AIM : Write SQL queries to CREATE TABLES for various databases using DDL commands (i.e.CREATE, ALTER, DROP, TRUNCATE).

CREATE TABLE: Creates a table with specified constraints

SYNTAX: CREATE TABLE tablename (column1 data_ type [constraint] [

, column2 data_ type [constraint]] [,

PRIMARY KEY (column1 [, column2])] [

, FOREIGN KEY (column1 [, column2]) REFERENCES tablename] [,CONSTRAINT

constraint])

```
C##554>SPOOL EXP_1.TXT
C##554>CREATE TABLE college(
  2  college_name VARCHAR(5),
  3  CLG_ID VARCHAR(5),
  4  place VARCHAR(5),
  5  std_strength NUMBER,
  6  total_branches NUMBER,
  7  PRIMARY KEY(clg_id)
  8  )
  9  /

Table created.
```

```
C##554>DESC college;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COLLEGE_NAME                                       VARCHAR2(5)
 CLG_ID                                    NOT NULL VARCHAR2(5)
 PLACE                                              VARCHAR2(5)
 STD_STRENGTH                                       NUMBER
 TOTAL_BRANCHES                                     NUMBER
```

ALTER TABLE : Used to add or modify table details like column names and data types, column constraints

```
C##554>ALTER TABLE college
  2  ADD clg_fee NUMBER NOT NULL;

Table altered.

C##554>DESC college
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COLLEGE_NAME                                       VARCHAR2(5)
 CLG_ID                                    NOT NULL VARCHAR2(5)
 PLACE                                              VARCHAR2(5)
 STD_STRENGTH                                       NUMBER
 TOTAL_BRANCHES                                     NUMBER
 CLG_FEE                                   NOT NULL NUMBER
```

```
C##554>ALTER TABLE college
  2  DROP COLUMN total_branches;

Table altered.

C##554>DESC college;
 Name                                    Null?    Type
 ----------------------------------------- -------- ----------------------------
 COLLEGE_NAME                                       VARCHAR2(5)
 CLG_ID                                  NOT NULL VARCHAR2(5)
 PLACE                                              VARCHAR2(5)
 STD_STRENGTH                                       NUMBER
 CLG_FEE                                 NOT NULL NUMBER
```

DROP TABLE: Deletes the specified table.

SYNTAX: DROP TABLE table_name;

```
C##554>CREATE TABLE product(
  2  p_name VARCHAR(10) NOT NULL,
  3  p_id NUMBER NOT NULL,
  4  PRIMARY KEY(p_id)
  5  );

Table created.
```

```
C##554>DROP TABLE product;

Table dropped.

C##554>DESC product;
ERROR:
ORA-04043: object product does not exist
```

```
C##554>ALTER TABLE college
  2  ADD clgs_fee NUMBER NOT NULL;

Table altered.

C##554>DESC college;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COLLEGE_NAME                                       VARCHAR2(5)
 CLG_ID                                    NOT NULL VARCHAR2(5)
 PLACE                                              VARCHAR2(5)
 STD_STRENGTH                                       NUMBER
 CLG_FEE                                   NOT NULL NUMBER
 CLGS_FEE                                  NOT NULL NUMBER
```

RENAME TABLE: To rename table_name, column_name

SYNTAXES: RENAME new_table_name TO old_table_name

```
C##554>RENAME college to data;

Table renamed.

C##554>desc data;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COLLEGE_NAME                                       VARCHAR2(5)
 CLG_ID                                    NOT NULL VARCHAR2(5)
 PLACE                                              VARCHAR2(5)
 STD_STRENGTH                                       NUMBER
 CLG_FEE                                   NOT NULL NUMBER
 CLGS_FEE                                  NOT NULL NUMBER
```

TRUNCATE TABLE: To remove all rows in a specified table.

