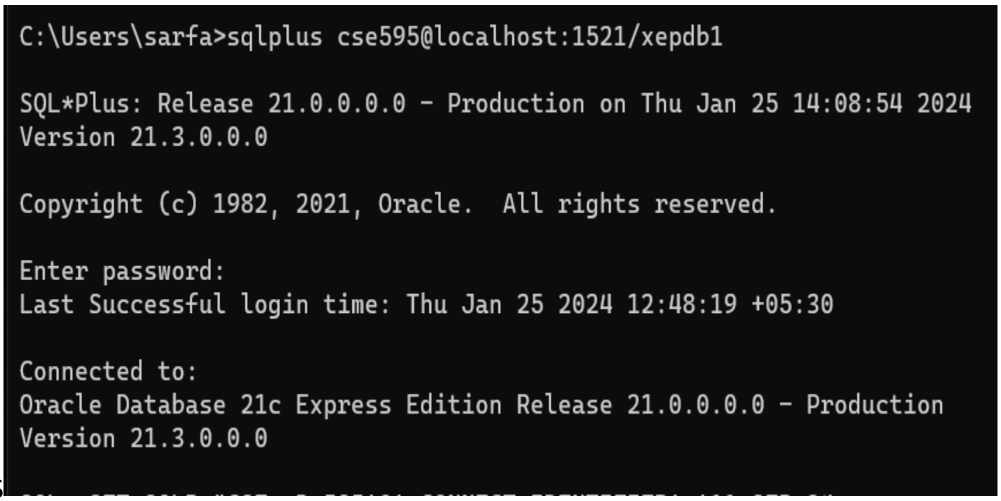
EXPERIMENT-1

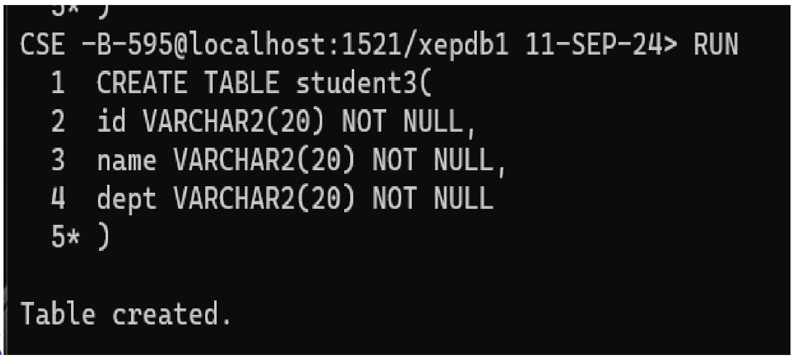
WRITE SQL QUERIES TO CREATE TABLES FOR VARIOUS DATABASES USING DDL COMMANDS (CREATE, ALTER, DROP, TRUNCATE)



```
C:\Users\sarfa>sqlplus cse595@localhost:1521/xepdb1

SQL*Plus: Release 21.0.0.0.0 - Production on Thu Jan 25 14:08:54 2024
Version 21.3.0.0.0

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

Enter password:
Last Successful login time: Thu Jan 25 2024 12:48:19 +05:30

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

S

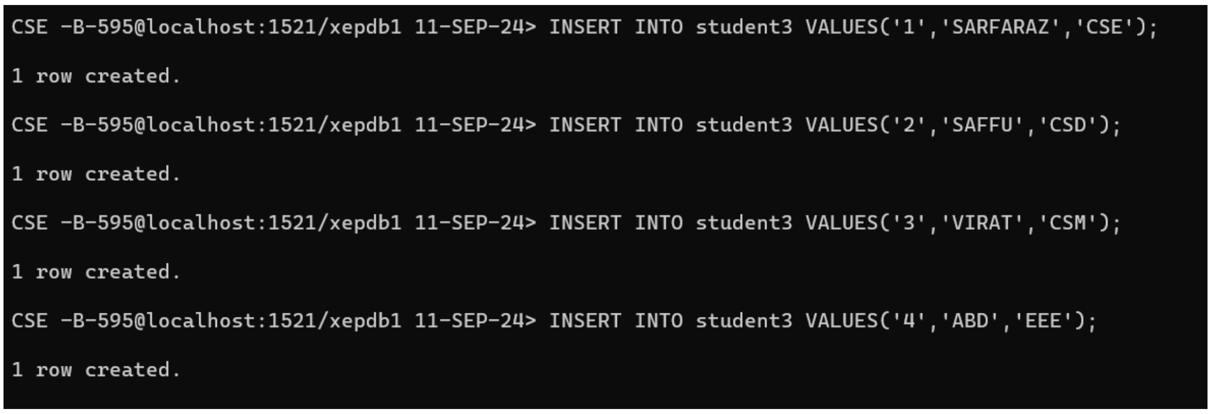
CREATE TABLE:



```
CSE -B-595@localhost:1521/xepdb1 11-SEP-24> RUN
  1  CREATE TABLE student3(
  2  id VARCHAR2(20) NOT NULL,
  3  name VARCHAR2(20) NOT NULL,
  4  dept VARCHAR2(20) NOT NULL
  5* )

Table created.
```

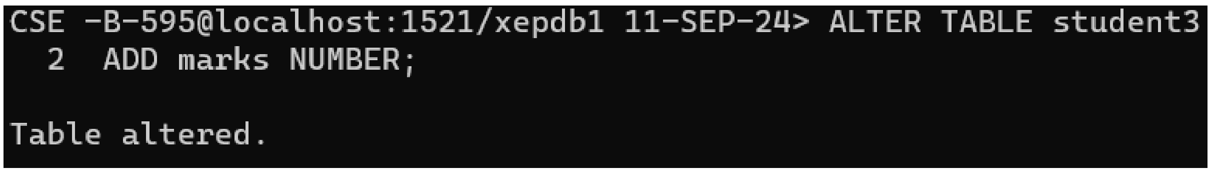
R

INSERTING VALUES INTO TABLE:



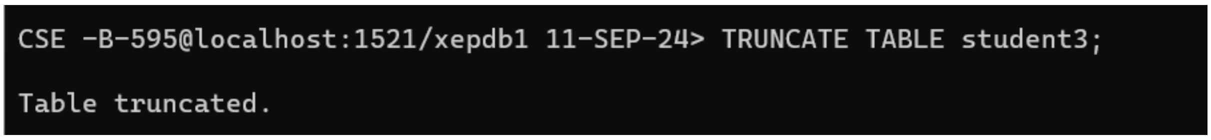
```
CSE -B-595@localhost:1521/xepdb1 11-SEP-24> INSERT INTO student3 VALUES('1','SARFARAZ','CSE');

1 row created.

CSE -B-595@localhost:1521/xepdb1 11-SEP-24> INSERT INTO student3 VALUES('2','SAFFU','CSD');

1 row created.

CSE -B-595@localhost:1521/xepdb1 11-SEP-24> INSERT INTO student3 VALUES('3','VIRAT','CSM');

1 row created.

CSE -B-595@localhost:1521/xepdb1 11-SEP-24> INSERT INTO student3 VALUES('4','ABD','EEE');

1 row created.
```

ALTER TABLE:



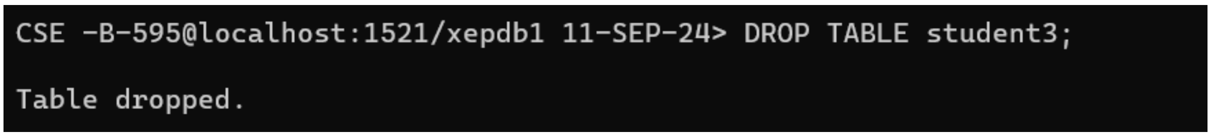
```
CSE -B-595@localhost:1521/xepdb1 11-SEP-24> ALTER TABLE student3
  2  ADD marks NUMBER;

Table altered.
```

TABLE TRUNCATED:



```
CSE -B-595@localhost:1521/xepdb1 11-SEP-24> TRUNCATE TABLE student3;

Table truncated.
```

TABLE DROPPED:

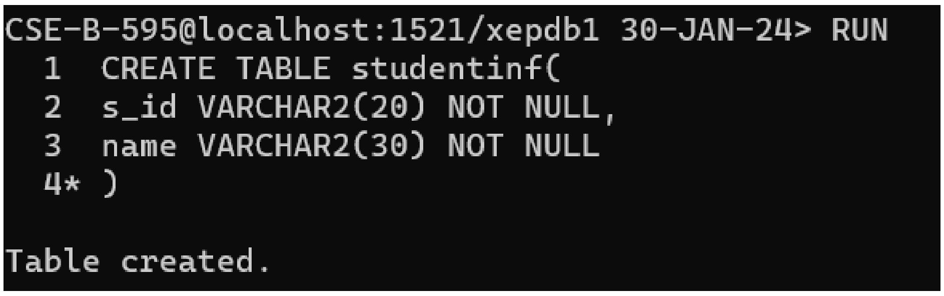


```
CSE -B-595@localhost:1521/xepdb1 11-SEP-24> DROP TABLE student3;

