor.contact%type;
dep doctor.dept_no%type;
begin
did := '&did';
select doc_name into dna from doctor where doc_id = did;
select gender into gen from doctor where doc_id = did;
select dob into dob from doctor where doc_id = did;
select specialist into spe from doctor where doc_id = did;
select contact into con from doctor where doc_id = did;
select dept_no into dep from doctor where doc_id = did;
dbms_output.put_line('Doctor ID:');
dbms_output.put_line(did);
dbms_output.put_line('Doctor Name: ');
dbms_output.put_line(dna);
dbms_output.put_line('Doctor Gender: ');
dbms_output.put_line(gen);
dbms_output.put_line('Specialization: ');
dbms_output.put_line(spe);
dbms_output.put_line('Contact: ');
dbms_output.put_line(con);
dbms_output.put_line('Department Number: ');
dbms_output.put_line(dep);
end;
```

## Output:

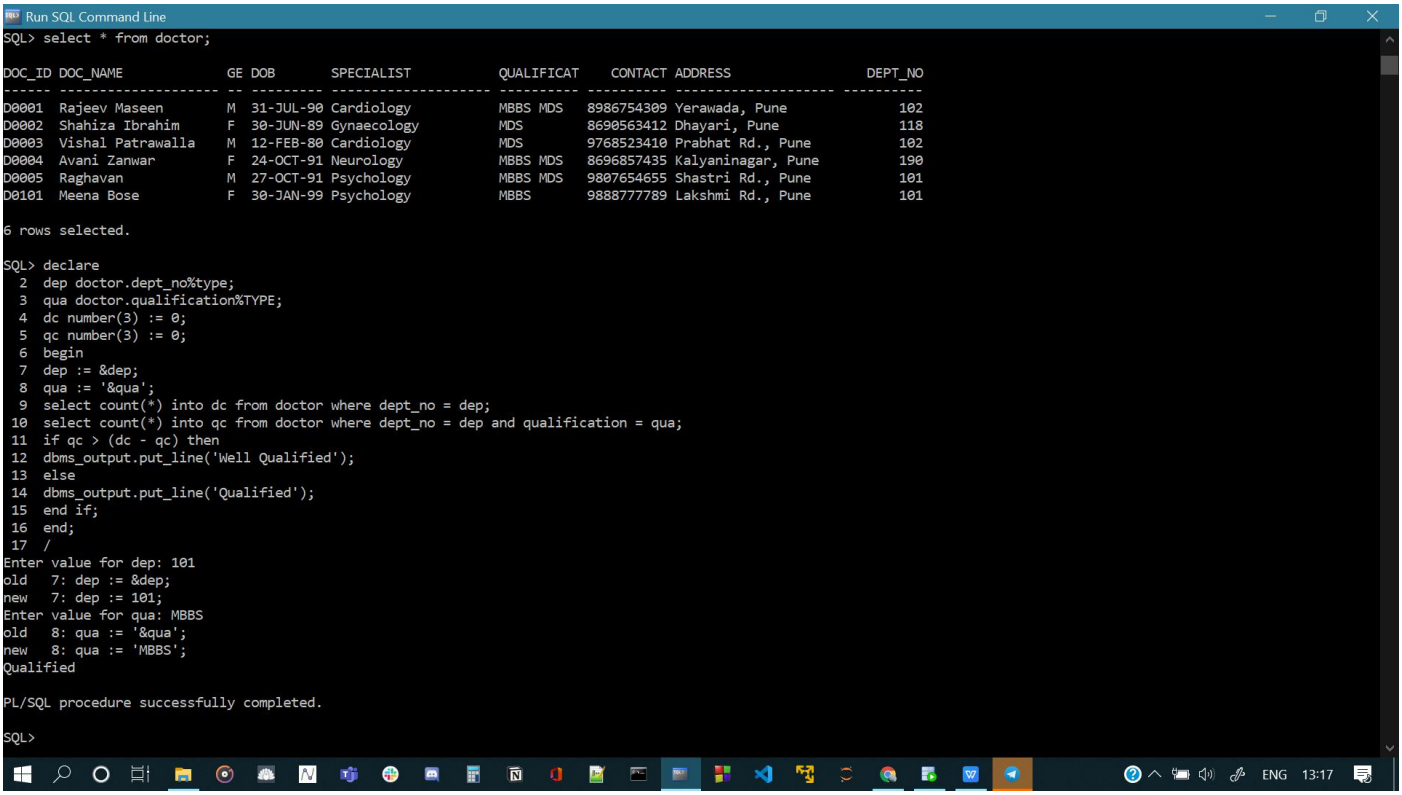
```
Run SQL Command Line
  9  begin
 10  did := '&did';
 11  select doc_name into dna from doctor where doc_id = did;
 12  select gender into gen from doctor where doc_id = did;
 13  select dob into dob from doctor where doc_id = did;
 14  select specialist into spe from doctor where doc_id = did;
 15  select contact into con from doctor where doc_id = did;
 16  select dept_no into dep from doctor where doc_id = did;
 17  dbms_output.put_line('Doctor ID:');
 18  dbms_output.put_line(did);
 19  dbms_output.put_line('Doctor Name: ');
 20  dbms_output.put_line(dna);
 21  dbms_output.put_line('Doctor Gender: ');
 22  dbms_output.put_line(gen);
 23  dbms_output.put_line('Specialization: ');
 24  dbms_output.put_line(spe);
 25  dbms_output.put_line('Contact: ');
 26  dbms_output.put_line(con);
 27  dbms_output.put_line('Department Number: ');
 28  dbms_output.put_line(dep);
 29  end;
 30  /
Enter value for did: D0001
old  10: did := '&did';
new  10: did := 'D0001';
Doctor ID:
D0001
Doctor Name:
Rajeev Maseen
Doctor Gender:
M
Specialization:
Cardiology
Contact:
8986754309
Department Number:
102

PL/SQL procedure successfully completed.