SYNTAX: TRUNCATE TABLE table_name

```
C##554>TRUNCATE TABLE data;

Table truncated.
```

EXPERIMENT-3

Aim: To implement a view level design using CREATE VIEW,ALTER VIEW and DELETE VIEW ddl commands.

```
C##554>create table studentss(
  2  name varchar(10),
  3  roll_no NUMBER,
  4  sec VARCHAR(5),
  5  Branch VARCHAR(10),
  6  id_no NUMBER,
  7  PRIMARY KEY(ID_NO)
  8  );

Table created.
```

By using insert command we can insert values in a tables

```
C##554>INSERT INTO studentss VALUES('Tauheed',547,'A','CSE',1);

1 row created.

C##554>INSERT INTO studentss VALUES('Rehan',554,'A','CSE',2);

1 row created.
```

```
C##554>INSERT INTO studentss VALUES('Navya',555,'A','CSE',3);

1 row created.

C##554>INSERT INTO studentss VALUES('kavya',565,'A','CSD',4);

1 row created.

C##554>INSERT INTO studentss VALUES('keerthi',665,'A','CSM',5);

1 row created.
```

Creating view councellor:

```
C##554>create view counsellor as select name,roll_no,id_no from studentss;

View created.
```

Inserting values into councellor:

```
C##554>INSERT INTO counsellor VALUES('sasi',543,6);

1 row created.

C##554>INSERT INTO counsellor VALUES('jaggu',544,7);

1 row created.

C##554>INSERT INTO counsellor VALUES('neha',559,8);

1 row created.
```

```
C##554>select * from counsellor;

NAME          ROLL_NO      ID_NO
---------- ---------- ----------
Tauheed           547          1
Rehan             554          2
Navya             555          3
kavya             565          4
keerthi           665          5
sasi              543          6
jaggu             544          7
neha              559          8

8 rows selected.
```

Selecting specific row :

```
C##554>select * from counsellor where id_no=4;

NAME          ROLL_NO      ID_NO
---------- ---------- ----------
kavya             565          4
```

Update :

```
C##554>update counsellor set name ='jagan' where id_no=2;

1 row updated.

C##554>select * from counsellor;

NAME          ROLL_NO      ID_NO
---------- ---------- ----------
Tauheed           547          1
jagan             554          2
Navya             555          3
kavya             565          4
keerthi           665          5
sasi              543          6
jaggu             544          7
neha              559          8

8 rows selected.
```

truncate or drop view:

```
C##554>drop view counsellor;

View dropped.
```

EXPERIMENT-5

Aim: write SQL queries for the aggregate functions(sum,count,min,max,avg)

Creating a table:

```
C##554>CREATE TABLE student(
  2  name VARCHAR(10),
  3  age NUMBER,
  4  subject VARCHAR(15),
  5  marks NUMBER
  6  );

Table created.
```

```
C##554>INSERT INTO student VALUES('tauheed',22,'maths',30);

1 row created.

C##554>INSERT INTO student VALUES('para',32,'oop',33);

1 row created.

C##554>INSERT INTO student VALUES('jagaa',12,'dbms',20);

1 row created.

C##554>INSERT INTO student VALUES('kiran',24,'english',24);

1 row created.

C##554>INSERT INTO student VALUES('arjun',34,'SE',27);

1 row created.
```

Selecting table :

```
C##554>select * from student;

NAME              AGE SUBJECT              MARKS
---------- ---------- --------------- ----------
tauheed            22 maths                   30
para               32 oop                     33
jagaa              12 dbms                    20
kiran              24 english                 24
arjun              34 SE                      27
```

Sum();

```
C##554>select sum(marks) from student;

SUM(MARKS)
----------
       134
```

AVG:

```
AVG(MARKS)
----------
      26.8

C##554>select min(marks) from student;
```

MIN:

```
MIN(MARKS)
----------
        20

C##554>select max(marks) from student;

MAX(MARKS)
----------
        33

C##554>select count(marks) from student;

COUNT(MARKS)
------------
           5
```

EXPERIMENT-5

Aim: write SQL queries for the aggregate functions(sum,count,min,max,avg)

Creating a table:

```
C##554>CREATE TABLE student(
  2  name VARCHAR(10),
  3  age NUMBER,
  4  subject VARCHAR(15),
  5  marks NUMBER
  6  );

Table created.
```

```
C##554>INSERT INTO student VALUES('tauheed',22,'maths',30);

1 row created.

C##554>INSERT INTO student VALUES('para',32,'oop',33);

1 row created.

C##554>INSERT INTO student VALUES('jagaa',12,'dbms',20);

1 row created.

C##554>INSERT INTO student VALUES('kiran',24,'english',24);

1 row created.

C##554>INSERT INTO student VALUES('arjun',34,'SE',27);

1 row created.
```

