

SLIP-1

Q3. Consider the following entities and their relationships.

Client (client_no, client_name, address, birthdate)

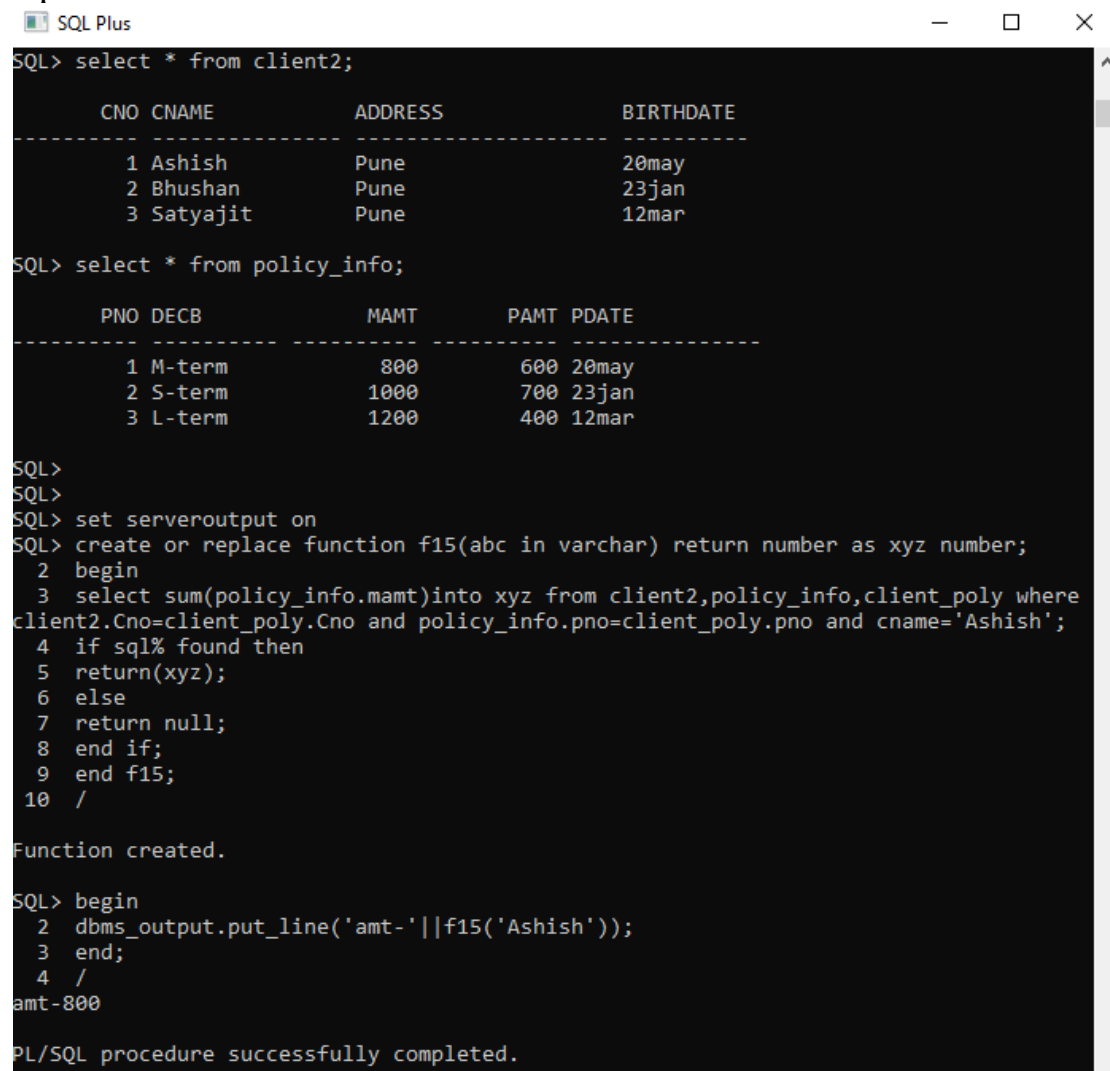
Policy_info (policy_no, desc, maturity_amt, prem_amt, date)

Relation between Client and Policy_info is Many to Many

Constraint: Primary key, prem_amt and maturity_amt should be > 0.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a function which will return total maturity amount of policies of a particular client.



```
SQL> select * from client2;
```

CNO	CNAME	ADDRESS	BIRTHDATE
1	Ashish	Pune	20may
2	Bhushan	Pune	23jan
3	Satyajit	Pune	12mar

```
SQL> select * from policy_info;
```

PNO	DECB	MAMT	PAMT	PDATE
1	M-term	800	600	20may
2	S-term	1000	700	23jan
3	L-term	1200	400	12mar

```
SQL>
SQL>
SQL> set serveroutput on
SQL> create or replace function f15(abc in varchar) return number as xyz number;
2 begin
3 select sum(policy_info.mamt)into xyz from client2,policy_info,client_poly where
client2.Cno=client_poly.Cno and policy_info.pno=client_poly.pno and cname='Ashish';
4 if sql% found then
5 return(xyz);
6 else
7 return null;
8 end if;
9 end f15;
10 /

Function created.

SQL> begin
2 dbms_output.put_line('amt-'||f15('Ashish'));
3 end;
4 /
amt-800

PL/SQL procedure successfully completed.
```

2. Write a cursor which will display policy date wise client details.

```
SQL Plus
SQL> set serveroutput on
SQL> declare
  2 cursor c1 is select  cname, address, pdate from client1, poly, cp
  where client1.cno=cp.cno and  poly.pno=cp.pno order by pdate;
  3 c c1%rowtype;
  4
  5 begin
  6 open c1;
  7 loop
  8 fetch c1 into c;
  9 exit when c1%notfound;
 10 dbms_output.put_line(c.cname||' '||c.address||' '||c.pdate);
 11 end loop;
 12 close c1;
 13 end;
 14 /
shreepune23jan

PL/SQL procedure successfully completed.

SQL>
```

SLIP-2

Q3. Consider the following Item_Supplier database

Item (itemno, itemname)

Supplier (supplier_No , supplier_name, address, city)

Relationship between Item and Supplier is many-to-many with descriptive attribute rate and quantity

Constraints: itemno ,supplier_No primary key

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write function to print the total number of suppliers of a particular item.

```
Select SQL Plus
SQL> select * from item;

   INO INAME
-----
     1 soap
     2 waffer
     3 biscuit

SQL> select * from supplier;

   SNO SNAME      ADDRESS      CITY
-----
   101 Ashish     loni         pune
   102 Satyajit   kand         pune
   103 Bhushan    pashan       pune

SQL> select * from i_s;

   INO      SNO      RATE      QUANTITY
-----
     1      101      250         3
     1      102      420         2
     3      103      180         7

SQL> set serveroutput on
SQL> create or replace function f1(abc in varchar) return number as xyz number;
2  begin
3  select sum(supplier.sno) into xyz from item,supplier,i_s where item.ino=i_s.ino and supplier.sno=i_s.sno and iname='soap';
4  if sql %found then
5  return (xyz);
6  else
7  return null;
8  end if;
9  end f1;
10 /

Function created.

SQL> begin
2  dbms_output.put_line ('item'||f1('soap'));
3  end;
4  /
item203
```

2. Write a trigger which will fire before insert or update on rate and quantity less than or equal to zero. (Raise user defined exception and give appropriate message)

```
SQL Plus
PL/SQL procedure successfully completed.

SQL> create or replace trigger t1 before insert or update on i_s
  2  for each row
  3  begin
  4  if (:new.quantity <= 0) then
  5  raise_application_error(-20001, 'quantity > 0');
  6  end if;
  7  end;
  8  /

Trigger created.

SQL>
```

```
SQL Plus

SQL> insert into i_s values (4,104,-100,-10);
insert into i_s values (4,104,-100,-10)
*
ERROR at line 1:
ORA-20001: quantity > 0
ORA-06512: at "SYSTEM.T1", line 3
ORA-04088: error during execution of trigger 'SYSTEM.T1'
```

SLIP-3

Q3. Consider the following entities and their relationship. [40]

Newspaper (name,language , publisher , cost)

Cities (pincode , city, state)

Relationship between Newspaper and Cities is many-to-many with descriptive attribute daily required

Constraints: name and pincode primary key

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1)Write a trigger which will fire before insert on the cities table which check that the pincode must be of 6 digit. (Raise user defined exception and give appropriate message).

```
SQL Plus
SQL> select * from newspaper;

NAME          LANGUAG  PUBLI      COST
-----
Punetimes     Marathi  abc        5
AajkaAnand    Hindi   qwe        3
Indiatimes     English xyz        7

SQL> select * from citys;

PINCODE CITY  ST
-----
411015 Pune  MH
411038 Pune  MH
411045 Pune  MH

SQL> select * from nc;

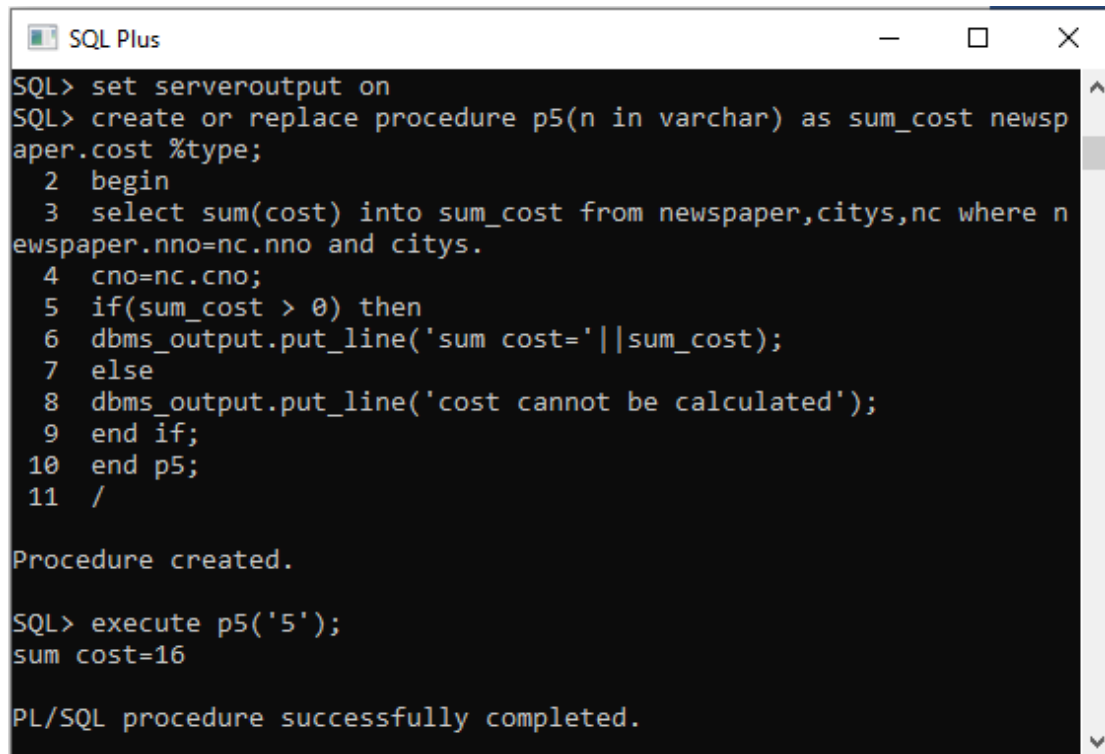
NAME          PINCODE DAILY
-----
Punetimes     411015 yes
AajkaAnand    411038 yes
Indiatimes     411045 yes

SQL> CREATE OR REPLACE TRIGGER check_pincode
2 BEFORE INSERT ON citys
3 FOR EACH ROW
4 DECLARE
5     pincode_length NUMBER(2);
6 BEGIN
7     pincode_length := LENGTH(:NEW.pincode);
8     IF (pincode_length <> 6) THEN
9         RAISE_APPLICATION_ERROR(-20001, 'Pincode must be 6 digits');
10    END IF;
11 END;
12 /

Trigger created.
```

```
SQL Plus
SQL> insert into citys values (41105,'Pune','MH');
insert into citys values (41105,'Pune','MH')
*
ERROR at line 1:
ORA-20001: Pincode must be 6 digits
ORA-06512: at "SYSTEM.CHECK_PINCODE", line 6
ORA-04088: error during execution of trigger 'SYSTEM.CHECK_PINCODE'
```

