EXPERIMENT- 6

IMPLEMENTATION OF BUILT IN FUNCTIONS IN

RDBMS USING TEMP TABLES.

AIM:

To implement built in functions in RDBMS using temp tables.

THEORY:

CONCAT- Adds two or more strings together.

LENGTH- Returns the length of a string.

REVERSE- Reverses a string and returns the value.

SQRT- Returns the square root of a number.

FLOOR- This function rounds the specified number down and returns the largest number that is less than or equal to the specified number.

CEIL- This function rounds the specified number up and returns the smallest number that is greater than or equal to the specified number.

ABS- Returns the absolute value of a number.

Questions.

Build in Functions

1)Find the Ceil value of 8.29

2) Find the floor value of9.76

3) Find the square root of100

4) Demonstrate the use of least & greatest funs in string

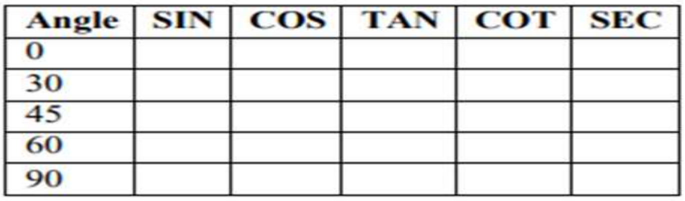
5) Display the systemtime

6) Use Select To_Char ( ) to display date and time in different formats

7) Find the abs value of 8.29

8) Create a table named angle

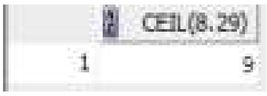
. Create a table named angle



Angle	SIN	COS	TAN	COT	SEC
0					
30					
45					
60					
90					

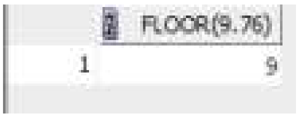
Queries and Output:

1)Find the Ceil value of8.29
select ceil(8.29) from dual;



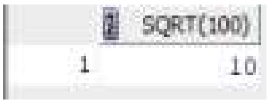
```
  CEIL(8.29)
1          9
```

2) Find the floor value of9.76
select floor(9.76) from dual;



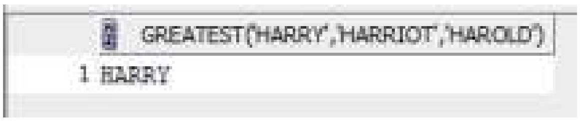
```
  FLOOR(9.76)
1           9
```

3) Find the square root of100
select sqrt(100) from dual;



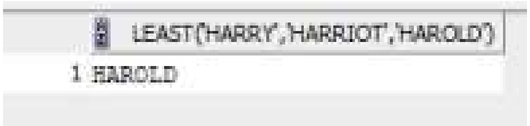
```
  SQRT(100)
1        10
```

4) Demonstrate the use of least & greatest functions in a string
SELECT GREATEST ('HARRY', 'HARRIOT', 'HAROLD') FROM DUAL;



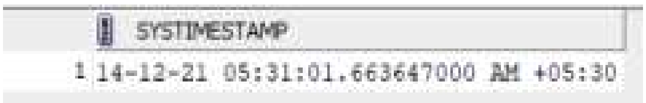
```
  GREATEST('HARRY','HARRIOT','HAROLD')
1 HARRY
```

SELECT LEAST ('HARRY', 'HARRIOT', 'HAROLD') FROM DUAL;



```
  LEAST('HARRY','HARRIOT','HAROLD')
1 HAROLD
```

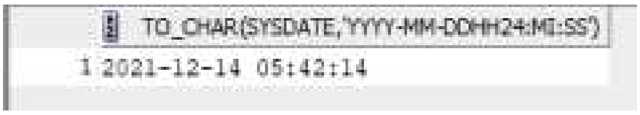
5) Display the systemtime
select SYSTIMESTAMP from dual;



```
  SYSTIMESTAMP
1 14-12-21 05:31:01.663647000 AM +05:30
```

6) . Use Select To_Char ( ) to display date and time in differentformats

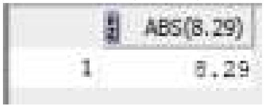
```
SELECT
        TO_CHAR(SYSDATE, 'YYYY-MM-DD HH24:MI:SS')
FROM
 dual;
```



```
     TO_CHAR(SYSDATE,'YYYY-MM-DDHH24:MI:SS')
   1 2021-12-14 05:42:14
```

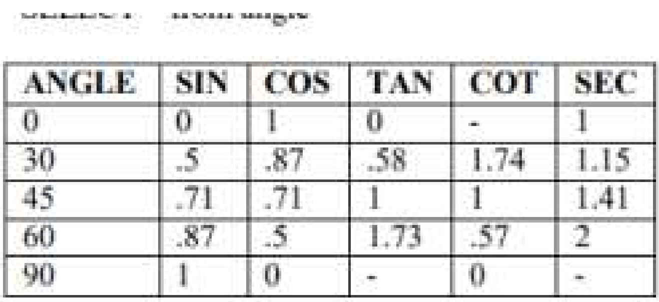
7) Find the abs value of8.29

```
select abs(8.29) from dual;
```



```
     ABS(8.29)
   1      8.29
```

8) . Create a table named angle



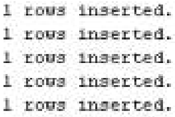
SELECT * from angle

ANGLE	SIN	COS	TAN	COT	SEC
0	0	1	0	-	1
30	.5	.87	.58	1.74	1.15
45	.71	.71	1	1	1.41
60	.87	.5	1.73	.57	2
90	1	0	-	0	-

```
create table angle(angle int,sin decimal(4,2),cos decimal(4,2),tan decimal(4,2),cot decimal(4,2),sec decimal(4,2));
```

```
table ANGLE created.
```

```
insert into angle(angle) values(0);
 insert into angle(angle) values(30);
insert into angle(angle) values(45);
insert into angle(angle) values(60);
 insert into angle(angle) values(90);
```



```
1 rows inserted.
1 rows inserted.
1 rows inserted.
1 rows inserted.
1 rows inserted.
```

```
UPDATE angle SET sin=sin(angle*(3.14/180));
```

```
5 rows updated.
```

```
UPDATE angle SET cos=cos(angle*(3.14/180));
```

```
5 rows updated.
```

```
UPDATE angle SET tan=(sin/cos) where cos!=0;
```

```
4 rows updated.
```

