## 1. Write SQL queries to CREATE TABLES for various databases using DDL COMMANDS (i.e., CREATE, ALTER, DROP, TRUNCATE).

**AIM:** To design databases using DDL commands like CREATE, ALTER, DROP and TRUNCATE.

**CREATING A TABLE:**

```
C:\Users\mahij>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Sun Jan 14 16:55:14 2024
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Enter user-name: system
Enter password:
Last Successful login time: Sun Jan 14 2024 16:52:43 +05:30

Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> SET SQLP "CSE-B-5A2'@'_CONNECT_IDENTIFIER' '_DATE> "
CSE-B-5A2@XE 14-JAN-24>
```

A table contains six constraints:

primary key constraints.

foreign key constraints.

Not null constraints.

check constraints.

Unique constraints.

Ref constraints.

**Primary key and not null constraints:**

```
CSE-B-5A2@XE 14-JAN-24> CREATE TABLE persons(
  2  per_id NUMBER,
  3  first_name VARCHAR2(50) NOT NULL,
  4  last_name VARCHAR2(50) NOT NULL,
  5  PRIMARY KEY(per_id)
  6  );

Table created.
```

**Foreign key:**

```
CSE-B-5A2@XE 14-JAN-24> run
  1  CREATE TABLE studentss(
  2  stu_id int PRIMARY KEY,
  3  f_name VARCHAR2(250) NOT NULL,
  4  l_name VARCHAR2(250),
  5  age int
  6* )

Table created.
```

```
CSE-B-5A2@XE 14-JAN-24> run
  1  CREATE TABLE orderss(
  2  or_id int PRIMARY KEY,
  3  or_num int NOT NULL,
  4  stu_id int REFERENCES studentss(stu_id)
  5* )

Table created.
```

**Unique constraints:**

```
CSE-B-5A2@XE 14-JAN-24> CREATE TABLE clientss(
  2  cli_id NUMBER,
  3  fr_name VARCHAR2(50) NOT NULL,
  4  lt_name VARCHAR2(50) NOT NULL,
  5  email VARCHAR2(255) NOT NULL UNIQUE,
  6  phone VARCHAR2(50)
  7  );

Table created.
```

**Check constraints:**

```
CSE-B-5A2@XE 14-JAN-24> CREATE TABLE partss(
  2  part_id NUMBER,
  3  part_name VARCHAR2(50) NOT NULL,
  4  buy_price NUMBER(9, 2) CHECK(buy_price>0),
  5  PRIMARY KEY(part_id)
  6  );

Table created.
```

**DROPPING A TABLE:**

```
CSE-B-5A2@XE 14-JAN-24> DROP TABLE partss
  2  ;

Table dropped.

CSE-B-5A2@XE 14-JAN-24> DROP TABLE persons;

Table dropped.
```

**ALTERING A TABLE:**

```
CSE-B-5A2@XE 14-JAN-24>  CREATE TABLE persons(
  2  per_id NUMBER,
  3  first_name VARCHAR2(50) NOT NULL,
  4  last_name VARCHAR2(50) NOT NULL,
  5  PRIMARY KEY(per_id)
  6  );

Table created.

CSE-B-5A2@XE 14-JAN-24> ALTER TABLE persons
  2  ADD(
  3  phone VARCHAR2(50),
  4  email VARCHAR2(250)
  5  );

Table altered.
```

```
CSE-B-5A2@XE 14-JAN-24> DESC persons;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PER_ID                                    NOT NULL NUMBER
 FIRST_NAME                                NOT NULL VARCHAR2(50)
 LAST_NAME                                 NOT NULL VARCHAR2(50)
 PHONE                                              VARCHAR2(50)
 EMAIL                                              VARCHAR2(250)
```

```
CSE-B-5A2@XE 14-JAN-24> ALTER TABLE persons ADD birthdate DATE NULL;

Table altered.

CSE-B-5A2@XE 14-JAN-24> ALTER TABLE persons MODIFY birthdate DATE NOT NULL;

Table altered.

CSE-B-5A2@XE 14-JAN-24> ALTER TABLE persons MODIFY(
  2  phone VARCHAR2(50) NOT NULL,
  3  email VARCHAR2(250) NOT NULL
  4  );

Table altered.

CSE-B-5A2@XE 14-JAN-24> DESC persons;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PER_ID                                    NOT NULL NUMBER
 FIRST_NAME                                NOT NULL VARCHAR2(50)
 LAST_NAME                                 NOT NULL VARCHAR2(50)
 PHONE                                     NOT NULL VARCHAR2(50)
 EMAIL                                     NOT NULL VARCHAR2(250)
 BIRTHDATE                                 NOT NULL DATE
```

```
CSE-B-5A2@XE 14-JAN-24> ALTER TABLE persons
  2  RENAME COLUMN first_name TO forename;

Table altered.

CSE-B-5A2@XE 14-JAN-24> ALTER TABLE persons RENAME TO people;

Table altered.

CSE-B-5A2@XE 14-JAN-24> DESC persons;
ERROR:
ORA-04043: object persons does not exist


CSE-B-5A2@XE 14-JAN-24> DESC people;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PER_ID                                    NOT NULL NUMBER
 FORENAME                                  NOT NULL VARCHAR2(50)
 LAST_NAME                                 NOT NULL VARCHAR2(50)
```

**TRUNCATING A TABLE:**

```
CSE-B-5A2@XE 14-JAN-24> CREATE TABLE quotations(
  2  quotation_no NUMERIC,
  3  customer_id NUMERIC NOT NULL,
  4  valid_from DATE NOT NULL,
  5  valid_to DATE NOT NULL,
  6  PRIMARY KEY(quotation_no)
  7  );

Table created.

CSE-B-5A2@XE 14-JAN-24> CREATE TABLE quotation_item(
  2  quotation_no NUMERIC,
  3  item_no NUMERIC,
  4  product_id NUMERIC NOT NULL,
  5  qty NUMERIC NOT NULL,
  6  price NUMERIC(9, 2) NOT NULL
  7  );

Table created.
```

```
CSE-B-5A2@XE 14-JAN-24> TRUNCATE TABLE quotations CASCADE;

Table truncated.
```

**RENAMING A TABLE:**

```
CSE-B-5A2@XE 14-JAN-24> RENAME quotations to lines;

Table renamed.
```

## 2. Write SQL queries to MANIPULATE TABLES for various databases using DML commands (i.e., INSERT, SELECT, UPDATE, DELETE).