2) Write a procedure to calculate city wise total cost of each newspaper



```
SQL> set serveroutput on
SQL> create or replace procedure p5(n in varchar) as sum_cost newsp
aper.cost %type;
  2  begin
  3  select sum(cost) into sum_cost from newspaper,citys,nc where n
ewspaper.nno=nc.nno and citys.
  4  cno=nc.cno;
  5  if(sum_cost > 0) then
  6  dbms_output.put_line('sum cost='||sum_cost);
  7  else
  8  dbms_output.put_line('cost cannot be calculated');
  9  end if;
 10  end p5;
 11  /

Procedure created.

SQL> execute p5('5');
sum cost=16

PL/SQL procedure successfully completed.
```

SLIP-4

Q3 Consider the following entities and their relationships. [40]

Client (client_no, client_name, address, birthdate)

Policy_info (policy_no, desc, maturity_amt, prem_amt, date)

Relation between Client and Policy_info is Many to Many

Constraint: Primary key, prem_amt and maturity_amt should be > 0.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a procedure which will display all policy details having premium amount less than 5000.

```
SQL Plus
SQL> select * from clients;
-----
CNO CNAME   ADDR   BDATE
-----
1 Ashish   Pune   20-05-2004
2 Satyajit Pune   13-02-2002
3 Bhushan  Pune   01-04-2003

SQL> select * from policies;
-----
PNO DISC   MATR   PAMT PDATE
-----
1 life     10000   2000 10-5-2003
2 health   8000    1500 03-8-2002
3 life     11500   1200 22-3-2004

SQL> select * from c_p;
-----
CNO   PNO
-----
1     1
2     2
3     3

SQL> set serveroutput on
SQL> create or replace procedure p4(n in varchar)as cursor c4 is select pamt,pdate from clients,policies,c_p where clients.cno=c_p.cno and policies.pno=c_p.pno and pam
< 5000;
2 c c4 %rowtype;
3 begin
4 open c4;
5 dbms_output.put_line('pamt'||' '||'pdate');
6 loop
7 fetch c4 into c;
8 exit when c4 %notfound;
9 if(c.pamt=n) then
10 dbms_output.put_line(c.pamt||' '||c.pdate);
11 end if;
12 end loop;
13 close c4;
14 end;
15 /
Procedure created.

SQL> set serveroutput on;
SQL> begin
2 p4('2000');
3 end;
4 /
pamtpdate
200010-5-2003

PL/SQL procedure successfully completed.
```

2) Write a trigger which will fire before insert or update on policy_info having maturity amount less than premium amount. (Raise user defined exception and give appropriate message).

```
SQL Plus
SQL> CREATE OR REPLACE TRIGGER check_maturity_amount
 2 BEFORE INSERT OR UPDATE ON policies
 3 FOR EACH ROW
 4 DECLARE
 5     pamt NUMBER;
 6     mamt NUMBER;
 7 BEGIN
 8     pamt := :NEW.pamt;
 9     mamt := :NEW.mamt;
10     IF pamt <= 0 OR mamt <= 0 THEN
11         RAISE_APPLICATION_ERROR(-20001, 'Pamt and mamt should be greater t
han 0. ');
12     END IF;
13     IF mamt < pamt THEN
14         RAISE_APPLICATION_ERROR(-20002, 'Mamt should not be less than pamt
. ');
15     END IF;
16 END;
17 /

Trigger created.

SQL> insert into policies values (4,'health',1500,5000,10-06-2003);
insert into policies values (4,'health',1500,5000,10-06-2003)
*
ERROR at line 1:
ORA-20002: Mamt should not be less than pamt.
ORA-06512: at "SYSTEM.CHECK_MATURITY_AMOUNT", line 11
ORA-04088: error during execution of trigger 'SYSTEM.CHECK_MATURITY_AMOUNT'
```


SLIP-5

Q3 Consider the following entities and their relationships. [40]

Library(Lno, Lname, Location, Librarian, no_of_books)

Book(Bid, Bname, Author_Name, Price, publication)

Relation between Library and Book is one to many.

Constraint: Primary key, Price should not be null.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a function which will accept publication name from user and display total price of books of that publication.

```
SQL Plus
SQL> select * from library;

  L_NO L_NAME      LOCATION      LIBRARIAN
-----
10 1 francis      pune          Ashish
7  2 las_vegas    pune          Bhushan
5  3 Tokyo        pune          Satyajit

SQL> select * from books;

  B_ID B_NAME      AUTHOR_NAME      PRICE
-----
1 1 wonderland    alice            520
2 2 imagica       smile            450
3 3 bala          slice            370

SQL> set serveroutput on;
SQL> create or replace function fun1(pn in varchar) return number as pm number;
2  begin
3  select sum(books.price) into pm from library,books where library.l_no=books.l_no and publication='vision';
4  if sql %found then
5  return(pm);
6  else
7  return null;
8  end if;
9  end;
10 /

Function created.

SQL> begin
2  dbms_output.put_line('price- '||fun5('nirali'));
3  end;
4  /
price-520

PL/SQL procedure successfully completed.
```

2) Write a cursor which will display library wise book details.(Use Parameterized Cursor)

```
SQL Plus
SQL> set serveroutput on
SQL> declare
2  cursor c1(yyyy Library.l_name %type)is select l_name,b_name from Library,Books  where Library.l_no= Books.l_no order by l_name;
3  c c1%rowtype;
4  begin
5  open c1('&l_name ');
6  loop
7  fetch c1 into c;
8  exit when c1%notfound;
9  dbms_output.put_line(c.l_name||' '||c.b_name);
10 end loop;
11 close c1;
12 end;
13 /
Enter value for l_name: Ashish
old 5: open c1('&l_name ');
new 5: open c1('Ashish ');
Tokyobala
franciswonderland
las_vegasimagica
PL/SQL procedure successfully completed.
```

SLIP-6

Q3 Consider the following entities and their relationships. [40]

Employee (emp_id, emp_name, address)

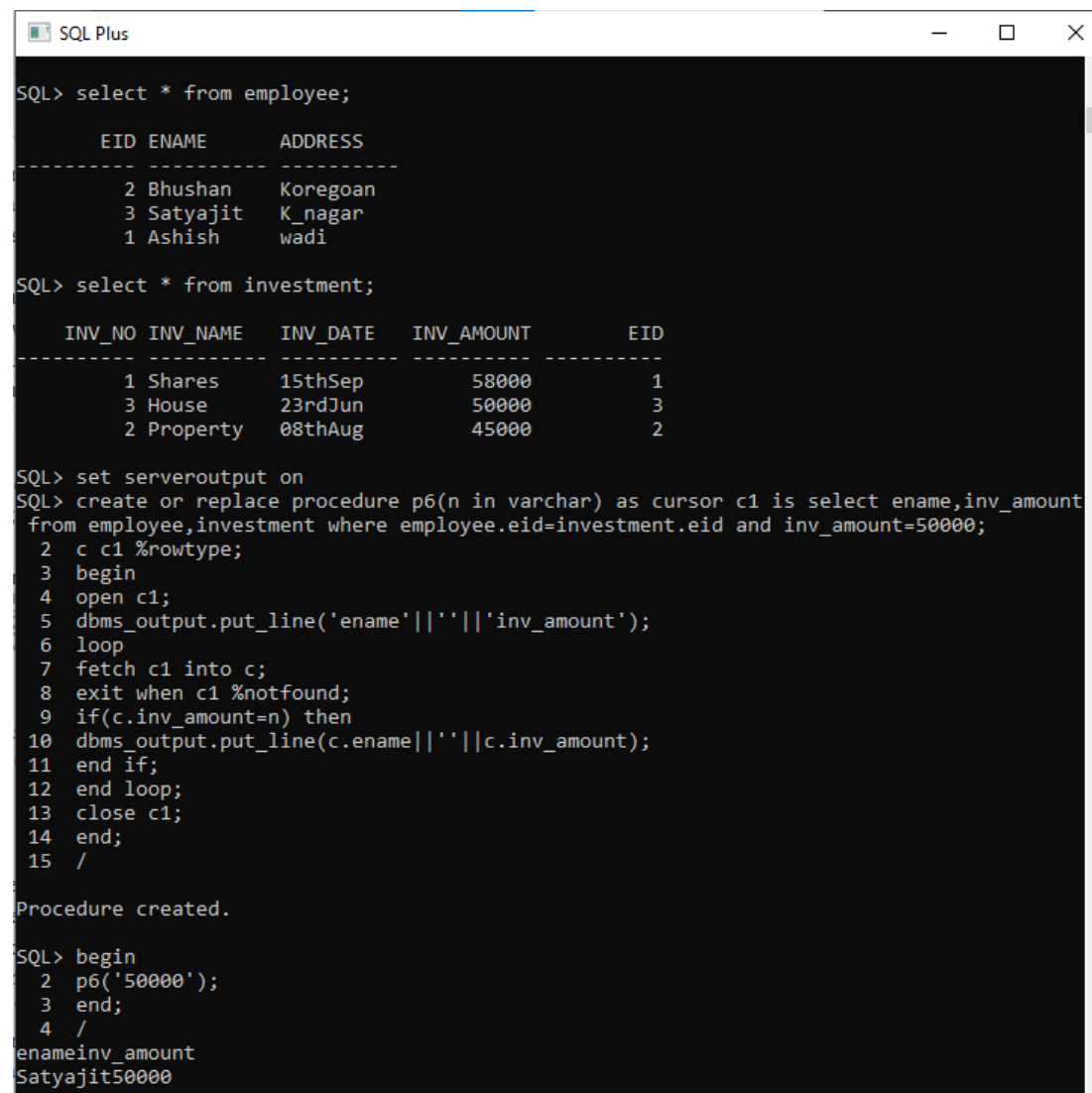
Investment (inv_no, inv_name, inv_date, inv_amount)

Relation between Employee and Investment is One to Many.