UPDATE angle SET sec=(1/cos) where cos!=0;

```
4 rows updated.
```

UPDATE angle SET cot=(cos/sin) where sin!=0;

9. Use the system table DUAL for the following questions:

1. Find the reverse of the string 'nmutuAotedOehT'
select reverse('nmutuAotedOehT') from dual;

2.Use LTRIM function on '123231xyzTech' so as to obtain the output Tech' select
ltrim('123231xyzTech','123231xyz') from dual;

3. Use RTRIM function on Computer ' to remove the trailing spaces. select
rtrim('Computer ') from dual;

4 Perform RPAD on 'computer' to obtain the output as'computerXXXX' select
rpad('computer',12,'x') from dual; 5 Find the length of the string 'Database
ManagementSystems'. select length('Database ManagementSystems') from dual;
6.Concatenate the strings 'Julius' and'Caesar'
select concat('Julius','Ceaser') from dual;

7.Use SUBSTR function to retrieve the substring 'is' from the string 'India is my country' select
substr('India is my country',7,2) from dual;

RESULT: Successfully implemented the built in function in RDBMS using Temp tables and Output is verified.CO2 is attained.

EXPERIMENT.NO- 10

IMPLEMENTATION OF SQL TCL COMMANDS

AIM:

To implement SQL TCL commands.

THEORY:

In SQL, TCL stands for Transaction control language.

A single unit of work in a database is formed after the consecutive execution of commands is known as a transaction.

There are certain commands present in SQL known as TCL commands that help the user manage the transactions that take place in a database.

COMMIT. ROLLBACK and SAVEPOINT are the most commonly used TCL commands in SQL.

Questions:

Create a table bank with bankname,headoffice,branch and branchcode as attributes and perform commit,savepoint and rollback commands.

Queries and Output:

TCL commands

Sample Table

```
CREATE TABLE BANK(bankcode varchar(3),
        bankname varchar(30) not null,
        headoffice varchar(30),
        branches int not null check(branches>0),
        primary key(bankcode)
        );


insert into BANK values('SBT','SBI Bank','Delhi',30);
insert into BANK values('CNB','Canara Bank','Ernakulam',20);
```

insert into BANK values('SIB','South Indian Bank','Madras',30);

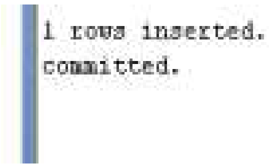
insert into BANK values('AXB','Axis Bank','Kottayam',15); insert

into BANK values('FDB','Federal Bank','Ernakulam',25);

1.COMMIT

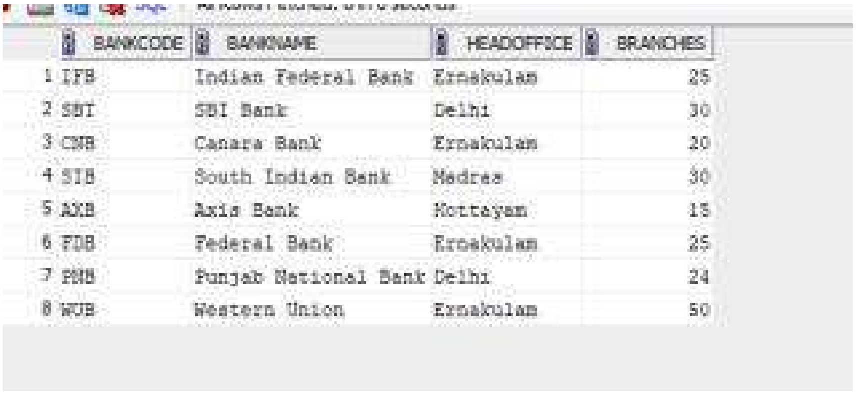
insert into BANK values('IFB','Indian Federal Bank','Ernakulam',25);

commit;



```
1 rows inserted.
committed.
```

select * from BANK;

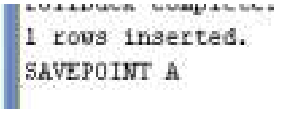


	BANKCODE	BANKNAME	HEADOFFICE	BRANCHES
1	IFB	Indian Federal Bank	Ernakulam	25
2	SBT	SBI Bank	Delhi	30
3	CNB	Canara Bank	Ernakulam	20
4	SIB	South Indian Bank	Madras	30
5	AXB	Axis Bank	Kottayam	15
6	FDB	Federal Bank	Ernakulam	25
7	PNB	Punjab National Bank	Delhi	24
8	WUB	Western Union	Ernakulam	50

2.SAVEPOINT

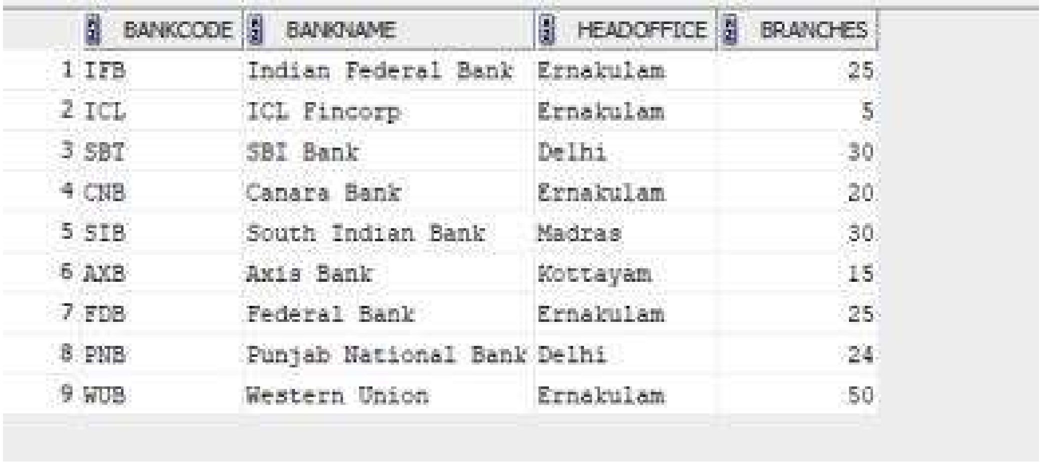
insert into BANK values('ICL','ICL Fincorp','Ernakulam',5);