SQL>
```

**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'.**

**Code:**

```
declare
dep doctor.dept_no%type;
qua doctor.qualification%TYPE;
dc number(3) := 0;
qc number(3) := 0;
begin
dep := &dep;
qua := '&qua';
select count(*) into dc from doctor where dept_no = dep;
select count(*) into qc from doctor where dept_no = dep and qualification = qua;
if qc > (dc - qc) then
dbms_output.put_line('Well Qualified');
else
dbms_output.put_line('Qualified');
end if;
end;
```

**Output:**



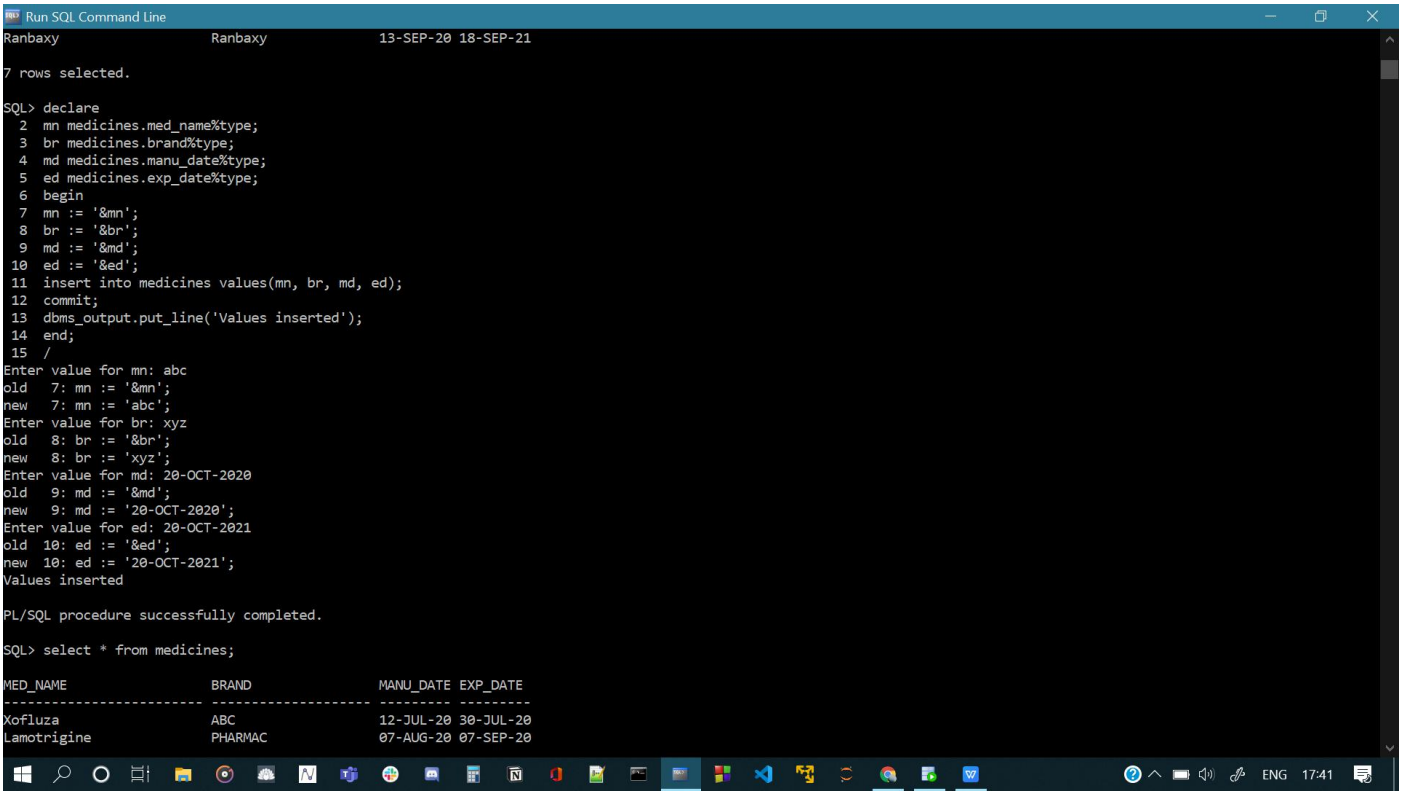
```
Run SQL Command Line
SQL> select * from doctor;

DOC_ID DOC_NAME             GE DOB       SPECIALIST           QUALIFICAT    CONTACT ADDRESS                DEPT_NO
------ -------------------- -- --------- -------------------- ---------- ---------- -------------------- ----------
D0001  Rajeev Maseen        M  31-JUL-90 Cardiology           MBBS MDS   8986754309 Yerawada, Pune              102
D0002  Shahiza Ibrahim      F  30-JUN-89 Gynaecology          MDS        8690563412 Dhayari, Pune               118
D0003  Vishal Patrawalla    M  12-FEB-80 Cardiology           MDS        9768523410 Prabhat Rd., Pune           102
D0004  Avani Zanwar         F  24-OCT-91 Neurology            MBBS MDS   8696857435 Kalyaninagar, Pune          190
D0005  Raghavan             M  27-OCT-91 Psychology           MBBS MDS   9807654655 Shastri Rd., Pune           101
D0101  Meena Bose           F  30-JAN-99 Psychology           MBBS       9888777789 Lakshmi Rd., Pune           101

6 rows selected.

SQL> declare
  2  dep doctor.dept_no%type;
  3  qua doctor.qualification%TYPE;
  4  dc number(3) := 0;
  5  qc number(3) := 0;
  6  begin
  7  dep := &dep;
  8  qua := '&qua';
  9  select count(*) into dc from doctor where dept_no = dep;
 10  select count(*) into qc from doctor where dept_no = dep and qualification = qua;
 11  if qc > (dc - qc) then
 12  dbms_output.put_line('Well Qualified');
 13  else
 14  dbms_output.put_line('Qualified');
 15  end if;
 16  end;
 17  /
Enter value for dep: 101
old   7: dep := &dep;
new   7: dep := 101;
Enter value for qua: MBBS
old   8: qua := '&qua';
new   8: qua := 'MBBS';
Qualified

PL/SQL procedure successfully completed.