Constraint: Primary key, inv_amount should be > 0.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a procedure which will display details of employees invested amount in "Mutual Fund"



```
SQL> select * from employee;

  EID  ENAME      ADDRESS
-----
     2  Bhushan    Koregoan
     3  Satyajit    K_nagar
     1  Ashish      wadi

SQL> select * from investment;


  INV_NO  INV_NAME  INV_DATE  INV_AMOUNT  EID
-----
     1  Shares    15thSep    58000        1
     3  House     23rdJun    50000        3
     2  Property  08thAug    45000        2

SQL> set serveroutput on
SQL> create or replace procedure p6(n in varchar) as cursor c1 is select ename,inv_amount
from employee,investment where employee.eid=investment.eid and inv_amount=50000;
  2  c c1 %rowtype;
  3  begin
  4  open c1;
  5  dbms_output.put_line('ename'||' '||'inv_amount');
  6  loop
  7  fetch c1 into c;
  8  exit when c1 %notfound;
  9  if(c.inv_amount=n) then
10  dbms_output.put_line(c.ename||' '||c.inv_amount);
11  end if;
12  end loop;
13  close c1;
14  end;
15  /

Procedure created.

SQL> begin
  2  p6('50000');
  3  end;
  4  /
enameinv_amount
Satyajit50000
```

2) Write a cursor which will display date wise investment details.



```
SQL Plus
PL/SQL procedure successfully completed.
SQL> set serveroutput on
SQL> declare
  2 cursor c1 is select inv_date,inv_no,inv_name,inv_amount from employee,investment whe
re employee.eid=investment.eid;
  3 c c1 %rowtype;
  4 begin
  5 open c1;
  6 loop
  7 fetch c1 into c;
  8 exit when c1 %notfound;
  9 dbms_output.put_line(c.inv_date||' '||c.inv_no||' '||c.inv_name||' '||c.inv_amount);
 10 end loop;
 11 close c1;
 12 end;
 13 /
15thSep1Shares58000
23rdJun3House50000
08thAug2Property45000
PL/SQL procedure successfully completed.
```

SLIP-7

Q3 Consider the following entities and their relationships. [40]

Bill (billno, day, tableno, total)

Menu (dish_no, dish_desc, price)

The relationship between Bill and Menu is Many to Many with quantity as descriptive attribute.

Constraint: Primary key, price should be > 0.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a procedure to display menu details having price between 200 to 500 which were order on 'Saturday' .

```
SQL Plus
SQL> select * from bill;

  BILL_NO DAY          TABLE_NO    TOTAL
-----
      1 saturday         101         240
      2 Monday           102         280
      3 Wednesday        103         200

SQL> select * from menu;

  DISH_NO DISH_DESCR    PRICE
-----
      1 paneer          200
      2 Shurma          320
      3 p_handi         420

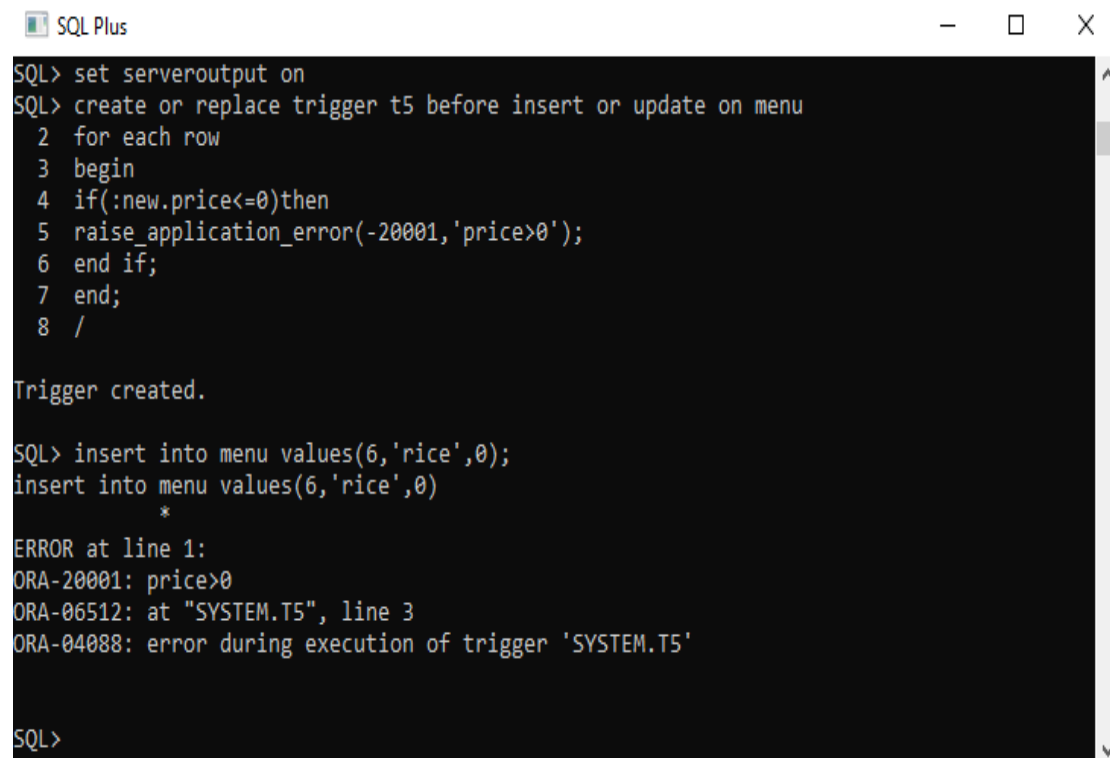
SQL> set serveroutput on;
SQL> create or replace procedure p1(a in varchar) as cursor
c1 is select menu.dish_no,menu.price,bill.day from bill,menu
,bm where bill.bill_no=bm.bill_no and menu.dish_no=bm.dish_n
o and price between 200 and 500 and day='saturday';
2  c c1 %rowtype;
3  begin
4  open c1;
5  dbms_output.put_line('dish_no'||' '||'price'||' '||'day')
;
6  loop
7  fetch c1 into c;
8  exit when c1 %notfound;
9  if(c.day=a)then
10 dbms_output.put_line(c.dish_no||' '||c.price||' '||c.day)
;
11 end if;
12 end loop;
13 close c1;
14 end;
15 /

Procedure created.
```

```
SQL Plus
SQL> begin
2  p1('saturday');
3  end;
4  /
dish_nopriceday
1200saturday

PL/SQL procedure successfully completed.
```

2) Write a trigger which will fire before insert or update on Menu having price less than or equal to zero. (Raise user defined exception and give appropriate message)



```
SQL> set serveroutput on
SQL> create or replace trigger t5 before insert or update on menu
  2  for each row
  3  begin
  4  if(:new.price<=0)then
  5  raise_application_error(-20001,'price>0');
  6  end if;
  7  end;
  8  /

Trigger created.

SQL> insert into menu values(6,'rice',0);
insert into menu values(6,'rice',0)
      *
ERROR at line 1:
ORA-20001: price>0
ORA-06512: at "SYSTEM.T5", line 3
ORA-04088: error during execution of trigger 'SYSTEM.T5'

SQL>
```

SLIP-8

Q3 Consider the following entities and their relationships. [40]

Plan (plan_no, plan_name, nooffreecalls, freecalltime, fix_amt)

Customer (cust_no, cust_name, mobile_no)

Relation between Plan and Customer is One to Many.

Constraint: Primary key, fix_amt should be greater than 0.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a function which will accept plan number from user and display all the details of the selected plan

```
Select SQL Plus
SQL> select * from plan;

      PNO PNAME      NC      FCT      F_AMT
-----
      1 Summer      10      8min      460
      2 Winter       8     10min      500
      3 Rainy       12     14min      450

SQL> select * from cust;

      CNO CNAME      MNO      PNO
-----
      1 Ashish      7770070406      1
      2 Bhushan     9879864535      2
      3 Satyajit    9763162617      3

SQL> set serveroutput on
SQL> create or replace function fun1(nocomp in varchar) return varchar as detalis varchar(
10);
 2 begin
 3 select ( plan.pname) into detalis from plan, cust where plan.pno=cust.pno and plan.pno
='1';
 4 if sql %found then
 5 return(detalis);
 6 else
 7 return null;
 8 end if;
 9 end fun1;
10 /

Function created.

SQL> begin
 2 dbms_output.put_line('detalis-'||fun1('1'));
 3 end;
 4 /
detalis-Summer

PL/SQL procedure successfully completed.
```

2) Write a cursor which will display customer wise plan details.(Use Parameterized Cursor)

```
SQL Plus
SQL> set serveroutput on
SQL> declare
  2 cursor c1(yyyy cust.cname %type)is select cname,pname from plan,cust whe
re plan.pno=cust.pno order by cname;
  3 c c1%rowtype;
  4 begin
  5 open c1('&cname ');
  6 loop
  7 fetch c1 into c;
  8 exit when c1%notfound;
  9 dbms_output.put_line(c.cname||' '||c.pname);
 10 end loop;
 11 close c1;
 12 end;
 13 /
Enter value for cname: Ashish
old  5: open c1('&cname ');
new  5: open c1('Ashish ');
AshishSummer
BhushanWinter
SatyajitRainy

PL/SQL procedure successfully completed.
```


SLIP-9

Q3 Consider the following entities and their relationships. [40]

Project (pno, pname, start_date, budget, status)

Department (dno, dname, HOD, loc)

The relationship between Project and Department is Many to One.

Constraint: Primary key.

