

# INDEX

Sr.No	Name of Practical	Date	Page No.	Remark
1	Create table DONAR			
2	Insert a Values in DONAR Table			
	<b>Perform the following Queries on DONAR table</b>			
3	Find all donars whose name starts between alphabets “A” to “S”			
4	Find all donar who belongs to city CHANDRAPUR			
5	Find all donars who does not belongs to CHANDRAPUR City			
6	Find all donars who belongs to either CHANDRAPUR or NAGPUR City			
7	Find all donars whose city value contains NULL			
8	Arrange all donars in the sorted order whose age is between 18 and 22			
9	Diply all Male Donars			
10	Find all male donars have o+Ve blood group			
11	Find all donars who donated the blood between 25-AUG-11 and 26-AUG-11			
12	Find all donars who donated more than 100 ml of blood			
13	Find all female donars who belong to city CHANDRAPUR having blood group “A+VE”			
14	Find all female donars who belong to city CHANDRAPUR having blood group “A+VE”			
15	Display all donars according their age			
16	Display the donar list in recent order of donation date			
17	Display all distinct blood group type			
18	Update the age of all donars by 1			
19	Mr. RAJESH RAO changed his name as RAMESH RAO and he is shifted to DURGAPUR. Note the above changes in the table			
20	Display certain name whose first name starts with letter “A” and ends with “D” irrespective of caseletter.			
<b>PL/SQL Programs</b>				
1	Write a PL/SQL block to accept the marks of 3 papers and display the result if student score more then 35 marks in each paper out of maximum marks 100 than declare pass otherwise fail. And specify the class depending upon following conditions: Percentage                  Class >=75                              Distinction			

	<div> <div>&gt;=60</div> <div>&gt;=45</div> <div>Otherwise</div> </div> <div> <div>First</div> <div>Second</div> <div>Third</div> </div>			
<b>2</b>	Write a PL/SQL block to find factorial of a number and store the number and its factorial in a table called FACT which contains two column namely number and result			
<b>3</b>	Write a PL/SQL block to check number is palindrome or not			
<b>4</b>	Write a program to divide a number by character number. If any error occurs it should be handled properly and store the error number and its description in a table called ERRORH			
<b>5</b>	<p>Write a PL/SQL block to accept and insert a valid data into the table PATIENT, having following structure. Write an appropriate user defined exception.</p> <div> <div>PNAME</div> <div>AGE</div> <div>PRESCRIPTION</div> </div> <div> <div>VARCHAR(20)</div> <div>NUMBER(2)</div> <div>VARCHAR2(20)</div> </div>			
<b>6</b>	Write a PL/SQL block to display first 2 records of table FRUITS which contain only one column called name			
<b>7</b>	Write a PL/SQL Program to Reverse the String			
<b>8</b>	Write a PL/SQL Program to convert each digit of the number into words			

**Practical- 1) Create table DONAR with following attributes**

Name	Null?	Type
------	-------	------

DNO	NOT NULL	NUMBER(4)
DNAME		VARCHAR2(20)
CITY		VARCHAR2(20)
AGE		NUMBER(3)
SEX		VARCHAR2(5)
BG		VARCHAR2(5)
QUANTITY		NUMBER(4)
DOD		DATE

SQL> create table DONAR(Dno number(4) primary key,Dname varchar(20),City varchar(20),Age number(3),SEX VARCHAR(5),BG varchar(5),Quantity number(4),DOD DATE);

Table created.

SQL> DESC DONAR

Name	Null?	Type
------	-------	------

DNO	NOT NULL	NUMBER(4)
DNAME		VARCHAR2(20)
CITY		VARCHAR2(20)
AGE		NUMBER(3)
SEX		VARCHAR2(5)
BG		VARCHAR2(5)
QUANTITY		NUMBER(4)
DOD		DATE

**B) Insert the following values in DONAR Table with following values**

DNO	DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
101	RAJESH RAO	CHANDRAPUR	28	M	O+ve	100	25-AUG-11
102	ANAND SHARMA	NAGPUR	20	M	O+VE	200	26-AUG-11
103	VISHAL DESHPANDE	HYDERABAD	23	M	O+VE	250	26-AUG-11
104	SHRUTI RAKHUNDE	CHANDRAPUR	22	F	A+VE	100	27-AUG-11
105	ANUSHREE DHAKATE	CHANDRAPUR	22	F	A-VE	200	26-AUG-11
106	VIJETA DHAKATE	BALLARPUR	22	F	O+VE	100	25-AUG-11
107	AAMIR TAJA	CHANDRAPUR	21	M	O+VE	250	27-AUG-11
108	AMIR KHAN	DURGAPUR	25	M	O+VE	100	25-AUG-11

```
SQL> insert into donar values(&dno,&dname,&city,&age,&sex,&BG,&QUANTITY,&DOD);
```