Table dropped.
```

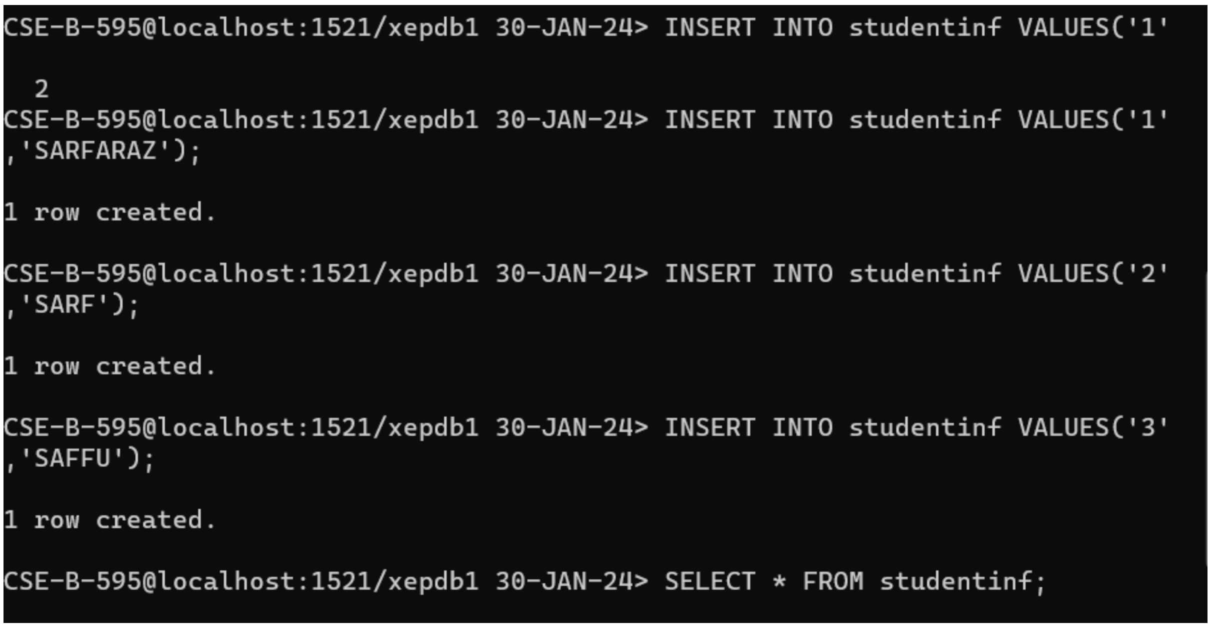
CONCLUSION: **THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.**

EXPERIMENT-2

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

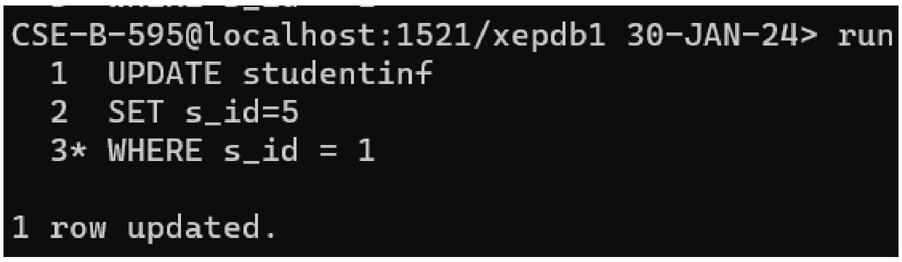


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> RUN
  1  CREATE TABLE studentinf(
  2  s_id VARCHAR2(20) NOT NULL,
  3  name VARCHAR2(30) NOT NULL
  4* )

Table created.
```



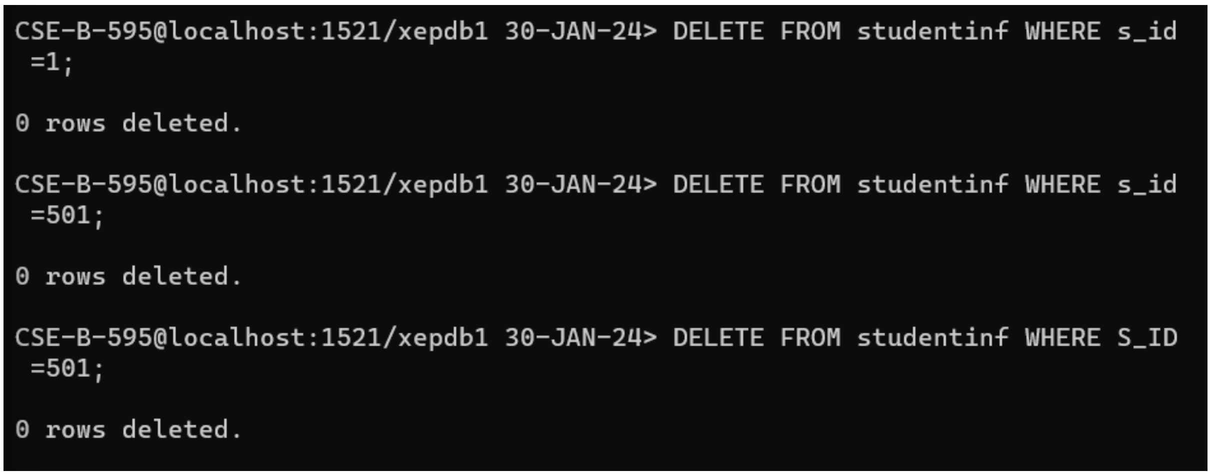
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> INSERT INTO studentinf VALUES('1'

  2
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> INSERT INTO studentinf VALUES('1'
,'SARFARAZ');

1 row created.

CSE-B-595@localhost:1521/xepdb1 30-JAN-24> INSERT INTO studentinf VALUES('2'
,'SARF');

1 row created.

CSE-B-595@localhost:1521/xepdb1 30-JAN-24> INSERT INTO studentinf VALUES('3'
,'SAFFU');

1 row created.

CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM studentinf;
```

Update



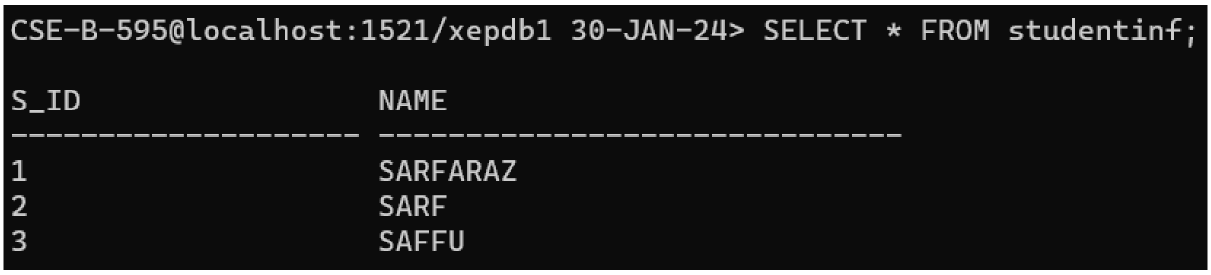
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> run
  1  UPDATE studentinf
  2  SET s_id=5
  3* WHERE s_id = 1

1 row updated.
```

DELETE



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> DELETE FROM studentinf WHERE s_id
 =1;

0 rows deleted.

CSE-B-595@localhost:1521/xepdb1 30-JAN-24> DELETE FROM studentinf WHERE s_id
 =501;

0 rows deleted.

CSE-B-595@localhost:1521/xepdb1 30-JAN-24> DELETE FROM studentinf WHERE S_ID
 =501;

0 rows deleted.
```

SELECT



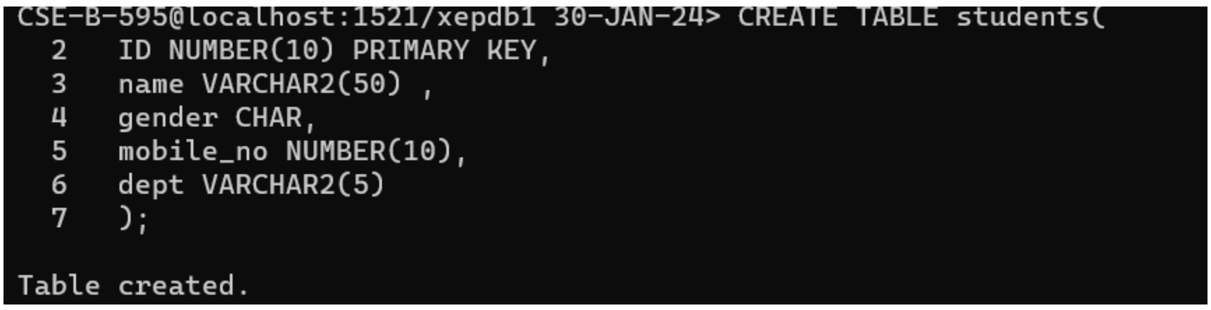
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM studentinf;

S_ID                 NAME
-------------------- ------------------------------
1                    SARFARAZ
2                    SARF
3                    SAFFU
```

CONCLUSION: THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.