Project Status Constraints: C – Completed,

P – Progressive,

I – Incomplete

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a function which accept department name and display total number of projects whose status is “p”(progressive).

```
SQL Plus
SQL> select * from project;

      PNO PNAME      S_DATE      BUDGET
-----
STATUS  DNO
-----
101 RDBMS      10-2-2015      9,000
102 Java       23-6-2015     10,000
103 Python     02-5-2015     11,500

SQL> select * from department;

      DNO DNAME      HOD      LOC
-----
1 Computer Ashish    Pune
2 Computer Bhushan  Pune
3 Computer Satyajit  Pune

SQL> set serveroutput on
SQL> create or replace function f1(xyz in varchar) return number as abc number;
2 begin
3 select count(project.pno) into abc from department, project where department.dno=project.dno and department.dname='computer' and project.status='P';
4 if sql %found then
5 return(abc);
6 else
7 return null;
8 end if;
9 end f1;
10 /

Function created.
```

```
SQL Plus
SQL> begin
2 dbms_output.put_line('project- ' || f1('computer'));
3 end;
4 /

project-3

PL/SQL procedure successfully completed.
```

2) Write a cursor which will display status wise project details of each department.

```
SQL> DECLARE
2   CURSOR project_cursor IS
3     SELECT d.dname, p.status, COUNT(*) AS project_count
4     FROM project p
5     JOIN project_department pd ON p.pno = pd.pno
6     JOIN department d ON pd.dno = d.dno
7     GROUP BY d.dname, p.status;
8
9   dept_name VARCHAR2(50);
10  proj_status VARCHAR2(20);
11  proj_count NUMBER(10);
12
13 BEGIN
14   OPEN project_cursor;
15
16   LOOP
17     FETCH project_cursor INTO dept_name, proj_status, proj_count;
18     EXIT WHEN project_cursor%NOTFOUND;
19
20     DBMS_OUTPUT.PUT_LINE('Department: ' || dept_name || ', Status: '
|| proj_status || ', Project Count: ' || proj_count);
21   END LOOP;
22
23   CLOSE project_cursor;
24 END;
25 /
Department: Computer, Status: P, Project Count: 3
PL/SQL procedure successfully completed.
```

SLIP-10

Q3 Consider the following entities and their relationships. [40]

Gym (Name, city, charges, scheme)

Member (ID, Name, phoneNo, address)

Relation between Gym and member is one to many.

Constraint: Primary Key, charges must be greater than 0.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a function which will accept member id and scheme from user and display charges paid by that member.

```
SQL Plus
SQL> select * from gym;

      GNO GNAME          CITY          CHARGES
-----
SCHEME
-----
      101 FIT_7          Pune          1000
xyz
      102 Grome          Pune          1200
abc
      103 Body_spa       Pune          800
qwr

SQL> select * from member;

      MID MNAME          PHONE ADDR          GNO
-----
      1 Ashish          7770070406 Vishrantwadi 101
      2 Bhushan          986543213 Koregoan     102
      3 Stayajit          7867843254 Karve_nagar  103

SQL> set serveroutput on
SQL> create or replace function f3(abc in varchar) return number as xyz number;
2 begin
3 select gym.charges into xyz from gym,member where gym.gno=member.gno and mid='2';
4 if sql %found then
5 return(xyz);
6 else
7 return null;
8 end if;
9 end f3;
10 /

Function created.

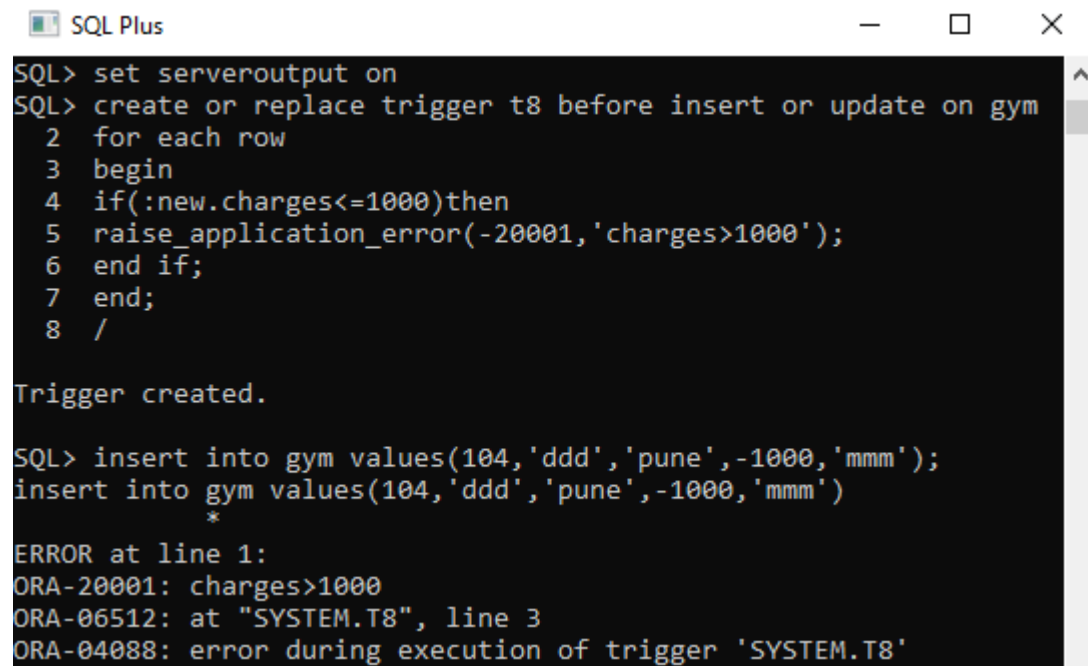
SQL> begin
2 dbms_output.put_line('gym-'||f3 ('2'));
3 end;
4 /
```

```
SQL Plus

4 /
gym-1200

PL/SQL procedure successfully completed.
```

2) Write a trigger which will fire before insert or update on Gym having charges less than 1000. (Raise user defined exception and give appropriate message)



```
SQL> set serveroutput on
SQL> create or replace trigger t8 before insert or update on gym
  2  for each row
  3  begin
  4  if(:new.charges<=1000)then
  5  raise_application_error(-20001,'charges>1000');
  6  end if;
  7  end;
  8  /

Trigger created.

SQL> insert into gym values(104,'ddd','pune',-1000,'mmm');
insert into gym values(104,'ddd','pune',-1000,'mmm')
*
```

ERROR at line 1:
ORA-20001: charges>1000
ORA-06512: at "SYSTEM.T8", line 3
ORA-04088: error during execution of trigger 'SYSTEM.T8'

SLIP-11

Q3 Consider the following entities and their relationships. [40]

Student (rollno, sname, class, timetable)

Lab (LabNo, LabName, capacity, equipment)

Relation between Student and Lab is Many to One.

Constraint: Primary Key, capacity should not be null.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a function which will accept Lab number from user and display total number of student allocated in that lab.

```
SQL Plus
SQL> select * from student10;

  ROLLNO  SNAME      CLASS      TIMETABLE
-----
  LABNO
-----
    101 Ashish        FY          Monday
    1
    102 Bhushan       FY          Monday
    2
    103 Satyajit     FY          Monday
    3

SQL> select * from lab;

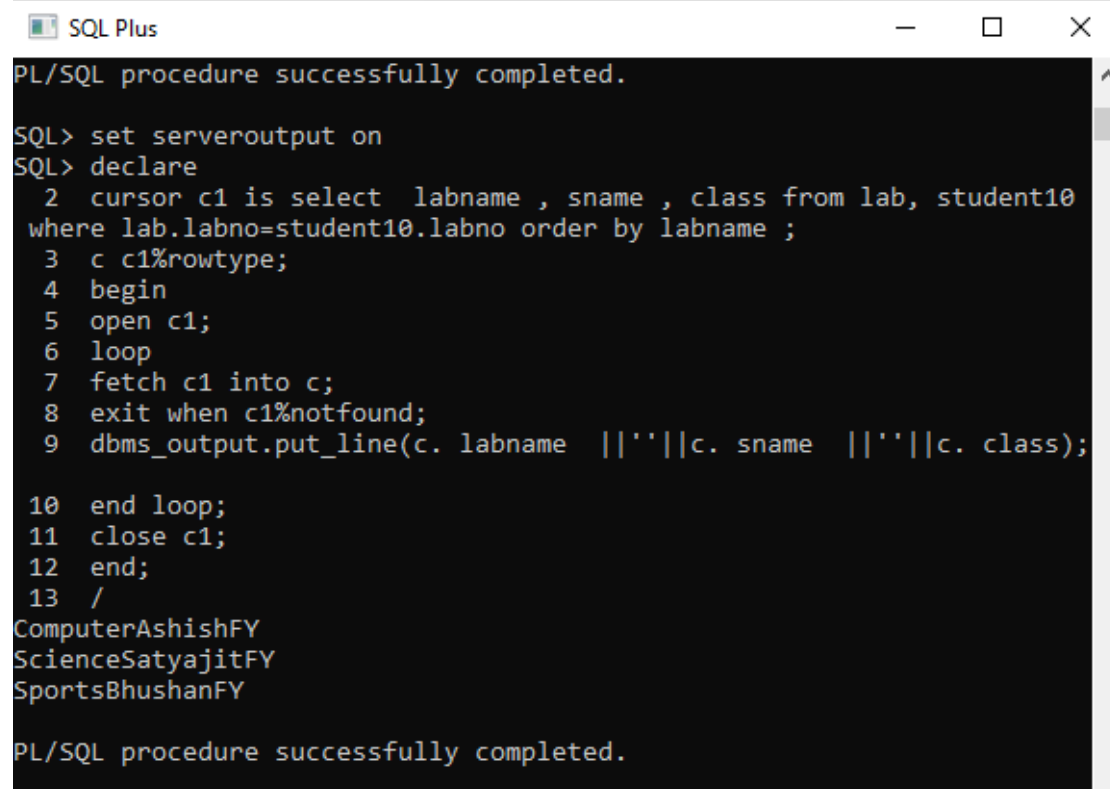
  LABNO LABNAME      CAPACITY EQUIPMENT
-----
    1 Computer        100 scope
    2 Sports          100 Football
    3 Science          100 Test_tube

SQL> set serveroutput on
SQL> create or replace function f2(abc in varchar) return number as xyz number;
  2 begin
  3 select count(student10.rollno) into xyz from student10,lab where lab.labno=student10.labno and lab.labno=1;
  4 if sql %found then
  5 return (xyz);
  6 else
  7 return null;
  8 end if;
  9 end f2;
 10 /

Function created.

SQL> begin
  2 dbms_output.put_line ('no of student'||f2('1'));
  3 end;
  4 /
no of student1
```

2) Write a cursor which will display lab wise student details.



```
SQL Plus
PL/SQL procedure successfully completed.

SQL> set serveroutput on
SQL> declare
  2  cursor c1 is select  labname , sname , class from lab, student10
  where lab.labno=student10.labno order by labname ;
  3  c c1%rowtype;
  4  begin
  5  open c1;
  6  loop
  7  fetch c1 into c;
  8  exit when c1%notfound;
  9  dbms_output.put_line(c. labname ||' '||c. sname ||' '||c. class);

 10  end loop;
 11  close c1;
 12  end;
 13  /
ComputerAshishFY
ScienceSatyajitFY
SportsBhushanFY

PL/SQL procedure successfully completed.
```

SLIP-12

Q3 Consider the following entities and their relationships. [40]

Wholesaler (w_no, w_name, address, city)

Product (product_no, product_name, rate)

Relation between Wholesaler and Product is Many to Many with quantity as descriptive attribute.