Enter value for dno: 101

Enter value for dname: RAJESH RAO

Enter value for city: CHANDRAPUR

Enter value for age: 28

Enter value for sex: M

Enter value for bg: O+ve

Enter value for quantity: 100

Enter value for dod: 25-AUG-11

```
old 1: insert into donar values(&dno,&dname,&city,&age,&sex,&BG,&QUANTITY,&DOD')
```

```
new 1: insert into donar values(101,'RAJESH RAO','CHANDRAPUR',28,'M','O+ve',100,'25-AUG-11')
```

1 row created.

```
SQL> /
```

Enter value for dno: 102

Enter value for dname: ANAND SHARMA

Enter value for city: NAGPUR

Enter value for age: 20

Enter value for sex: M

Enter value for bg: O+VE

Enter value for quantity: 200

Enter value for dod: 26-AUG-11

```
old 1: insert into donar values(&dno,&dname,&city,&age,&sex,&BG,&QUANTITY,&DOD')
```

```
new 1: insert into donar values(102,'ANAND SHARMA','NAGPUR',20,'M','O+VE',200,'26-AUG-11')
```

1 row created.

```
SQL> /
```

Enter value for dno: 103

Enter value for dname: VISHAL DESHPANDE

Enter value for city: HYDERABAD

Enter value for age: 23

Enter value for sex: M

Enter value for bg: O+VE

Enter value for quantity: 250

Enter value for dod: 26-AUG-11

```
old 1: insert into donar values(&dno,&dname,&city,&age,&sex,&BG,&QUANTITY,&DOD')
```

```
new 1: insert into donar values(103,'VISHAL DESHPANDE','HYDERABAD',23,'M','O+VE',250,'26-AUG-11')
```

1 row created.

```
SQL> /
```

Enter value for dno: 104

Enter value for dname: SHRUTI RAKHUNDE

Enter value for city: CHANDRAPUR

Enter value for age: 22

Enter value for sex: F

Enter value for bg: A+VE

Enter value for quantity: 100

Enter value for dod: 27-AUG-11

old 1: insert into donar values(&dno,&dname,&city,&age,&sex,&BG,&QUANTITY,&DOD')

new 1: insert into donar values(104,'SHRUTI RAKHUNDE','CHANDRAPUR',22,'F','A+VE',100,'27-AUG-11')

1 row created.

SQL> /

Enter value for dno: 105

Enter value for dname: ANUSHREE DHAKATE

Enter value for city: CHANDRAPUR

Enter value for age: 22

Enter value for sex: F

Enter value for bg: A-VE

Enter value for quantity: 200

Enter value for dod: 26-AUG-11

old 1: insert into donar values(&dno,&dname,&city,&age,&sex,&BG,&QUANTITY,&DOD')

new 1: insert into donar values(105,'ANUSHREE DHAKATE','CHANDRAPUR',22,'F','A-VE',200,'26-AUG-11')

1 row created.

SQL> /

Enter value for dno: 106

Enter value for dname: VIJETA DHAKATE

Enter value for city: BALLARPUR

Enter value for age: 22

Enter value for sex: F

Enter value for bg: O+VE

Enter value for quantity: 100

Enter value for dod: 25-AUG-11

old 1: insert into donar values(&dno,&dname,&city,&age,&sex,&BG,&QUANTITY,&DOD')

new 1: insert into donar values(106,'VIJETA DHAKATE','BALLARPUR',22,'F','O+VE',100,'25-AUG-11')

1 row created.

SQL> /

Enter value for dno: 107

Enter value for dname: AAMIR TAJA

Enter value for city: CHANDRAPUR

Enter value for age: 21

Enter value for sex: M

Enter value for bg: O+VE

Enter value for quantity: 250

Enter value for dod: 27-AUG-11

old 1: insert into donar values(&dno,&dname,&city,&age,&sex,&BG,&QUANTITY,&DOD')

new 1: insert into donar values(107,'AAMIR TAJA','CHANDRAPUR',21,'M','O+VE',250,'27-AUG-11')

1 row created.