SQL>
```

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

**Code:**

```
SQL> declare
  2  mn medicines.med_name%type;
  3  br medicines.brand%type;
  4  md medicines.manu_date%type;
  5  ed medicines.exp_date%type;
  6  begin
  7  mn := '&mn';
  8  br := '&br';
  9  md := '&md';
 10  ed := '&ed';
 11  insert into medicines values(mn, br, md, ed);
 12  commit;
 13  dbms_output.put_line('Values inserted');
 14  end;
 15  /
```

**Output:**



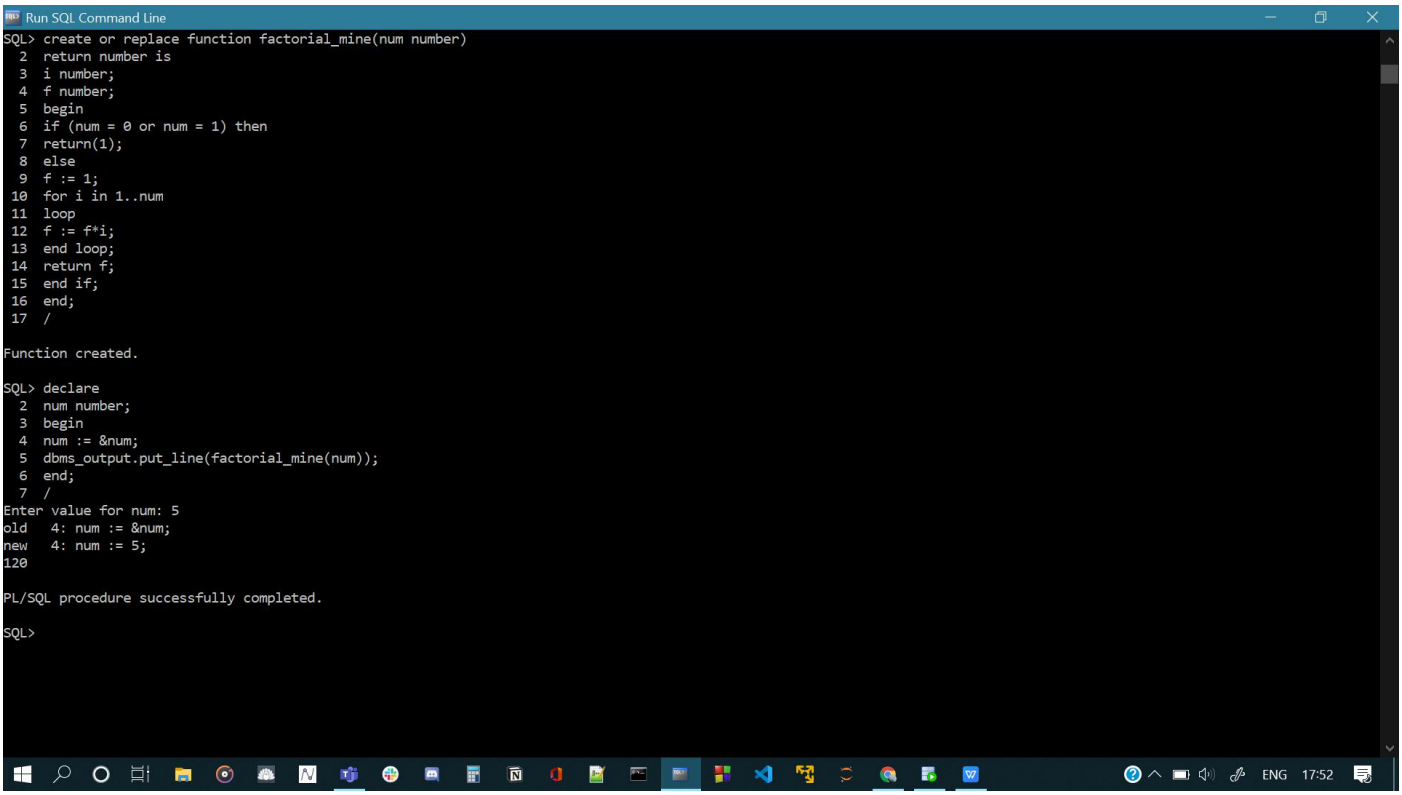
```
Run SQL Command Line
Ranbaxy                  Ranbaxy              13-SEP-20 18-SEP-21

7 rows selected.

SQL> declare
  2  mn medicines.med_name%type;
  3  br medicines.brand%type;
  4  md medicines.manu_date%type;
  5  ed medicines.exp_date%type;
  6  begin
  7  mn := '&mn';
  8  br := '&br';
  9  md := '&md';
 10  ed := '&ed';
 11  insert into medicines values(mn, br, md, ed);
 12  commit;
 13  dbms_output.put_line('Values inserted');
 14  end;
 15  /
Enter value for mn: abc
old   7: mn := '&mn';
new   7: mn := 'abc';
Enter value for br: xyz
old   8: br := '&br';
new   8: br := 'xyz';
Enter value for md: 20-OCT-2020
old   9: md := '&md';
new   9: md := '20-OCT-2020';
Enter value for ed: 20-OCT-2021
old  10: ed := '&ed';
new  10: ed := '20-OCT-2021';
Values inserted

PL/SQL procedure successfully completed.

SQL> select * from medicines;

MED_NAME                 BRAND                MANU_DATE EXP_DATE
------------------------ -------------------- --------- ---------
Xofluza                  ABC                  12-JUL-20 30-JUL-20
Lamotrigine              PHARMAC              07-AUG-20 07-SEP-20
```

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

### Code:

```
SQL> create or replace function factorial_mine(num number)
  2  return number is
  3  i number;
  4  f number;
  5  begin
  6  if (num = 0 or num = 1) then
  7  return(1);
  8  else
  9  f := 1;
 10  for i in 1..num
 11  loop
 12  f := f*i;
 13  end loop;
 14  return f;
 15  end if;
 16  end;
 17  /