Constraint: Primary key, rate should be > 0.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a function which will accept wholesaler name from user and will display total number of items supplied by him.

```
SQL Plus
SQL> select * from product;

  PNO PNAME      RATE
-----
    1 Computer    1000
    2 Laptop     1000
    3 Mobile     1800

SQL> select * from wholesaler;

  WNO WNAME      ADDRESS  CITY
-----
   101 Ashish    Dhanori   Pune
   102 Bhushan   Kalas     Pune
   103 Satyajit  Paud      Pune

SQL> select * from wholesaler_product;

  WNO      PNO NO_OF_ITEM
-----
   101      1         4
   102      2         5
   103      3         7

SQL> set serveroutput on
SQL> create or replace function fun1(pn in varchar) return number as pm number;
  2 begin
  3   select sum(wholesaler_product.no_of_item) into pm from wholesaler,product,wholesaler_product where
  4     wholesaler.wno=wholesaler_product.wno and product.pno=wholesaler_product.pno and wholesaler.wname='Ashish';
  5   if sql %found then
  6     return(pm);
  7   else
  8     return null;
  9   end if;
 10 end;

Function created.

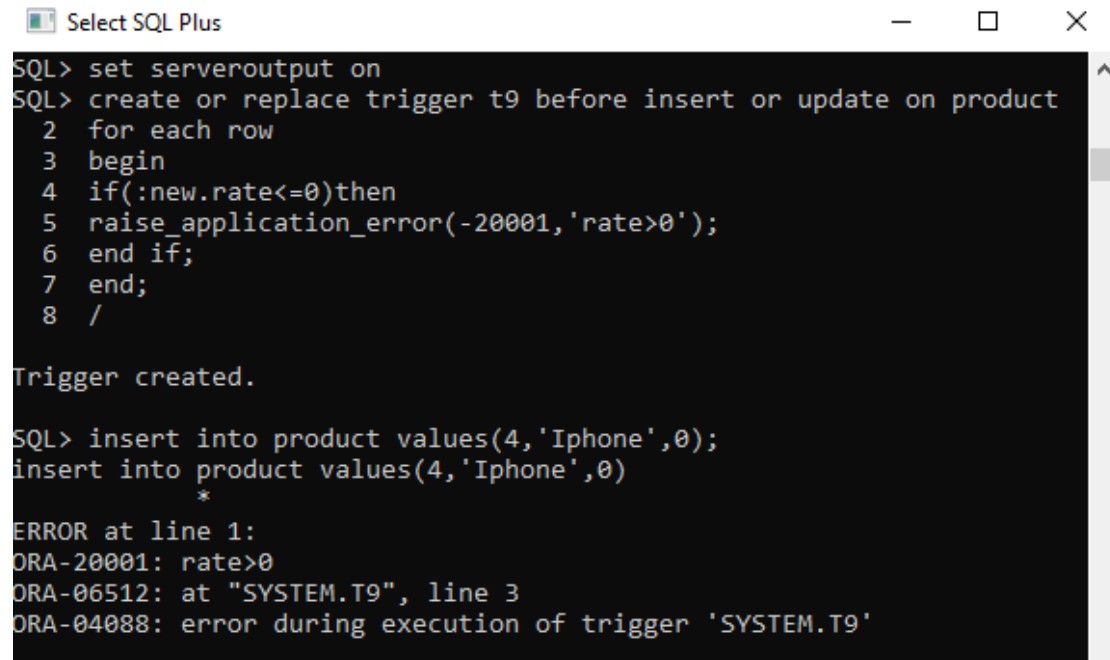
SQL> begin
  2   dbms_output.put_line('no_of_item-'||fun1('Ashish'));
  3 end;
```

```
SQL Plus

  4 /
no_of_item-4

PL/SQL procedure successfully completed.
```

2) Write a trigger which will fire before insert or update on product having rate less than or equal to zero (Raise user defined exception and give appropriate message).



```
SQL> set serveroutput on
SQL> create or replace trigger t9 before insert or update on product
  2  for each row
  3  begin
  4  if (:new.rate<=0)then
  5  raise_application_error(-20001,'rate>0');
  6  end if;
  7  end;
  8  /

Trigger created.

SQL> insert into product values(4,'Iphone',0);
insert into product values(4,'Iphone',0)
*
ERROR at line 1:
ORA-20001: rate>0
ORA-06512: at "SYSTEM.T9", line 3
ORA-04088: error during execution of trigger 'SYSTEM.T9'
```


SLIP-13

Q3 Consider the following entities and their relationships. [40]

Country (CId, CName , no_of_states, area, location, population)

Citizen(Id, Name, mother_toung, state_name)

Relation between Country and Citizen is one to many.

Constraint: Primary key, area should not be null.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a function which will display name of the country having minimum population.

```
SQL Plus
SQL> select * from country13;

-----
CID CNAME   NO_OF_STAT AREA      LOCATION  POPULATION
-----
1 India    10         Wadi      Pune      20000
2 Sri_Lanka 3          Dhanori   Mumbai    8000
3 Pakistan 6          Lohegoan  Hyderabad 12000

SQL> select * from citizen13;

-----
ID NAME      MOTHER_TOU STATE_NAME  CID
-----
1 Ashish     Hindi       Maharastra  1
2 Bhushan    German      Tokyo       2
3 Satyajit   English     Las_Vegas   3

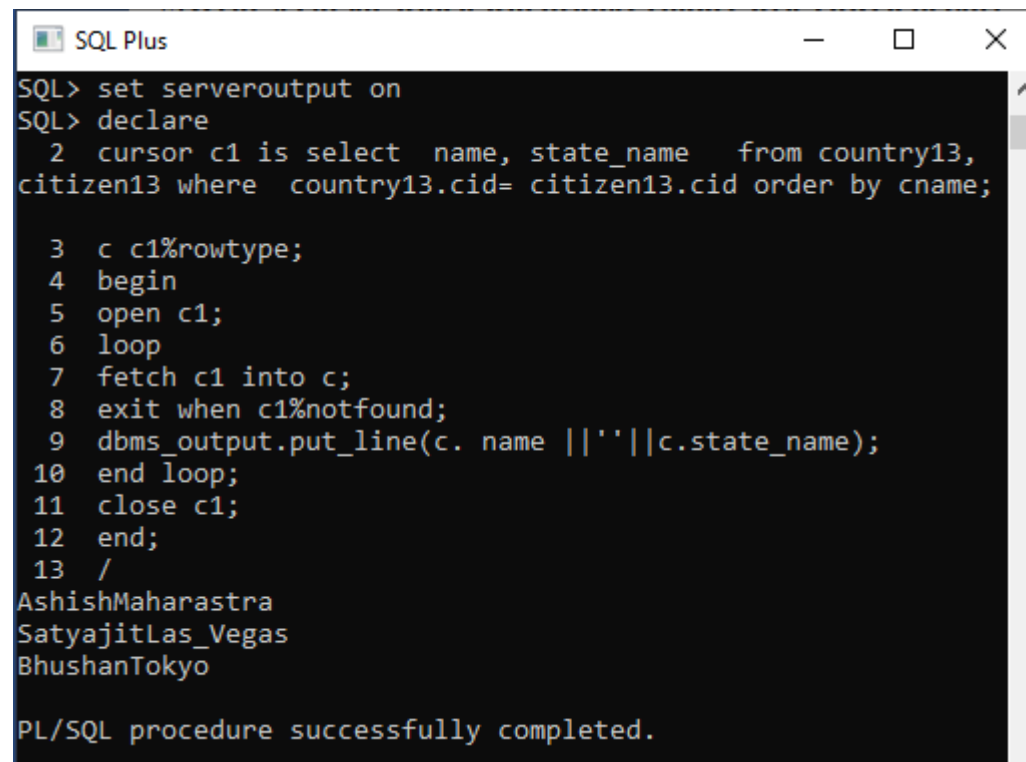
SQL> set serveroutput on
SQL> create or replace function f1(dn in varchar) return varchar as dmn varchar(10);
2 begin
3 select(country13.cname) into dmn from country13 where population=(select min(population) from country13);
4 if sql %found then
5 return (dmn);
6 else
7 return null;
8 end if;
9 end f1;
10 /

Function created.

SQL> begin
2 dbms_output.put_line('name-'||f1('cname'));
3 end;
4 /
name-Sri_Lanka

PL/SQL procedure successfully completed.
```

2) Write a cursor which will display county wise citizen details.

A screenshot of a SQL Plus window titled "SQL Plus". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main area is a black terminal with white text. The text shows SQL commands being entered and the output of a PL/SQL procedure. The commands include setting serveroutput on, declaring a cursor, and a loop that fetches data from a table and prints it. The output shows three lines of data: "AshishMaharashtra", "SatyajitLas_Vegas", and "BhushanTokyo". The final line indicates the procedure completed successfully.

```
SQL> set serveroutput on
SQL> declare
  2 cursor c1 is select name, state_name from country13,
citizen13 where country13.cid= citizen13.cid order by cname;

  3 c c1%rowtype;
  4 begin
  5 open c1;
  6 loop
  7 fetch c1 into c;
  8 exit when c1%notfound;
  9 dbms_output.put_line(c.name ||' '||c.state_name);
10 end loop;
11 close c1;
12 end;
13 /
AshishMaharashtra
SatyajitLas_Vegas
BhushanTokyo

PL/SQL procedure successfully completed.
```

SLIP-14

Q3 Consider the following entities and their relationships. [40]

College (code, college_name, address)

Teacher (teacher_id, teacher_name, Qualification, specialization, salary, Desg)

Relation between Teacher and College is Many to One.