SAVEPOINT A;



```
rollback complete.
1 rows inserted.
SAVEPOINT A
```

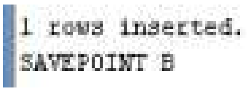
select * from BANK;



	BANKCODE	BANKNAME	HEADOFFICE	BRANCHES
1	IFB	Indian Federal Bank	Ernakulam	25
2	ICL	ICL Fincorp	Ernakulam	5
3	SBT	SBI Bank	Delhi	30
4	CNB	Canara Bank	Ernakulam	20
5	SIB	South Indian Bank	Madras	30
6	AXB	Axis Bank	Kottayam	15
7	FDB	Federal Bank	Ernakulam	25
8	PNB	Punjab National Bank	Delhi	24
9	WUB	Western Union	Ernakulam	50

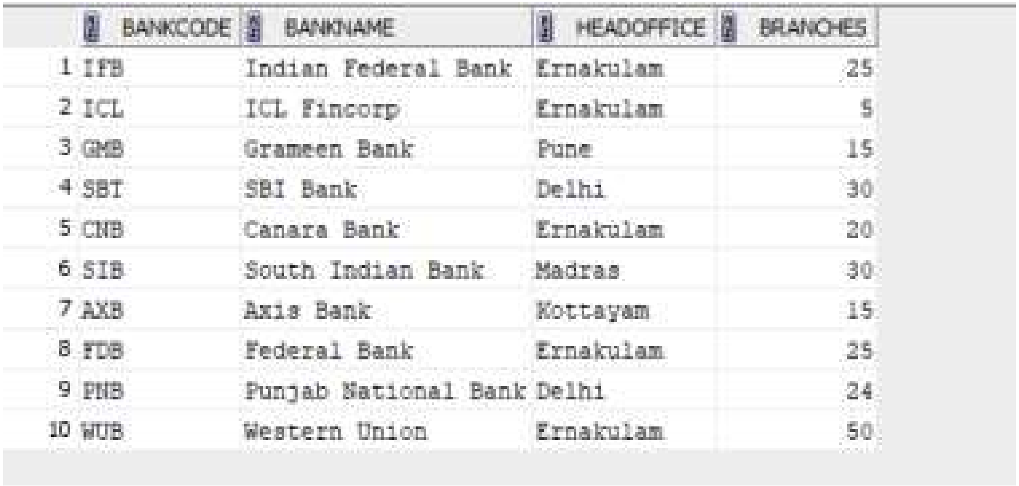
insert into BANK values('GMB','Grameen Bank','Pune',15);

SAVEPOINT B;



```
1 rows inserted.
SAVEPOINT B
```

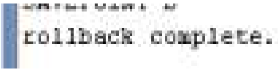
select * from BANK;



	BANKCODE	BANKNAME	HEADOFFICE	BRANCHES
1	IFB	Indian Federal Bank	Ernakulam	25
2	ICL	ICL Fincorp	Ernakulam	5
3	GMB	Grameen Bank	Pune	15
4	SBT	SBI Bank	Delhi	30
5	CNB	Canara Bank	Ernakulam	20
6	SIB	South Indian Bank	Madras	30
7	AXB	Axis Bank	Kottayam	15
8	FDB	Federal Bank	Ernakulam	25
9	PNB	Punjab National Bank	Delhi	24
10	WUB	Western Union	Ernakulam	50

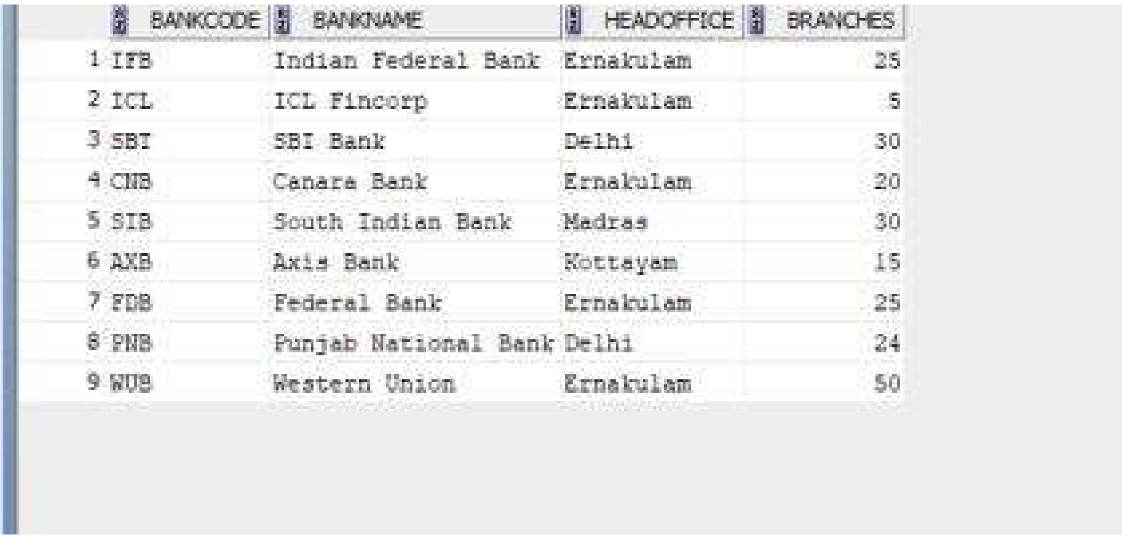
3.ROLLBACK

ROLLBACK TO B;



```
rollback complete.
```

select * from BANK;



	BANKCODE	BANKNAME	HEADOFFICE	BRANCHES
1	IFB	Indian Federal Bank	Ernakulam	25
2	ICL	ICL Fincorp	Ernakulam	5
3	SBI	SBI Bank	Delhi	30
4	CNB	Canara Bank	Ernakulam	20
5	SIB	South Indian Bank	Madras	30
6	AXB	Axis Bank	Kottayam	15
7	FDB	Federal Bank	Ernakulam	25
8	PNB	Punjab National Bank	Delhi	24
9	WUB	Western Union	Ernakulam	50

RESULT:

Implementation of SQL TCL commands is successfully done and Output is verified. CO2 is attained.