EXPERIMENT-3

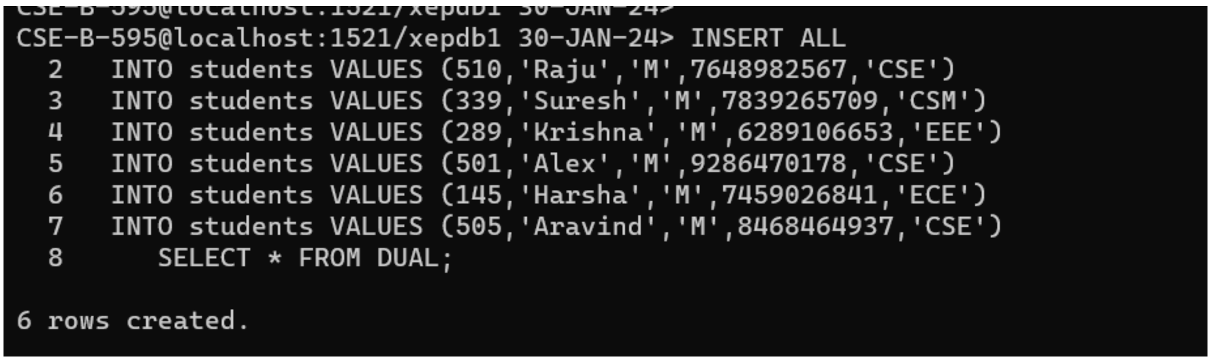
WRITE SQL QUERIES TO VIEWS FOR VARIOUS DATABASES (CREATE VIEW, ALTER VIEW, AND DELETE VIEW)

Table creation



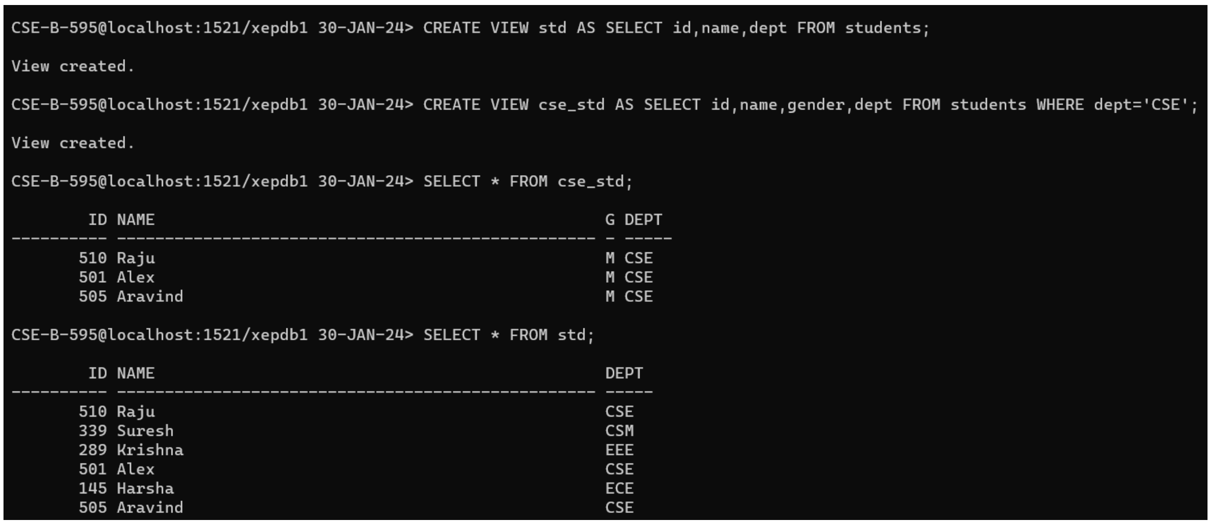
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> CREATE TABLE students(
     ID NUMBER(10) PRIMARY KEY,
     name VARCHAR2(50) ,
     gender CHAR,
     mobile_no NUMBER(10),
     dept VARCHAR2(5)
     );

Table created.
```

Inserting Values



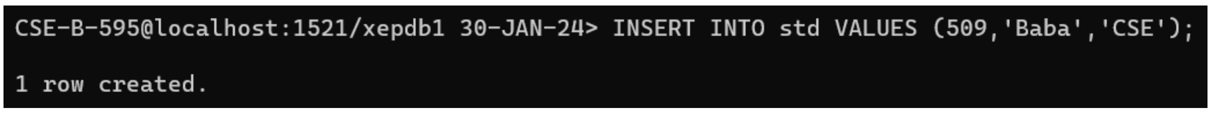
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24>
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> INSERT ALL
     INTO students VALUES (510,'Raju','M',7648982567,'CSE')
     INTO students VALUES (339,'Suresh','M',7839265709,'CSM')
     INTO students VALUES (289,'Krishna','M',6289106653,'EEE')
     INTO students VALUES (501,'Alex','M',9286470178,'CSE')
     INTO students VALUES (145,'Harsha','M',7459026841,'ECE')
     INTO students VALUES (505,'Aravind','M',8468464937,'CSE')
        SELECT * FROM DUAL;

 6 rows created.
```

Creating View



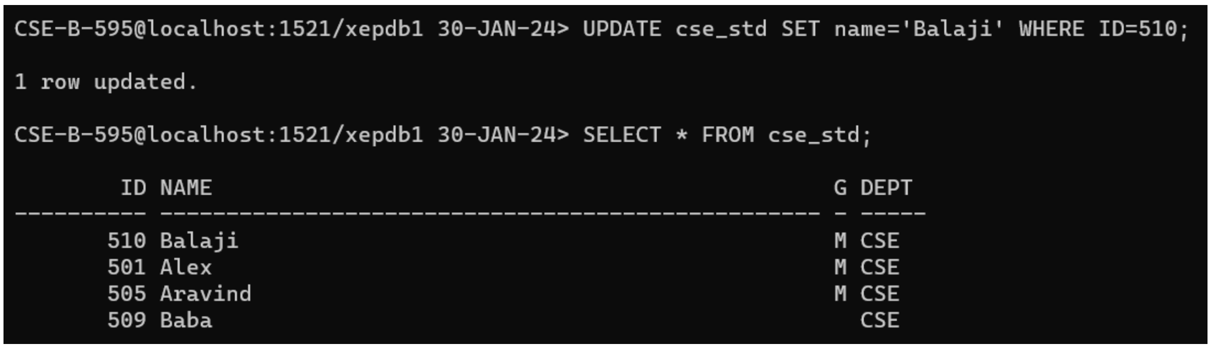
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> CREATE VIEW std AS SELECT id,name,dept FROM students;

View created.

CSE-B-595@localhost:1521/xepdb1 30-JAN-24> CREATE VIEW cse_std AS SELECT id,name,gender,dept FROM students WHERE dept='CSE';

View created.

CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM cse_std;

        ID NAME                                               G DEPT
---------- -------------------------------------------------- - -----
       510 Raju                                               M CSE
       501 Alex                                               M CSE
       505 Aravind                                            M CSE

CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM std;

        ID NAME                                               DEPT
---------- -------------------------------------------------- -----
       510 Raju                                               CSE
       339 Suresh                                             CSM
       289 Krishna                                            EEE
       501 Alex                                               CSE
       145 Harsha                                             ECE
       505 Aravind                                            CSE
```

Inserting Values into VIEWS



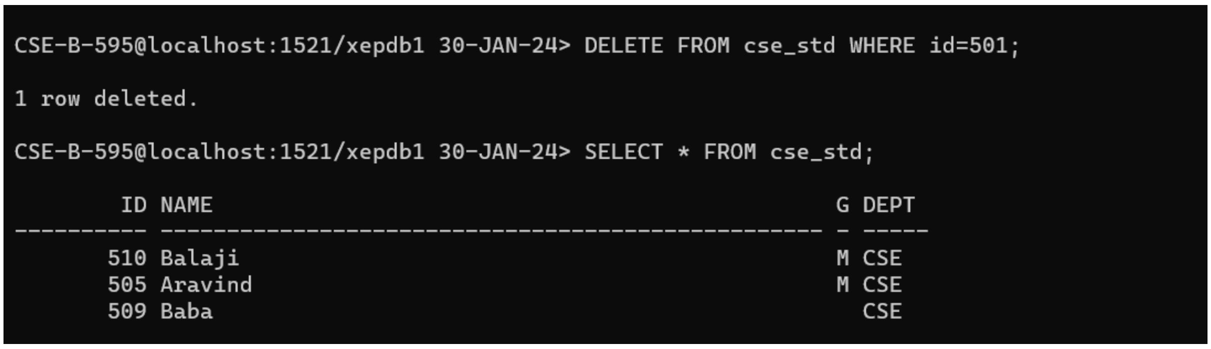
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> INSERT INTO std VALUES (509,'Baba','CSE');

1 row created.
```

Update VIEWS



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> UPDATE cse_std SET name='Balaji' WHERE ID=510;

1 row updated.

CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM cse_std;

        ID NAME                                               G DEPT
---------- -------------------------------------------------- - -----
       510 Balaji                                             M CSE
       501 Alex                                               M CSE
       505 Aravind                                            M CSE
       509 Baba                                                 CSE
```

DELETE VIEWS



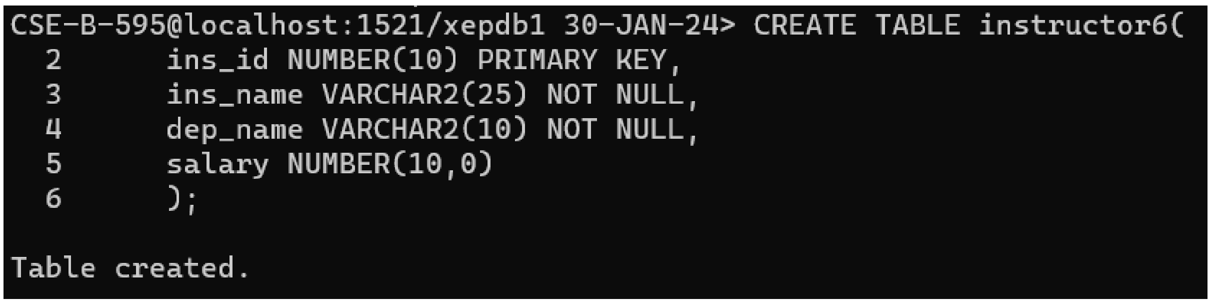
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> DELETE FROM cse_std WHERE id=501;

1 row deleted.

CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM cse_std;

        ID NAME                                               G DEPT
---------- -------------------------------------------------- - -----
       510 Balaji                                             M CSE
       505 Aravind                                            M CSE
       509 Baba                                                 CSE
```

CONCLUSION: THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.

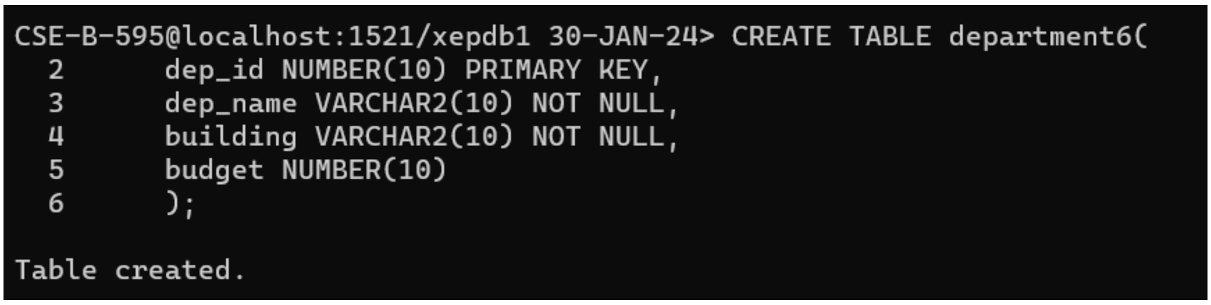
EXPERINMENT-4

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

Creating tables

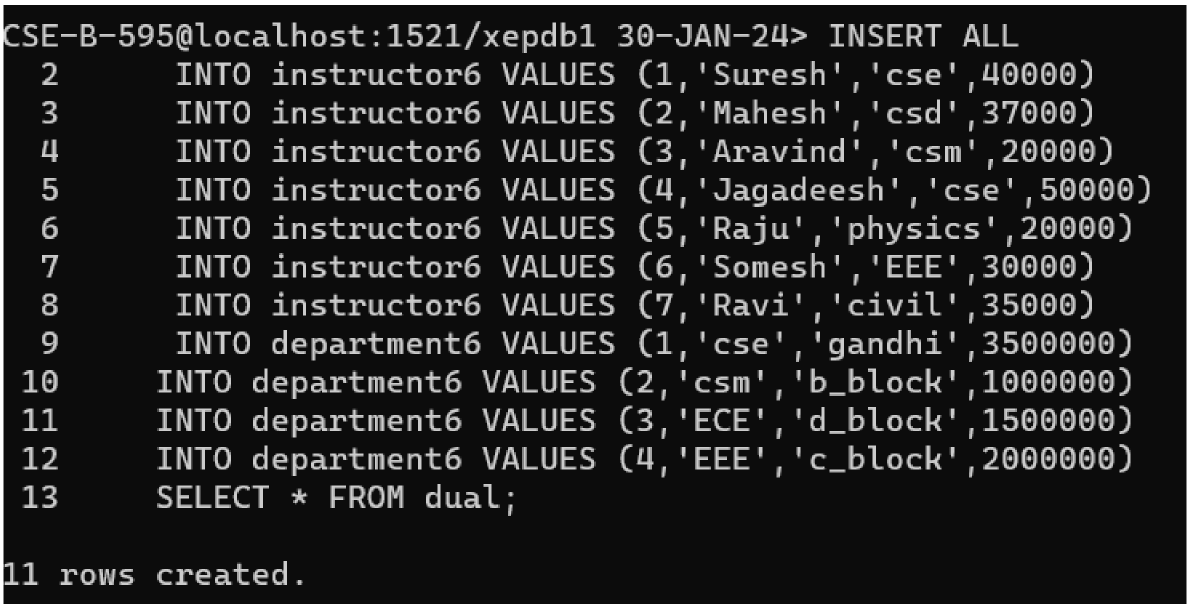


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> CREATE TABLE instructor6(
        ins_id NUMBER(10) PRIMARY KEY,
        ins_name VARCHAR2(25) NOT NULL,
        dep_name VARCHAR2(10) NOT NULL,
        salary NUMBER(10,0)
        );

Table created.
```

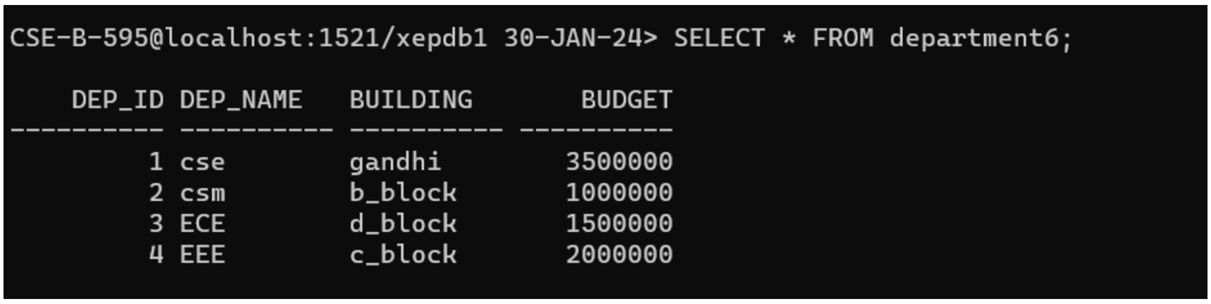


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> CREATE TABLE department6(
        dep_id NUMBER(10) PRIMARY KEY,
        dep_name VARCHAR2(10) NOT NULL,
        building VARCHAR2(10) NOT NULL,
        budget NUMBER(10)
        );

Table created.
```

Inserting Values

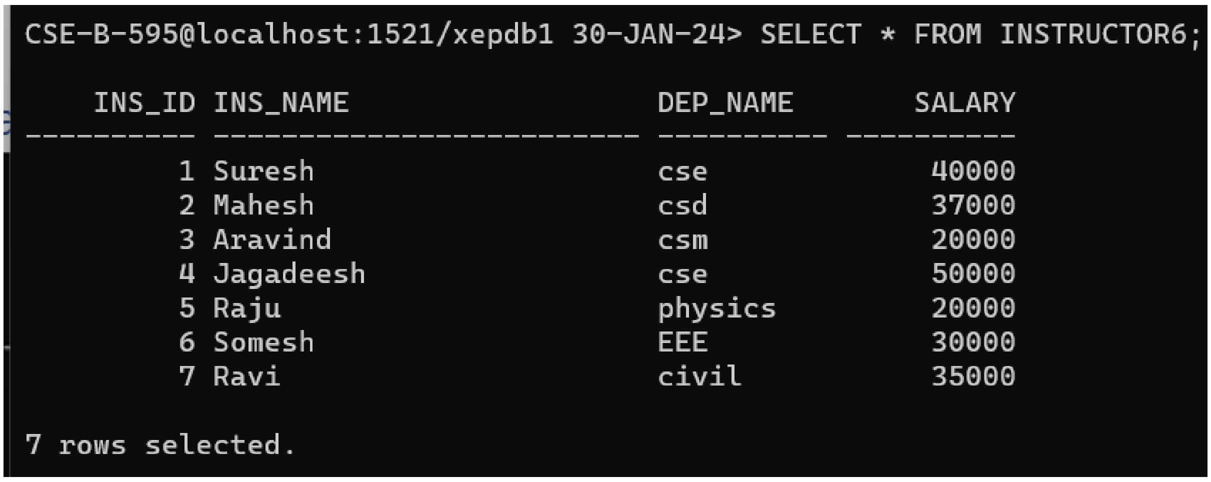


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> INSERT ALL
         INTO instructor6 VALUES (1,'Suresh','cse',40000)
         INTO instructor6 VALUES (2,'Mahesh','csd',37000)
         INTO instructor6 VALUES (3,'Aravind','csm',20000)
         INTO instructor6 VALUES (4,'Jagadeesh','cse',50000)
         INTO instructor6 VALUES (5,'Raju','physics',20000)
         INTO instructor6 VALUES (6,'Somesh','EEE',30000)
         INTO instructor6 VALUES (7,'Ravi','civil',35000)
         INTO department6 VALUES (1,'cse','gandhi',3500000)
        INTO department6 VALUES (2,'csm','b_block',1000000)
        INTO department6 VALUES (3,'ECE','d_block',1500000)
        INTO department6 VALUES (4,'EEE','c_block',2000000)
        SELECT * FROM dual;

