EXPERIMENT-1

WRITE SQL QUERIES TO CREATE TABLES FOR VARIOUS DATABASES USING DDL COMMANDS (CREATE, ALTER, DROP, TRUNCATE)

```
CSE-B-572@XE 20-SEP-23> CREATE TABLE persons(
  2  person_id NUMBER,
  3  first_name VARCHAR2(10) NOT NULL,
  4  last_name VARCHAR2(15) NOT NULL,
  5  PRIMARY KEY(person_id)
  6  );

Table created.
```

```
CSE-B-572@XE 20-SEP-23> INSERT INTO persons VALUES(101,'Pranavi','Sharma');

1 row created.

CSE-B-572@XE 20-SEP-23> INSERT INTO persons VALUES(102,'Priya','Birla');

1 row created.

CSE-B-572@XE 20-SEP-23> INSERT INTO persons VALUES(103,'Armaan','Poddar');

1 row created.

CSE-B-572@XE 20-SEP-23> Select * from persons;

 PERSON_ID FIRST_NAME LAST_NAME
---------- ---------- ---------------
       101 Pranavi    Sharma
       102 Priya      Birla
       103 Armaan     Poddar
```

```
CSE-B-572@XE 20-SEP-23> ALTER TABLE persons
  2  ADD birth_date DATE;

Table altered.

CSE-B-572@XE 20-SEP-23> desc persons;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------------
 PERSON_ID                                 NOT NULL NUMBER
 FIRST_NAME                                NOT NULL VARCHAR2(10)
 LAST_NAME                                 NOT NULL VARCHAR2(15)
 BIRTH_DATE                                         DATE
```

```
CSE-B-572@XE 20-SEP-23> ALTER TABLE persons
  2  ADD (
  3  phone VARCHAR2(10),
  4  email VARCHAR2(20)
  5  );

Table altered.

CSE-B-572@XE 20-SEP-23> desc persons;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PERSON_ID                                 NOT NULL NUMBER
 FIRST_NAME                                NOT NULL VARCHAR2(10)
 LAST_NAME                                 NOT NULL VARCHAR2(15)
 BIRTH_DATE                                         DATE
 PHONE                                              VARCHAR2(10)
 EMAIL                                              VARCHAR2(20)
```

```
CSE-B-572@XE 20-SEP-23> ALTER TABLE persons
  2  RENAME COLUMN person_id TO p_id;

Table altered.

CSE-B-572@XE 20-SEP-23> desc persons;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 P_ID                                      NOT NULL NUMBER
 FIRST_NAME                                NOT NULL VARCHAR2(10)
 LAST_NAME                                 NOT NULL VARCHAR2(15)
 BIRTH_DATE                                         DATE
 PHONE                                              VARCHAR2(10)
 EMAIL                                              VARCHAR2(20)
```

```
CSE-B-572@XE 20-SEP-23> ALTER TABLE persons
  2  DROP COLUMN email;

Table altered.

CSE-B-572@XE 20-SEP-23> desc persons;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 P_ID                                      NOT NULL NUMBER
 FIRST_NAME                                NOT NULL VARCHAR2(10)
 LAST_NAME                                 NOT NULL VARCHAR2(15)
 BIRTH_DATE                                         DATE
 PHONE                                              VARCHAR2(10)
```

```
CSE-B-572@XE 20-SEP-23> DROP TABLE persons;

Table dropped.

CSE-B-572@XE 20-SEP-23> desc persons;
ERROR:
ORA-04043: object persons does not exist
```

```
CSE-B-572@XE 20-SEP-23> CREATE TABLE quotations (
  2  quotation_no NUMERIC NOT NULL,
  3  customer_id NUMERIC NOT NULL,
  4  valid_form DATE NOT NULL,
  5  valid_to DATE NOT NULL,
  6  PRIMARY KEY(quotation_no)
  7  );

Table created.

CSE-B-572@XE 20-SEP-23> desc quotations;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 QUOTATION_NO                              NOT NULL NUMBER(38)
 CUSTOMER_ID                               NOT NULL NUMBER(38)
 VALID_FORM                                NOT NULL DATE
 VALID_TO                                  NOT NULL DATE
```

```
CSE-B-572@XE 20-SEP-23> TRUNCATE TABLE quotations;

Table truncated.
```

```
CSE-B-572@XE 20-SEP-23> RENAME quotations TO quotations1;

Table renamed.

CSE-B-572@XE 20-SEP-23> desc quotations;
ERROR:
ORA-04043: object quotations does not exist


CSE-B-572@XE 20-SEP-23> desc quotations1;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 QUOTATION_NO                              NOT NULL NUMBER(38)
 CUSTOMER_ID                               NOT NULL NUMBER(38)
 VALID_FORM                                NOT NULL DATE
 VALID_TO                                  NOT NULL DATE
```

EXPERIMENT-2

WRITE SQL QUERIES TO MANIPULATE TABLES FOR VARIOUS DATABASES USING DML COMMANDS (INSERT, SELECT, UPDATE, DELETE)

```
CSE-B-572@XE 28-SEP-23> CREATE TABLE student_info (
  2  s_id VARCHAR2(10) NOT NULL,
  3  s_name VARCHAR2(10) NOT NULL,
  4  s_branch VARCHAR2(15) NOT NULL,
  5  PRIMARY KEY(s_id)
  6  );

Table created.

CSE-B-572@XE 28-SEP-23> desc student_info;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 S_ID                                      NOT NULL VARCHAR2(10)
 S_NAME                                    NOT NULL VARCHAR2(10)
 S_BRANCH                                  NOT NULL VARCHAR2(15)
```

```
CSE-B-572@XE 28-SEP-23> CREATE TABLE orders (
  2  cid NUMBER PRIMARY KEY,
  3  oid NUMBER,
  4  ono NUMBER
  5  );

Table created.

CSE-B-572@XE 28-SEP-23> desc orders;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 CID                                       NOT NULL NUMBER
 OID                                                NUMBER
 ONO                                                NUMBER
```

```
CSE-B-572@XE 28-SEP-23> INSERT INTO orders VALUES(1,101,501);

1 row created.

CSE-B-572@XE 28-SEP-23> INSERT INTO orders VALUES(2,201,601);

1 row created.

CSE-B-572@XE 28-SEP-23> INSERT INTO orders VALUES(3,401,501);

1 row created.

CSE-B-572@XE 28-SEP-23> INSERT INTO orders VALUES(4,501,601);

1 row created.

CSE-B-572@XE 28-SEP-23> DESC orders;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CID                                       NOT NULL NUMBER
 OID                                                NUMBER
 ONO                                                NUMBER

CSE-B-572@XE 28-SEP-23> SELECT * FROM orders;

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

Update