EXPERIMENT- 11

IMPLEMENTATION OF DCL COMMANDS

AIM:

To implement DCL commands.

THEORY:

TDCL (Data Control Language):

DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions, and other controls of the database system.

List of DCL commands:

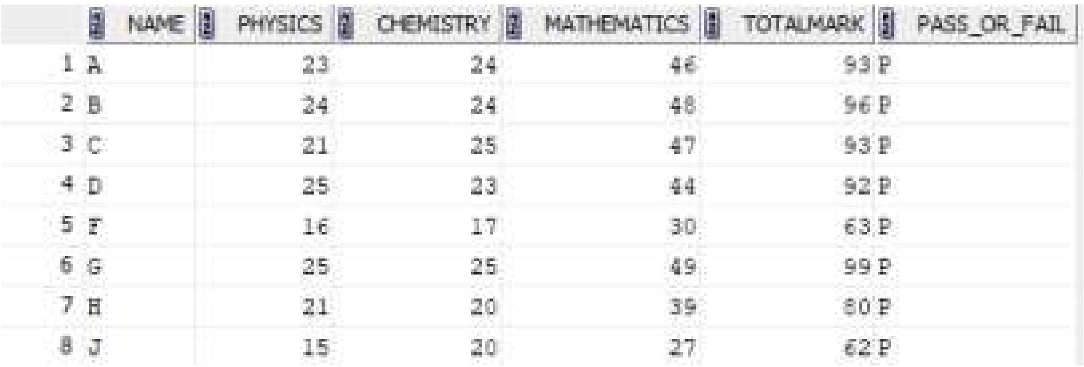
GRANT: This command gives users access privileges to the database.

REVOKE: This command withdraws the user's access privileges given by using the GRANT command.

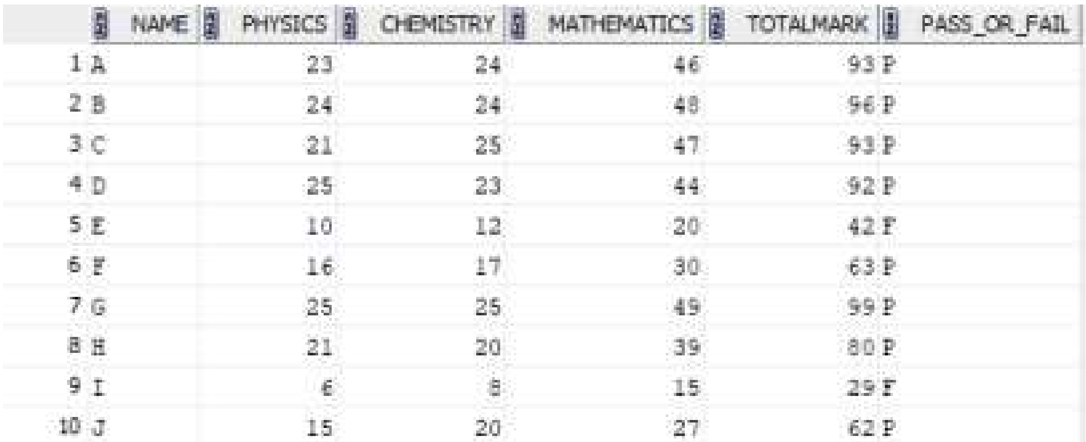
Queries and Output:

ROLLBACK,COMMIT,SAVEPOINT

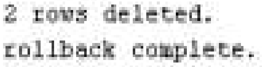
```
ROLLBACK
select *from STUDENT
delete from STUDENT where PASS_OR_FAIL='F';
ROLLBACK;
```



	NAME	PHYSICS	CHEMISTRY	MATHEMATICS	TOTALMARK	PASS_OR_FAIL
1	A	23	24	46	93	P
2	B	24	24	48	96	P
3	C	21	25	47	93	P
4	D	25	23	44	92	P
5	F	16	17	30	63	P
6	G	25	25	49	99	P
7	H	21	20	39	80	P
8	J	15	20	27	62	P



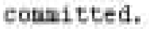
	NAME	PHYSICS	CHEMISTRY	MATHEMATICS	TOTALMARK	PASS_OR_FAIL
1	A	23	24	46	93	P
2	B	24	24	48	96	P
3	C	21	25	47	93	P
4	D	25	23	44	92	P
5	E	10	12	20	42	F
6	F	16	17	30	63	P
7	G	25	25	49	99	P
8	H	21	20	39	80	P
9	I	6	8	15	29	F
10	J	15	20	27	62	P



```
2 rows deleted.
rollback complete.
```

select *from STUDENT

commit



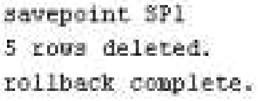
```
committed.
```

SAVEPOINT

savepoint SP1;

delete from STUDENT where TOTALMARK>85;

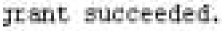
ROLLBACK TO SP1;



```
savepoint SP1
5 rows deleted.
rollback complete.
```