SQL> /

Enter value for dno: 108

Enter value for dname: AMIR KHAN

Enter value for city: DURGAPUR

Enter value for age: 25

Enter value for sex: M

Enter value for bg: O+VE

Enter value for quantity: 100

Enter value for dod: 25-AUG-11

old 1: insert into donar values(&dno,&dname,&city,&age,&sex,&BG,&QUANTITY,&DOD')

new 1: insert into donar values(108,'AMIR KHAN','DURGAPUR',25,'M','O+VE',100,'25-AUG-11')

1 row created.

SQL> SELECT \* FROM DONAR;

DNO	DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
101	RAJESH RAO	CHANDRAPUR	28	M	O+ve	100	25-AUG-11
102	ANAND SHARMA	NAGPUR	20	M	O+VE	200	26-AUG-11
103	VISHAL DESHPANDE	HYDERABAD	23	M	O+VE	250	26-AUG-11
104	SHRUTI RAKHUNDE	CHANDRAPUR	22	F	A+VE	100	27-AUG-11
105	ANUSHREE DHAKATE	CHANDRAPUR	22	F	A-VE	200	26-AUG-11
106	VIJETA DHAKATE	BALLARPUR	22	F	O+VE	100	25-AUG-11
107	AAMIR TAJA	CHANDRAPUR	21	M	O+VE	250	27-AUG-11
108	AMIR KHAN	DURGAPUR	25	M	O+VE	100	25-AUG-11

8 rows selected.

### C) Perform following queries on above table

1) Find all donars whose name starts between alphabets "A" to "S"

SQL> select \* from donar where dname between 'A' AND 'S';

DNO	DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
101	RAJESH RAO	CHANDRAPUR	28	M	O+ve	100	25-AUG-11
102	ANAND SHARMA	NAGPUR	20	M	O+VE	200	26-AUG-11
105	ANUSHREE DHAKATE	CHANDRAPUR	22	F	A-VE	200	26-AUG-11
107	AAMIR TAJA	CHANDRAPUR	21	M	O+VE	250	27-AUG-11
108	AMIR KHAN	DURGAPUR	25	M	O+VE	100	25-AUG-11

**2) Find all donar who belongs to city CHANDRAPUR**

SQL> select \* from donar where city='CHANDRAPUR';

DNO	DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
101	RAJESH RAO	CHANDRAPUR	28	M	O+ve	100	25-AUG-11
104	SHRUTI RAKHUNDE	CHANDRAPUR	22	F	A+VE	100	27-AUG-11
105	ANUSHREE DHAKATE	CHANDRAPUR	22	F	A-VE	200	26-AUG-11
107	AAMIR TAJA	CHANDRAPUR	21	M	O+VE	250	27-AUG-11

**3) Find all donars who does not belongs to CHANDRAPUR City**

SQL> select \* from donar where city not in ('CHANDRAPUR');

DNO	DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
102	ANAND SHARMA	NAGPUR	20	M	O+VE	200	26-AUG-11
103	VISHAL DESHPANDE	HYDERABAD	23	M	O+VE	250	26-AUG-11
106	VIJETA DHAKATE	BALLARPUR	22	F	O+VE	100	25-AUG-11
108	AMIR KHAN	DURGAPUR	25	M	O+VE	100	25-AUG-11

**4) Find all donars who belongs to either CHANDRAPUR or NAGPUR City**

SQL> SELECT \* FROM DONAR WHERE CITY IN('CHANDRAPUR','NAGPUR');

DNO DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
-----------	------	-----	-----	----	----------	-----



102	ANAND SHARMA	NAGPUR	20	M	O+VE	200	26-AUG-11
103	VISHAL DESHPANDE	HYDERABAD	23	M	O+VE	250	26-AUG-11
107	AAMIR TAJA	CHANDRAPUR	21	M	O+VE	250	27-AUG-11
108	AMIR KHAN	DURGAPUR	25	M	O+VE	100	25-AUG-11

**9) Find all donars who donated the blood between 25-AUG-11 and 26-AUG-11**

SQL> select \* from donar where dod between '25-AUG-11' AND '26-AUG-11';

DNO	DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
101	RAJESH RAO	CHANDRAPUR	28	M	O+ve	100	25-AUG-11
102	ANAND SHARMA	NAGPUR	20	M	O+VE	200	26-AUG-11
103	VISHAL DESHPANDE	HYDERABAD	23	M	O+VE	250	26-AUG-11
105	ANUSHREE DHAKATE		22	F	A-VE	200	26-AUG-11
106	VIJETA DHAKATE	BALLARPUR	22	F	O+VE	100	25-AUG-11
108	AMIR KHAN	DURGAPUR	25	M	O+VE	100	25-AUG-11

6 rows selected.