```
CSE-B-572@XE 28-SEP-23> UPDATE orders
  2  SET cid = 6
  3  WHERE oid = 501;

1 row updated.

CSE-B-572@XE 28-SEP-23> SELECT * FROM orders
  2  WHERE oid = 501;

       CID        OID        ONO
---------- ---------- ----------
         6        501        601
```

```
CSE-B-572@XE 28-SEP-23> UPDATE orders
  2  SET cid = 7,ono = 601
  3  WHERE oid = 501;

1 row updated.

CSE-B-572@XE 28-SEP-23> select * from orders
  2  WHERE oid = 501;

       CID        OID        ONO
---------- ---------- ----------
         7        501        601
```

```
CSE-B-572@XE 28-SEP-23> UPDATE orders
  2  SET cid = cid*2;

4 rows updated.

CSE-B-572@XE 28-SEP-23> select * from orders;

       CID        OID        ONO
---------- ---------- ----------
         2        101        501
         4        201        601
         6        401        501
        14        501        601
```

DELETE

```
        14        501        601

CSE-B-572@XE 28-SEP-23> DELETE FROM orders WHERE cid = 1;

0 rows deleted.

CSE-B-572@XE 28-SEP-23> DELETE FROM orders WHERE cid = 4;

1 row deleted.

CSE-B-572@XE 28-SEP-23> select * from orders;

       CID        OID        ONO
---------- ---------- ----------
         2        101        501
         6        401        501
        14        501        601
```

```
CSE-B-572@XE 28-SEP-23> DELETE FROM orders;

3 rows deleted.

CSE-B-572@XE 28-SEP-23> select * from orders;

no rows selected
```

SELECT

```
CSE-B-572@XE 28-SEP-23> CREATE TABLE department (
  2  did NUMBER(3),
  3  dname VARCHAR2(5),
  4  description VARCHAR2(20)
  5  );

Table created.

CSE-B-572@XE 28-SEP-23> desc department;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DID                                                NUMBER(3)
 DNAME                                              VARCHAR2(5)
 DESCRIPTION                                        VARCHAR2(20)
```

```
CSE-B-572@XE 28-SEP-23> INSERT INTO department VALUES(10,'CSE','Computer Science');

1 row created.

CSE-B-572@XE 28-SEP-23> INSERT INTO department VALUES(20,'CSD','Data Science');

1 row created.

CSE-B-572@XE 28-SEP-23> INSERT INTO department VALUES(30,'CSM','Machine Learning');

1 row created.

CSE-B-572@XE 28-SEP-23> INSERT INTO department VALUES(30,'EEE','Electrial and Electronics');
INSERT INTO department VALUES(30,'EEE','Electrial and Electronics')
                                       *
ERROR at line 1:
ORA-12899: value too large for column "SYSTEM"."DEPARTMENT"."DESCRIPTION"
(actual: 25, maximum: 20)


CSE-B-572@XE 28-SEP-23> INSERT INTO department VALUES(30,'EEE','Electronics');

1 row created.

CSE-B-572@XE 28-SEP-23> SELECT * FROM department;

       DID DNAME DESCRIPTION
---------- ----- --------------------
        10 CSE   Computer Science
        20 CSD   Data Science
        30 CSM   Machine Learning
        30 EEE   Electronics
```

```
CSE-B-572@XE 28-SEP-23> SELECT did FROM department;

       DID
----------
        10
        20
        30
        30
        30

CSE-B-572@XE 28-SEP-23> SELECT did,dname FROM department;

       DID DNAME
---------- -----
        10 CSE
        20 CSD
        30 CSM
        30 EEE
```

```
CSE-B-572@XE 28-SEP-23> SELECT * FROM department;

       DID DNAME DESCRIPTION
---------- ----- --------------------
        10 CSE   Computer Science
        20 CSD   Data Science
        30 CSM   Machine Learning
        30 EEE   Electronics
```

EXPERIMENT-3

WRITE SQL QUERIES TO VIEWS FOR VARIOUS DATABASES (CREATE VIEW, ALTER VIEW, AND DELETE VIEW)

Table creation

```
CSE-B-572@XE 17-OCT-23> CREATE TABLE scholar (
  2  scholar_id NUMBER NOT NULL,
  3  scholar_name VARCHAR2(20) NOT NULL,
  4  branch VARCHAR2(20) NOT NULL,
  5  building VARCHAR2(15) NOT NULL,
  6  PRIMARY KEY(scholar_id)
  7  );

Table created.
```

Inserting Values

```
CSE-B-572@XE 17-OCT-23> INSERT INTO scholar VALUES(571,'Armaan','CSE','Bblock');

1 row created.

CSE-B-572@XE 17-OCT-23> INSERT INTO scholar VALUES(572,'Abhira','CSM','Ablock');

1 row created.

CSE-B-572@XE 17-OCT-23> INSERT INTO scholar VALUES(573,'Roohi','CSD','Bblock');

1 row created.

CSE-B-572@XE 17-OCT-23> INSERT INTO scholar VALUES(574,'Rohith','EEE','Cblock');

1 row created.