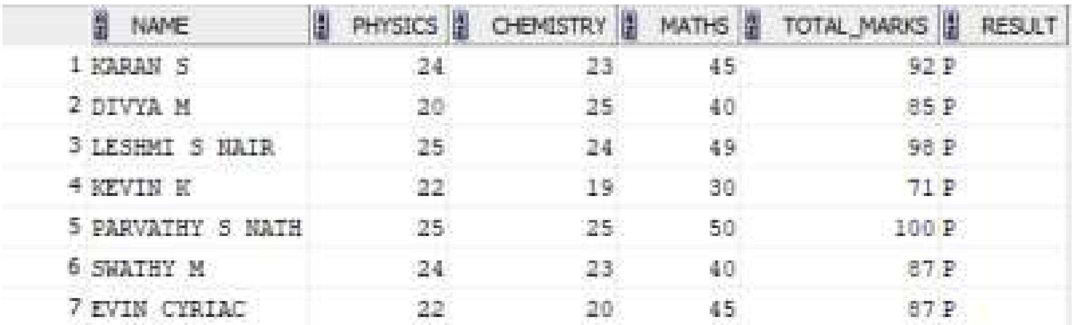
GRANT,REVOKE

grant select on STUDENT to C19CSB14



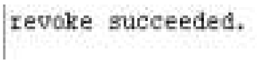
```
grant succeeded.
```

select *from C19CSB14.STUDENT



	NAME	PHYSICS	CHEMISTRY	MATHS	TOTAL_MARKS	RESULT
1	KARAN S	24	23	45	92	P
2	DIVYA M	20	25	40	85	P
3	LESHMI S NAIR	25	24	49	98	P
4	KEVIN K	22	19	30	71	P
5	PARVATHY S NATH	25	25	50	100	P
6	SWATHY M	24	23	40	87	P
7	EVIN CYRIAC	22	20	45	87	P

revoke select on student from c19csb14



```
revoke succeeded.
```

RESULT:

Successfully implemented DCL commands and CO2 is attained.

EXPERIMENT.NO- 12

CREATION OF VIEWS AND ASSERTIONS.

AIM:

To create views and assertions.

THEORY:

Views in SQL

Views in SQL are considered as a virtual table. A view also contains rows and columns.

To create the view, we can select the fields from one or more tables present in the database.

A view can either have specific rows based on certain conditions or all the rows of a table.

An assertion is a statement in SQL that ensures a certain condition will always exist in the database. Assertions are like column and table constraints, except that they are specified separately from table definitions.

Questions:

QUESTION

i)Create a table named Bank with the following attributes

-bankcode (To be set as Primary Key, type= varchar(3))

-bankname (Should not beNULL)

-headoffice

-branches (Integer value greater tha nZero)

Populate the database.

Make sure that all constraints are working properly.

All constraints have to be set after creating the table.

ii)Create a table named Branch with the following attribute

-branchid (To be set as PrimaryKey)

-branchname (Set Default value as 'New Delhi') –

bankid (Foreign Key:- Refers to bank code of Bank table)

Populate the database. Make sure that all constraints are working properly

iii) Delete the bank with bank code 'SBT' and make sure that the corresponding entries are getting deleted from the related tables.

iv) Drop the Primary Key in branch using ALTER command

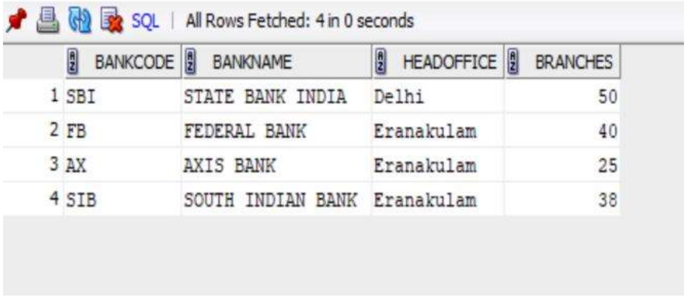
v) Create a View named bank_head office to hold the details of all bank whose head office at Ernakulam.

vi) Create a View named bank_branch to hold the details of all bank who have branches at kottayam.

Queries and Output:

```
1) create table Bank(bankcode varchar(3),
            bankname varchar(30) not null,
            headoffice varchar(15),
            branches int not null check(branches>0),
            primary key(bankcode));
```

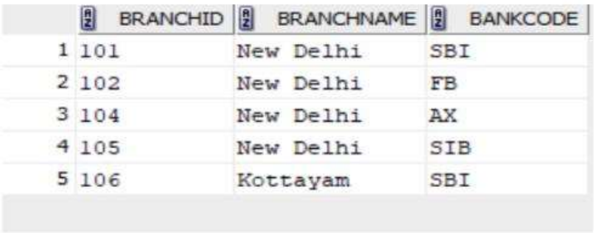
```
insert into Bank values('SBI','STATE BANK INDIA','Delhi','50'); insert into Bank values('FB','FEDERAL
BANK','Eranakulam','40'); insert into Bank values('SBT','STATE BANK
TRAVANCORE','Kottayam','35'); insert into Bank values('AX','AXIS BANK ','Eranakulam','25'); insert
into Bank values('SIB','SOUTH INDIAN BANK ','Eranakulam','38'); select *from Bank;
```



	BANKCODE	BANKNAME	HEADOFFICE	BRANCHES
1	SBI	STATE BANK INDIA	Delhi	50
2	FB	FEDERAL BANK	Eranakulam	40
3	AX	AXIS BANK	Eranakulam	25
4	SIB	SOUTH INDIAN BANK	Eranakulam	38

```
2) create table Branch(branchid varchar(3),

        branchname varchar(30) DEFAULT'New Delhi',

        bankcode varchar(3),

        primary key(branchid)

        );
```

```
ALTER TABLE Branch

ADD FOREIGN KEY(bankcode) references Bank(bankcode);
      insert into Branch values('101', DEFAULT,'SBI'); insert into Branch

  values('102', DEFAULT,'FB'); insert into Branch values('103', DEFAULT,'SBT');

 insert into Branch values('104', DEFAULT,'AX'); insert into Branch values('105',

                              DEFAULT,'SIB');

 select *from Branch;
```



	BRANCHID	BRANCHNAME	BANKCODE
1	101	New Delhi	SBI
2	102	New Delhi	FB
3	104	New Delhi	AX
4	105	New Delhi	SIB
5	106	Kottayam	SBI

```
3) delete from Bank where bankcode='SBT'; delete from Branch where bankcode='SBT'; insert into
Branch values('106','Kottayam','SBI');
   select *from Branch;
```