11 rows created.
```

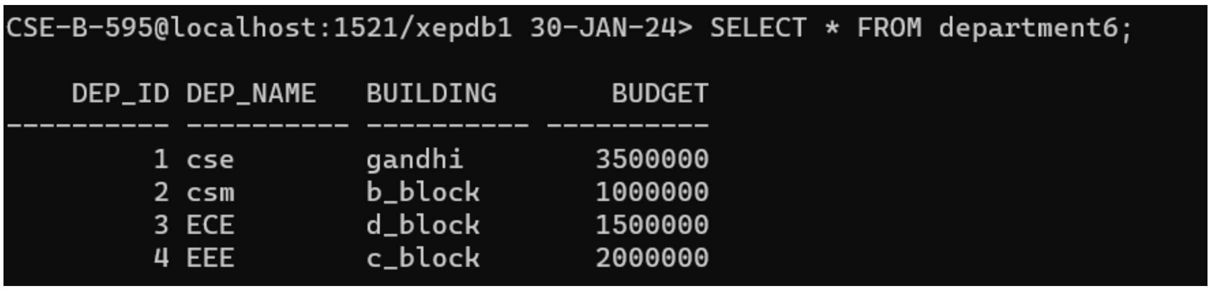


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM department6;

    DEP_ID DEP_NAME   BUILDING       BUDGET
---------- ---------- ---------- ----------
         1 cse        gandhi        3500000
         2 csm        b_block       1000000
         3 ECE        d_block       1500000
         4 EEE        c_block       2000000
```

View the data

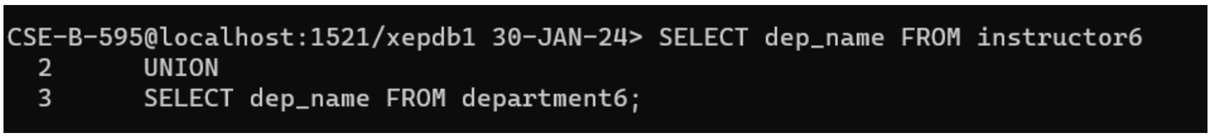


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM INSTRUCTOR6;

    INS_ID INS_NAME                  DEP_NAME       SALARY
---------- ------------------------- ---------- ----------
         1 Suresh                    cse             40000
         2 Mahesh                    csd             37000
         3 Aravind                   csm             20000
         4 Jagadeesh                 cse             50000
         5 Raju                      physics         20000
         6 Somesh                    EEE             30000
         7 Ravi                      civil           35000

7 rows selected.
```



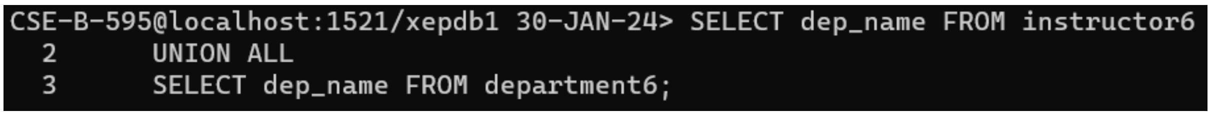
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM department6;

    DEP_ID DEP_NAME   BUILDING       BUDGET
---------- ---------- ---------- ----------
         1 cse        gandhi        3500000
         2 csm        b_block       1000000
         3 ECE        d_block       1500000
         4 EEE        c_block       2000000
```

UNION operation



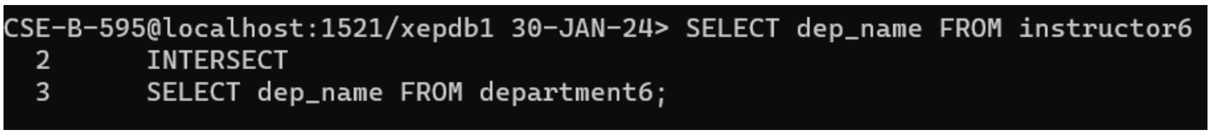
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT dep_name FROM instructor6
  2      UNION
  3      SELECT dep_name FROM department6;
```

UNION ALL operation



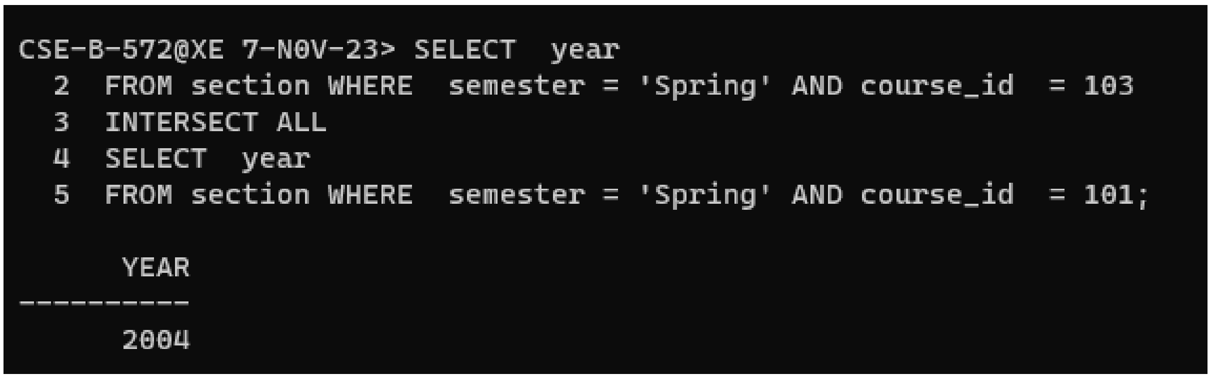
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT dep_name FROM instructor6
  2      UNION ALL
  3      SELECT dep_name FROM department6;
```

INTERSECT operation



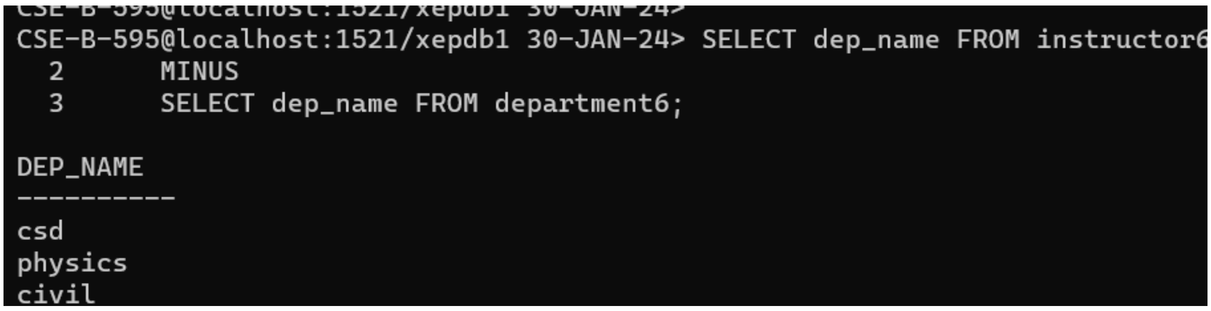
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT dep_name FROM instructor6
  2      INTERSECT
  3      SELECT dep_name FROM department6;
```

INTERSECT ALL operation



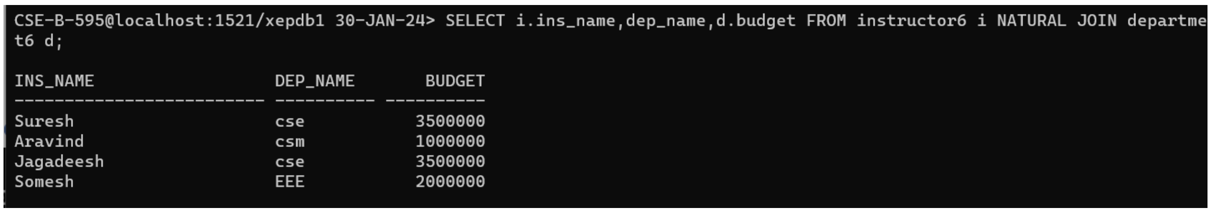
```
CSE-B-572@XE 7-NOV-23> SELECT  year
  2  FROM section WHERE  semester = 'Spring' AND course_id  = 103
  3  INTERSECT ALL
  4  SELECT  year
  5  FROM section WHERE  semester = 'Spring' AND course_id  = 101;

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

MINUS operation



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24>
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT dep_name FROM instructor6
  2         MINUS
  3         SELECT dep_name FROM department6;

DEP_NAME
----------
csd
physics
civil
```

MINUS ALL operation



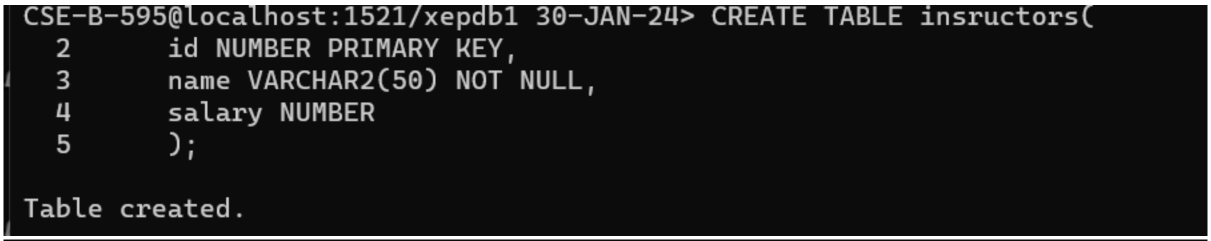
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT i.ins_name,dep_name,d.budget FROM instructor6 i NATURAL JOIN departme
t6 d;

INS_NAME                  DEP_NAME       BUDGET
------------------------- ---------- ----------
Suresh                    cse           3500000
Aravind                   csm           1000000
Jagadeesh                 cse           3500000
Somesh                    EEE           2000000
```

CONCLUSION: THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.