Selecting table :

```
C##554>select * from student;

NAME              AGE SUBJECT              MARKS
---------- ---------- --------------- ----------
tauheed            22 maths                   30
para               32 oop                     33
jagaa              12 dbms                    20
kiran              24 english                 24
arjun              34 SE                      27
```

Sum();

```
C##554>select sum(marks) from student;

SUM(MARKS)
----------
       134
```

AVG:

```
AVG(MARKS)
----------
      26.8

C##554>select min(marks) from student;
```

MIN:

```
MIN(MARKS)
----------
        20

C##554>select max(marks) from student;

MAX(MARKS)
----------
        33

C##554>select count(marks) from student;

COUNT(MARKS)
------------
           5
```

EXPERIMENT-6

AIM: TO WRITE SQL QUERIES TO PERFORM SPECIAL OPERATIONS (i.e LIKE, BETWEEN, ISNULL, ISNOTNULL)

```
C##554>CREATE TABLE students_in(
  2  name varchar2(10) not null,
  3  r_no varchar(5) not null,
  4  branch varchar2(5) null,
  5  block varchar2(6) null,
  6  fee number not null,
  7  primary key(name)
  8  )
  9  /

Table created.
```

```
C##554>INSERT INTO students_in VALUES('arjun',534,'CSE','B',250000);

1 row created.

C##554>INSERT INTO students_in VALUES('arun',524,'CSE','A',280000);

1 row created.

C##554>INSERT INTO students_in VALUES('rehan',544,'CSE','B',289000);

1 row created.

C##554>INSERT INTO students_in VALUES('feroz',644,'CSE','B',289000);

1 row created.
```

```
C##554>INSERT INTO students_in VALUES('fezz',644,'','',2877000);

1 row created.
```

```
C##554>select * from students_in;

NAME       R_NO  BRANC BLOCK         FEE
---------- ----- ----- ------ ----------
arjun      534   CSE   B          250000
arun       524   CSE   A          280000
rehan      544   CSE   B          289000
feroz      644   CSE   B          289000
fezz       644                   2877000

C##554>select * from students_in where branch is null;

NAME       R_NO  BRANC BLOCK         FEE
---------- ----- ----- ------ ----------
fezz       644                   2877000

C##554>select * from students_in where branch is not null;

NAME       R_NO  BRANC BLOCK         FEE
---------- ----- ----- ------ ----------
arjun      534   CSE   B          250000
arun       524   CSE   A          280000
rehan      544   CSE   B          289000
feroz      644   CSE   B          289000
```

```
C##554>select * from students_in where fee between 200000 and 300000;

NAME       R_NO  BRANC BLOCK         FEE
---------- ----- ----- ------ ----------
arjun      534   CSE   B          250000
arun       524   CSE   A          280000
rehan      544   CSE   B          289000
feroz      644   CSE   B          289000

C##554>select * from students_in where branch like 'cse%';

no rows selected

C##554>select * from students_in where block like 'B%';

NAME       R_NO  BRANC BLOCK         FEE
---------- ----- ----- ------ ----------
arjun      534   CSE   B          250000
rehan      544   CSE   B          289000
feroz      644   CSE   B          289000

C##554>select * from students_in where block like 'A%';

NAME       R_NO  BRANC BLOCK         FEE
---------- ----- ----- ------ ----------
arun       524   CSE   A          280000

C##554>SELECT * FROM students_in where exists (select name from students_in);

NAME       R_NO  BRANC BLOCK         FEE
---------- ----- ----- ------ ----------
arjun      534   CSE   B          250000
arun       524   CSE   A          280000
rehan      544   CSE   B          289000
feroz      644   CSE   B          289000
fezz       644                   2877000
```