4) ALTER TABLE Branch

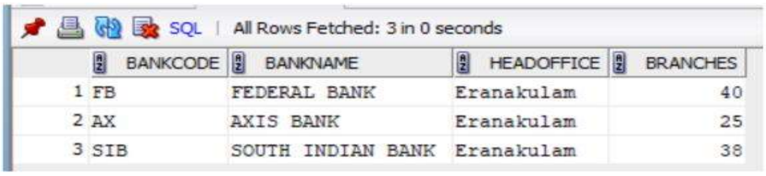
DROP PRIMARY KEY;

5) CREATE VIEW bank_headoffice AS

SELECT bankcode,bankname,headoffice,branches FROM Bank

WHERE headoffice='Eranakulam';

select *from Bank_headoffice;



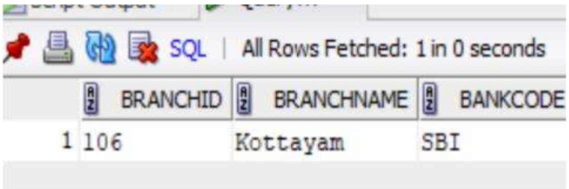
All Rows Fetched: 3 in 0 seconds

	BANKCODE	BANKNAME	HEADOFFICE	BRANCHES
1	FB	FEDERAL BANK	Eranakulam	40
2	AX	AXIS BANK	Eranakulam	25
3	SIB	SOUTH INDIAN BANK	Eranakulam	38

6) CREATE VIEW bank_branch AS SELECT

branchid,branchname,bankcode FROM Branch

WHERE branchname='Kottayam'; select *from bank_branch;



All Rows Fetched: 1 in 0 seconds

	BRANCHID	BRANCHNAME	BANKCODE
1	106	Kottayam	SBI

RESULT:
Creation of views and assertions have been done and the output has been verified and CO2 is attained.

EXPERIMENT- 13

IMPLEMENTATION OF CONTROL STATEMENTS USING PL/SQL

WITH EXCEPTION HANDLING

AIM:

Implementation of Control Statements using PL/SQL with Exception handling.

THEORY:

PL/SQL is basically a procedural language, which provides the functionality of decision making, iteration and many more features of procedural programming languages. PL/SQL can execute a number of queries in one block using single command

Syntax:

```
DECLARE

<declarations section>

BEGIN

<executable command(s)>

EXCEPTION

<exception handling>

END;
```

Questions:

1. PL/SQL program to find factorials of a number.
2. PL/SQL program to find the greatest of three numbers.
3. PL/SQL program to implement a calculator.
4. PL/SQL program to generate fibonacci series.
5. PL/SQL program to show divide by zero exception.
6. PL/SQL program to show no data found exception

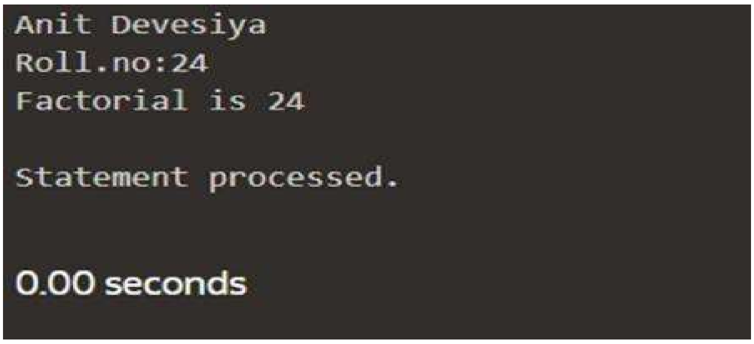
Queries and Output:

a) FACTORIAL

PROGRAM:

```
declare
f number:=1;
a number;
i number:=1;
begin
a:=:a;
if(a=0)
then
dbms_output.put_line('Factorial is: 1');
else
while(i<=a)
loop
f:=f*i;
i:=i+1;
end loop;
dbms_output.put_line('Anit Devesiya ');
dbms_output.put_line('Roll.no:24');
dbms_output.put_line('Factorial is '||f);
end if;
end;
```

OUTPUT:



```
Anit Devesiya
Roll.no:24
Factorial is 24

Statement processed.


0.00 seconds
```

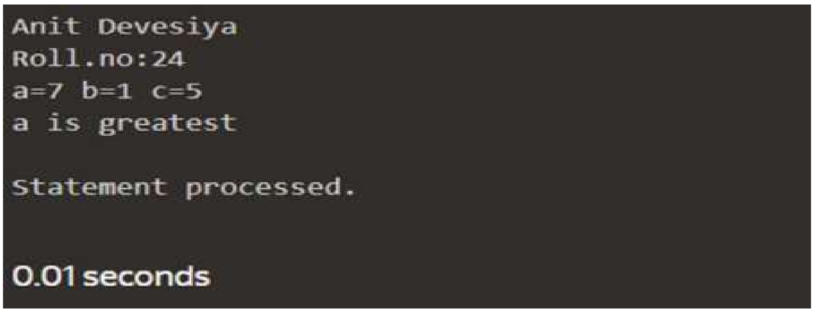
b) TO FIND THE GREATEST OF THREE NUMBERS

PROGRAM:

```
declare
a number:=7;
b number:=1;
c number:=5;
begin
```

```
dbms_output.put_line('Anit Devesiya ');
dbms_output.put_line('Roll.no:24');
dbms_output.put_line('a='||a||' b='||b||' c='||c);
if a>b AND a>c
then
dbms_output.put_line('a is greatest');
else
if b>a AND b>c
then
dbms_output.put_line('b is greatest');
else
dbms_output.put_line('c is greatest');
end if;
end if;
end;
```

OUTPUT:



```
Anit Devesiya
Roll.no:24
a=7 b=1 c=5
a is greatest

Statement processed.

0.01 seconds
```