**10) Find all donars who donated more than 100 ml of blood**

SQL> select \* from donar where quantity > 100;

DNO	DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
102	ANAND SHARMA	NAGPUR	20	M	O+VE	200	26-AUG-11
103	VISHAL DESHPANDE	HYDERABAD	23	M	O+VE	250	26-AUG-11
105	ANUSHREE DHAKATE		22	F	A-VE	200	26-AUG-11
107	AAMIR TAJA	CHANDRAPUR	21	M	O+VE	250	27-AUG-11

**11) Find all female donars who belong to city CHANDRAPUR having blood group "A+VE"**

SQL> SELECT \* FROM DONAR WHERE SEX='F' AND CITY='CHANDRAPUR';

DNO	DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
104	SHRUTI RAKHUNDE	CHANDRAPUR	22	F	A+VE	100	27-AUG-11

**12) Display all donars according their age**

SQL> SELECT \* FROM DONAR ORDER BY AGE;

DNO	DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
102	ANAND SHARMA	NAGPUR	20	M	O+VE	200	26-AUG-11

107 AAMIR TAJA	CHANDRAPUR	21	M	O+VE	250	27-AUG-11
104 SHRUTI RAKHUNDE	CHANDRAPUR	22	F	A+VE	100	27-AUG-11
106 VIJETA DHAKATE	BALLARPUR	22	F	O+VE	100	25-AUG-11
105 ANUSHREE DHAKATE		22	F	A-VE	200	26-AUG-11
103 VISHAL DESHPANDE	HYDERABAD	23	M	O+VE	250	26-AUG-11
108 AMIR KHAN	DURGAPUR	25	M	O+VE	100	25-AUG-11
101 RAJESH RAO	CHANDRAPUR	28	M	O+ve	100	25-AUG-11

8 rows selected.

### 13) Display the donar list in recent order of donation date

SQL> SELECT \* FROM DONAR ORDER BY DOD DESC;

DNO	DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
104	SHRUTI RAKHUNDE	CHANDRAPUR	22	F	A+VE	100	27-AUG-11
107	AAMIR TAJA	CHANDRAPUR	21	M	O+VE	250	27-AUG-11
103	VISHAL DESHPANDE	HYDERABAD	23	M	O+VE	250	26-AUG-11
105	ANUSHREE DHAKATE		22	F	A-VE	200	26-AUG-11
102	ANAND SHARMA	NAGPUR	20	M	O+VE	200	26-AUG-11
108	AMIR KHAN	DURGAPUR	25	M	O+VE	100	25-AUG-11
106	VIJETA DHAKATE	BALLARPUR	22	F	O+VE	100	25-AUG-11
101	RAJESH RAO	CHANDRAPUR	28	M	O+ve	100	25-AUG-11

8 rows selected.

### 14) Display all distinct blood group type

SQL> SELECT DISTINCT BG FROM DONAR;

```

BG
----
O+ve
A-VE
A+VE
O+VE

```

### 15) Update the age of all donars by 1

SQL> UPDATE DONAR SET AGE=AGE+1;

8 rows updated.

SQL> SELECT \* FROM DONAR;

DNO	DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
101	RAJESH RAO	CHANDRAPUR	29	M	O+ve	100	25-AUG-11
102	ANAND SHARMA	NAGPUR	21	M	O+VE	200	26-AUG-11
103	VISHAL DESHPANDE	HYDERABAD	24	M	O+VE	250	26-AUG-11
104	SHRUTI RAKHUNDE	CHANDRAPUR	23	F	A+VE	100	27-AUG-11
105	ANUSHREE DHAKATE		23	F	A-VE	200	26-AUG-11
106	VIJETA DHAKATE	BALLARPUR	23	F	O+VE	100	25-AUG-11
107	AAMIR TAJA	CHANDRAPUR	22	M	O+VE	250	27-AUG-11
108	AMIR KHAN	DURGAPUR	26	M	O+VE	100	25-AUG-11

8 rows selected.