CSE-B-572@XE 17-OCT-23> SELECT * FROM scholar;

SCHOLAR_ID SCHOLAR_NAME         BRANCH               BUILDING
---------- -------------------- -------------------- ---------------
       571 Armaan               CSE                  Bblock
       572 Abhira               CSM                  Ablock
       573 Roohi                CSD                  Bblock
       574 Rohith               EEE                  Cblock
```

Creating View

```
CSE-B-572@XE 17-OCT-23> CREATE VIEW professor AS
  2  SELECT scholar_id,scholar_name,branch,building FROM scholar;

View created.
```

Inserting Values into VIEWS

```
CSE-B-572@XE 17-OCT-23> INSERT INTO professor VALUES(575,'Dev','CIVIL','Cblock');

1 row created.

CSE-B-572@XE 17-OCT-23> INSERT INTO professor VALUES(576,'Sonakshi','CSE','Bblock');

1 row created.

CSE-B-572@XE 17-OCT-23> INSERT INTO professor VALUES(577,'Akshara','CSM','Ablock');

1 row created.

CSE-B-572@XE 17-OCT-23> SELECT * FROM professor;

SCHOLAR_ID SCHOLAR_NAME         BRANCH               BUILDING
---------- -------------------- -------------------- ---------------
       571 Armaan               CSE                  Bblock
       572 Abhira               CSM                  Ablock
       573 Roohi                CSD                  Bblock
       574 Rohith               EEE                  Cblock
       575 Dev                  CIVIL                Cblock
       576 Sonakshi             CSE                  Bblock
       577 Akshara              CSM                  Ablock

7 rows selected.
```

Update VIEWS

```
CSE-B-572@XE 17-OCT-23> UPDATE professor SET scholar_name = 'Abhimanyu' WHERE scholar_id = 577;

1 row updated.

CSE-B-572@XE 17-OCT-23> SELECT * FROM professor;

SCHOLAR_ID SCHOLAR_NAME         BRANCH               BUILDING
---------- -------------------- -------------------- ---------------
       571 Armaan               CSE                  Bblock
       572 Abhira               CSM                  Ablock
       573 Roohi                CSD                  Bblock
       574 Rohith               EEE                  Cblock
       575 Dev                  CIVIL                Cblock
       576 Sonakshi             CSE                  Bblock
       577 Abhimanyu            CSM                  Ablock

7 rows selected.
```

DELETE VIEWS

```
CSE-B-572@XE 17-OCT-23> DROP VIEW professor;

View dropped.

CSE-B-572@XE 17-OCT-23> SELECT * FROM professor;
SELECT * FROM professor
              *
ERROR at line 1:
ORA-00942: table or view does not exist


CSE-B-572@XE 17-OCT-23> SELECT * FROM scholar;

SCHOLAR_ID SCHOLAR_NAME         BRANCH               BUILDING
---------- -------------------- -------------------- ---------------
       571 Armaan               CSE                  Bblock
       572 Abhira               CSM                  Ablock
       573 Roohi                CSD                  Bblock
       574 Rohith               EEE                  Cblock
       575 Dev                  CIVIL                Cblock
       576 Sonakshi             CSE                  Bblock
       577 Abhimanyu            CSM                  Ablock

7 rows selected.
```

EXPERINMENT-4

WRITE SQL QUERIES TO PERFORM RELATIONAL SET OPERATIONS (UNION, UNION ALL, CROSS JOIN, NATURAL JOIN, MINUS, INTERSECT, INTERSECT ALL, MINUS ALL)

Creating tables

```
CSE-B-572@XE 7-N0V-23> CREATE TABLE section (
  2  course_id NUMBER PRIMARY KEY,
  3  semester VARCHAR2(20) NOT NULL,
  4  year NUMBER NOT NULL,
  5  building VARCHAR2(10) NOT NULL
  6  );

Table created.
```

```
CSE-B-572@XE 7-N0V-23> CREATE TABLE class (
  2  course_name VARCHAR2(20) NOT NULL,
  3  semester VARCHAR2(20)  NOT NULL,
  4  course_id NUMBER REFERENCES section(course_id)
  5  );

Table created.
```

Inserting Values

```
CSE-B-572@XE 7-N0V-23> INSERT INTO section VALUES(101,'Spring',2004,'Ablock');

1 row created.

CSE-B-572@XE 7-N0V-23> INSERT INTO section VALUES(102,'Fall',2005,'Bblock');

1 row created.

CSE-B-572@XE 7-N0V-23> INSERT INTO section VALUES(103,'Spring',2004,'Ablock');

1 row created.

CSE-B-572@XE 7-N0V-23> INSERT INTO section VALUES(104,'Fall',2005,'Bblock');

1 row created.
```

```
CSE-B-572@XE 7-N0V-23> INSERT INTO class VALUES('Database','one',104);

1 row created.

CSE-B-572@XE 7-N0V-23> INSERT INTO class VALUES('DS','two',103);

1 row created.

CSE-B-572@XE 7-N0V-23> INSERT INTO class VALUES('PS','three',102);

1 row created.

CSE-B-572@XE 7-N0V-23> INSERT INTO class VALUES('ELESE','four',101);

1 row created.
```

View the data

```
CSE-B-572@XE 7-N0V-23> SELECT * FROM section;

 COURSE_ID SEMESTER                  YEAR BUILDING
---------- ------------------ ---------- ----------
       101 Spring                   2004 Ablock
       102 Fall                     2005 Bblock
       103 Spring                   2004 Ablock
       104 Fall                     2005 Bblock
```

```
CSE-B-572@XE 7-N0V-23> select * from class;

COURSE_NAME          SEMESTER              COURSE_ID
-------------------- -------------------- ----------
Database             one                         104
DS                   two                         103
PS                   three                       102
ELESE                four                        101
```

UNION operation