SQL> declare
  2  num number;
  3  begin
  4  num := &num;
  5  dbms_output.put_line(factorial_mine(num));
  6  end;
  7  /
```

### Output:

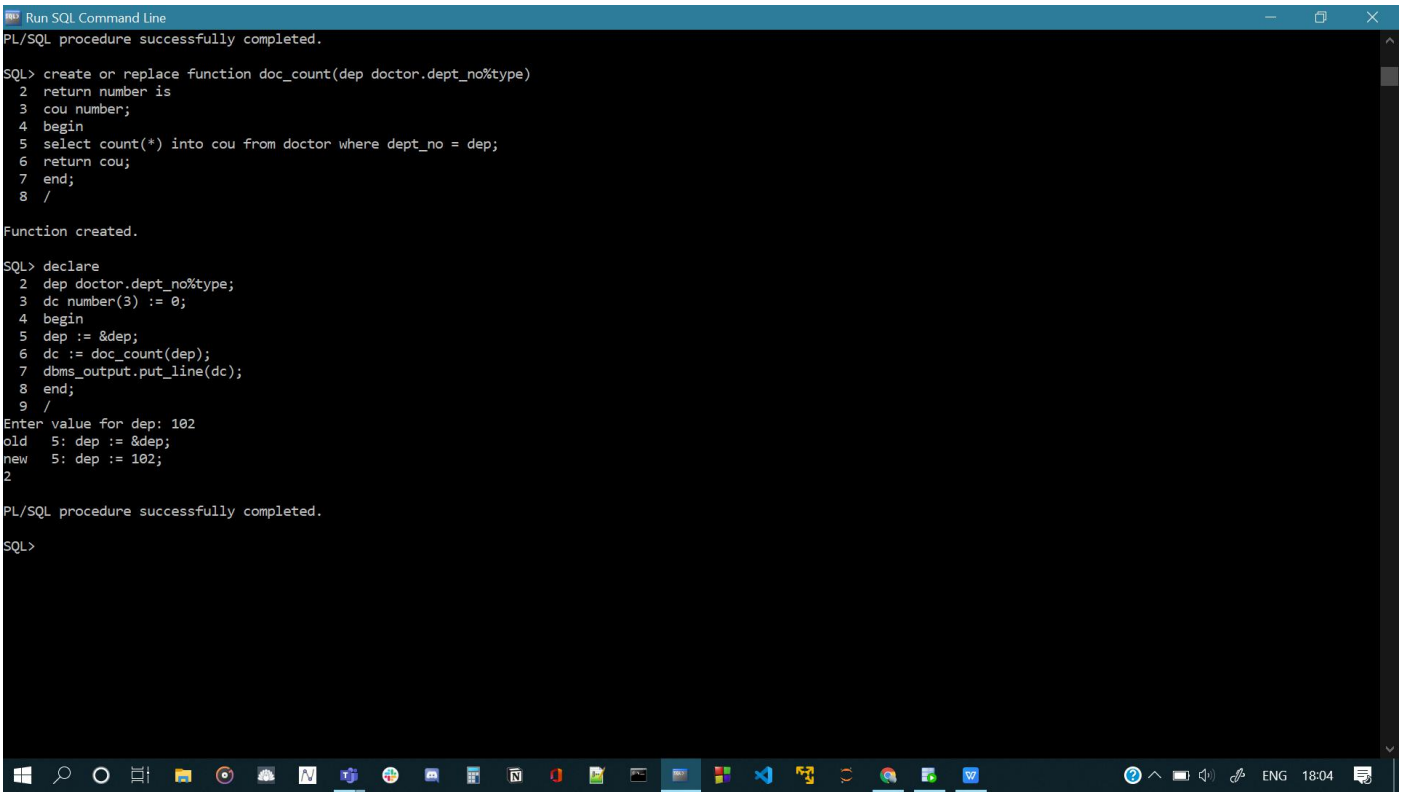


**6. Create a function DOC_COUNT to find the number of doctors in the given department. Use the department name as the input parameter for the function.**

**Code:**

```
create or replace function doc_count(dep doctor.dept_no%type)
return number is
cou number;
begin
select count(*) into cou from doctor where dept_no = dep;
return cou;
end;
declare
dep doctor.dept_no%type;
dc number(3) := 0;
begin
dep := &dep;
dc := doc_count(dep);
dbms_output.put_line(dc);
end;
```

**Output:**



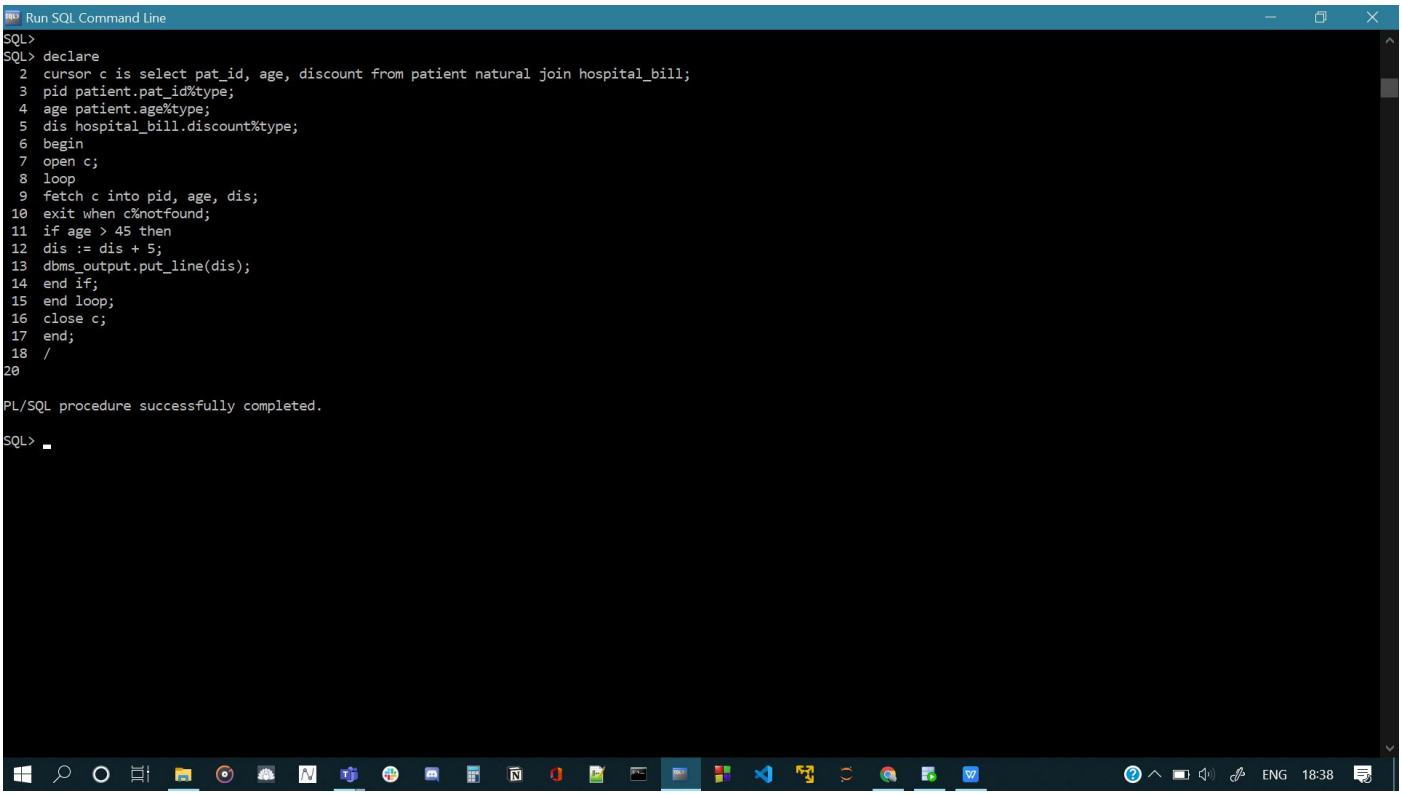
# Cursors:

## 1. Write a CURSOR to give 5% additional discount to all senior citizen patients.

### Code:

```
SQL> declare
  2  cursor c is select pat_id, age, discount from patient natural join hospital_bill;
  3  pid patient.pat_id%type;
  4  age patient.age%type;
  5  dis hospital_bill.discount%type;
  6  begin
  7  open c;
  8  loop
  9  fetch c into pid, age, dis;
 10  exit when c%notfound;
 11  if age > 45 then
 12  dis := dis + 5;
 13  dbms_output.put_line(dis);
 14  end if;
 15  end loop;
 16  close c;
 17  end;
 18  /
```

### Output:



**2. Write a CURSOR to change the department number from 1 as 5 for all doctors with a qualification 'MD'.**

**Code:**

```
declare
cursor c1 is select doc_id, qualification, dept_no from doctor;
did doctor.doc_id%type;
qua doctor.qualification%type;
dep doctor.dept_no%type;
begin
open c1;
loop
fetch c1 into did, qua, dep;
exit when c1%notfound;
if (qua = 'MD' and dep = 1) then
update doctor set dept_no = 5 where doc_id = did;
dbms_output.put_line('Updated');
end if;
end loop;
close c1;
end;
```

**Output:**

```
Run SQL Command Line
D0102  name                M  21-OCT-80 Psychology          MD        9999999999 address                     1

7 rows selected.

SQL> declare
  2  cursor c1 is select doc_id, qualification, dept_no from doctor;
  3  did doctor.doc_id%type;
  4  qua doctor.qualification%type;
  5  dep doctor.dept_no%type;
  6  begin
  7  open c1;
  8  loop
  9  fetch c1 into did, qua, dep;
 10  exit when c1%notfound;
 11  if (qua = 'MD' and dep = 1) then
 12  update doctor set dept_no = 5 where doc_id = did;
 13  dbms_output.put_line('Updated');
 14  end if;
 15  end loop;
 16  close c1;
 17  end;
 18  /
Updated

PL/SQL procedure successfully completed.

SQL> select * from doctor;

DOC_ID DOC_NAME             GE DOB       SPECIALIST           QUALIFICAT    CONTACT ADDRESS                DEPT_NO
------ -------------------- -- --------- -------------------- ---------- ---------- -------------------- ----------
D0001  Rajeev Maseen        M  31-JUL-90 Cardiology           MBBS MDS   8986754309 Yerawada, Pune              102
D0002  Shahiza Ibrahim      F  30-JUN-89 Gynaecology          MDS        8690563412 Dhayari, Pune               118
D0003  Vishal Patrawalla    M  12-FEB-80 Cardiology           MDS        9768523410 Prabhat Rd., Pune           102
D0004  Avani Zanwar         F  24-OCT-91 Neurology            MBBS MDS   8696857435 Kalyaninagar, Pune          190
D0005  Raghavan             M  27-OCT-91 Psychology           MBBS MDS   9807654655 Shastri Rd., Pune           101
D0101  Meena Bose           F  30-JAN-99 Psychology           MBBS       9888777789 Lakshmi Rd., Pune           101
D0102  name                 M  21-OCT-80 Psychology           MD         9999999999 address                       5

7 rows selected.

SQL>
```

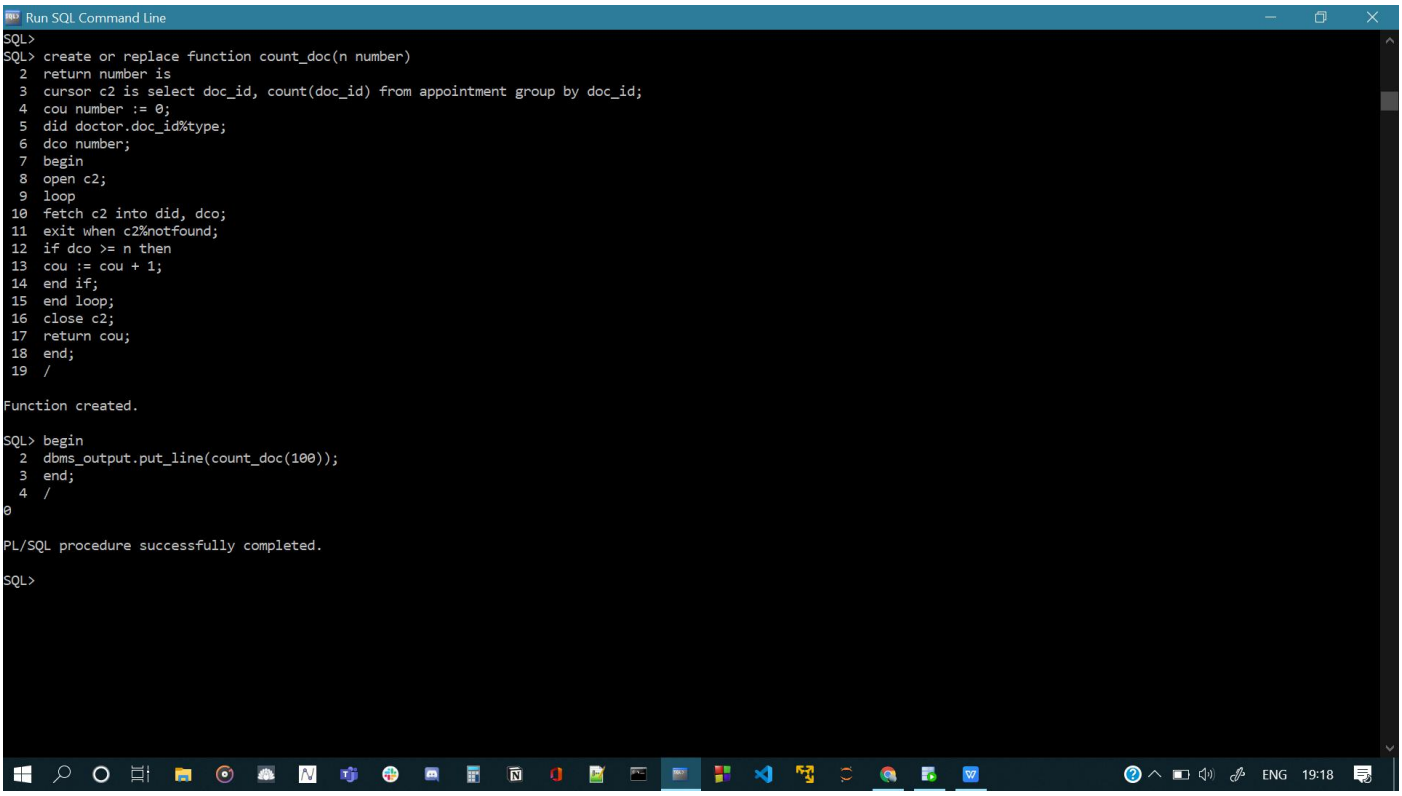
# Functions and Procedures:

**1. Write a 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.**

**Code:**