**AIM:** To MANIPULATE and QUERY databases using DML commands like INSERT, SELECT, UPDATE and DELETE.

**INSERTING VALUES INTO A TABLE:**

```
CSE-B-5A2@XE 22-SEP-23> CREATE TABLE ORDER_list(
  2  cid NUMBER PRIMARY KEY,
  3  oid NUMBER,
  4  ono NUMBER
  5  );

Table created.

CSE-B-5A2@XE 22-SEP-23> INSERT INTO order_list VALUES(1,101,501);

1 row created.

CSE-B-5A2@XE 22-SEP-23> INSERT INTO order_list VALUES(2,201,601);

1 row created.

CSE-B-5A2@XE 22-SEP-23> INSERT INTO order_list VALUES(3,301,701);

1 row created.

CSE-B-5A2@XE 22-SEP-23> INSERT INTO order_list VALUES(4,401,801);

1 row created.
```

**SELECTING FROM TABLE:**

```
CSE-B-5A2@XE 22-SEP-23> SELECT * FROM order_list;

       CID        OID        ONO
---------- ---------- ----------
         1        101        501
         2        201        601
         3        301        701
         4        401        801
```

```
CSE-B-5A2@XE 22-SEP-23> CREATE TABLE customerss(
  2  c_id NUMBER PRIMARY KEY,
  3  o_id NUMBER,
  4  o_no NUMBER
  5  );

Table created.

CSE-B-5A2@XE 22-SEP-23> INSERT INTO customerss VALUES(5,20,30);

1 row created.

CSE-B-5A2@XE 22-SEP-23> INSERT INTO customerss VALUES(8,21,31);

1 row created.

CSE-B-5A2@XE 22-SEP-23> SELECT * FROM customerss;

      C_ID       O_ID       O_NO
---------- ---------- ----------
         5         20         30
         8         21         31
```

```
CSE-B-5A2@XE 22-SEP-23> TRUNCATE TABLE order_list;

Table truncated.

CSE-B-5A2@XE 22-SEP-23> SELECT * FROM order_list;

no rows selected

CSE-B-5A2@XE 22-SEP-23> DESC order_list;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 CID                                       NOT NULL NUMBER
 OID                                                NUMBER
 ONO                                                NUMBER
```

```
CSE-B-5A2@XE 22-SEP-23> INSERT INTO order_list(cid, ono)
  2  SELECT c_id,o_no FROM customerss;

2 rows created.

CSE-B-5A2@XE 22-SEP-23> SELECT * FROM order_list;

       CID        OID        ONO
---------- ---------- ----------
         5                    30
         8                    31
```

## UPDATING A TABLE:

```
CSE-B-5A2@XE 22-SEP-23> UPDATE order_list
  2  SET cid=9
  3  WHERE ono=30;

1 row updated.

CSE-B-5A2@XE 22-SEP-23> SELECT * FROM order_list;

       CID        OID        ONO
---------- ---------- ----------
         9                    30
         8                    31
```

**DELETING A TABLE:**

```
CSE-B-5A2@XE 22-SEP-23> DELETE
  2  FROM order_list
  3  WHERE ono=31;

1 row deleted.

CSE-B-5A2@XE 22-SEP-23> SELECT * FROM order_list;

       CID        OID        ONO
---------- ---------- ----------
         9                    30
```

## 3. Create various SQL view queries of various databases (CREATE VIEW, ALTER VIEW, DELETE VIEW).

**Aim:** To create various SQL view queries of various databases like CREATE VIEW, ALTER VIEW, DELETE VIEW.

**CREATING TABLE:**

```
CSE-B-5A2@XE 12-DEC-23> CREATE TABLE scholar(
  2  sch_id NUMBER NOT NULL,
  3  sch_name VARCHAR2(50) NOT NULL,
  4  branch VARCHAR2(20) NOT NULL,
  5  building VARCHAR2(20) NOT NULL,
  6  PRIMARY KEY(sch_id)
  7  );

Table created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO scholar VALUES(571, 'armaan','cse', 'b-block');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO scholar VALUES(572, 'abhira','csm', 'a-block');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO scholar VALUES(573, 'roohi','csd', 'b-block');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO scholar VALUES(574, 'rohith','eee', 'c-block');

1 row created.
```

```
CSE-B-5A2@XE 12-DEC-23> SELECT * FROM scholar;

    SCH_ID SCH_NAME
---------- --------------------------------------------------
BRANCH               BUILDING
-------------------- --------------------
       571 armaan
cse                  b-block

       572 abhira
csm                  a-block

       573 roohi
csd                  b-block


    SCH_ID SCH_NAME
---------- --------------------------------------------------
BRANCH               BUILDING
-------------------- --------------------
       574 rohith
eee                  c-block
```

## CREATE VIEW:

```
CSE-B-5A2@XE 12-DEC-23> CREATE VIEW professor AS
  2  SELECT sch_id, sch_name, branch, building FROM scholar;

View created.
```

```
CSE-B-5A2@XE 12-DEC-23> INSERT INTO professor VALUES(575,'dev','civil','c-block');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO professor VALUES(576,'sonakshi','cse','b-block');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO professor VALUES(577,'akshara','csm','a-block');

1 row created.
```

```
CSE-B-5A2@XE 12-DEC-23> SELECT * FROM professor;

    SCH_ID SCH_NAME
---------- --------------------------------------------------
BRANCH               BUILDING
-------------------- --------------------
       571 armaan
cse                  b-block

       572 abhira
csm                  a-block

       573 roohi
csd                  b-block


    SCH_ID SCH_NAME
---------- --------------------------------------------------
BRANCH               BUILDING
-------------------- --------------------
       574 rohith
eee                  c-block

       575 dev
civil                c-block

       576 sonakshi
cse                  b-block


    SCH_ID SCH_NAME
---------- --------------------------------------------------
BRANCH               BUILDING
-------------------- --------------------
       577 akshara
csm                  a-block


7 rows selected.
```

## UPDATE VIEW:

```
CSE-B-5A2@XE 12-DEC-23> UPDATE professor SET sch_name = 'Abhimanyu' WHERE sch_id = 577;

1 row updated.

CSE-B-5A2@XE 12-DEC-23> SELECT * FROM professor;
```

```
CSE-B-5A2@XE 12-DEC-23> SELECT * FROM professor;

    SCH_ID SCH_NAME
---------- --------------------------------------------------
BRANCH               BUILDING
-------------------- --------------------
       571 armaan
cse                  b-block

       572 abhira
csm                  a-block

       573 roohi
csd                  b-block


    SCH_ID SCH_NAME
---------- --------------------------------------------------
BRANCH               BUILDING
-------------------- --------------------
       574 rohith
eee                  c-block

       575 dev
civil                c-block

       576 sonakshi
cse                  b-block


    SCH_ID SCH_NAME
---------- --------------------------------------------------
BRANCH               BUILDING
-------------------- --------------------
       577 Abhimanyu
csm                  a-block
```

## DELETING VIEW:

```
CSE-B-5A2@XE 12-DEC-23> DROP VIEW professor;

View dropped.

CSE-B-5A2@XE 12-DEC-23>
```

## 4. Write SQL queries to perform RELATIONAL SET OPERATIONS (i.e., UNION, UNION ALL, INTERSECT, INSERSECT ALL, MINUS, CROSS JOIN, NATURAL JOIN).