```
CSE-B-572@XE 7-N0V-23> SELECT  course_id
  2  FROM section WHERE  semester = 'Spring' AND year = 2004
  3  UNION
  4  SELECT  course_id
  5  FROM section WHERE  semester = 'Fall' AND year = 2005;

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

UNION ALL operation

```
CSE-B-572@XE 7-N0V-23> SELECT  course_id
  2  FROM section WHERE  semester = 'Spring' AND year = 2004
  3  UNION ALL
  4  SELECT  course_id
  5  FROM section WHERE  semester = 'Fall' AND year = 2005;

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

INTERSECT operation

```
CSE-B-572@XE 7-N0V-23> SELECT  year
  2  FROM section WHERE  semester = 'Spring' AND course_id  = 101
  3  INTERSECT
  4  SELECT  year
  5  FROM section WHERE  semester = 'Fall' AND course_id  = 102;

no rows selected
```

```
CSE-B-572@XE 7-N0V-23> SELECT  year
  2  FROM section WHERE  semester = 'Spring' AND course_id  = 101
  3  INTERSECT
  4  SELECT  year
  5  FROM section WHERE  semester = 'Spring' AND course_id  = 103;

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

INTERSECT ALL operation

```
CSE-B-572@XE 7-N0V-23> SELECT  year
  2  FROM section WHERE  semester = 'Spring' AND course_id  = 103
  3  INTERSECT ALL
  4  SELECT  year
  5  FROM section WHERE  semester = 'Spring' AND course_id  = 101;

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

MINUS operation

```
CSE-B-572@XE 7-N0V-23> SELECT course_id
  2  FROM section WHERE  semester = 'Spring' AND year = 2004
  3  MINUS
  4  SELECT course_id
  5  FROM section WHERE  semester = 'Fall' AND year = 2005;

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

MINUS ALL operation

```
CSE-B-572@XE 7-N0V-23> SELECT course_id
  2  FROM section WHERE  semester = 'Fall' AND year = 2005
  3  MINUS ALL
  4  SELECT course_id
  5  FROM section WHERE  semester = 'Spring' AND year = 2004;

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

EXPERIMENT -5

WRITE SQL QUERIES TO PERFORM SPECIAL OPERATIONS (IS NULL, BETWEEN, LIKE, IN, EXISTS)

Creating Tables

```
CSE-B-572@XE 14-NOV-23> CREATE TABLE instructor (
  2  in_id VARCHAR2(10) NOT NULL,
  3  name VARCHAR2(10),
  4  salary NUMBER
  5  );

Table created.
```

```
CSE-B-572@XE 14-NOV-23> CREATE TABLE department_info (
  2  d_name VARCHAR2(10) NOT NULL,
  3  building VARCHAR2(10),
  4  semester VARCHAR2(20),
  5  year NUMBER
  6  );

Table created.
```

Intersecting Values

```
CSE-B-572@XE 14-NOV-23> INSERT INTO instructor VALUES('A571','Neil',NULL);

1 row created.

CSE-B-572@XE 14-NOV-23> INSERT INTO instructor VALUES('A572','Nikitha',40000);

1 row created.

CSE-B-572@XE 14-NOV-23> INSERT INTO instructor VALUES('A573','Nikhil',50000);

1 row created.

CSE-B-572@XE 14-NOV-23> INSERT INTO instructor VALUES('A574','Namitha',NULL);

1 row created.
```

```
CSE-B-572@XE 14-NOV-23> INSERT INTO department_info VALUES('CSE','Ablock','Spring',2004);

1 row created.

CSE-B-572@XE 14-NOV-23> INSERT INTO department_info VALUES('CSM','Bblock','Fall',2005);

1 row created.

CSE-B-572@XE 14-NOV-23> INSERT INTO department_info VALUES('CSD','Ablock','Spring',2006);

1 row created.

CSE-B-572@XE 14-NOV-23> INSERT INTO department_info VALUES('EEE','Cblock','Star',2007);

1 row created.
```

Viewing the data

```
CSE-B-572@XE 14-NOV-23> select * from instructor;

IN_ID      NAME           SALARY
---------- ---------- ----------
A571       Neil
A572       Nikitha         40000
A573       Nikhil          50000
A574       Namitha
```

```
CSE-B-572@XE 14-NOV-23> select * from department_info;

D_NAME     BUILDING   SEMESTER                  YEAR
---------- ---------- -------------------- ----------
CSE        Ablock     Spring                     2004
CSM        Bblock     Fall                       2005
CSD        Ablock     Spring                     2006
EEE        Cblock     Star                       2007
```

IS NULL

```
CSE-B-572@XE 14-NOV-23> SELECT * FROM instructor WHERE salary IS NULL;

IN_ID      NAME           SALARY
---------- ---------- ----------
A571       Neil
A574       Namitha
```

BETWEEN

```
CSE-B-572@XE 14-NOV-23> SELECT * FROM instructor WHERE salary>=20000 AND salary<=50000;

IN_ID      NAME           SALARY
---------- ---------- ----------
A572       Nikitha         40000
A573       Nikhil          50000
```

LIKE

```
CSE-B-572@XE 14-NOV-23> SELECT * FROM instructor WHERE name LIKE 'N%';

IN_ID      NAME           SALARY
---------- ---------- ----------
A571       Neil
A572       Nikitha         40000
A573       Nikhil          50000
A574       Namitha
```

```
CSE-B-572@XE 14-NOV-23> SELECT * FROM instructor WHERE name LIKE '%a';

IN_ID      NAME           SALARY
---------- ---------- ----------
A572       Nikitha         40000
A574       Namitha
```

```
CSE-B-572@XE 14-NOV-23> SELECT * FROM instructor WHERE name LIKE '_______';

IN_ID      NAME           SALARY
---------- ---------- ----------
A572       Nikitha         40000
A574       Namitha
```

IN

```
CSE-B-572@XE 14-NOV-23> SELECT * FROM department_info WHERE d_name IN ('CSE','CSD','IT');