Constraint: Primary Key, qualification should not be null.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a procedure which will accept teacher name from user and display his/her college details.

```
SQL Plus
SQL> select * from teacher2;

```

TID	TNAME	QUALIFICATION	SPECIAL
100	aaa	m.bba	english
15000	head_of_department		
200	Ashish	bba	english
18000	Teacher		
300	Bhushan	b.com	english
25000	Principal		

```
SQL> select * from college20;

```

TID	CID	CNAME	ADDR
	1	mmm	pune
100	2	xyz	pune
200	3	abc	pune
300			

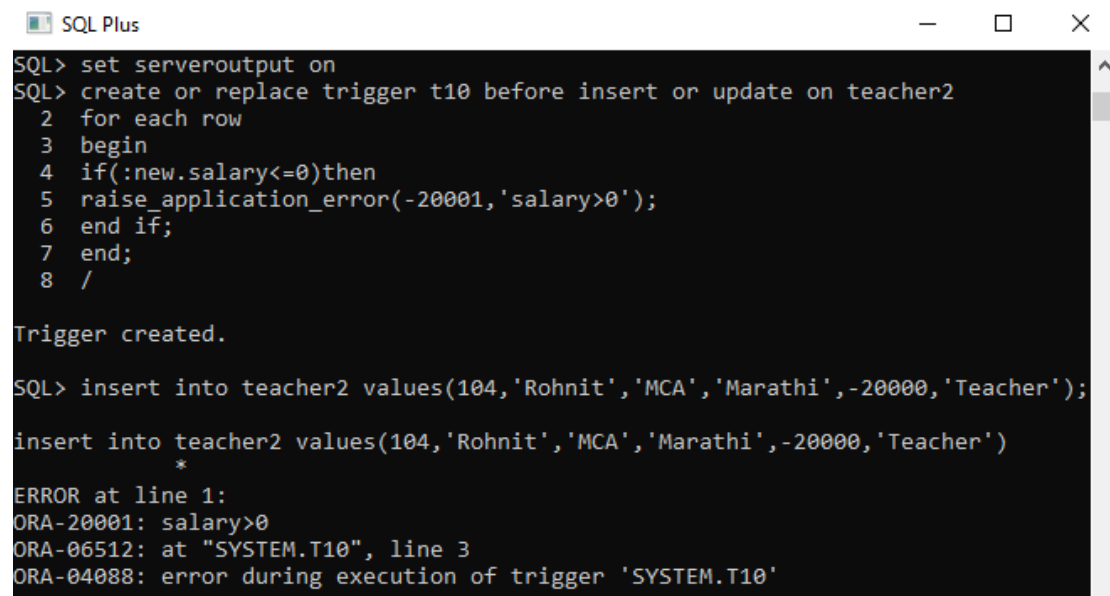
```
SQL Plus
SQL> set serveroutput on
SQL> create or replace procedure p1(n in varchar) as cursor
c1 is select tname,cname from college20,teacher2 where coll
ege20.tid=teacher2.tid and tname='aaa';
2 c c1 %rowtype;
3 begin
4 open c1;
5 dbms_output.put_line('tname'||' '||'cname');
6 loop
7 fetch c1 into c;
8 exit when c1 %notfound;
9 if(c.tname=n)then
10 dbms_output.put_line(c.tname ||' '||c.cname);
11 end if;
12 end loop;
13 close c1;
14 end;
15 /

Procedure created.

SQL> begin
2 p1('aaa');
3 end;
4 /
tnamecname
aaammm

PL/SQL procedure successfully completed.
```

2) Write a trigger which will fire before insert or update on Teacher having salary less than or equal to zero (Raise user defined exception and give appropriate message)



The screenshot shows a SQL Plus window with a black background and white text. The window title is 'SQL Plus'. The SQL prompt is 'SQL>'. The user has entered the following SQL code to create a trigger named 't10' that fires before insert or update on the 'teacher2' table. The trigger checks if the salary is less than or equal to zero and raises a user-defined exception 'ORA-20001: salary>0'. The user then attempts to insert a record with a salary of -20000, which results in an error.

```
SQL> set serveroutput on
SQL> create or replace trigger t10 before insert or update on teacher2
  2  for each row
  3  begin
  4  if (:new.salary <= 0) then
  5  raise_application_error(-20001, 'salary>0');
  6  end if;
  7  end;
  8  /

Trigger created.

SQL> insert into teacher2 values(104,'Rohnit','MCA','Marathi',-20000,'Teacher');

insert into teacher2 values(104,'Rohnit','MCA','Marathi',-20000,'Teacher')
*
ERROR at line 1:
ORA-20001: salary>0
ORA-06512: at "SYSTEM.T10", line 3
ORA-04088: error during execution of trigger 'SYSTEM.T10'
```

SLIP-15

Q3 Consider the following entities and their relationships. [40]

Driver (driver_id, driver_name, address)

Car (license_no, model, year)

Relation between Driver and Car is Many to Many with date and time as descriptive attribute.

Constraint: Primary key, driver_name should not be null.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a function which will display the total number of person who are using "Swift" car.



```
SQL> select * from driver;

  DID DNAME      ADDRESS
-----
   1 Ashish      Pune
   2 Bhushan     Pune
   3 Satyajit    Pune

SQL> select * from car;

  LNO MODEL      YEAR
-----
  101 Swift      2000
  102 Roll_Royce 2008
  103 Benz       2003

SQL> select * from dc;

  DID      LNO T_DATE      TIME
-----
   1      101 15-2-2000  10aAM
   2      102 12-3-2007   4PM
   3      103 27-12-2003   3AM

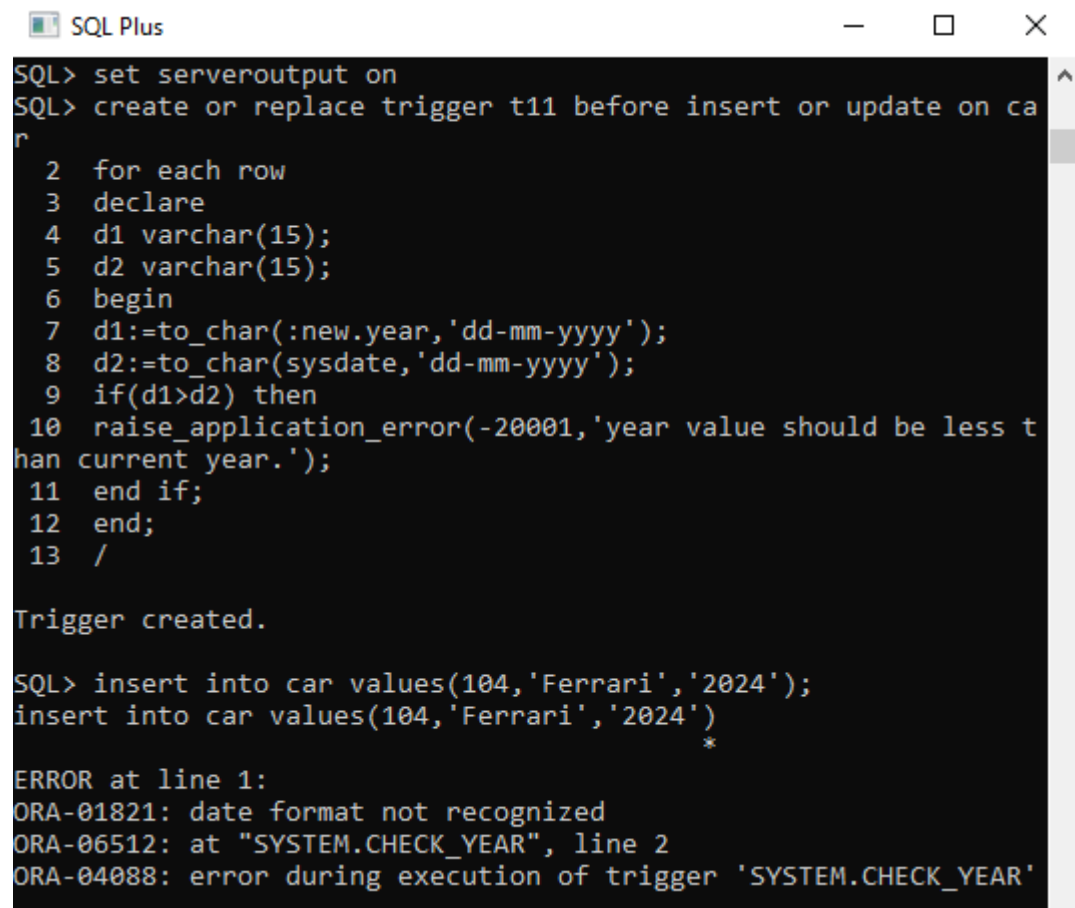
SQL> set serveroutput on
SQL> create or replace function fun2(bcd in varchar) return number
as efg number;
2 begin
3 select sum(driver.did) into efg from driver, car, dc where
driver.did=dc.did and car.lno=dc.lno and car.model='swift';
4 if sql %found then
5 return (efg);
6 else
7 return null;
8 end if;
9 end fun2;
10 /

Function created.

SQL> begin
2 dbms_output.put_line('person- ' || fun2('Swift'));
3 end;
```

```
4 /
person-
PL/SQL procedure successfully completed.
```

2) Write a trigger which will fire before insert or update on year. If year value is more than current year. (Raise user defined exception and give appropriate message)

A screenshot of a SQL Plus window titled "SQL Plus". The window has standard window controls (minimize, maximize, close) in the top right corner. The main area is a black terminal with white text. The text shows the creation of a trigger named 't11' that fires before insert or update on a table named 'car'. The trigger code includes a loop 'for each row', variable declarations for 'd1' and 'd2' as 'varchar(15)', and logic to compare the new year with the current year. If the new year is greater, it raises an application error with message 'year value should be less than current year.'. After the trigger is created, an insert statement is executed, which results in an error because the year '2024' is not recognized as a valid date format.

```
SQL> set serveroutput on
SQL> create or replace trigger t11 before insert or update on ca
r
  2  for each row
  3  declare
  4  d1 varchar(15);
  5  d2 varchar(15);
  6  begin
  7  d1:=to_char(:new.year,'dd-mm-yyyy');
  8  d2:=to_char(sysdate,'dd-mm-yyyy');
  9  if(d1>d2) then
 10  raise_application_error(-20001,'year value should be less t
han current year.');
```

```
11  end if;
12  end;
13  /

Trigger created.

SQL> insert into car values(104,'Ferrari','2024');
insert into car values(104,'Ferrari','2024')
*
```

```
ERROR at line 1:
ORA-01821: date format not recognized
ORA-06512: at "SYSTEM.CHECK_YEAR", line 2
ORA-04088: error during execution of trigger 'SYSTEM.CHECK_YEAR'
```

SLIP-16

Q3 Consider the following entities and their relationships. [40]

Game (game_name, no_of_players, coach_name)

Player (pid, pname, address, club_name)

Relation between Game and Player is Many to Many.