**Aim:** TO write SQL queries to perform RELATIONAL SET OPERATIONS like UNION, UNION ALL, INTERSECT, INSERSECT ALL, MINUS, CROSS JOIN, NATURAL JOIN.

**CREATING TABLES:**

```
CSE-B-5A2@XE 14-NOV-23> CREATE TABLE section(
  2  course_id NUMBER PRIMARY KEY,
  3  sem VARCHAR2(20) NOT NULL,
  4  year NUMBER NOT NULL,
  5  building VARCHAR2(10) NOT NULL
  6  );

Table created.

CSE-B-5A2@XE 14-NOV-23> CREATE TABLE class(
  2  course_name VARCHAR2(20) NOT NULL,
  3  sem VARCHAR2(20) NOT NULL,
  4  course_id NUMBER REFERENCES section(course_id)
  5  );

Table created.
```

**INSERTING VALUES:**

```
CSE-B-5A2@XE 14-NOV-23> INSERT INTO section VALUES(101,'spring',2004,'a-block');

1 row created.

CSE-B-5A2@XE 14-NOV-23> INSERT INTO section VALUES(102,'fall',2005,'b-block');

1 row created.

CSE-B-5A2@XE 14-NOV-23> INSERT INTO section VALUES(103,'spring',2004,'a-block');

1 row created.

CSE-B-5A2@XE 14-NOV-23> INSERT INTO section VALUES(104,'fall',2005,'b-block');

1 row created.
```

```
CSE-B-5A2@XE 14-NOV-23> INSERT INTO class VALUES('database', 'one',104)
  2  ;

1 row created.

CSE-B-5A2@XE 14-NOV-23> INSERT INTO class VALUES('ds', 'two',103);

1 row created.

CSE-B-5A2@XE 14-NOV-23> INSERT INTO class VALUES('ps', 'three',102);

1 row created.

CSE-B-5A2@XE 14-NOV-23> INSERT INTO class VALUES('elese', 'four',101);

1 row created.

CSE-B-5A2@XE 14-NOV-23> SELECT*FROM section;

 COURSE_ID SEM                  YEAR BUILDING
---------- ------------------- ---------- ----------
       101 spring                    2004 a-block
       102 fall                      2005 b-block
       103 spring                    2004 a-block
       104 fall                      2005 b-block
```

```
CSE-B-5A2@XE 14-NOV-23> select*from class;

COURSE_NAME          SEM                   COURSE_ID
-------------------- -------------------- ----------
database             one                         104
ds                   two                         103
ps                   three                       102
elese                four                        101

CSE-B-5A2@XE 14-NOV-23>
```

**UNION OPERATION:**

```
CSE-B-5A2@XE 14-NOV-23> SELECT course_id
  2  FROM section WHERE sem='spring' AND year=2004
  3  UNION
  4  SELECT course_id
  5  FROM section WHERE sem='fall' AND year=2005;

 COURSE_ID
----------
       101
       103
       102
       104
```

**UNION ALL OPERATION:**

```
CSE-B-5A2@XE 14-NOV-23> SELECT course_id
  2  FROM section WHERE sem='spring' AND year=2004
  3  UNION ALL
  4  SELECT course_id
  5  FROM section WHERE sem='fall' AND year=2005;

 COURSE_ID
----------
       101
       103
       102
       104
```

**INTERSECT OPERATION:**

```
CSE-B-5A2@XE 14-NOV-23> run;
  1  SELECT year
  2  FROM section WHERE sem='spring' AND course_id=101
  3  INTERSECT
  4  SELECT year
  5* FROM section WHERE sem='fall' AND course_id=102

no rows selected
```

```
CSE-B-5A2@XE 07-NOV-23> SELECT year
  2  FROM section WHERE sem='spring' AND course_id=101
  3  INTERSECT
  4  SELECT year
  5  FROM section WHERE sem='spring' AND course_id=103;

      YEAR
----------
      2004
```

**INTERSECT ALL OPERATION:**

```
CSE-B-5A2@XE 07-NOV-23> run;
  1  SELECT year
  2  FROM section WHERE sem='spring' AND course_id=103
  3  INTERSECT ALL
  4  SELECT year
  5* FROM section WHERE sem='spring' AND course_id=101

      YEAR
----------
      2004
```

**MINUS OPERATION:**

```
CSE-B-5A2@XE 07-NOV-23> run;
  1  SELECT course_id
  2  FROM section WHERE sem='spring' AND year=2004
  3  MINUS
  4  SELECT course_id
  5* FROM section WHERE sem='fall' AND year=2005

 COURSE_ID
----------
       101
       103
```

**MINUS ALL OPERATION:**

```
CSE-B-5A2@XE 07-NOV-23> SELECT course_id
  2  FROM section WHERE sem='fall' AND year=2005
  3  MINUS ALL
  4  SELECT course_id
  5  FROM section WHERE sem='spring' AND year=2004;

 COURSE_ID
----------
       102
       104
```