EXPERIMENT-7

AIM: Write SQL queries to perform JOIN OPERATIONS (i.e. CONDITIONAL JOIN, EQUI JOIN,LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN)

```
C##554>CREATE TABLE studentt(
  2  name varchar(10),
  3  roll_no number,
  4  dept varchar(10),
  5  primary key(name)
  6  )
  7  /

Table created.
```

```
C##554>INSERT INTO studentt values('tauheed',547,'cse');

1 row created.

C##554>INSERT INTO studentt values('tara',447,'cse');

1 row created.

C##554>INSERT INTO studentt values('sara',545,'cse');

1 row created.

C##554>INSERT INTO studentt values('neha',745,'cse');

1 row created.
```

```
C##554>select * from studentt;

NAME          ROLL_NO DEPT
---------- ---------- ----------
tauheed           547 cse
tara              447 cse
sara              545 cse
neha              745 cse
```

```
C##554>CREATE TABLE Library(
  2  roll_no NUMBER,
  3  book varchar(10)
  4  );

Table created.
```

```
C##554>INSERT INTO Library values(547,'dbms');

1 row created.

C##554>INSERT INTO Library values(559,'java');

1 row created.

C##554>INSERT INTO Library values(555,'maths');

1 row created.

C##554>INSERT INTO Library values(558,'se');

1 row created.
```

```
C##554>select * from library;

   ROLL_NO BOOK
---------- ----------
       547 dbms
       559 java
       555 maths
       558 se

C##554>select * from studentt join library on studentt.roll_no=library.roll_no;

NAME          ROLL_NO DEPT          ROLL_NO BOOK
---------- ---------- ---------- ---------- ----------
tauheed           547 cse               547 dbms

C##554>select * from studentt join library using (roll_no);

   ROLL_NO NAME       DEPT       BOOK
---------- ---------- ---------- ----------
       547 tauheed    cse        dbms

C##554>select * from studentt NATURAL LEFT OUTER JOIN LIBRARY;

   ROLL_NO NAME       DEPT       BOOK
---------- ---------- ---------- ----------
       547 tauheed    cse        dbms
       745 neha       cse
       545 sara       cse
       447 tara       cse
```

```
C##554>select * from studentt NATURAL RIGHT OUTER JOIN LIBRARY;

   ROLL_NO NAME       DEPT       BOOK
---------- ---------- ---------- ----------
       547 tauheed    cse        dbms
       559                       java
       558                       se
       555                       maths

C##554>select * from studentt NATURAL FULL OUTER JOIN LIBRARY;;
select * from studentt NATURAL FULL OUTER JOIN LIBRARY;
                                                       *
ERROR at line 1:
ORA-00933: SQL command not properly ended


C##554>select * from studentt NATURAL FULL OUTER JOIN LIBRARY;

   ROLL_NO NAME       DEPT       BOOK
---------- ---------- ---------- ----------
       547 tauheed    cse        dbms
       559                       java
       555                       maths
       558                       se
       745 neha       cse
       545 sara       cse
       447 tara       cse

7 rows selected.
```

EXPERIMENT-8

AIM : Write SQL queries to perform ORACLE BUILT-IN FUNCTIONS (i.e. DATE, TIME).

Built-in Functions 1. Character Functions I. Case-conversion functions II. Character manipulation functions

2. Number Functions

3. DATE functions

CREATING TABLE :

```
C##554>CREATE TABLE names(
  2  first_name VARCHAR(20) NOT NULL,
  3  last_name VARCHAR(20) NOT NULL
  4  );

Table created.
```

```
C##554>INSERT ALL
  2  INTO names values('tauheed','steeve')
  3  INTO names values('neha','angel')
  4  INTO names values('rehu','swag')
  5  INTO names values('navya','white')
  6  select * from dual;

4 rows created.

C##554>select lower(first_name) from names;

LOWER(FIRST_NAME)
--------------------
tauheed
neha
rehu
navya

C##554>select upper(first_name) from names;

UPPER(FIRST_NAME)
--------------------
TAUHEED
NEHA
REHU
NAVYA
```

```
C##554>select initcap(first_name) from names;

INITCAP(FIRST_NAME)
--------------------
Tauheed
Neha
Rehu
Navya

C##554>select CONCAT(first_name,last_name) from names;

CONCAT(FIRST_NAME,LAST_NAME)
----------------------------------------
tauheedsteeve
nehaangel
rehuswag
navyawhite

C##554>select substr(first_name,1,4) from names;

SUBSTR(FIRST_NAM
----------------
tauh
neha
rehu
navy

C##554>select length(first_name) from names;

LENGTH(FIRST_NAME)
------------------
                 7
                 4
                 4
                 5
```

```
C##554>select instr(first_name,'ta') from names;

INSTR(FIRST_NAME,'TA')
----------------------
                     1
                     0
                     0
                     0

C##554>select trim('A' from first_name) from names;

TRIM('A'FROMFIRST_NA
--------------------
tauheed
neha
rehu
navya
```

EXPERIMENT-9

AIM : Write SQL queries to perform KEY CONSTRAINTS (i.e. PRIMARY KEY, FOREIGN KEY,UNIQUE NOT NULL, CHECK, DEFAULT).