Constraint: Primary key, no_of_players should be > 0.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a procedure which will display games details having number of players more than 5.

```
SQL Plus
SQL> select * from game2;

  GID GNAME          NO_OF_PLAYERS COACH_NAME
-----
    1 football         11          aaa
    2 cricket          12          xyz
    3 hockey            6          qwe

SQL> select * from player;

  PID PNAME          ADDR CLUB_NAME
-----
  100 mmm            pune  pfc
  101 ashish         pune  cbi
  102 bhushan        pune  cid

SQL Plus
SQL> set serveroutput on
SQL> create or replace procedure p3(n in varchar)as cursor c
3 is select gname,coach_name from game2,player,gp where game
2.gid=gp.gid and player.pid=gp.pid and no_of_players > 5;
  2 c c3 %rowtype;
  3 begin
  4 open c3;
  5 dbms_output.put_line('gname'||' '||'coach_name');
  6 loop
  7 Fetch c3 into c;
  8 exit when c3 %notfound;
  9 if(c.gname=n)then
 10 dbms_output.put_line(c.gname||' '||c.coach_name);
 11 end if;
 12 end loop;
 13 close c3;
 14 end;
 15 /

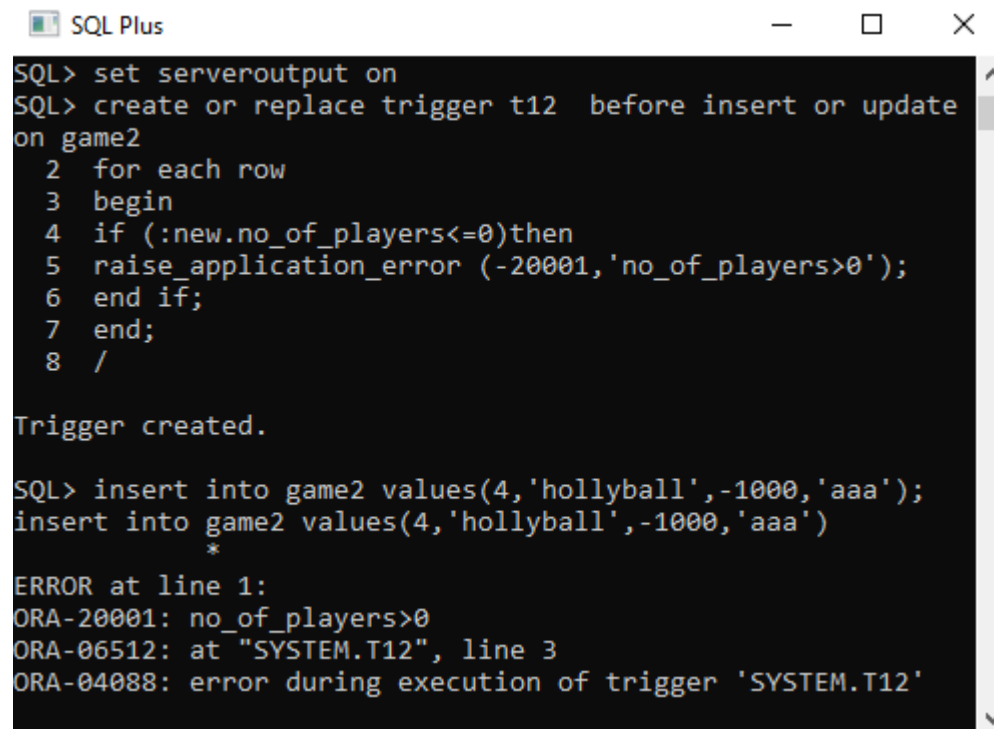
Procedure created.

SQL> begin
  2 p3('football');
  3 end;
  4 /
gnamecoach_name
footballaaa

PL/SQL procedure successfully completed.

SQL> _
```

2) Write a trigger which will fire before insert or update on Game having no_of_players less than or equal to zero. (Raise user defined exception and give appropriate message)



The screenshot shows a SQL Plus window with a black background and white text. The window title is 'SQL Plus'. The text inside shows the following SQL commands and their output:

```
SQL> set serveroutput on
SQL> create or replace trigger t12 before insert or update
on game2
  2 for each row
  3 begin
  4 if (:new.no_of_players<=0)then
  5 raise_application_error (-20001,'no_of_players>0');
  6 end if;
  7 end;
  8 /

Trigger created.

SQL> insert into game2 values(4,'hollyball',-1000,'aaa');
insert into game2 values(4,'hollyball',-1000,'aaa')
*
```

ERROR at line 1:
ORA-20001: no_of_players>0
ORA-06512: at "SYSTEM.T12", line 3
ORA-04088: error during execution of trigger 'SYSTEM.T12'

SLIP-17

Q3. Consider the following Item_Supplier database [40]

Company (name , address , city , phone , share_value)

Person (pname ,pcity)

Relationship between Company and Person is M to M relationship with descriptive

attribute No_of_shares i

Constraints: name,pname primary key

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a trigger before insert or update on No_of_shares field should not be zero.(Raise user defined exception and give appropriate message)

```
SQL Plus
SQL> select * from company0;

  CID CNAME      ADDR      CITY
-----
  1 Solanki_Ltd  Wadi      Pune
  2 Gaikwad_Ltd Bhiwabdi  Mumbai
  3 Patale_Ltd  Koregaon  Pune

  PHONE SHARE_VALUE
-----
  7770070406 280
  9876567897 200
  8976523432 320

SQL> select * from person3;

  PNO PNAME      PCITY
-----
  11 Ashish      Pimpri
  12 Bhushan   Pune
  13 Satyajit  Chincwad

SQL> select * from com_per;

  CID      PNO NO_OF_SHARES
-----
  1        11        50
  2        12        30
  3        13        20

SQL> set serveroutput on
SQL> create or replace trigger t13 before insert or update on com_per
  2 for each row
  3 begin
  4 if(:new.no_of_shares<=0)then
  5 raise_application_error(-20001,'no_of_shares>0');
  6 end if;
  7 end;
  8 /

Trigger created.
```

```
SQL Plus
SQL> insert into com_per values(6,16,-1000);
insert into com_per values(6,16,-1000)
*
ERROR at line 1:
ORA-20001: no_of_shares>0
ORA-06512: at "SYSTEM.T13", line 3
ORA-04088: error during execution of trigger 'SYSTEM.T13'
```

2) Write a function to display total no_of_shares of a specific person.

```
SQL Plus
SQL> select * from company0;

  CID CNAME          ADDR          CITY
-----
  1 ashish          koregaon      pune
1122334455          200
  2 bhushi          khese         pune
1234567890          300
  3 satya           pimpri        pune
9876543210          100

SQL> select * from person3;

  PNO PNAME          PCITY
-----
  11 solanki         pimpri
  12 gaikwad         pimpri
  13 pathale         pimpri

SQL> select * from com_per;

  CID      PNO NO_OF_SHARES
-----
  1        11         20
  2        12         30
  3        13         10

SQL Plus
SQL> set serveroutput on
SQL> create or replace function f2(xyz in varchar) return number as abc nu
mber;
  2 begin
  3   select sum(person3.pno) into abc from company0,person3,com_per where
company0.cid=com_per.cid  and person3.pno=com_per.pno and pname='ashish';
  4   if sql %found then
  5     return(abc);
  6   else
  7     return null;
  8   end if;
  9   end f2;
10  /

Function created.

SQL>
SQL> begin
  2   dbms_output.put_line('company- 20'||f2('ashish'));
  3   end;
  4   /
company- 20

PL/SQL procedure successfully completed.
```

SLIP-18

Q3. Consider the following entities and their relationship. [40]

Student (s_reg_no, s_name, s_class)

Competition (comp_no, comp_name, comp_type)

Relationship between Student and Competition is many-to-many with descriptive attribute rank and year.

Constraints: primary key, foreign key, primary key for third table(s_reg_no, comp_no, year), s_name and comp_name should not be null, comp_type can be sports or academic.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a function which will accept s_reg_no of student and returns total number of competition in which student has participated in a given year.

```
SQL Plus
SQL> select * from competition;

  COMPNO  COMPNAME  COMPTYPE
-----
    101   Cricket   Sports
    102    Maths    Quiz
    103    Race     Sports

SQL> select * from student1competition;

  SREGNO  COMPNO  YEAR
-----
      1    101  2021
      2    102  2022
      3    103  2023

SQL> set serveroutput on
SQL> create or replace function fun1(nocomp in varchar) return number as nofcomp number;
  2 begin
  3   select count(competition.compno) into nofcomp from student1competition, competition, student1competition where student1.sregno=student1competition.sregno and competition.compno=student1competition.compno and student1.sregno=1;
  4   if sql %found then
  5     return(nofcomp);
  6   else
  7     return null;
  8   end if;
  9   end fun1;
 10 /

Function created.
```

```
SQL Plus
SQL> begin
  2   dbms_output.put_line('no of competition-'||fun1(2021));
  3 end;
  4 /

no of competition-1

PL/SQL procedure successfully completed.
```

2) Write a cursor which will display year wise details of competitions.
(Use parameterized cursor)

```
SQL Plus
SQL> set serveroutput on
SQL> declare
  2  cursor c1(yyyy student1competition.year%type)is select compname,compt
type,year from student1,competition,student1competition where student1.sreg
no=student1competition.sregno and competition.compno=student1competition.c
ompno order by year;
  3  c c1%rowtype;
  4  begin
  5  open c1('&yyyy');
  6  loop
  7  fetch c1 into c;
  8  exit when c1%notfound;
  9  dbms_output.put_line(c.compname||' '||c.comptype||' '||c.year);
 10  end loop;
 11  close c1;
 12  end;
 13  /
Enter value for yyyy:
old  5: open c1('&yyyy');
new  5: open c1('');
Cricket Sports 2023
Race Sports 2023
Maths Quiz 2023

PL/SQL procedure successfully completed.
```

SLIP-19

Q3 Consider the following entities and their relationships. [40]

Game (game_name, no_of_players, coach_name)

Player (pid, pname, address, club_name)

Relation between Game and Player is Many to Many.