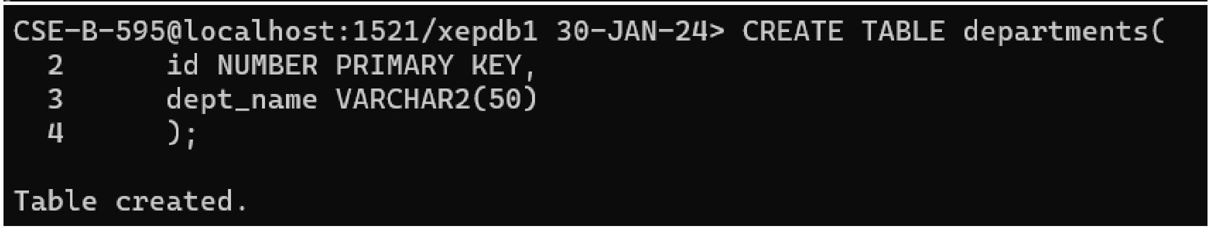
EXPERIMENT -5

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

Creating Tables

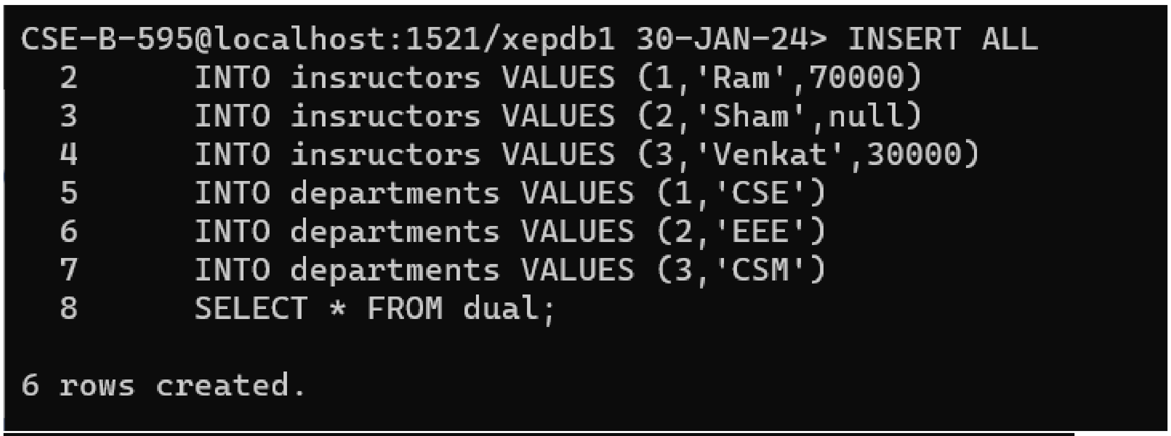


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> CREATE TABLE insructors(
        id NUMBER PRIMARY KEY,
        name VARCHAR2(50) NOT NULL,
        salary NUMBER
        );

Table created.
```

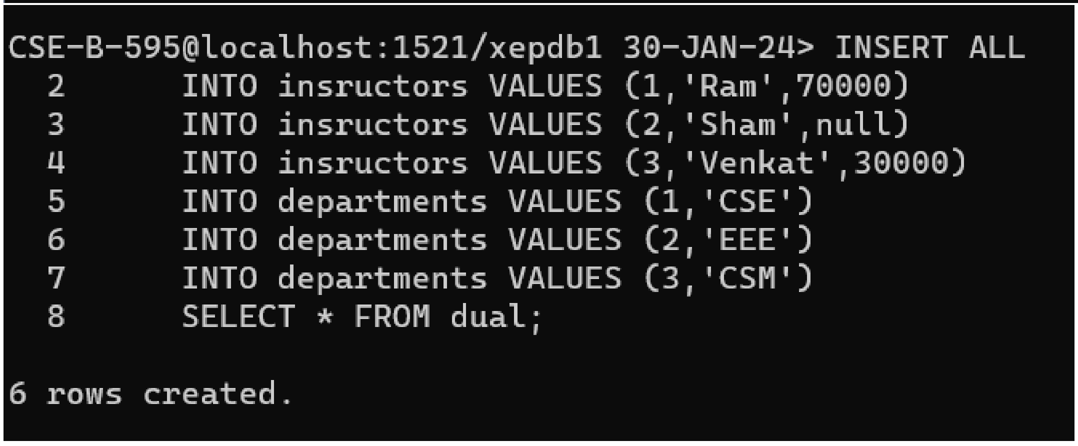


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> CREATE TABLE departments(
        id NUMBER PRIMARY KEY,
        dept_name VARCHAR2(50)
        );

Table created.
```

Intersecting Values

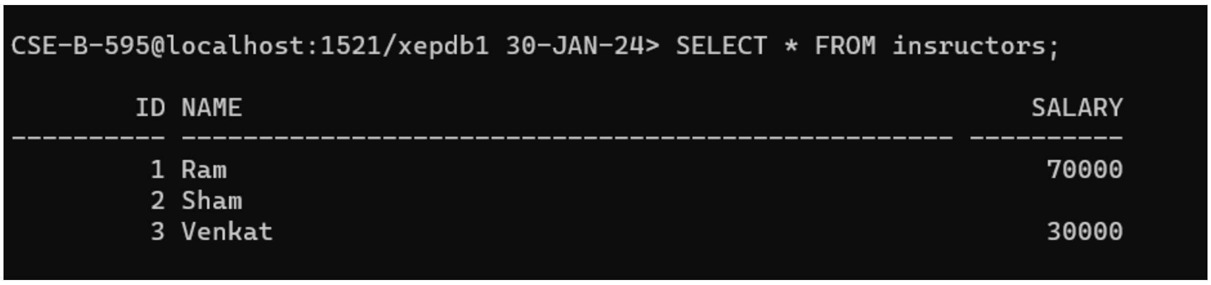


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> INSERT ALL
        INTO insructors VALUES (1,'Ram',70000)
        INTO insructors VALUES (2,'Sham',null)
        INTO insructors VALUES (3,'Venkat',30000)
        INTO departments VALUES (1,'CSE')
        INTO departments VALUES (2,'EEE')
        INTO departments VALUES (3,'CSM')
        SELECT * FROM dual;

6 rows created.
```

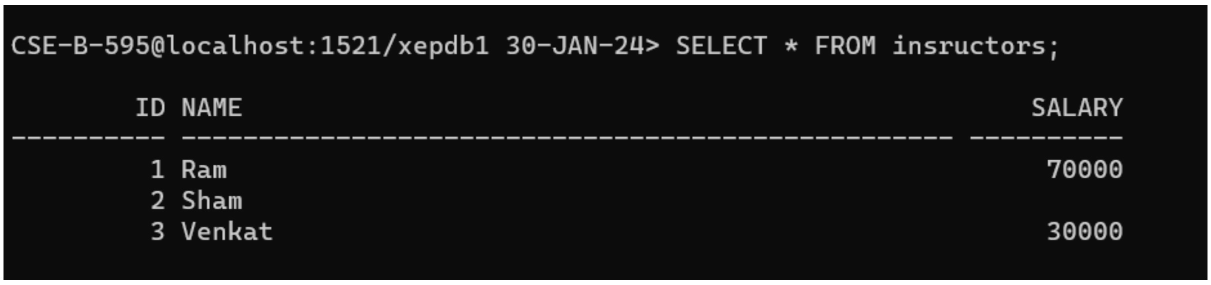


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> INSERT ALL
        INTO insructors VALUES (1,'Ram',70000)
        INTO insructors VALUES (2,'Sham',null)
        INTO insructors VALUES (3,'Venkat',30000)
        INTO departments VALUES (1,'CSE')
        INTO departments VALUES (2,'EEE')
        INTO departments VALUES (3,'CSM')
        SELECT * FROM dual;

6 rows created.
```

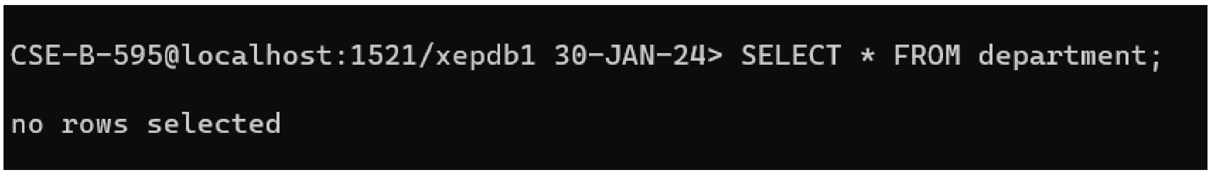
Viewing the data



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM insructors;