**16) Mr. RAJESH RAO changed his name as RAMESH RAO and he is shifted to DURGAPUR. Note the above changes in the table**

SQL> UPDATE DONAR SET DNAME='RAMESH RAO',CITY='DURGAPUR' WHERE DNO=101;

1 row updated.

SQL> SELECT \* FROM DONAR;

DNO	DNAME	CITY	AGE	SEX	BG	QUANTITY	DOD
101	RAMESH RAO	DURGAPUR	29	M	O+ve	100	25-AUG-11
102	ANAND SHARMA	NAGPUR	21	M	O+VE	200	26-AUG-11
103	VISHAL DESHPANDE	HYDERABAD	24	M	O+VE	250	26-AUG-11
104	SHRUTI RAKHUNDE	CHANDRAPUR	23	F	A+VE	100	27-AUG-11
105	ANUSHREE DHAKATE		23	F	A-VE	200	26-AUG-11
106	VIJETA DHAKATE	BALLARPUR	23	F	O+VE	100	25-AUG-11
107	AAMIR TAJA	CHANDRAPUR	22	M	O+VE	250	27-AUG-11
108	AMIR KHAN	DURGAPUR	26	M	O+VE	100	25-AUG-11

8 rows selected.

**17) Display certain name whose first name starts with letter "A" and ends with "D" irrespective of caseletter.**

SQL> select \* from donar where dname like'A%D';

no rows selected

## PL/SQL Programs

**1)Que: Write a PL/SQL block to accept the marks of 3 papers and display the result if student score more then 35 marks in each paper out of maximum marks 100 than declare pass otherwise fail. And specify the class depending upon following conditions:**

<b>Percentage</b>	<b>Class</b>
<b>&gt;=75</b>	<b>Distinction</b>
<b>&gt;=60</b>	<b>First</b>

**>=45**  
**Otherwise**

**Second**  
**Third.**

SQL>ED PL1

SET SERVEROUTPUT ON

SET VERIFY OFF

CLEAR SCREEN

DECLARE

PAPER1 INTEGER:=&PAPER1;

PAPER2 INTEGER:=&PAPER2;

PAPER3 INTEGER:=&PAPER3;

TOTALM INTEGER;

PER NUMBER(5,2);

CLASS CHAR(15);

BEGIN

IF(PAPER1>=35 AND PAPER2>=35 AND PAPER3>=35) THEN

DBMS\_OUTPUT.PUT\_LINE('PASS');

TOTALM:=PAPER1+PAPER2+PAPER3;

PER:=TOTALM/3;

IF(PER>=75) THEN

CLASS:='DISTINCTION';

ELSIF(PER>=60)THEN

CLASS:='FIRSTCLASS';

ELSIF(PER>=45)THEN

CLASS:='SECONDCLASS';

ELSE

CLASS:='THIRDCLASS';

END IF;

DBMS\_OUTPUT.PUT\_LINE('SCORED'|| PER||'% AND GOT'||CLASS);

ELSE

DBMS\_OUTPUT.PUT\_LINE('FAIL');

END IF;

END;

/

### **OUTPUT:**

SQL>@ PL1

Enter value for paper1: 75

Enter value for paper2: 75

Enter value for paper3: 75

PASS

SCORED75%AND GOTDISTINCTION

PL/SQL procedure successfully completed

.

**2)QUE:-/\*Write a PL/SQL block to find factorial of a number and store the number and its factorial in a table called FACT which contains two column namely number and result.\*/**

SQL>ED PL2

```
SET SERVEROUTPUT ON
SET VERIFY OFF
CLEAR SCREEN
DECLARE
    N INTEGER;
    F NUMBER:=1;
BEGIN
    N:=&N;
    FOR I IN REVERSE 1..N
LOOP
    F:=F*I;
END LOOP;
    DBMS_OUTPUT.PUT_LINE('FACTORIAL OF ' ||N|| ' IS ' ||F);
    INSERT INTO FACT VALUES(N,F);
END;
/
```

### **OUTPUT:-**

SQL>@PL2

Enter value for n: 5

FACTORIAL OF 5 IS 120

PL/SQL procedure successfully completed.