Constraint: Primary key, no_of_players should be > 0.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a function which will return total number of football players of "Sports Club".

```
SQL Plus
SQL> select * from game;

GAME_NAM NO_OF_PLAYER COACH_
-----
football      10 patil
cricket       12 khede
hockey         6 ashish

SQL> select * from player;

      PID PNAME  ADDRE CLUB_N
-----
      101 swarup pune   sports
      102 ankit  pune   quiz
      103 sahil  pune   quiz

SQL> select * from player_game;

      PID GAME_NAM
-----
      101 football
      102 cricket
      103 hockey
```

```
SQL Plus
SQL> CREATE OR REPLACE FUNCTION get_football_player_count
2  RETURN NUMBER
3  IS
4      v_count NUMBER := 0;
5  BEGIN
6      SELECT COUNT(DISTINCT p.pid) INTO v_count
7      FROM Player p
8      JOIN Player_Game pg ON p.pid = pg.pid
9      JOIN Game g ON pg.game_name = g.game_name
10     WHERE g.game_name = 'Football' AND p.club_name = 'Sports Club';
11
12     RETURN v_count;
13 END;
14 /

Function created.

SQL>
SQL> DECLARE
2     v_player_count NUMBER;
3  BEGIN
4     v_player_count := get_football_player_count();
5     DBMS_OUTPUT.PUT_LINE('Total number of football players
in Sports Club: 1' || v_player_count);
6  END;
7  /

Total number of football players in Sports Club: 10

PL/SQL procedure successfully completed.
```

2) Write a cursor which will display club wise details of players.

```
SQL Plus
SQL> DECLARE
2   CURSOR c_player_details IS
3     SELECT p.pname, p.address, p.club_name, g.game_name, g.coach_name
4     FROM Player p
5     JOIN Player_Game pg ON p.pid = pg.pid
6     JOIN Game g ON pg.game_name = g.game_name
7     ORDER BY p.club_name, p.pname;
8
9   v_pname Player.pname%TYPE;
10  v_address Player.address%TYPE;
11  v_club_name Player.club_name%TYPE;
12  v_game_name Game.game_name%TYPE;
13  v_coach_name Game.coach_name%TYPE;
14  v_current_club Player.club_name%TYPE := NULL;
15 BEGIN
16   OPEN c_player_details;
17   FETCH c_player_details INTO v_pname, v_address, v_club_name, v_game_name, v_coach_name;
18
19   DBMS_OUTPUT.PUT_LINE('Club-wise details of players:');
20   DBMS_OUTPUT.PUT_LINE('-----');
21
22   WHILE c_player_details%FOUND LOOP
23     IF v_current_club IS NULL OR v_current_club != v_club_name THEN
24       v_current_club := v_club_name;
25       DBMS_OUTPUT.PUT_LINE(v_current_club || ':');
26     END IF;
27
28     DBMS_OUTPUT.PUT_LINE('  ' || v_pname || ' (' || v_game_name || ' coached by ' || v_coach_name || ')');
29
30     FETCH c_player_details INTO v_pname, v_address, v_club_name, v_game_name, v_coach_name;
31   END LOOP;
32
33   CLOSE c_player_details;
34 END;
35 /
Club-wise details of players:
-----
quiz:
ankit (cricket coached by khede)
sahil (hockey coached by ashish)
sports:
swarup (football coached by patil)

PL/SQL procedure successfully completed.
```

SLIP-20

Q3 Consider the following entities and their relationships. [40]

Driver (driver_id, driver_name, address)

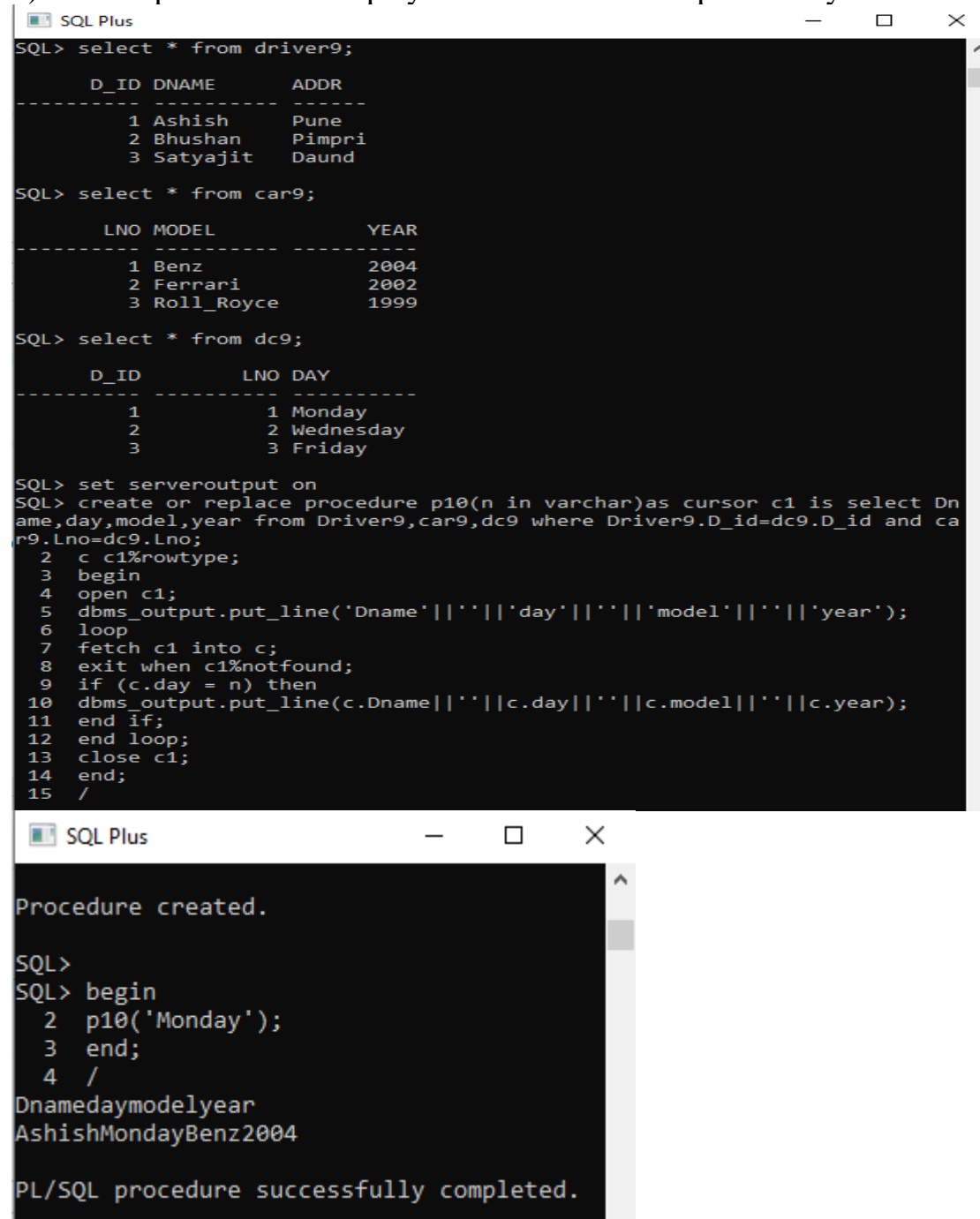
Car (license_no, model, year)

Relation between Driver and Car is Many to Many with date and time as descriptive attribute.

Constraint: Primary key, driver_name should not be null.

Create a RDB in 3NF and write PL/SQL blocks in Oracle for the following:

1) Write a procedure to display car details used on specific day.



```
SQL Plus
SQL> select * from driver9;

  D_ID DNAME      ADDR
-----
     1 Ashish      Pune
     2 Bhushan     Pimpri
     3 Satyajit     Daund

SQL> select * from car9;

  LNO MODEL          YEAR
-----
     1 Benz            2004
     2 Ferrari         2002
     3 Roll_Royce      1999

SQL> select * from dc9;

  D_ID      LNO DAY
-----
     1         1 Monday
     2         2 Wednesday
     3         3 Friday

SQL> set serveroutput on
SQL> create or replace procedure p10(n in varchar)as cursor c1 is select Dn
ame,day,model,year from Driver9,car9,dc9 where Driver9.D_id=dc9.D_id and ca
r9.Lno=dc9.Lno;
  2  c c1%rowtype;
  3  begin
  4  open c1;
  5  dbms_output.put_line('Dname'||' '||'day'||' '||'model'||' '||'year');
  6  loop
  7  fetch c1 into c;
  8  exit when c1%notfound;
  9  if (c.day = n) then
 10  dbms_output.put_line(c.Dname||' '||c.day||' '||c.model||' '||c.year);
 11  end if;
 12  end loop;
 13  close c1;
 14  end;
 15  /

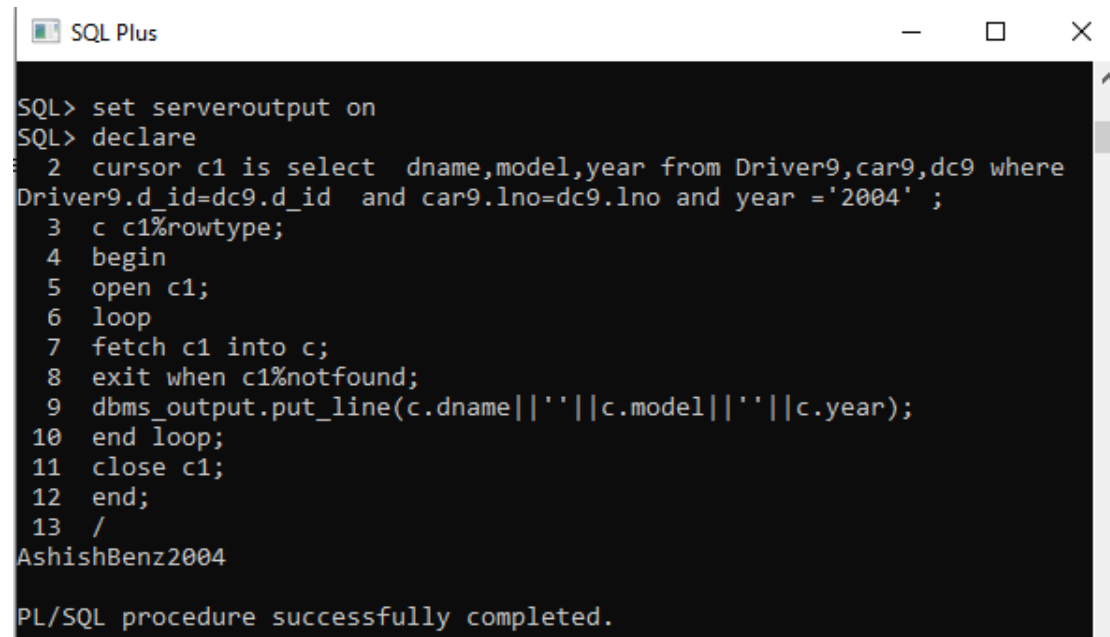
Procedure created.

SQL>
SQL> begin
  2  p10('Monday');
  3  end;
  4  /

Dnamedaymodelyear
AshishMondayBenz2004

PL/SQL procedure successfully completed.
```

2) Write a cursor which will display driver wise car details in the year 2004.

A screenshot of a SQL Plus window titled "SQL Plus". The window has a dark background with white text. The code is as follows:

```
SQL> set serveroutput on
SQL> declare
1  cursor c1 is select  dname,model,year from Driver9,car9,dc9 where
2  Driver9.d_id=dc9.d_id  and car9.lno=dc9.lno and year ='2004' ;
3  c c1%rowtype;
4  begin
5  open c1;
6  loop
7  fetch c1 into c;
8  exit when c1%notfound;
9  dbms_output.put_line(c.dname||' '||c.model||' '||c.year);
10 end loop;
11 close c1;
12 end;
13 /
```

The output of the code is "AshishBenz2004". At the bottom of the window, it says "PL/SQL procedure successfully completed.".