```
create or replace function count_doc(n number)
return number is
cursor c2 is select doc_id, count(doc_id) from appointment group by doc_id;
cou number := 0;
did doctor.doc_id%type;
dco number;
begin
open c2;
loop
fetch c2 into did, dco;
exit when c2%notfound;
if dco >= n then
cou := cou + 1;
end if;
end loop;
close c2;
return cou;
end;
```

**Output:**



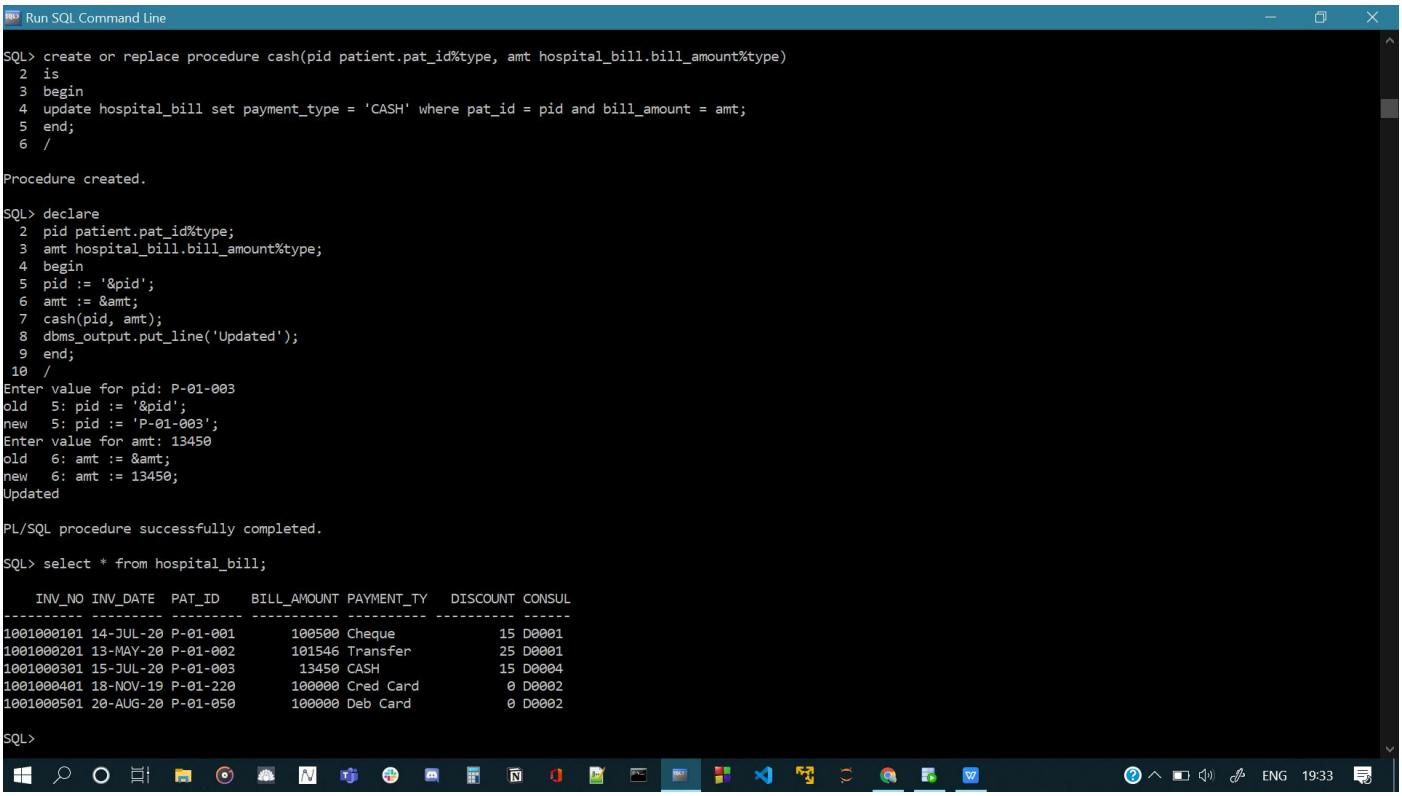
```
Run SQL Command Line
SQL>
SQL> create or replace function count_doc(n number)
  2  return number is
  3  cursor c2 is select doc_id, count(doc_id) from appointment group by doc_id;
  4  cou number := 0;
  5  did doctor.doc_id%type;
  6  dco number;
  7  begin
  8  open c2;
  9  loop
 10  fetch c2 into did, dco;
 11  exit when c2%notfound;
 12  if dco >= n then
 13  cou := cou + 1;
 14  end if;
 15  end loop;
 16  close c2;
 17  return cou;
 18  end;
 19  /

Function created.

SQL> begin
  2  dbms_output.put_line(count_doc(100));
  3  end;
  4  /
0

PL/SQL procedure successfully completed.

SQL>
```

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

**Code:**

```
create or replace procedure cash(pid patient.pat_id%type, amt hospital_bill.bill_amount%type)
is
begin
update hospital_bill set payment_type = 'CASH' where pat_id = pid and bill_amount = amt;
end;

declare
pid patient.pat_id%type;
amt hospital_bill.bill_amount%type;
begin
pid := '&pid';
amt := &amt;
cash(pid, amt);
dbms_output.put_line('Updated');
end;
```

**Output:**



```
Run SQL Command Line

SQL> create or replace procedure cash(pid patient.pat_id%type, amt hospital_bill.bill_amount%type)
  2  is
  3  begin
  4  update hospital_bill set payment_type = 'CASH' where pat_id = pid and bill_amount = amt;
  5  end;
  6  /

Procedure created.