        ID NAME                                                   SALARY
---------- -------------------------------------------------- ----------
         1 Ram                                                     70000
         2 Sham
         3 Venkat                                                  30000
```

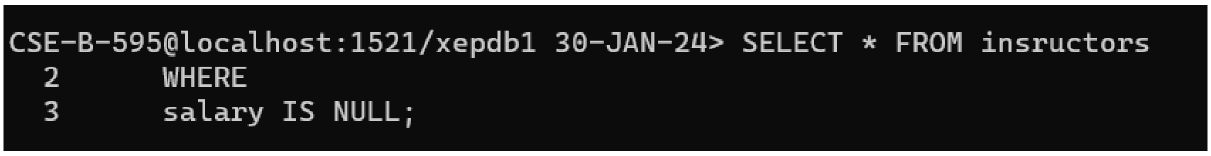


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM insructors;

        ID NAME                                                   SALARY
---------- -------------------------------------------------- ----------
         1 Ram                                                     70000
         2 Sham
         3 Venkat                                                  30000
```



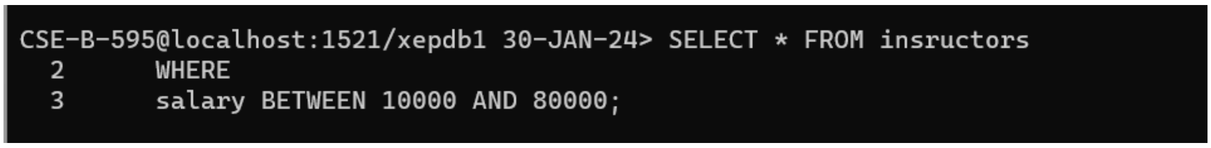
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM department;

no rows selected
```

IS NULL



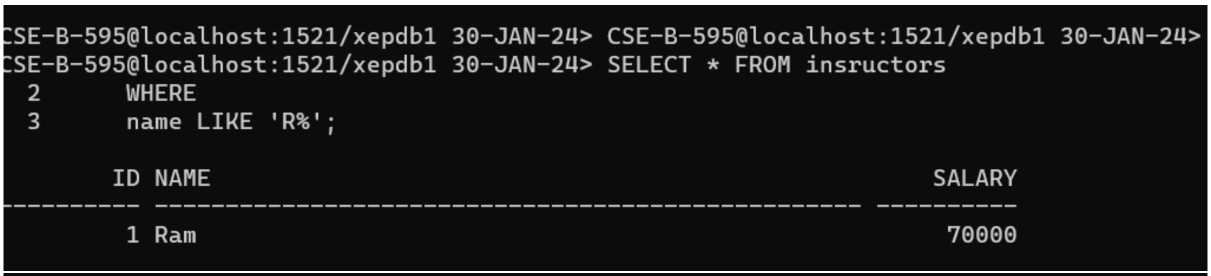
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM insructors
  2       WHERE
  3       salary IS NULL;
```

BETWEEN

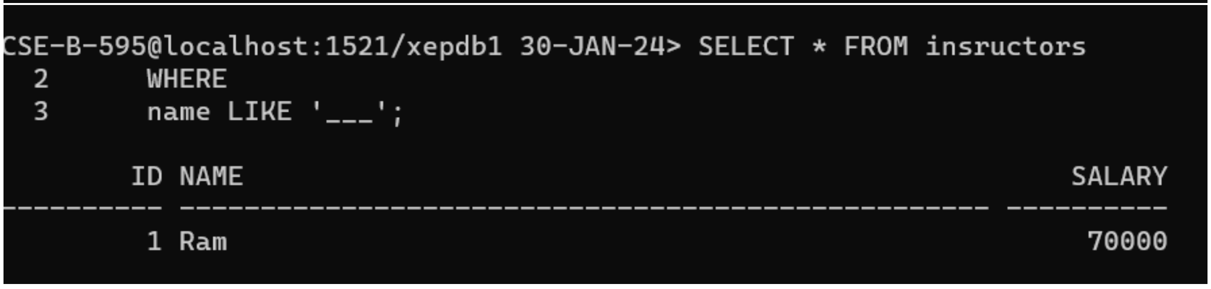


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM insructors
  2       WHERE
  3       salary BETWEEN 10000 AND 80000;
```

LIKE

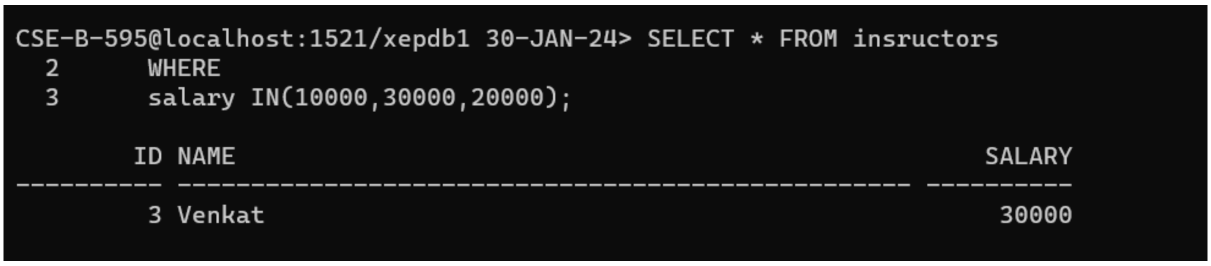


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> CSE-B-595@localhost:1521/xepdb1 30-JAN-24>
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM insructors
  2       WHERE
  3       name LIKE 'R%';

        ID NAME                                                   SALARY
---------- -------------------------------------------------- ----------
         1 Ram                                                     70000
```



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM insructors
  2       WHERE
  3       name LIKE '___';

        ID NAME                                                   SALARY
---------- -------------------------------------------------- ----------
         1 Ram                                                     70000
```

IN



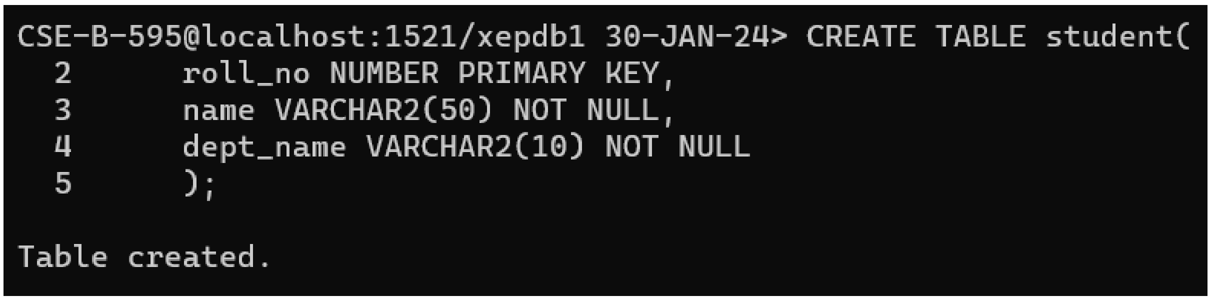
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM insructors
        WHERE
        salary IN(10000,30000,20000);

        ID NAME                                                   SALARY
---------- -------------------------------------------------- ----------
         3 Venkat                                                  30000
```

CONCLUSION: THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.

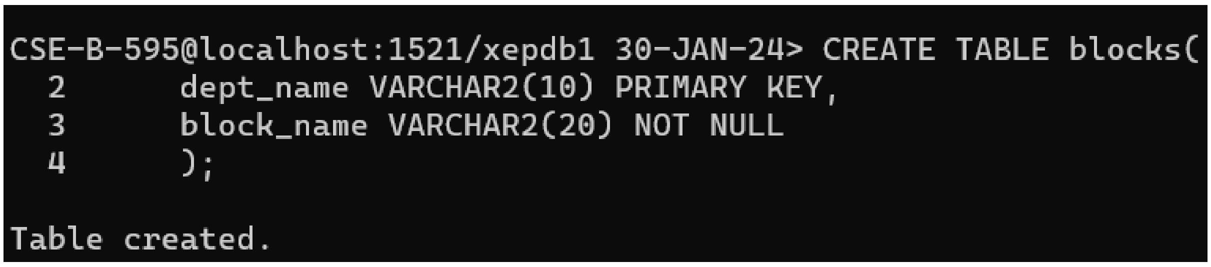
EXPERIMENT-6

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

Creating tables

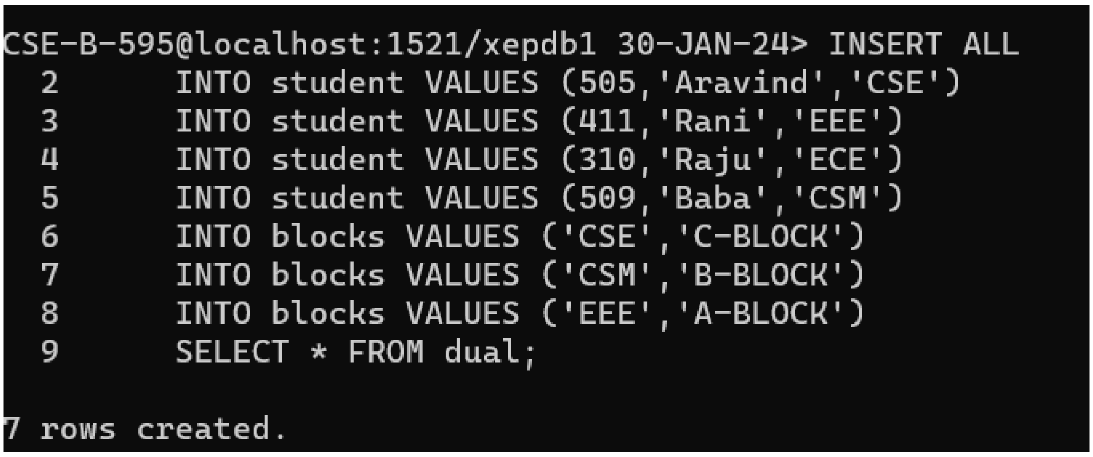


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> CREATE TABLE student(
        roll_no NUMBER PRIMARY KEY,
        name VARCHAR2(50) NOT NULL,
        dept_name VARCHAR2(10) NOT NULL
        );