SQL> SELECT\* FROM FACT;

FNUMBER	RESULT
---------	--------

5	120
---	-----

**3) Que: Write a PL/SQL block to check number is palindrome or not.**

SQL>ED PL3

```
SET SERVEROUTPUT ON
SET VERIFY OFF
CLEAR SCREEN
DECLARE
    N NUMBER;
    R NUMBER;
    S NUMBER:=0;
    TN NUMBER;
BEGIN
    N:=&N;
    TN:=N;
    WHILE(N>0)
    LOOP
        R:=MOD(N,10);
        S:=S*10+R;
        N:=FLOOR(N/10);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('REVERSE NUMBER IS '||S);
    IF(TN=S) THEN
        DBMS_OUTPUT.PUT_LINE('NUMBER '||TN||' IS PALINDROME');
    ELSE
        DBMS_OUTPUT.PUT_LINE('NUMBER '||TN||' IS NOT PALINDROME');
    END IF;
END;
/
```

**OUTPUT:**

SQL>@ PL3

```
Enter value for n: 121
REVERSE NUMBER IS 121
NUMBER 121 IS PALINDROME
PL/SQL procedure successfully completed.
```

SQL>@ PL3

```
Enter value for n: 122
REVERSE NUMBER IS 122
NUMBER 122 IS NOT PALINDROME

PL/SQL procedure successfully completed.
```

- 4) **Que:** Write a program to divide a number by character number. If any error occurs it should be handled properly and store the error number and its description in a table called **ERRORH**.

```
SQL> create table ERRORH
 2 (ERRNO NUMBER(10),
 3 ERR_DESC VARCHAR2(100));
```

Table created.

```
SQL> ED PL4
```

```
SET SERVEROUTPUT ON
SET VERIFY OFF
CLEAR SCREEN
DECLARE
    A NUMBER(6):=&A;
    B CHAR(6):='&B';
    C NUMBER(7);
    TERRNO          ERRORH.ERRNO%TYPE;
    TERR_DESC ERRORH.ERR_DESC%TYPE;
BEGIN
    C:=A/TO_NUMBER(B);
    DBMS_OUTPUT.PUT_LINE('DIVISION OF TWO NUMBER IS'||C);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('ERROR:- '||SQLCODE);
        DBMS_OUTPUT.PUT_LINE('DESCRIPTION OF ERROR'||SQLERRM);
        TERRNO:=SQLCODE;
        TERR_DESC:=SUBSTR(SQLERRM,10);
INSERT INTO ERRORH VALUES(TERRNO,TERR_DESC);
END;
/
```

### **OUTPUT:**

```
SQL> @PL4
Enter value for a: 5
Enter value for b: Y
ERROR:- -6502
DESCRIPTION OF ERRORORA-06502: PL/SQL: numeric or value error: character to
number conversion error
```

PL/SQL procedure successfully completed.

```
SQL> select * from ERRORH;
ERRNO  ERR_DESC
```

```
-----
-6502   : PL/SQL: numeric or value error: character to number conversion error
```

- 5) **Que:** Write a PL/SQL block to accept and insert a valid data into the table **PATIENT**, having following structure. Write an appropriate user defined exception.

<b>PNAME</b>	<b>VARCHAR(20)</b>
<b>AGE</b>	<b>NUMBER(2)</b>
<b>PRESCRIPTION</b>	<b>VARCHAR2(20)</b>

SQL>ED PL5

SET SERVEROUTPUT ON

SET VERIFY OFF

CLEAR SCREEN

DECLARE

TPNAME PATIENT.PNAME%TYPE;

TAGE PATIENT.AGE%TYPE;

TPRESCRIPTION PATIENT.PRESCRIPTION%TYPE;

CHECK\_CHAR EXCEPTION;

CHECK\_AGE EXCEPTION;

BEGIN

TPNAME:='&NAME';

IF TPNAME IS NULL THEN

RAISE CHECK\_CHAR;

END IF;

TAGE:=&AGE;

IF TAGE<=0 THEN

RAISE CHECK\_AGE;

ELSE

NULL;

END IF;

TPRESCRIPTION:='&PRESCRIPTION';

IF TPRESCRIPTION IS NULL THEN

RAISE CHECK\_CHAR;

END IF;