## 5. Write SQL queries to perform SPECIAL OPERATIONS (i.e., IS NULL, BETWEEN, LIKE, IN, EXISTS).

**Aim:** To Write SQL queries to perform SPECIAL OPERATIONS like IS NULL, BETWEEN, LIKE, IN, EXISTS.

**CREATING TABLES AND INSERTING:**

```
CSE-B-5A2@XE 07-NOV-23> CREATE TABLE faculty(
  2  id VARCHAR2(20) PRIMARY KEY,
  3  name VARCHAR2(50) NOT NULL,
  4  dept_name VARCHAR2(50),
  5  salary NUMBER
  6  );

Table created.
```

```
CSE-B-5A2@XE 07-NOV-23> CREATE TABLE departments(
  2  id VARCHAR2(20),
  3  dept_name VARCHAR2(30) NOT NULL,
  4  section VARCHAR2(10)
  5  );

Table created.
```

```
CSE-B-5A2@XE 07-NOV-23> INSERT INTO faculty VALUES('101','rahu','phy',30000);

1 row created.

CSE-B-5A2@XE 07-NOV-23> INSERT INTO faculty VALUES('102','raghu','chy',35000);

1 row created.

CSE-B-5A2@XE 07-NOV-23> INSERT INTO faculty VALUES('103','rafi','eee',34000);

1 row created.

CSE-B-5A2@XE 07-NOV-23> INSERT INTO faculty VALUES('104','ramu','ece',37000);

1 row created.
```

```
CSE-B-5A2@XE 07-NOV-23> INSERT INTO faculty VALUES('105','latha','null',NULL);

1 row created.
```

```
CSE-B-5A2@XE 07-NOV-23> INSERT INTO departments VALUES('101','mech','a');

1 row created.

CSE-B-5A2@XE 07-NOV-23> INSERT INTO departments VALUES('102','civil','b');

1 row created.

CSE-B-5A2@XE 07-NOV-23> INSERT INTO departments VALUES('103','eee','b');

1 row created.

CSE-B-5A2@XE 07-NOV-23> INSERT INTO departments VALUES('104','ece','a');

1 row created.

CSE-B-5A2@XE 07-NOV-23>
```

## IS NULL / IS NOT NULL operation:

```
CSE-B-5A2@XE 07-NOV-23> SELECT * FROM faculty WHERE salary IS NULL;

ID                   NAME
-------------------- --------------------------------------------------
DEPT_NAME                                          SALARY
-------------------------------------------------- ----------
105                  latha
null


CSE-B-5A2@XE 07-NOV-23> SELECT * FROM FACULTY WHERE salary is NOT NULL;

ID                   NAME
-------------------- --------------------------------------------------
DEPT_NAME                                          SALARY
-------------------------------------------------- ----------
101                  rahu
phy                                                 30000

102                  raghu
chy                                                 35000

103                  rafi
eee                                                 34000


ID                   NAME
-------------------- --------------------------------------------------
DEPT_NAME                                          SALARY
-------------------------------------------------- ----------
104                  ramu
ece                                                 37000


CSE-B-5A2@XE 07-NOV-23>
```

## BETWEEN OPERATION:

```
CSE-B-5A2@XE 14-NOV-23> SELECT * FROM faculty WHERE salary BETWEEN 25000 AND 35000;

ID                   NAME
-------------------- --------------------------------------------------
DEPT_NAME                                          SALARY
-------------------------------------------------- ----------
101                  rahu
phy                                                 30000

102                  raghu
chy                                                 35000

103                  rafi
eee                                                 34000
```

## LIKE OPERATION:

```
CSE-B-5A2@XE 14-NOV-23> SELECT * FROM faculty WHERE name LIKE 'r%';

ID                   NAME
-------------------- --------------------------------------------------
DEPT_NAME                                          SALARY
-------------------------------------------------- ----------
101                  rahu
phy                                                 30000

102                  raghu
chy                                                 35000

103                  rafi
eee                                                 34000


ID                   NAME
-------------------- --------------------------------------------------
DEPT_NAME                                          SALARY
-------------------------------------------------- ----------
104                  ramu
ece                                                 37000


CSE-B-5A2@XE 14-NOV-23> SELECT * FROM faculty WHERE name LIKE '____';

ID                   NAME
-------------------- --------------------------------------------------
DEPT_NAME                                          SALARY
-------------------------------------------------- ----------
101                  rahu
phy                                                 30000

103                  rafi
eee                                                 34000

104                  ramu
ece                                                 37000
```

## IN OPERATION:

```
CSE-B-5A2@XE 14-NOV-23> SELECT * FROM faculty WHERE dept_name IN('eee','ece');

ID                   NAME
-------------------- --------------------------------------------------
DEPT_NAME                                          SALARY
-------------------------------------------------- ----------
103                  rafi
eee                                                 34000

104                  ramu
ece                                                 37000
```

## EXISTS OPERATION:

```
CSE-B-5A2@XE 14-NOV-23> SELECT * FROM departments WHERE dept_name EXISTS(SELECT
  2  dept_name FROM faculty
  3  );
```

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

**Aim:** To write SQL queries to perform JOIN OPERATIONS like CONDITIONAL JOIN, EQUI JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN.

**CREATING TABLE:**

```
  1  CREATE TABLE countriess1(
  2  country_id VARCHAR2(10) NOT NULL,
  3  country_name VARCHAR2(50),
  4  region_id NUMBER,
  5  PRIMARY KEY(country_id)
  6* )

Table created.

CSE-B-5A2@XE 21-NOV-23> INSERT INTO countriess1 VALUES('AR','ARGENTINA',2);

1 row created.

CSE-B-5A2@XE 21-NOV-23> INSERT INTO countriess1 VALUES('IN','INDIA',3);

1 row created.

CSE-B-5A2@XE 21-NOV-23> INSERT INTO countriess1 VALUES('NL','NETHERLANDS',1);

1 row created.

CSE-B-5A2@XE 21-NOV-23> INSERT INTO countriess1 VALUES('AU','AUSTRALIA',3);

1 row created.

CSE-B-5A2@XE 21-NOV-23> SELECT * FROM countries;

COUNTRY_ID COUNTRY_NAME                                        REGION_ID
---------- ---------------------------------------------------- ----------
AR         ARGENTINA                                                    2
IN         INDIA                                                        3
NL         AUSTRALIA                                                    1
AU         AUSTRALIA                                                    3
```

```
CSE-B-5A2@XE 21-NOV-23> CREATE TABLE locationss1(
  2  street VARCHAR2(50) NOT NULL,
  3  location_id int PRIMARY KEY,
  4  city VARCHAR2(20),
  5  country_id REFERENCES countriess1(country_id)
  6  );

Table created.
```

```
CSE-B-5A2@XE 21-NOV-23> INSERT INTO locationss1 VALUES('ramnagar',1000,'IN','ATP');

1 row created.

CSE-B-5A2@XE 21-NOV-23> INSERT INTO locationss1 VALUES('gandhinagar',1001,'IN','bombay');

1 row created.

CSE-B-5A2@XE 21-NOV-23> INSERT INTO locationss1 VALUES('pieter',1002,'NC','utrecht');

1 row created.

CSE-B-5A2@XE 21-NOV-23> INSERT INTO locationss1 VALUES('street',1003,'AU','sydney');

1 row created.
```