D_NAME     BUILDING   SEMESTER                  YEAR
---------- ---------- -------------------- ----------
CSE        Ablock     Spring                     2004
CSD        Ablock     Spring                     2006
```

EXPERIMENT-6

WRITE SQL QUERIES TO PERFORM JOIN OPERATIONS (CONDITIONAL JOIN, EQUI JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN)

Creating tables

```
CSE-B-572@XE 21-NOV-23> CREATE TABLE countries (
  2  country_id VARCHAR2(20) NOT NULL,
  3  c_name VARCHAR2(50),
  4  region_id NUMBER,
  5  PRIMARY KEY(country_id)
  6  );

Table created.
```

```
CSE-B-572@XE 21-NOV-23> CREATE TABLE locations_info (
  2  l_id int PRIMARY KEY,
  3  street VARCHAR2(20) NOT NULL,
  4  country_id VARCHAR2(20) NOT NULL,
  5  city VARCHAR2(20),
  6  FOREIGN KEY(country_id) REFERENCES countries(country_id)
  7  );

Table created.
```

Inserting Values

```
CSE-B-572@XE 21-NOV-23> INSERT INTO countries VALUES('AR','Argentina',3);

1 row created.

CSE-B-572@XE 21-NOV-23> INSERT INTO countries VALUES('IN','India',2);

1 row created.

CSE-B-572@XE 21-NOV-23> INSERT INTO countries VALUES('AU','Australia',1);

1 row created.

CSE-B-572@XE 21-NOV-23> INSERT INTO countries VALUES('NL','NetherLands',4);

1 row created.
```

```
CSE-B-572@XE 21-NOV-23> INSERT INTO locations_info VALUES(101,'RamNagar','IN','ATP');

1 row created.

CSE-B-572@XE 21-NOV-23> INSERT INTO locations_info VALUES(102,'GandhiNagar','IN','Banglore');

1 row created.

CSE-B-572@XE 21-NOV-23> INSERT INTO locations_info VALUES(103,'VictoriaStreet','AU','Sydney');

1 row created.

CSE-B-572@XE 21-NOV-23> INSERT INTO locations_info VALUES(104,'SomnathNagar','IN','Kadapa');

1 row created.
```

Viewing the data

```
CSE-B-572@XE 21-NOV-23> SELECT * FROM locations_info;

      L_ID STREET               COUNTRY_ID          CITY
---------- -------------------- ------------------- --------------------
       101 RamNagar             IN                  ATP
       102 GandhiNagar          IN                  Banglore
       103 VictoriaStreet       AU                  Sydney
       104 SomnathNagar         IN                  Kadapa
```

```
CSE-B-572@XE 21-NOV-23> SELECT * FROM countries;

COUNTRY_ID           C_NAME                REGION_ID
-------------------- -------------------- ----------
AR                   Argentina                     3
IN                   India                         2
AU                   Australia                     1
NL                   NetherLands                   4
```

CONDITIONAL JOIN

```
CSE-B-572@XE 21-NOV-23> SELECT *
  2  FROM locations_info join countries
  3  ON
  4  locations_info.country_id = countries.country_id;
```

```
CSE-B-572@XE 21-NOV-23> SELECT * FROM
  2  locations_info natural LEFT OUTER JOIN countries;

no rows selected
```

EXPERIMENT-7

WRITE SQL QUERIES TO PERFORM AGGREGATE OPERATIONS (SUM, COUNT, AVG, MIN, MAX)

Creating tables

```
CSE-B-572@XE 21-NOV-23> CREATE TABLE employee_info (
  2  e_id int PRIMARY KEY,
  3  e_name VARCHAR2(10) NOT NULL,
  4  salary NUMBER(9,2),
  5  branch_name VARCHAR2(15) NOT NULL
  6  );

Table created.
```

```
CSE-B-572@XE 21-NOV-23> select  * from countries
  2  natural RIGHT OUTER JOIN locations_info;

no rows selected
```

Inserting values

```
CSE-B-572@XE 21-NOV-23> INSERT INTO employee_info VALUES(571,'Priya',90000,'CSE');

1 row created.

CSE-B-572@XE 21-NOV-23> INSERT INTO employee_info VALUES(572,'Anirudh',95000,'CSD');

1 row created.

CSE-B-572@XE 21-NOV-23> INSERT INTO employee_info VALUES(573,'Ishikha',80000,'CSE');

1 row created.

CSE-B-572@XE 21-NOV-23> INSERT INTO employee_info VALUES(573,'Dev',85000,'CSD');
INSERT INTO employee_info VALUES(573,'Dev',85000,'CSD')
*
ERROR at line 1:
ORA-00001: unique constraint (SYSTEM.SYS_C008417) violated


CSE-B-572@XE 21-NOV-23> INSERT INTO employee_info VALUES(574,'Dev',85000,'CSD');

1 row created.
```

Viewing the data

```
CSE-B-572@XE 21-NOV-23> select * from employee_info;

      E_ID E_NAME         SALARY BRANCH_NAME
---------- ---------- ---------- --------------
       571 Priya           90000 CSE
       572 Anirudh         95000 CSD
       573 Ishikha         80000 CSE
       574 Dev             85000 CSD
```

SUM

To find salary(sum of salaries) of CSE branch

```
CSE-B-572@XE 21-NOV-23> SELECT SUM(salary) as salary
  2  FROM employee_info
  3  WHERE
  4  branch_name = 'CSE';

    SALARY
----------
    170000
```

To find salary (sum of salaries) of CSD branch

```
CSE-B-572@XE 21-NOV-23> SELECT SUM(salary) as salary
  2  FROM employee_info
  3  WHERE
  4  branch_name = 'CSD';

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

COUNT

To find number of employees in the company

```
CSE-B-572@XE 21-NOV-23> SELECT COUNT(e_id)
  2  FROM employee_info;

COUNT(E_ID)
-----------
          4
```