INSERT INTO PATIENT VALUES(TPNAME,TAGE,TPRESCRIPTION);

EXCEPTION

WHEN CHECK\_CHAR THEN

DBMS\_OUTPUT.PUT\_LINE('CAN NOT BE BLANK');

WHEN CHECK\_AGE THEN

DBMS\_OUTPUT.PUT\_LINE('AGE SHOULD BE > 0');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('ERROR:-'||SQLCODE);

DBMS\_OUTPUT.PUT\_LINE('DESCRIPTION OF ERROR'||SQLERRM);

END;

/

### **OUTPUT:**

SQL>@PL5

Enter value for name: RAM

Enter value for age: 25

Enter value for prescription: CROCIN

PL/SQL procedure successfully completed.



SQL>@PL5

Enter value for name: RAJ

Enter value for age: -25

Enter value for prescription: DISPRIN

AGE SHOULD BE > 0

PL/SQL procedure successfully completed.

SQL>@PL5

Enter value for name: RAHUL

Enter value for age: 30

Enter value for prescription:

CAN NOT BE BLANK

PL/SQL procedure successfully completed.

SQL> SELECT \* FROM PATIENT;

PNAME	AGE	PRESCRIPTION
-------	-----	--------------

-----

RAM	25	CROCIN
-----	----	--------

**6) Que:-/\*Write a PL/SQL block to display first 2 records of table FRUITS which contain only one column called name.\*/\***

```
SQL> ED PL5
SET SERVEROUTPUT ON
SET VERIFY OFF
CLEAR SCREEN
DECLARE
    CURSOR CFRUITS IS SELECT NAME FROM FRUITS;
    CNAME VARCHAR2(20);
BEGIN
    OPEN CFRUITS;
    DBMS_OUTPUT.PUT_LINE('FRUITS ARE');
    DBMS_OUTPUT.PUT_LINE('*****');
    FOR I IN 1..2
LOOP
    FETCH CFRUITS INTO CNAME;
    DBMS_OUTPUT.PUT_LINE(CNAME);
END LOOP;
    DBMS_OUTPUT.PUT_LINE('*****');
    CLOSE CFRUITS;
EXCEPTION
    WHEN CURSOR_ALREADY_OPEN THEN
        DBMS_OUTPUT.PUT_LINE('CURSOR ALREADY OPEN');
    END;
/
```

**OUTPUT:-**

```
SQL> SELECT* FROM FRUITS;
```

```
NAME
-----
```

```
APPLE
MANGO
ORANGE
```

```
SQL>@ PL5
```

```
FRUITS ARE
*****
APPLE
MANGO
*****
```

PL/SQL procedure successfully completed.

### Que 7. Write a PL/SQL Program to Reverse the String

DECLARE

-- declare variable str , len

-- and str1 of datatype varchar

str VARCHAR(20) := 'skeegrofskeeg';

len NUMBER;

str1 VARCHAR(20);

BEGIN

-- Here we find the length of string

len := Length(str);

-- here we starting a loop from max len to 1

FOR i IN REVERSE 1.. len LOOP

-- assigning the reverse string in str1

str1 := str1

|| Substr(str, i, 1);

END LOOP;

dbms\_output.Put\_line('Reverse of string is '

|| str1);

END;

-- Program End

Result :-

Input: skeegrofskeeg

Output: geeksforgeeks

**8) Write a PL/SQL Program to convert each digit of the number into words**

```
DECLARE
    num          INTEGER;
    num_to_word  VARCHAR2(100);
    str          VARCHAR2(100);
    len          INTEGER;
    c            INTEGER;
BEGIN
    num := 123456;
    len := Length(num);
    dbms_output.Put_line('Entered Number is: '
                          ||num);

    FOR i IN 1..len LOOP
        c := Substr(num, i, 1);

        SELECT Decode(c, 0, 'Zero ',
                      1, 'One ',
                      2, 'Two ',
                      3, 'Three ',
                      4, 'Four ',
                      5, 'Five ',
                      6, 'Six ',
                      7, 'Seven ',
                      8, 'Eight ',
                      9, 'Nine ')
        INTO str
        FROM dual;
        num_to_word := num_to_word
                      ||str;
    END LOOP;
    dbms_output.Put_line('Number to words: '
                          ||num_to_word);
END;
```

**Output :**

Entered Number is: 123456

Number to words: One Two Three Four Five Six