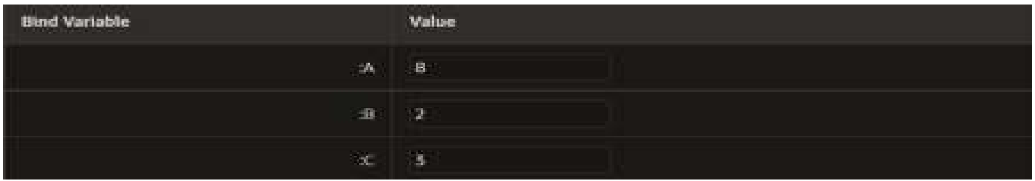
c) TO IMPLEMENT A CALCULATOR

PROGRAM:

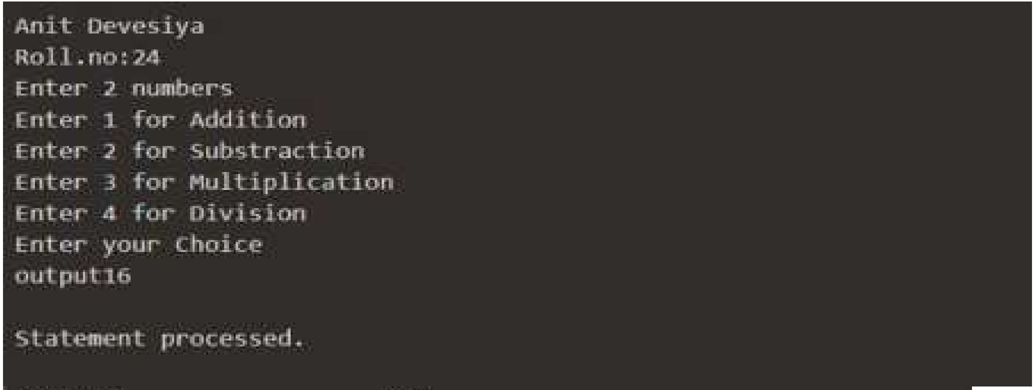
```
declare
a number(3);
b number(3);
c number(3);
d number(3);
begin
dbms_output.put_line('Anit Devesiya ');
dbms_output.put_line('Roll.no:24');
dbms_output.put_line('Enter 2 numbers');
 a:=:a;
b:=:b;
dbms_output.put_line('Enter 1 for Addition');
 dbms_output.put_line('Enter 2 for Substraction');
dbms_output.put_line('Enter 3 for Multiplication');
dbms_output.put_line('Enter 4 for Division');
dbms_output.put_line('Enter your Choice');
```

```
 c:=:c;
case c
WHEN 1 THEN d:=:a+b ;
WHEN 2 THEN d:=:a-b ;
WHEN 3 THEN d:=:a*b ;
WHEN 4 THEN d:=:a/b ;
end case;
dbms_output.put_line('output'||d);
end;
```

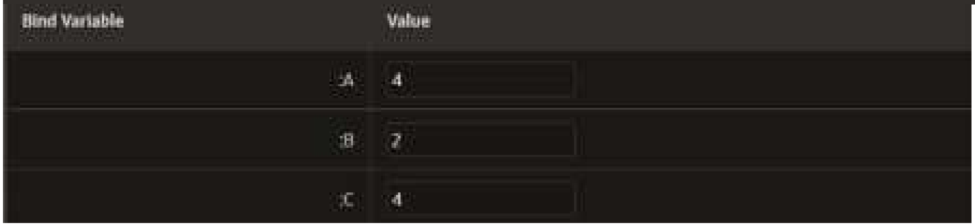
OUTPUT:



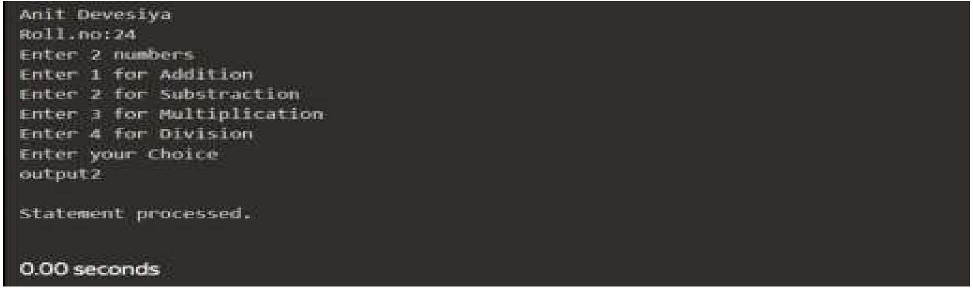
Bind Variable	Value
:A	8
:B	2
:C	3



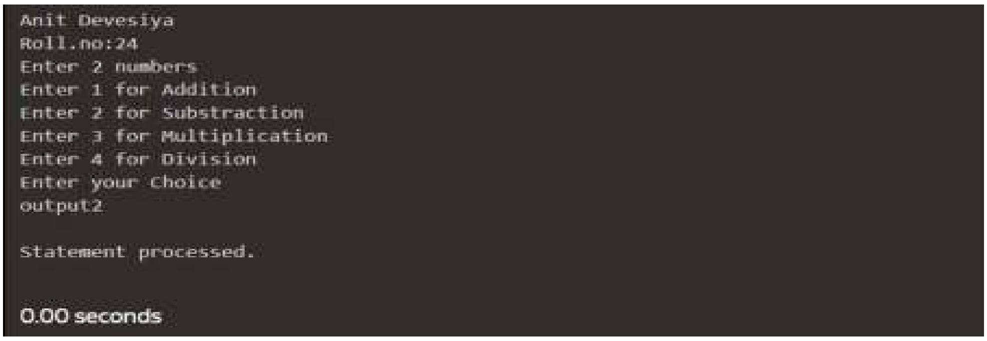
```
Anit Devesiya
Roll.no:24
Enter 2 numbers
Enter 1 for Addition
Enter 2 for Substraction
Enter 3 for Multiplication
Enter 4 for Division
Enter your Choice
output16

Statement processed.
```



Bind Variable	Value
:A	4
:B	2
:C	4



```
Anit Devesiya
Roll.no:24
Enter 2 numbers
Enter 1 for Addition
Enter 2 for Substraction
Enter 3 for Multiplication
Enter 4 for Division
Enter your Choice
output2

Statement processed.

0.00 seconds
```