## CONDITIONAL JOIN:

```
CSE-B-5A2@XE 21-NOV-23> SELECT * FROM locationss1 JOIN countriess1
  2  ON
  3  locationss1.country_id=countriess1.country_id;

STREET                                             LOCATION_ID
-------------------------------------------------- -----------
COUNTRY_ID           CITY                  COUNTRY_ID
-------------------- --------------------- ----------
COUNTRY_NAME                                        REGION_ID
-------------------------------------------------- ----------
ramnagar                                                 1000
IN                   ATP                   IN
INDIA                                                       3

gandhinagar                                              1001
IN                   bombay                IN
INDIA                                                       3

STREET                                             LOCATION_ID
-------------------------------------------------- -----------
COUNTRY_ID           CITY                  COUNTRY_ID
-------------------- --------------------- ----------
COUNTRY_NAME                                        REGION_ID
-------------------------------------------------- ----------

street                                                   1003
AU                   sydney                AU
AUSTRALIA                                                   3
```

## NATURAL LEFT OUTER JOIN:

```
CSE-B-5A2@XE 21-NOV-23> SELECT * FROM locationss1 NATURAL LEFT OUTER JOIN countriess1;

COUNTRY_ID           STREET
-------------------- --------------------------------------------------
LOCATION_ID CITY
----------- --------------------
COUNTRY_NAME                                       REGION_ID
-------------------------------------------------- ----------
IN                   ramnagar
       1000 ATP
INDIA                                                       3

IN                   gandhinagar
       1001 bombay
INDIA                                                       3

COUNTRY_ID           STREET
-------------------- --------------------------------------------------
LOCATION_ID CITY
----------- --------------------
COUNTRY_NAME                                       REGION_ID
-------------------------------------------------- ----------

AU                   street
       1003 sydney
AUSTRALIA                                                   3

NC                   pieter
       1002 utrecht

COUNTRY_ID           STREET
-------------------- --------------------------------------------------
LOCATION_ID CITY
----------- --------------------
COUNTRY_NAME                                       REGION_ID
-------------------------------------------------- ----------
```

## NATURAL RIGHT OUTER JOIN:

```
CSE-B-5A2@XE 21-NOV-23> SELECT * from countriess1 NATURAL RIGHT OUTER JOIN locationss1;

COUNTRY_ID           COUNTRY_NAME
-------------------- --------------------------------------------------
 REGION_ID STREET                                          LOCATION_ID
---------- -------------------------------------------------- -----------
CITY
--------------------
IN                   INDIA
         3 ramnagar                                             1000
ATP

IN                   INDIA
         3 gandhinagar                                          1001
bombay

COUNTRY_ID           COUNTRY_NAME
-------------------- --------------------------------------------------
 REGION_ID STREET                                          LOCATION_ID
---------- -------------------------------------------------- -----------
CITY
--------------------

AU                   AUSTRALIA
         3 street                                               1003
sydney

NC
           pieter                                               1002

COUNTRY_ID           COUNTRY_NAME
-------------------- --------------------------------------------------
 REGION_ID STREET                                          LOCATION_ID
---------- -------------------------------------------------- -----------
CITY
--------------------
utrecht
```

```
---------- -------------------------------------------------- -----------
CITY
--------------------
utrecht


CSE-B-5A2@XE 21-NOV-23> SELECT location_id,city,country_id FROM locationss1 EQUI JOIN WHERE locationss1.country_id=countriess1.country_ID;
SELECT location_id,city,country_id FROM locationss1 EQUI JOIN WHERE locationss1.country_id=countriess1.country_ID
                                                                  *
```

**7. Write SQL queries to perform AGGREGATE OPERATIONS (i.e., SUM, COUNT, AVG, MIN, MAX).**

**Aim:** To Write SQL queries to perform AGGREGATE OPERATIONS i.e., SUM, COUNT, AVG, MIN, MAX.

**CREATING TABLE:**

```
CSE-B-5A2@XE 07-NOV-23> CREATE TABLE offices(
  2  emp_id int PRIMARY KEY,
  3  emp_name VARCHAR2(30) NOT NULL,
  4  salary NUMBER(9, 2),
  5  branch_name VARCHAR2(30)
  6  );

Table created.
```

**INSERTING:**

```
CSE-B-5A2@XE 07-NOV-23> INSERT INTO offices VALUES(101,'steve','90000','spring');

1 row created.

CSE-B-5A2@XE 07-NOV-23> INSERT INTO offices VALUES(102,'max','95000','fall');

1 row created.

CSE-B-5A2@XE 07-NOV-23> INSERT INTO offices VALUES(103,'will','93000','spring');

1 row created.

CSE-B-5A2@XE 07-NOV-23> INSERT INTO offices VALUES(104,'joyce','85000','fall');

1 row created.

CSE-B-5A2@XE 07-NOV-23> SELECT * FROM offices;
```

**SUM:**

To find salary (sum of salaries) of fall branch.

```
CSE-B-5A2@XE 07-NOV-23> SELECT SUM(salary) AS salary FROM offices
  2  WHERE branch_name = 'fall';

    SALARY
----------
    180000
```

To find salary (sum of salaries) of spring branch.

```
CSE-B-5A2@XE 07-NOV-23> SELECT SUM(salary) AS salary FROM offices
  2  WHERE branch_name = 'spring';

    SALARY
----------
    183000
```

**COUNT:**

To find the number of employees in the company.

```
CSE-B-5A2@XE 07-NOV-23> SELECT COUNT(emp_id) FROM offices
  2  ;

COUNT(EMP_ID)
-------------
           4
```

```
CSE-B-5A2@XE 07-NOV-23> SELECT COUNT(emp_id) AS spring_emp
  2  FROM offices
  3  WHERE branch_name='spring';

SPRING_EMP
----------
         2
```

**AVERAGE:**

To find average salary of fall brnach

```
CSE-B-5A2@XE 07-NOV-23> SELECT avg(salary) AS avg_salary FROM offices
  2  WHERE branch_name='fall';

AVG_SALARY
----------
     90000
```

To find average salary of spring branch

```
CSE-B-5A2@XE 07-NOV-23> SELECT avg(salary) AS avg_salary FROM offices
  2  WHERE branch_name='spring';

AVG_SALARY
----------
     91500
```

**MIN:**

```
CSE-B-5A2@XE 07-NOV-23> SELECT MIN(salary) AS min_salary
  2  FROM offices
  3  WHERE branch_name='fall';

MIN_SALARY
----------
     85000

CSE-B-5A2@XE 07-NOV-23> SELECT MIN(salary) AS min_salary
  2  FROM offices
  3  WHERE branch_name='spring';

MIN_SALARY
----------
     90000
```

**MAX:**

```
CSE-B-5A2@XE 07-NOV-23> SELECT MAX(salary) AS max_salary
  2  FROM offices
  3  WHERE branch_name='spring';

MAX_SALARY
----------
     93000

CSE-B-5A2@XE 07-NOV-23> SELECT MAX(salary) AS max_salary
  2  FROM offices
  3  WHERE branch_name='fall';

MAX_SALARY
----------
     95000
```