SQL> declare
  2  pid patient.pat_id%type;
  3  amt hospital_bill.bill_amount%type;
  4  begin
  5  pid := '&pid';
  6  amt := &amt;
  7  cash(pid, amt);
  8  dbms_output.put_line('Updated');
  9  end;
 10  /
Enter value for pid: P-01-003
old   5: pid := '&pid';
new   5: pid := 'P-01-003';
Enter value for amt: 13450
old   6: amt := &amt;
new   6: amt := 13450;
Updated

PL/SQL procedure successfully completed.

SQL> select * from hospital_bill;

    INV_NO INV_DATE  PAT_ID    BILL_AMOUNT PAYMENT_TY   DISCOUNT CONSUL
---------- --------- --------- ----------- ---------- ---------- ------
1001000101 14-JUL-20 P-01-001       100500 Cheque             15 D0001
1001000201 13-MAY-20 P-01-002       101546 Transfer           25 D0001
1001000301 15-JUL-20 P-01-003        13450 CASH               15 D0004
1001000401 18-NOV-19 P-01-220       100000 Cred Card           0 D0002
1001000501 20-AUG-20 P-01-050       100000 Deb Card            0 D0002

SQL>
```

# Triggers:

**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.**

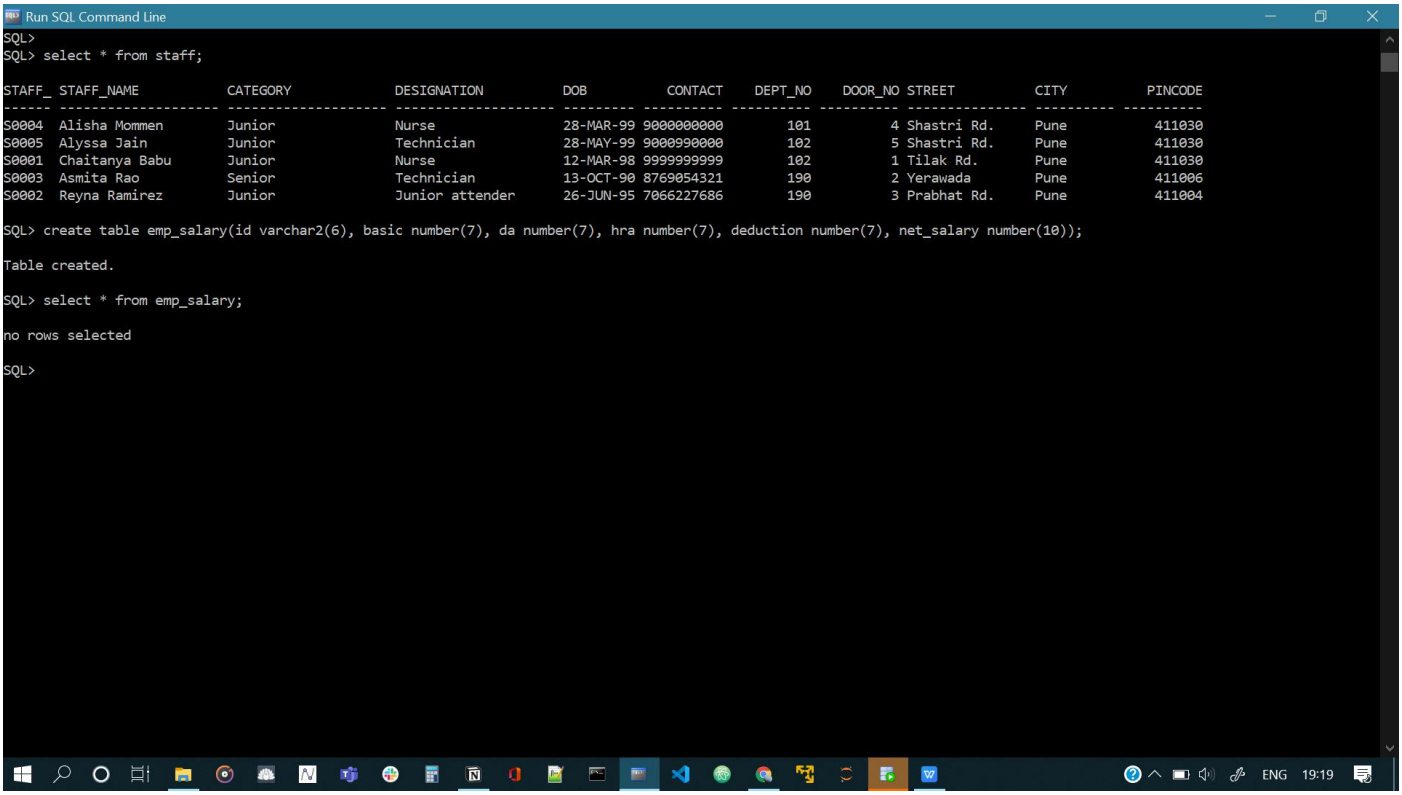
**Code:**

```
create or replace trigger age_patient
before insert on patient
for each row
declare
a patient.age%type;
pid patient.pat_id%type;
begin
pid := :old.pat_id;
a := 2020 - extract(year from :old.dob);
update patient set age = a where pat_id = pid;
end;
```

**Output:**

```
Run SQL Command Line

SQL> create or replace trigger age_patient
  2  before insert on patient
  3  for each row
  4  declare
  5  a patient.age%type;
  6  pid patient.pat_id%type;
  7  begin
  8  pid := :old.pat_id;
  9  a := 2020 - extract(year from :old.dob);
 10  update patient set age = a where pat_id = pid;
 11  end;
 12  /

Trigger created.