To find the number of employees in each branch

```
CSE-B-572@XE 21-NOV-23> SELECT COUNT(e_id) as CSE_emp
  2  FROM employee_info
  3  WHERE
  4  branch_name = 'CSE';

   CSE_EMP
----------
         2

CSE-B-572@XE 21-NOV-23> SELECT COUNT(e_id) as CSD_emp
  2  FROM employee_info
  3  WHERE
  4  branch_name = 'CSD';

   CSD_EMP
----------
         2
```

AVERAGE

To find average salary of CSE branch

```
CSE-B-572@XE 21-NOV-23> SELECT avg(salary) as avg_salary
  2  FROM employee_info
  3  WHERE
  4  branch_name = 'CSE';

AVG_SALARY
----------
     85000
```

To find average salary of CSD branch

```
CSE-B-572@XE 21-NOV-23> SELECT avg(salary) as avg_salary
  2  FROM employee_info
  3  WHERE
  4  branch_name = 'CSD';

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

MIN

```
CSE-B-572@XE 21-NOV-23> SELECT MIN(salary) as min_salary
  2  FROM employee_info
  3  WHERE
  4  branch_name = 'CSD';

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

CSE-B-572@XE 21-NOV-23> SELECT MIN(salary) as min_salary
  2  FROM employee_info
  3  WHERE
  4  branch_name = 'CSE';

MIN_SALARY
----------
     80000
```

MAX

```
CSE-B-572@XE 21-NOV-23> SELECT MAX(salary) as max_salary
  2  FROM employee_info
  3  WHERE
  4  branch_name = 'CSE';

MAX_SALARY
----------
     90000

CSE-B-572@XE 21-NOV-23> SELECT MAX(salary) as max_salary
  2  FROM employee_info
  3  WHERE
  4  branch_name = 'CSD';

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

EXPERIMENT-8

WRITE SQL QUERIES TO PERFORM ORACLE BUILT IN FUNCTIONS (DATE, TIME)

DATE FUNCTIONS

```
CSE-B-572@XE 30-JAN-24> SELECT ADD_MONTHS(SYSDATE,+2) FROM DUAL;

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

CSE-B-572@XE 30-JAN-24> SELECT ADD_MONTHS(SYSDATE,-2) FROM DUAL;

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

CSE-B-572@XE 30-JAN-24> SELECT SYSDATE
  2  FROM DUAL;

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

CSE-B-572@XE 30-JAN-24> SELECT SYSDATE+10 FROM DUAL;

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

CSE-B-572@XE 30-JAN-24> SELECT SYSDATE-10 FROM DUAL;

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

```
CSE-B-572@XE 30-JAN-24> SELECT LAST_DAY(SYSDATE) FROM DUAL;

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

```
CSE-B-572@XE 30-JAN-24> SELECT NEXT_DAY(SYSDATE,'MONDAY') FROM DUAL;

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

CSE-B-572@XE 30-JAN-24> SELECT NEXT_DAY(SYSDATE,'FRIDAY') FROM DUAL;

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

TIME FUNCTIONS

```
CSE-B-572@XE 30-JAN-24> SELECT
  2  CURRENT_TIMESTAMP(3)
  3  FROM
  4  DUAL;

CURRENT_TIMESTAMP(3)
---------------------------------------------------------------------------
30-JAN-24 03.42.18.836 PM +05:30

CSE-B-572@XE 30-JAN-24> SELECT MONTHS_BETWEEN('27-NOV-24','05-NOV-23')
  2  FROM DUAL;

MONTHS_BETWEEN('27-NOV-24','05-NOV-23')
---------------------------------------
                             12.7096774

CSE-B-572@XE 30-JAN-24> SELECT MONTHS_BETWEEN('27-NOV-24','27-NOV-23')
  2  FROM DUAL;

MONTHS_BETWEEN('27-NOV-24','27-NOV-23')
---------------------------------------
                                     12

CSE-B-572@XE 30-JAN-24> SELECT MONTHS_BETWEEN('27-NOV-23','27-NOV-24')
  2  FROM DUAL;

MONTHS_BETWEEN('27-NOV-23','27-NOV-24')
---------------------------------------
                                    -12
```

EXPERIMENT-9

Write SQL Queries to perform KEY CONSTRAINTS(PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, CHECK, DEFAULT)

PRIMARY KEY

```
CSE-B-572@XE 28-NOV-23> CREATE TABLE country (
  2  c_id VARCHAR2(10) NOT NULL,
  3  c_name VARCHAR2(50),
  4  region_id NUMBER,
  5  PRIMARY KEY(c_id)
  6  );

Table created.

CSE-B-572@XE 28-NOV-23> INSERT INTO country VALUES('IN','India',1);

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO country VALUES('AR','Argentina',3);

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO country VALUES('AU','Australia',2);

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO country VALUES('NL','NetherLand',4);

1 row created.

CSE-B-572@XE 28-NOV-23> SELECT *  FROM country;

C_ID       C_NAME                                             REGION_ID
---------- -------------------------------------------------- ----------
IN         India                                                       1
AR         Argentina                                                   3
AU         Australia                                                   2
NL         NetherLand                                                  4
```

FOREIGN KEY

```
CSE-B-572@XE 28-NOV-23> CREATE TABLE locations (
  2  l_id int PRIMARY KEY,
  3  street VARCHAr2(50) NOT NULL,
  4  c_id VARCHAR2(10) NOT NULL,
  5  city VARCHAR2(20),
  6  FOREIGN KEY(c_id) REFERENCES country(c_id)
  7  );

Table created.

CSE-B-572@XE 28-NOV-23> INSERT INTO locations VALUES(2001,'GandhiNagar','IN','ATP');

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO locations VALUES(2002,'RamNagar','IN','Banglore');

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO locations VALUES(2003,'VictoriaStreet','AU','Sydney');

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO locations VALUES(2004,'PieterBaugh','NL','NetherLands');

1 row created.