## 8. Write SQL queries to perform QRACLE BUILT IN FUNCTIONS (i.e., DATE, TIME).

**AIM:** TO write SQL queries to perform ORACLE BUILT IN FUNCTIONS like DATE, TIME.

**DATE FUNCTIONS:**

```
CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT SYSDATE FROM DUAL;

SYSDATE
---------
30-JAN-24

CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT SYSDATE+10 FROM DUAL;

SYSDATE+1
---------
09-FEB-24
```

**ADD MONTHS ():**

```
CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT ADD_MONTHS(SYSDATE,+2) FROM DUAL;

ADD_MONTH
---------
30-MAR-24

CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT ADD_MONTHS(SYSDATE,-2) FROM DUAL;

ADD_MONTH
---------
30-NOV-23
```

```
CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT SYSDATE
  2  FROM DUAL;

SYSDATE
---------
30-JAN-24
```

```
CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT SYSDATE+10 FROM DUAL;

SYSDATE+1
---------
09-FEB-24

CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT SYSDATE-10 FROM DUAL;

SYSDATE-1
---------
20-JAN-24
```

## LAST DAY:

```
CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT LAST_DAY(SYSDATE)
  2  FROM DUAL;

LAST_DAY(
---------
31-JAN-24
```

## NEXT DAY:

```
CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT NEXT_DAY(SYSDATE,'MONDAY')
  2  FROM DUAL;

NEXT_DAY(
---------
05-FEB-24

CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT NEXT_DAY(SYSDATE,'FRIDAY')
  2  FROM DUAL;

NEXT_DAY(
---------
02-FEB-24
```

```
CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT
  2  CURRENT_TIMESTAMP(3)
  3  FROM DUAL;

CURRENT_TIMESTAMP(3)
---------------------------------------------------------------------------
30-JAN-24 08.17.47.710 PM +05:30
```

## MONTHS BETWEEN:

```
CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT MONTHS_BETWEEN('05-JAN-24','05-JAN-23')
  2  FROM DUAL;

MONTHS_BETWEEN('05-JAN-24','05-JAN-23')
---------------------------------------
                                     12

CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT MONTHS_BETWEEN('05-JAN-23','05-JAN-24')
  2  FROM DUAL;

MONTHS_BETWEEN('05-JAN-23','05-JAN-24')
---------------------------------------
                                    -12
```

## ROUND ():

```
CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT ROUND(45.626,2) FROM DUAL;

ROUND(45.626,2)
---------------
          45.63

CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT ROUND(45.626,0) FROM DUAL;

ROUND(45.626,0)
---------------
             46

CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT ROUND(45.626,-1) FROM DUAL;

ROUND(45.626,-1)
----------------
              50

CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT ROUND(45.626,-2) FROM DUAL;

ROUND(45.626,-2)
----------------
               0
```

## TRUNC ():

```
CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT TRUNC(45.626,2) FROM DUAL;

TRUNC(45.626,2)
---------------
          45.62

CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT TRUNC(45.626,0) FROM DUAL;

TRUNC(45.626,0)
---------------
             45

CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT TRUNC(45.626,-1) FROM DUAL;

TRUNC(45.626,-1)
----------------
              40

CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT TRUNC(45.626,-2) FROM DUAL;

TRUNC(45.626,-2)
----------------
               0
```

**TIME FUNCTIONS:**

CURRENT TIMESTAMP ():

```
CSE-B-5A2@_CONNECT_IDENTIFIED 30-JAN-24>SELECT CURRENT_TIMESTAMP(3) FROM DUAL
  2  ;

CURRENT_TIMESTAMP(3)
---------------------------------------------------------------------------
30-JAN-24 08.35.00.548 PM +05:30
```

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

**Aim:** To perform KEY CONSTRAINTS using SQL queries.

**PRIMARY KEY:**

```
CSE-B-5A2@XE 28-NOV-2023> CREATE TABLE countries2(
  2  country_id VARCHAR2(10) NOT NULL,
  3  country_name VARCHAR2(50),
  4  region_id NUMBER,
  5  PRIMARY KEY(country_id)
  6  );

Table created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO countries2 VALUES('AR','Argentina',2);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO countries2 VALUES('IN','India',3);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO countries2 VALUES('NL','NetherLands',1);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO countries2 VALUES('AU','Australia',3);

1 row created.
```

SELECTING VALUES:

```
CSE-B-5A2@XE 28-NOV-2023> SELECT * FROM countries;

COUNTRY_ID COUNTRY_NAME                                    REGION_ID
---------- ---------------------------------------------- ----------
AR         ARGENTINA                                               2
IN         INDIA                                                   3
NL         AUSTRALIA                                               1
AU         AUSTRALIA                                               3
```

**FOREIGN KEYS:**

```
CSE-B-5A2@XE 28-NOV-2023> CREATE TABLE locations3(
  2  location_id int PRIMARY KEY,
  3  street VARCHAR2(50) NOT NULL,
  4  country_id REFERENCES countries2(country_id),
  5  city VARCHAR2(20)
  6  );

Table created.
```

INSERTING VALUES:

```
CSE-B-5A2@XE 28-NOV-2023> INSERT INTO locations3 VALUES(1000,'RamNagar','IN','Atp');

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO locations3 VALUES(1001,'GandhiNagar','IN','Bombay');

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO locations3 VALUES(1002,'VictoriaStreet','AU','Sydney');

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO locations3 VALUES(1003,'PieterBreugh','NL','NETHERLANDS');

1 row created.
```

SELECTING VALUES:

```
CSE-B-5A2@XE 28-NOV-2023> SELECT * FROM locations3;

LOCATION_ID STREET                                         COUNTRY_ID
----------- ---------------------------------------------- ----------
CITY
--------------------
       1000 RamNagar                                       IN
Atp

       1001 GandhiNagar                                    IN
Bombay

       1002 VictoriaStreet                                 AU
Sydney


LOCATION_ID STREET                                         COUNTRY_ID
----------- ---------------------------------------------- ----------
CITY
--------------------
       1003 PieterBreugh                                   NL
NETHERLANDS
```

**UNIQUE KEY:**

```
CSE-B-5A2@XE 28-NOV-2023> CREATE TABLE employee(
  2  emp_id int UNIQUE,
  3  name VARCHAR2(20),
  4  salary NUMBER(9,2)
  5  );

Table created.
```

INSETING VALUES:

```
Table created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO employee VALUES(100,'Steve',75000);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO employee VALUES(101,'Nancy',70000);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO employee VALUES(102,'Robin',72000);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO employee VALUES(103,'Jonathon',70000);

1 row created.
```