Types of SQL Constraints. 1. NOT NULL - Ensures that a column cannot have a NULL value

2. UNIQUE - Ensures that all values in a column are different

3. PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely I identifies each row in a table

4. FOREIGN KEY - Uniquely identifies a row/record in another table

5. CHECK - Ensures that all values in a column satisfies a specific condition

6. DEFAULT - Sets a default value for a column when no value is specified

.NOT NULL Constraint Example:

```
C##554>CREATE TABLE order1(
  2  id NUMBER PRIMARY KEY,
  3  product_name varchar2(50) not null,
  4  quantity number
  5  );

Table created.
```

```
C##554>insert into order1 values(1,'agarbathi',30);

1 row created.

C##554>insert into order1 values(4,'',30);
insert into order1 values(4,'',30)
                          *
ERROR at line 1:
ORA-01400: cannot insert NULL into ("C##554"."ORDER1"."PRODUCT_NAME")


C##554>create table emploueess(
  2  /
create table emploueess(
                       *
ERROR at line 1:
ORA-00904: : invalid identifier
```

```
C##554>create table employeess(
  2  id number primary key,
  3  name varchar(50) not null,
  4  e_mail varchar2(50) unique
  5  )
  6  /

Table created.

C##554>insert into employees values(547,'tau','shaik@gmail.com');
insert into employees values(547,'tau','shaik@gmail.com')
            *
ERROR at line 1:
ORA-00913: too many values
```

```
C##554>CREATE TABLE stud(
  2  id number primary key,
  3  first_name varchar(20) not null,
  4  last_name varchar(20) not null
  5  );

Table created.

C##554>insert into stud values(547,'harry','potter');

1 row created.

C##554>CREATE TABLE ondons(
```

```
C##554>CREATE TABLE orders(
  2  id number primary key,
  3  order_num number not null,
  4  stud_id number references stud(id)
  5  );

Table created.

C##554>insert into orders values(11,2,111);
insert into orders values(11,2,111)
*
ERROR at line 1:
ORA-02291: integrity constraint (C##554.SYS_C008368) violated - parent key not
found


C##554>CREATE TABLE parts1(
  2  part_id number primary key,
  3  part_name varchar2(50) not null,
  4  buy_price number(9,2) check(buy_price>0)
  5  );

Table created.

C##554>insert into parts1 values(1,'agarbatti',879);

1 row created.
```

AIM: To write a PL/SQL program for calculating the factorial of a given number.

Source code

```
C##554>SET SERVEROUT ON
C##554>SET VERIFY OFF
C##554>DECLARE
  2  fact number:=1;
  3  n number;
  4  BEGIN
  5  n:=&n;
  6  WHILE n>0 LOOP
  7  fact:=n*fact;
  8  n:=n-1;
  9  END LOOP;
 10  DBMS_OUTPUT.PUT_LINE(fact);
 11  END;
 12  /
Enter value for n: 6
720

PL/SQL procedure successfully completed.
```

Conclusion : The pl/sql program is successfully executed.

AIM: Write a PL/SQL program for finding the given number is prime number or not.

SOURCE CODE:

```
C##554>SET SERVEROUT ON
C##554>SET VERIFY OFF
C##554>DECLARE
  2  n number;
  3  i number;
  4  temp number;
  5  BEGIN
  6  n:=&n;
  7  i:=2;
  8  temp:=1;
  9  for i in 2..n/2
 10  loop
 11  if mod(n,i)=0
 12  then
 13  temp:=0;
 14  exit;
 15  end if;
 16  end loop;
 17  if temp=1
 18  then
 19  DBMS_OUTPUT.PUT_LINE(n||' is a prime number');
 20  else
 21  DBMS_OUTPUT.PUT_LINE(n||' is not a prime number');
 22  end if;
 23  end;
 24  /
Enter value for n: 78
78 is not a prime number

PL/SQL procedure successfully completed.
```

CONCLUSION: The pl/sql program is successfully executed

AIM: Write a PL/SQL program for displaying the Fibonacci series up to an integer.