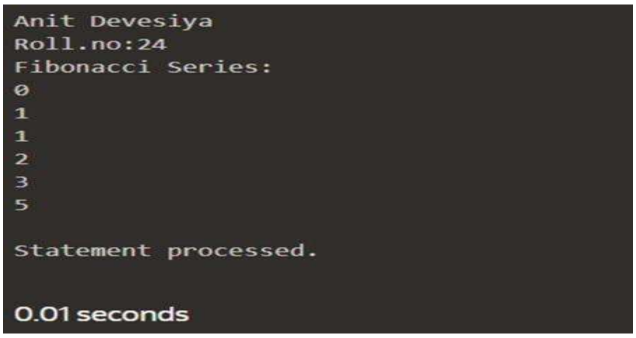
```
Anit Devesiya
Roll.no:24
Enter 2 numbers
Enter 1 for Addition
Enter 2 for Substraction
Enter 3 for Multiplication
Enter 4 for Division
Enter your Choice
output2

Statement processed.

0.00 seconds
```

d) TO GENERATE FIBONACCI SERIES

PROGRAM:

```
declare
first number := 0;
second number := 1;
temp number;
n number := 5;
i number;
begin
dbms_output.put_line('Anit Devesiya ');
 dbms_output.put_line('Roll.no:24');
dbms_output.put_line('Fibonacci Series:');
 dbms_output.put_line(first);
dbms_output.put_line(second);
for i in 2..n
loop
temp:=first+second;
first := second;
second := temp;
dbms_output.put_line(temp);
end loop;
end;
```

OUTPUT:

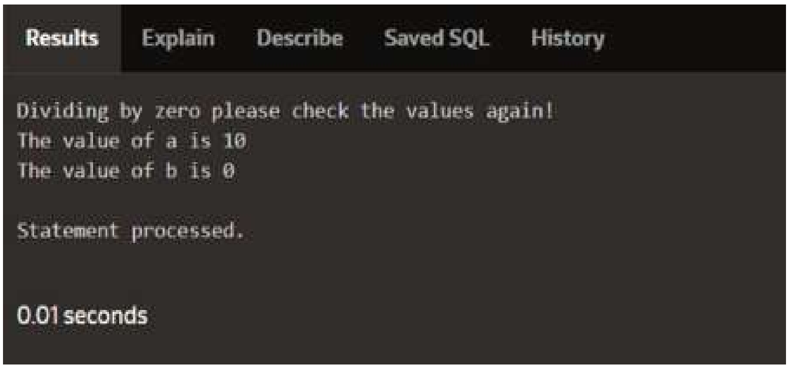


```
Anit Devesiya
Roll.no:24
Fibonacci Series:
0
1
1
2
3
5

Statement processed.

0.01 seconds
```

e) To show if a number is divided by zero

```
DECLARE
a int := 10;
b int := 0;
answer int;
BEGIN
answer:=a/b;
dbms_output.put_line('The Result after Division is '||answer);
exception
WHEN zero_divide THEN
dbms_output.put_line('Dividing by zero please check the values again!');
dbms_output.put_line('The value of a is '||a);
dbms_output.put_line('The value of b is '||b);
END;
```

OUTPUT:



```
Dividing by zero please check the values again!
The value of a is 10
The value of b is 0

Statement processed.

0.01 seconds
```

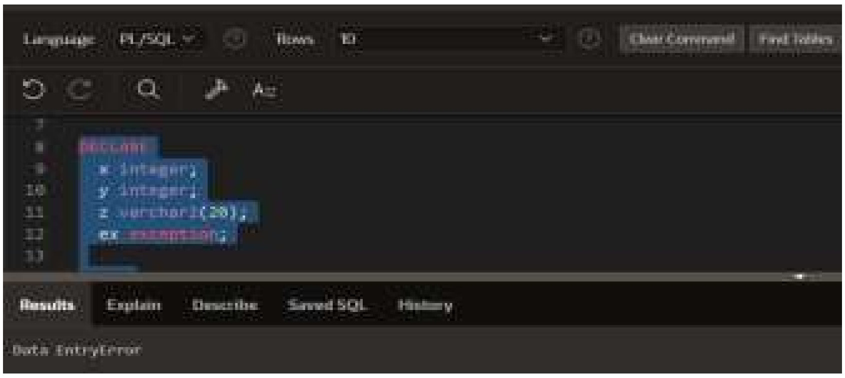
f) To show no data found exception.

PROGRAM

```
set serveroutput on;
CREATE TABLE ebill(
cname varchar(20),
prevreading varchar(20),
currreading varchar(20)
);
DECLARE
x integer;
y integer;
z varchar2(20);
ex exception;
BEGIN
x := :prevreading;
y := :currreading;
z := :cname;
```

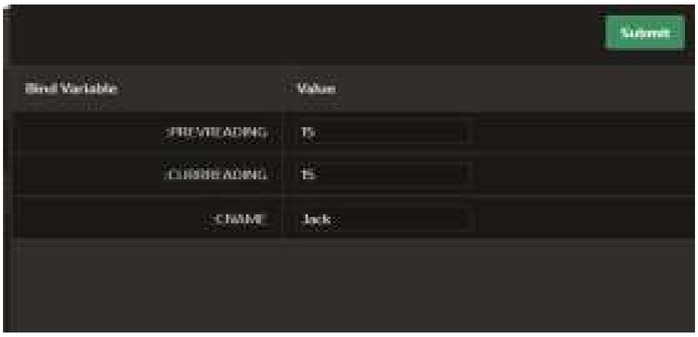
```
if(x = y) then
raise ex;
else
INSERT INTO ebill VALUES(z,y,x);
end if;
EXCEPTION
WHEN ex then
dbms_output.put_line('Data EntryError');
END;
```

OUTPUT:



```
DECLARE
  x integer;
  y integer;
  z varchar2(20);
  ex exception;
```

```
Data EntryError
```



Bind Variable	Value
:PREVREADING	15
:CURRREADING	15
:CNAME	Jack

RESULT:

Successfully implemented control structure using PL/SQL with exception handling and CO4 is attained.