**NOT NULL:**

```
CSE-B-5A2@XE 28-NOV-2023> CREATE TABLE student(
  2  ename VARCHAR2(20) NOT NULL,
  3  id int NOT NULL,
  4  age int
  5  );

Table created.
```

INSERTING VALUES:

```
CSE-B-5A2@XE 28-NOV-2023> INSERT INTO student VALUES('Steve',101,25);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO student VALUES('Nancy',102,23);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO student VALUES('Robin',103,24);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO student VALUES('Will',104,23);

1 row created.
```

SELECTING VALUES:

```
CSE-B-5A2@XE 28-NOV-2023> SELECT * FROM student;

ENAME                          ID        AGE
-------------------- ---------- ----------
Steve                         101         25
Nancy                         102         23
Robin                         103         24
Will                          104         23
```

**CHECK KEY:**

```
CSE-B-5A2@XE 28-NOV-2023> CREATE table stud1(
  2  id int PRIMARY KEY,
  3  name VARCHAR2(20),
  4  marks int CHECK(marks<=100)
  5  );

Table created.
```

INSERTING VALUES:

```
CSE-B-5A2@XE 28-NOV-2023> INSERT INTO stud1 VALUES(101,'Steve',99);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO stud1 VALUES(102,'nancy',95);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO stud1 VALUES(103,'Jonathon',97);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO stud1 VALUES(104,'Robin',96);

1 row created.
```

SELECTING VALUES:

```
CSE-B-5A2@XE 28-NOV-2023> SELECT * FROM stud1;

        ID NAME                          MARKS
---------- -------------------- ----------
       101 Steve                        99
       102 nancy                        95
       103 Jonathon                     97
       104 Robin                        96
```

**DEFAULT KEY:**

```
CSE-B-5A2@XE 28-NOV-2023> CREATE TABLE emp(
  2  id int PRIMARY KEY,
  3  name VARCHAR2(50),
  4  salary NUMBER(9,2) DEFAULT '0'
  5  );

Table created.
```

INSERTING VALUES:

```
CSE-B-5A2@XE 28-NOV-2023> INSERT INTO emp VALUES(101,'Steve',90000);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO emp VALUES(102,'Nancy',90000);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO emp VALUES(103,'Jonathon',88000);

1 row created.

CSE-B-5A2@XE 28-NOV-2023> INSERT INTO emp(id,name) VALUES(104,'Robin');

1 row created.
```

SELECTING VALUES:

```
CSE-B-5A2@XE 28-NOV-2023> SELECT * FROM emp;

        ID NAME                                               SALARY
---------- -------------------------------------------------- ----------
       101 Steve                                               90000
       102 Nancy                                               90000
       103 Jonathon                                            88000
       104 Robin                                                   0
```

**10. Write a PL/SQL program for calculating the factorial of a given number.**

```
CSE-B-5A2@XE 23-NOV-23> SET SERVEROUT ON;
CSE-B-5A2@XE 23-NOV-23> DECLARE
  2  fac NUMBER :=1;
  3  n NUMBER := 10;
  4  BEGIN
  5  WHILE n > 0 LOOP
  6  fac:=n*fac;
  7  n:=n-1;
  8  END LOOP;
  9  DBMS_OUTPUT.PUT_LINE(FAC);
 10  END;
 11  /
3628800

PL/SQL procedure successfully completed.
```

**11. Write a PL/SQL program to finding the given number is prime number or not.**

```
CSE-B-5A2@XE 28-NOV-23> DECLARE
  2  n NUMBER;
  3  i NUMBER;
  4  temp NUMBER;
  5  BEGIN
  6  n := 13;
  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  /
13 is a prime number

PL/SQL procedure successfully completed.
```

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

```
CSE-B-5A2@XE 05-DEC-23> DECLARE
  2  FIRST NUMBER := 0;
  3  SECOND NUMBER := 1;
  4  TEMP NUMBER;
  5  N NUMBER := 5;
  6  I NUMBER;
  7  BEGIN
  8  DBMS_OUTPUT.PUT_LINE('SERIES:');
  9  DBMS_OUTPUT.PUT_LINE(FIRST);
 10  DBMS_OUTPUT.PUT_LINE(SECOND);
 11  FOR I IN 2..N
 12  LOOP
 13  TEMP:=FIRST+SECOND;
 14  FIRST := SECOND;
 15  SECOND := TEMP;
 16  DBMS_OUTPUT.PUT_LINE(TEMP);
 17  END LOOP;
 18  END;
 19  /
SERIES:
0
1
1
2
3
5

PL/SQL procedure successfully completed.
```

## 13. Write a PL/SQL program to implement stored procedure on table.

### SYNTAX:

CREATE [OR REPLACE] PROCEDURE procedure_name

[ (parameter [,parameter]) ]

(IS | AS)

[declaration_section]

BEGIN

executable_section

[EXCEPTION

exception_section]

END [procedure_name];

```
CSE-B-5A2@XE 05-DEC-23> CREATE TABLE SAILORS(
  2  ID NUMBER(10) PRIMARY KEY,
  3  NAME VARCHAR2(100)
  4  );

Table created.

CSE-B-5A2@XE 05-DEC-23> CREATE OR REPLACE PROCEDURE INSERTUSER
  2  (ID IN NUMBER,
  3  NAME IN VARCHAR2)
  4  IS
  5  BEGIN
  6  INSERT INTO SAILORS VALUES(ID,NAME);
  7  DBMS_OUTPUT.PUT_LINE('RECORD INSERTED SUCCESSFULLY');
  8  END;
  9  /

Procedure created.

CSE-B-5A2@XE 05-DEC-23> DECLARE
  2  CNT NUMBER;
  3  BEGIN
  4  INSERTUSER(101,'NARASIMHA');
  5  SELECT COUNT(*) INTO CNT FROM SAILORS;
  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.
```

**14. Write PL/SQL program to implement stored function on table.**

**SYNTAX:**

CREATE [OR REPLACE] FUNCTION function_name

[ (parameter [,parameter]) ]

RETURN return_datatype

(IS | AS)

[declaration_section]

BEGIN

executable_section

[EXCEPTION

exception_section]

END [procedure_name];

```
CSE-B-5A2@XE 05-DEC-23> CREATE OR REPLACE FUNCTION ADDER(N1 IN NUMBER, N2 IN NUMBER)
  2  RETURN NUMBER
  3  IS
  4  N3 NUMBER(8);
  5  BEGIN
  6  N3 :=N1+N2;
  7  RETURN N3;
  8  END;
  9  /

Function created.

CSE-B-5A2@XE 05-DEC-23> DECLARE
  2  N3 NUMBER(2);
  3  BEGIN
  4  N3 := ADDER(11,22);
  5  DBMS_OUTPUT.PUT_LINE('ADDITION IS: ' || N3);
  6  END;
  7  /
ADDITION IS: 33

PL/SQL procedure successfully completed.
```