SQL> select * from patient;

PAT_ID    PAT_NAME             DOB       GE    CONTACT SYMPTOMS            STREET          CITY          PINCODE        AGE
--------- -------------------- --------- -- ---------- -------------------- --------------- ---------- ---------- ----------
P-01-001  Nina Torev           12-JUN-10 F  8897889709 Chest congestion     Tilak Rd.       Pune           411030         10
P-01-002  Kalyani Shah         14-JUL-01 F  9878987890 Loose motions        MG Rd.          Pune           411001         19
P-01-003  Ajit Adani           25-DEC-70 M  8976567890 Cold and weakness    Kalyaninagar    Pune           411006         50
P-03-004  Gayle                01-JAN-01 M  9990009990 Cold and weakness    Lakshmi Rd.     Pune           411030         19
P-01-101  Mani                 01-JAN-02 F  9991009990 Cold and weakness    Lakshmi Rd.     Pune           411030         18
P-01-220  Steve                01-JAN-03 M  9991109990 Cold and weakness    Lakshmi Rd.     Pune           411030         17
P-01-050  Karthik              01-JAN-04 M  9991119990 Cold and weakness    Lakshmi Rd.     Pune           411030         16

7 rows selected.

SQL> insert into patient(pat_id, pat_name, dob, gender, contact, symptoms, street, city, pincode) values('P-02-002', 'name', '13-JUL-1980','F', 9900990099, 'symptoms', 'str
eet', 'city', 411004);

1 row created.

SQL> select * from patient;

PAT_ID    PAT_NAME             DOB       GE    CONTACT SYMPTOMS            STREET          CITY          PINCODE        AGE
--------- -------------------- --------- -- ---------- -------------------- --------------- ---------- ---------- ----------
P-01-001  Nina Torev           12-JUN-10 F  8897889709 Chest congestion     Tilak Rd.       Pune           411030         10
P-01-002  Kalyani Shah         14-JUL-01 F  9878987890 Loose motions        MG Rd.          Pune           411001         19
```

**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**



```
Run SQL Command Line
SQL>
SQL> select * from staff;

STAFF_ STAFF_NAME           CATEGORY            DESIGNATION         DOB         CONTACT    DEPT_NO    DOOR_NO STREET          CITY          PINCODE
------ -------------------- -------------------- -------------------- --------- ---------- ---------- ---------- --------------- ---------- ----------
S0004  Alisha Mommen        Junior              Nurse               28-MAR-99 9000000000        101          4 Shastri Rd.     Pune           411030
S0005  Alyssa Jain          Junior              Technician          28-MAY-99 9000990000        102          5 Shastri Rd.     Pune           411030
S0001  Chaitanya Babu       Junior              Nurse               12-MAR-98 9999999999        102          1 Tilak Rd.       Pune           411030
S0003  Asmita Rao           Senior              Technician          13-OCT-90 8769054321        190          2 Yerawada        Pune           411006
S0002  Reyna Ramirez        Junior              Junior attender     26-JUN-95 7066227686        190          3 Prabhat Rd.     Pune           411004

SQL> create table emp_salary(id varchar2(6), basic number(7), da number(7), hra number(7), deduction number(7), net_salary number(10));

Table created.

SQL> select * from emp_salary;

no rows selected

SQL>
```

**1. 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.**

**Code:**

```
create or replace trigger insert_emp_salary
after insert on staff
for each row
enable
begin
if :new.Designation = 'Staff nurse' then
insert into emp_salary values (:new.Staff_Id, 6000, 6000, 2000, 2, (6000 + 6000 + 2000 - (0.02*6000)));
ELSIF :new.Designation = 'Head nurse' then
insert into emp_salary values (:new.Staff_Id, 8000, 2500, 3000, 2, (8000 + 2500 + 3000 - (0.2*8000)));
ELSIF :new.Designation = 'Technician' THEN
INSERT INTO emp_salary   VALUES (:new.Staff_Id, 6000, 2000, 2000, 2, (6000 + 2000 + 2000 - (0.2*6000)));
ELSIF :new.Designation = 'Senior technician' THEN
INSERT INTO emp_salary VALUES (:new.Staff_Id, 9000, 2500, 3500, 2, (9000 + 2500 + 3500 - (0.2*9000)));
ELSIF :new.Designation = 'Junior attender' THEN
INSERT INTO emp_salary VALUES (:new.Staff_Id, 5000, 1500, 2000, 2, (5000 + 1500 + 2000 - (0.2*5000)));
ELSIF :new.Designation = 'Senior attender' THEN
INSERT INTO emp_salary VALUES (:new.Staff_Id, 6500, 2000, 2000, 2, (6500 + 2000 + 2000 - (0.2*6500)));
```

```
END IF;
END;
```

**Output:**

```
Run SQL Command Line

Table created.

SQL> select * from emp_salary;

no rows selected

SQL> create or replace trigger insert_emp_salary
  2  after insert on staff
  3  for each row
  4  enable
  5  begin
  6  if :new.Designation = 'Staff nurse' then
  7  insert into emp_salary values (:new.Staff_Id, 6000, 6000, 2000, 2, (6000 + 6000 + 2000 - (0.02*6000)));
  8  ELSIF :new.Designation = 'Head nurse' then
  9  insert into emp_salary values (:new.Staff_Id, 8000, 2500, 3000, 2, (8000 + 2500 + 3000 - (0.2*8000)));
 10  ELSIF :new.Designation = 'Technician' THEN
 11  INSERT INTO emp_salary  VALUES (:new.Staff_Id, 6000, 2000, 2000, 2, (6000 + 2000 + 2000 - (0.2*6000)));
 12  ELSIF :new.Designation = 'Senior technician' THEN
 13  INSERT INTO emp_salary VALUES (:new.Staff_Id, 9000, 2500, 3500, 2, (9000 + 2500 + 3500 - (0.2*9000)));
 14  ELSIF :new.Designation = 'Junior attender' THEN
 15  INSERT INTO emp_salary VALUES (:new.Staff_Id, 5000, 1500, 2000, 2, (5000 + 1500 + 2000 - (0.2*5000)));
 16  ELSIF :new.Designation = 'Senior attender' THEN
 17  INSERT INTO emp_salary VALUES (:new.Staff_Id, 6500, 2000, 2000, 2, (6500 + 2000 + 2000 - (0.2*6500)));
 18  END IF;
 19  END;
 20  /

Trigger created.

SQL> INSERT INTO staff VALUES ('S0006', 'Jay', 'Attender', 'Junior attender', '27-JAN-1999', 9999990097, 101, 6, 'Ganeshnagar', 'Pune', 411041);

1 row created.

SQL> select * from emp_salary;

ID          BASIC         DA        HRA  DEDUCTION NET_SALARY
------ ---------- ---------- ---------- ---------- ----------
S0006        5000       1500       2000          2       7500

SQL> _
```