SOURCE CODE:

```
C##554>SET SERVEROUT ON
C##554>SET VERIFY OFF
C##554>DECLARE
  2  FIRST NUMBER:=0;
  3  SECOND NUMBER:=1;
  4  N NUMBER;
  5  TEMP NUMBER;
  6  I NUMBER;
  7  BEGIN
  8  N:=&N;
  9  DBMS_OUTPUT.PUT_LINE('SERIES: ');
 10  DBMS_OUTPUT.PUT_LINE(FIRST);
 11  DBMS_OUTPUT.PUT_LINE(SECOND);
 12  FOR I IN 2..N
 13  LOOP
 14  TEMP:=FIRST+SECOND;
 15  FIRST:=SECOND;
 16  SECOND:=TEMP;
 17  DBMS_OUTPUT.PUT_LINE(TEMP);
 18  END LOOP;
 19  END;
 20  /
Enter value for n: 8
SERIES:
0
1
1
2
3
5
8
13
21

PL/SQL procedure successfully completed.
```

CONCLUSION: The pl/sql program is successfully executed

Write PL/SQL program to implement Stored Procedure on table.

AIM: Write PL/SQL program to implement Stored Procedure on table.

PL/SQL Procedure: The PL/SQL stored procedure or simply a procedure is a PL/SQL block which performs one or more specific tasks. It is just like procedures in other programming languages. The procedure contains a header and a body.

EXAMPLE :

```
C##554>CREATE TABLE SAILOR(ID NUMBER(10) PRIMARY KEY,NAME VARCHAR2(100)

Table created.

C##554>CREATE OR REPLACE PROCEDURE INSERTUSER
  2  (ID IN NUMBER,
  3  NAME IN VARCHAR2)
  4  IS
  5  BEGIN
  6  INSERT INTO SAILOR VALUES(ID,NAME);
  7  DBMS_OUTPUT.PUT_LINE('RECORD INSERTED SUCCESSFULLY');
  8  END;
  9  /

Procedure created.
```

```
C##554>DECLARE
  2  CNT NUMBER;
  3  BEGIN
  4  INSERTUSER(101,'NARASIMHA');
  5  SELECT COUNT(*) INTO CNT FROM SAILOR;
  6  DBMS_OUTPUT.PUT_LINE(CNT||' RECORD IS INSERTED SUCCESSFULLY');
  7  END;
  8  /
RECORD INSERTED SUCCESSFULLY
1 RECORD IS INSERTED SUCCESSFULLY

PL/SQL procedure successfully completed.
```

```
C##554>DROP PROCEDURE insertuser;

Procedure dropped.

C##554>
```

AIM: TO Write PL/SQL program to implement Stored Function on table.

PL/SQL Function: The PL/SQL Function is very similar to PL/SQL Procedure. The main difference between procedure and a function is, a function must always return a value, and on the other hand a procedure may or may not return a value. Except this, all the other things of PL/SQL procedure are true for PL/SQL function to

```
##554>DECLARE
  2  N3 NUMBER(2);
  3  BEGIN
  4  N3:=ADDER(11,22);
  5  DBMS_OUTPUT.PUT_LINE('ADDITION IS: '||N3);
  6  END;
  7  /
DDITION IS: 33

L/SQL procedure successfully completed.

##554>CREATE FUNCTION fact(x number)
  2  RETURN number
  3  IS
  4  f number;
  5  BEGIN
  6  IF x=0 THEN
  7  f:=1;
  8  ELSE
  9  f:=x+fact(x-1);
 10  END IF;
 11  RETURN ;
 12  /

arning: Function created with compilation errors.
```

```
C##554>CREATE FUNCTION facts(x number)
  2  RETURN number
  3  IS
  4  f number;
  5  BEGIN
  6  IF x=0 THEN
  7  f:=1;
  8  ELSE
  9  f:=x+facts(x-1);
 10  END IF;
 11  RETURN f;
 12  END;
 13  /

Function created.

C##554>DECLARE
  2  num NUMBER;
  3  factorial number;
  4  BEGIN
  5  num:=6;
  6  factorial:=facts(num);
  7  DBMS_OUTPUT.PUT_LINE('factorial '|| num || ' is '  || factorial);
  8  END;
  9  /
factorial 6 is 22

PL/SQL procedure successfully completed.
```