CSE-B-572@XE 28-NOV-23> select * from locations;
```

```
CSE-B-572@XE 28-NOV-23> select * from locations;

      L_ID STREET                                            C_ID
---------- -------------------------------------------------- ----------
CITY
--------------------
      2001 GandhiNagar                                       IN
ATP

      2002 RamNagar                                          IN
Banglore

      2003 VictoriaStreet                                    AU
Sydney


      L_ID STREET                                            C_ID
---------- -------------------------------------------------- ----------
CITY
--------------------
      2004 PieterBaugh                                       NL
NetherLands
```

```
CSE-B-572@XE 28-NOV-23> COLUMN COMPANY_NAME l_id A10
SP2-0158: unknown COLUMN option "l_id"
CSE-B-572@XE 28-NOV-23> COLUMN l_id FORMAT A10
CSE-B-572@XE 28-NOV-23> COLUMN street FORMAT A10
CSE-B-572@XE 28-NOV-23> COLUMN c_id FORMAT A10
CSE-B-572@XE 28-NOV-23> SELECT  *  FROM locations;

      L_ID STREET     C_ID       CITY
---------- ---------- ---------- --------------------
########## GandhiNaga IN         ATP
           r

########## RamNagar   IN         Banglore
########## VictoriaSt AU         Sydney
           reet

########## PieterBaug NL         NetherLands
           h
```

UNIQUE KEY

```
CSE-B-572@XE 28-NOV-23> CREATE TABLE student (
  2  s_id INT UNIQUE,
  3  name VARCHAR2(10),
  4  branch VARCHAR2(10)
  5  );

Table created.

CSE-B-572@XE 28-NOV-23> INSERT INTO student VALUES(571,'Pranavi','CSE');

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO student VALUES(572,'Aditya','CSM');

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO student VALUES(573,'Arman','EEE');

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO student VALUES(574,'Abhira','ECE');

1 row created.

CSE-B-572@XE 28-NOV-23> select * from student;

      S_ID NAME       BRANCH
---------- ---------- ----------
       571 Pranavi    CSE
       572 Aditya     CSM
       573 Arman      EEE
       574 Abhira     ECE
```

NOT NULL

```
CSE-B-572@XE 28-NOV-23> CREATE TABLE emplyoee (
  2  e_name VARCHAR2(20) NOT NULL,
  3  e_id int NOT NULL,
  4  age int
  5  );

Table created.
```

```
CSE-B-572@XE 28-NOV-23> INSERT INTO emplyoee VALUES('Dev',204,24);

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO emplyoee VALUES('Sonakshi',205,23);

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO emplyoee VALUES('Anurag',206,26);

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO emplyoee VALUES('Prerana',207,26);

1 row created.

CSE-B-572@XE 28-NOV-23> select  *  from emplyoee;

E_NAME                    E_ID        AGE
-------------------- ---------- ----------
Dev                         204         24
Sonakshi                    205         23
Anurag                      206         26
Prerana                     207         26
```

CHECK

```
CSE-B-572@XE 28-NOV-23> CREATE TABLE staff (
  2  s_id int PRIMARY KEY,
  3  s_name VARCHAR2(10),
  4  salary int CHECK(salary<=90000)
  5  );

Table created.

CSE-B-572@XE 28-NOV-23> INSERT INTO staff VALUES(100,'Siddarth',85000);

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO staff VALUES(101,'Hasini',80000);

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO staff VALUES(103,'Karthik',75000);

1 row created.
```

```
CSE-B-572@XE 28-NOV-23> select * from staff;

      S_ID S_NAME         SALARY
---------- ---------- ----------
       100 Siddarth        85000
       101 Hasini          80000
       103 Karthik         75000
```

DEFAULT

```
CSE-B-572@XE 28-NOV-23> CREATE TABLE emp (
  2  id int PRIMARY KEY,
  3  name VARCHAR2(20),
  4  salary NUMERIC(9,2) DEFAULT '0'
  5  );

Table created.

CSE-B-572@XE 28-NOV-23> INSERT INTO emp VALUES(301,'Harish',80000);

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO emp VALUES(302,'Swathi',89000);

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO emp VALUES(303,'Harika',90000);

1 row created.

CSE-B-572@XE 28-NOV-23> INSERT INTO emp(id,name) VALUES(304,'Sarika');

1 row created.

CSE-B-572@XE 28-NOV-23> select * from emp;

        ID NAME                      SALARY
---------- -------------------- ----------
       301 Harish                    80000
       302 Swathi                    89000
       303 Harika                    90000
       304 Sarika                        0
```

EXPERIMENT-10

WRITE A PL/SQL PROGRAM FOR CALCULATING THE FACTORIAL OF A GIVEN NUMBER

```
CSE-B-572@XE 28-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  /

PL/SQL procedure successfully completed.
```

```
CSE-B-572@XE 28-NOV-23> set serveroutput on;
CSE-B-572@XE 28-NOV-23> /
3628800

PL/SQL procedure successfully completed.
```

EXPERIMENT-11

WRITE A PL/SQL PROGRAM FOR FINDING THE GIVEN NUMBER

IS PRIME OR NOT

```
CSE-B-572@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.
```

EXPERIMENT-12

WRITE A PL/SQL PROGRAM FOR DISPLAYING THE FIBONACCI SERIES UPTO AN INTEGER

```
CSE-B-572@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  /

PL/SQL procedure successfully completed.