## => Recursive Function

```
CSE-B-5A2@XE 05-DEC-23> run
  1  CREATE FUNCTION fact1(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 f;
 12* END;

Function created.

CSE-B-5A2@XE 05-DEC-23> DECLARE
  2  num number;
  3  factorial number;
  4  BEGIN
  5  num:= 6;
  6  factorial := fact1(num);
  7  dbms_output.put_line(' Factorial '|| num || ' is ' || factorial);
  8  END;
  9  /
Factorial 6 is 720

PL/SQL procedure successfully completed.
```

## DROP FUNCTION:

```
CSE-B-5A2@XE 05-DEC-23> DROP FUNCTION fact1;

Function dropped.
```

## 15. Write PL/SQL program to implement Trigger on table.

```
CSE-B-5A2@XE 05-DEC-23> SET SQLP "CSE-B-5A2'@'_CONNECT_IDENTIFIER' '12-DEC-23> "
CSE-B-5A2@XE 12-DEC-23> CREATE TABLE DEPARTMENT
  2  (DEPT_NAME VARCHAR2(50),
  3  building VARCHAR2(20),
  4  budget NUMERIC(12, 2) CHECK (budget > 0),
  5  PRIMARY KEY(dept_name)
  6  );

Table created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO department VALUES('biology', 'steve', '90000');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO department VALUES('comp sci', 'ziqui', '120000');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO department VALUES('Elec eng', 'jian', '100000');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO department VALUES('finance', 'xioa', '85000');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO department VALUES('history', 'xiochi', '50000');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO department VALUES('music', 'robin', '80000');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO department VALUES('physics', 'joyce', '70000');

1 row created.
```

```
CSE-B-5A2@XE 12-DEC-23> CREATE TABLE instructor(
  2  id VARCHAR2(30),
  3  name VARCHAR2(50) NOT NULL,
  4  dept_name VARCHAR2(20),
  5  salary NUMERIC(8, 2) CHECK (salary>29000),
  6  PRIMARY KEY(id),
  7  FOREIGN KEY(dept_name) REFERENCES department(dept_name)
  8  ON DELETE SET NULL
  9  );

Table created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO instructor VALUES('10101', 'joyce', 'comp sci', '65000');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO instructor VALUES('10102', 'robin', 'finance', '90000');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO instructor VALUES('10103', 'hopper', 'music', '40000');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO instructor VALUES('10104', 'will', 'physics', '95000');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO instructor VALUES('10105', 'mike', 'history', '65000');

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO instructor VALUES('10106', 'luka', 'physics', '87000');

1 row created.
```

```
CSE-B-5A2@XE 12-DEC-23> CREATE OR REPLACE TRIGGER display_salary_changes
  2  BEFORE UPDATE ON instructor
  3  FOR EACH ROW
  4  WHEN (NEW.ID = OLD.ID)
  5  DECLARE
  6  sal_diff number;
  7  BEGIN
  8  sal_diff := :NEW.salary - :OLD.salary;
  9  dbms_output.put_line('Old salary: ' || :OLD.salary);
 10  dbms_output.put_line('New salary: ' || :NEW.salary);
 11  dbms_output.put_line('Salary difference: ' || sal_diff);
 12  END;
 13  /

Trigger created.
```

```
CSE-B-5A2@XE 12-DEC-23> DECLARE
  2  total_rows number(2);
  3  BEGIN
  4  UPDATE instructor
  5  SET salary = salary + 5000;
  6  IF sql%notfound THEN
  7  dbms_output.put_line('no instructors updated');
  8  ELSIF sql%found THEN
  9  total_rows := sql%rowcount;
 10  dbms_output.put_line( total_rows || ' instructors updated ');
 11  END IF;
 12  END;
 13  /
Old salary: 65000
New salary: 70000
Salary difference: 5000
Old salary: 90000
New salary: 95000
Salary difference: 5000
Old salary: 40000
New salary: 45000
Salary difference: 5000
Old salary: 95000
New salary: 100000
Salary difference: 5000
Old salary: 65000
New salary: 70000
Salary difference: 5000
Old salary: 87000
New salary: 92000
Salary difference: 5000
6 instructors updated

PL/SQL procedure successfully completed.
```

## 16. Write PL/SQL program to implement cursor on table.

```
CSE-B-5A2@XE 12-DEC-23> CREATE TABLE customers(
  2  id NUMBER PRIMARY KEY,
  3  NAME VARCHAR2(20) NOT NULL,
  4  AGE NUMBER,
  5  ADDRESS VARCHAR2(20),
  6  salary NUMERIC(20, 2)
  7  );

Table created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO customers VALUES(1,'steve',24,'skull rock',29000);

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO customers VALUES(2,'robin',22,'upside down',31000);

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO customers VALUES(3,'nancy',21,'hawkins',33000);

1 row created.

CSE-B-5A2@XE 12-DEC-23> INSERT INTO customers VALUES(4,'john',23,'indiana',35000);

1 row created.
```

```
CSE-B-5A2@XE 12-DEC-23> DECLARE
  2  total_rows number(2);
  3  BEGIN
  4  UPDATE customers
  5  SET salary = salary + 5000;
  6  IF sql%notfound THEN
  7  dbms_output.put_line('no customers updated');
  8  ELSIF sql%found THEN
  9  total_rows := sql%rowcount;
 10  dbms_output.put_line( total_rows || ' customers updated ');
 11  END IF;
 12  END;
 13  /
4 customers updated

PL/SQL procedure successfully completed.
```

```
CSE-B-5A2@XE 12-DEC-23> DECLARE
  2  c_id customers.id%type;
  3  c_name customers.name%type;
  4  c_addr customers.address%type;
  5  CURSOR c_customers is
  6  SELECT id, name, address FROM customers;
  7  BEGIN
  8  OPEN c_customers;
  9  LOOP
 10  FETCH c_customers into c_id, c_name, c_addr;
 11  EXIT WHEN c_customers%notfound;
 12  dbms_output.put_line(c_id || ' ' || c_name || ' ' || c_addr);
 13  END LOOP;
 14  CLOSE c_customers;
 15  END;
 16  /
1 steve skull rock
2 robin upside down
3 nancy hawkins
4 john indiana

PL/SQL procedure successfully completed.
```