CSE-B-572@XE 05-DEC-23> set serveroutput on;
CSE-B-572@XE 05-DEC-23> /
SERIES:
0
1
1
2
3
5

PL/SQL procedure successfully completed.
```

EXPERIMENT-13

WRITE A PL/SQL PROGRAM TO IMPLEMENT STORED PROCEDURE ON TABLE

```
CSE-B-572@XE 05-DEC-23> CREATE TABLE sailor_info (
  2  ID NUMBER(10) PRIMARY KEY,
  3  NAME VARCHAR2(10)
  4  );

Table created.
```

PROCEDURE CREATION

```
CSE-B-572@XE 05-DEC-23> CREATE OR REPLACE PROCEDURE INSERTUSER
  2  (ID IN NUMBER,
  3  NAME IN VARCHAR2)
  4  IS
  5  BEGIN
  6  INSERT INTO sailor_info VALUES(ID,NAME);
  7  DBMS_OUTPUT.PUT_LINE('RECORD INSERTED SUCCESSFULLY');
  8  END;
  9  /

Procedure created.
```

EXECUTION PROCEDURE

```
CSE-B-572@XE 05-DEC-23> DECLARE
  2  CNT NUMBER;
  3  BEGIN
  4  INSERTUSER(101,'NARASIMHA');
  5  SELECT COUNT(*) INTO CNT FROM sailor_info;
  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.
```

DROP PROCEDURE

```
CSE-B-572@XE 05-DEC-23> DROP PROCEDURE INSERTUSER;

Procedure dropped.
```

EXPERIMENT-14

WRITE A PL/SQL PROGRAM TO IMPLEMENT STORED FUNCTION ON TABLE

FUNCTION CREATION

```
CSE-B-572@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.
```

EXECUTION PROCEDURE

```
CSE-B-572@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.
```

DROP FUNCTION

```
CSE-B-572@XE 05-DEC-23> DROP FUNCTION ADDER;

Function dropped.
```

RECURSIVE FUNCTION

```
CSE-B-572@XE 5-DEC-23> run
  1  CREATE FUNCTION factorial(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 * factorial(x-1);
 10  END IF;
 11  RETURN f;
 12* END;

Function created.
```

EXECUTION PROCEDURE

```
  8* END;
CSE-B-572@XE 5-DEC-23> run
  1  DECLARE
  2  num number;
  3  factorial1 number;
  4  BEGIN
  5  num:= 6;
  6  factorial1 := factorial(num);
  7  dbms_output.put_line(' Factorial1 '|| num || ' is ' || factorial1);
  8* END;

PL/SQL procedure successfully completed.
```

DROP FUNCTION

```
CSE-B-572@XE 5-DEC-23> DROP FUNCTION factorial;

Function dropped.
```

EXPERIMENT-15

WRITE A PL/SQL PROGRAM TO IMPLEMENT TRIGGER ON TABLE

CREATING TABLES

```
CSE-B-572@XE 12-DEC-23> CREATE TABLE instructor (
  2  id VARCHAR2(5),
  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) ON DELETE SET NULL
  8  );

Table created.
```

```
CSE-B-572@XE 12-DEC-23> CREATE TABLE department (
  2  dept_name VARCHAR2(20),
  3  building VARCHAR2(15),
  4  budget NUMERIC(12,2) CHECK(budget>0),
  5  PRIMARY KEY(dept_name)
  6  );

Table created.
```

INSERTING VALUES

```
CSE-B-572@XE 12-DEC-23> INSERT INTO department VALUES('CSE','Yashna',95000);

1 row created.

CSE-B-572@XE 12-DEC-23> INSERT INTO department VALUES('ECE','Yashna',85000);

1 row created.

CSE-B-572@XE 12-DEC-23> INSERT INTO department VALUES('EEE','Viraj',80000);

1 row created.

CSE-B-572@XE 12-DEC-23> INSERT INTO department VALUES('CSM','Viraj',80000);

1 row created.
```

```
CSE-B-572@XE 12-DEC-23> INSERT INTO instructor VALUES(101,'Priya','CSE',95000);

1 row created.

CSE-B-572@XE 12-DEC-23> INSERT INTO instructor VALUES(102,'Anirudh','ECE',90000);

1 row created.

CSE-B-572@XE 12-DEC-23> INSERT INTO instructor VALUES(103,'Kavya','EEE',89000);

1 row created.

CSE-B-572@XE 12-DEC-23> INSERT INTO instructor VALUES(104,'Abhay','CSM',88000);

1 row created.
```

CREATION OF TRIGGER

```
CSE-B-572@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.
```

EXECUTION OF TRIGGER

```
CSE-B-572@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  /

PL/SQL procedure successfully completed.
```

OUTPUT

```
CSE-B-572@XE 12-DEC-23> set serveroutput on;
CSE-B-572@XE 12-DEC-23> /
Old salary: 100000
New salary: 105000
Salary difference: 5000
Old salary: 95000
New salary: 100000
Salary difference: 5000
Old salary: 94000
New salary: 99000
Salary difference: 5000
Old salary: 93000
New salary: 98000
Salary difference: 5000
4 instructors updated

PL/SQL procedure successfully completed.
```

EXPERIMENT-16

WRITE A PL/SQL PROGRAM TO IMPLEMENT CURSOR ON TABLE

CREATING TABLES

```
CSE-B-572@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.
```

INSERTING VALUES

```
CSE-B-572@XE 12-DEC-23> INSERT INTO customers VALUES(1,'Priya',25,'SkullRock',29000);

1 row created.

CSE-B-572@XE 12-DEC-23> INSERT INTO customers VALUES(2,'Kavya',24,'Upsidedown',31000);

1 row created.

CSE-B-572@XE 12-DEC-23> INSERT INTO customers VALUES(3,'Viraj',21,'Hawkins',33000);

1 row created.

CSE-B-572@XE 12-DEC-23> INSERT INTO customers VALUES(4,'Anirudh',23,'Indiana',35000);

1 row created.
```

CREATE PROCEDURE

```
CSE-B-572@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-572@XE 12-DEC-23> SELECT * FROM customers;

        ID NAME                      AGE ADDRESS                SALARY
---------- -------------------- ---------- -------------------- ----------
         1 Priya                        25 SkullRock                 34000
         2 Kavya                        24 Upsidedown                36000
         3 Viraj                        21 Hawkins                   38000
         4 Anirudh                      23 Indiana                   40000
```

PROGRAM USING EXPLICIT CURSORS