Table created.
```



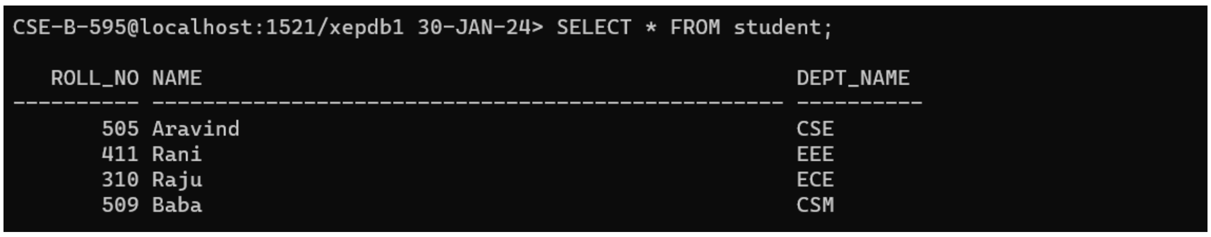
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> CREATE TABLE blocks(
        dept_name VARCHAR2(10) PRIMARY KEY,
        block_name VARCHAR2(20) NOT NULL
        );

Table created.
```

Inserting Values

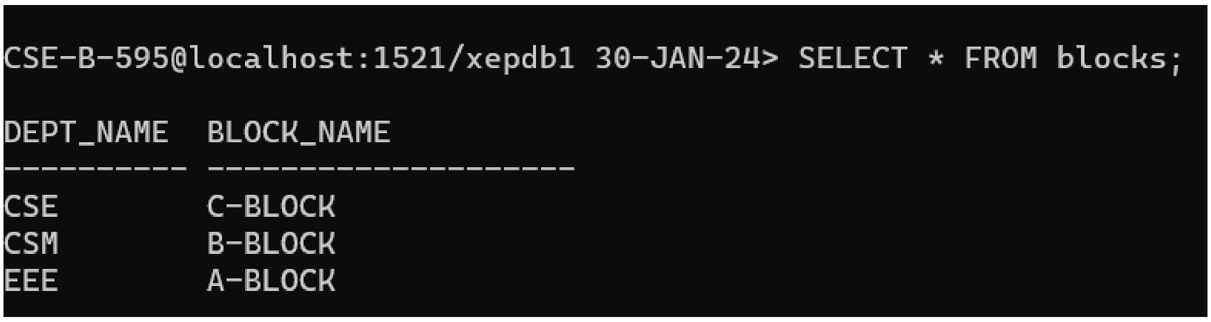


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> INSERT ALL
         INTO student VALUES (505,'Aravind','CSE')
         INTO student VALUES (411,'Rani','EEE')
         INTO student VALUES (310,'Raju','ECE')
         INTO student VALUES (509,'Baba','CSM')
         INTO blocks VALUES ('CSE','C-BLOCK')
         INTO blocks VALUES ('CSM','B-BLOCK')
         INTO blocks VALUES ('EEE','A-BLOCK')
         SELECT * FROM dual;

7 rows created.
```

Viewing the data

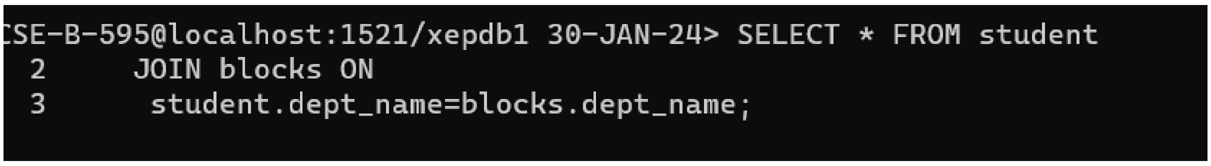


```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM student;

   ROLL_NO NAME                                               DEPT_NAME
---------- -------------------------------------------------- ----------
       505 Aravind                                            CSE
       411 Rani                                               EEE
       310 Raju                                               ECE
       509 Baba                                               CSM
```



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM blocks;

DEPT_NAME  BLOCK_NAME
---------- --------------------
CSE        C-BLOCK
CSM        B-BLOCK
EEE        A-BLOCK
```

CONDITIONAL JOIN



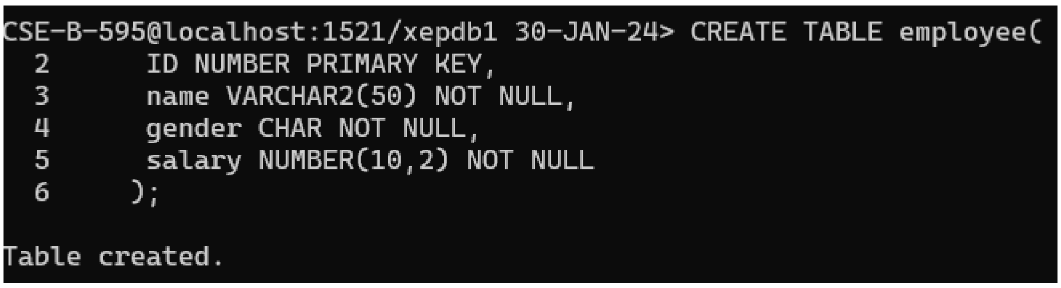
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM student
       JOIN blocks ON
        student.dept_name=blocks.dept_name;
```

CONCLUSION: THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.

EXPERIMENT-7

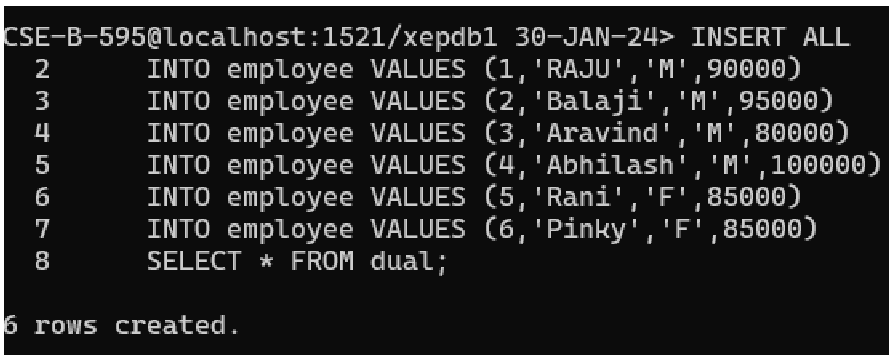
WRITE SQL QUERIES TO PERFORM AGGREGATE OPERATIONS (SUM, COUNT, AVG, MIN, MAX)

Creating tables



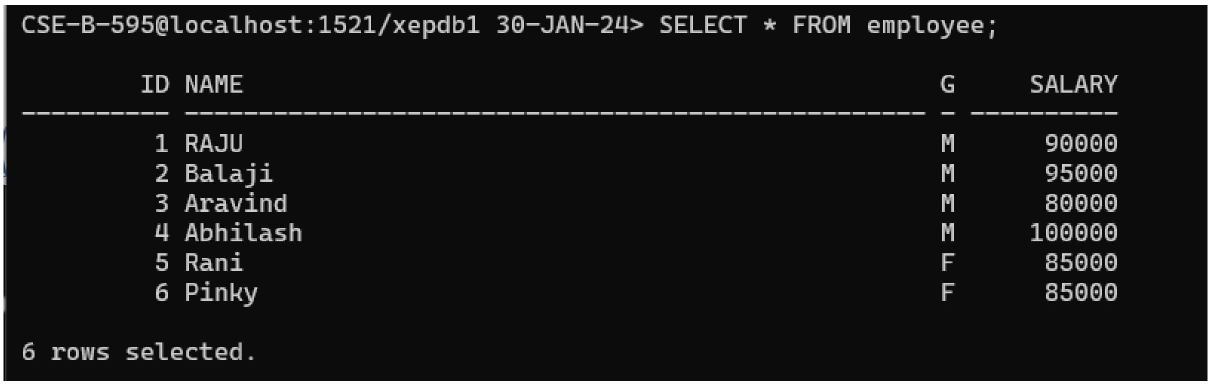
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> CREATE TABLE employee(
         ID NUMBER PRIMARY KEY,
         name VARCHAR2(50) NOT NULL,
         gender CHAR NOT NULL,
         salary NUMBER(10,2) NOT NULL
        );

Table created.
```

Inserting values



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> INSERT ALL
         INTO employee VALUES (1,'RAJU','M',90000)
         INTO employee VALUES (2,'Balaji','M',95000)
         INTO employee VALUES (3,'Aravind','M',80000)
         INTO employee VALUES (4,'Abhilash','M',100000)
         INTO employee VALUES (5,'Rani','F',85000)
         INTO employee VALUES (6,'Pinky','F',85000)
         SELECT * FROM dual;

6 rows created.
```

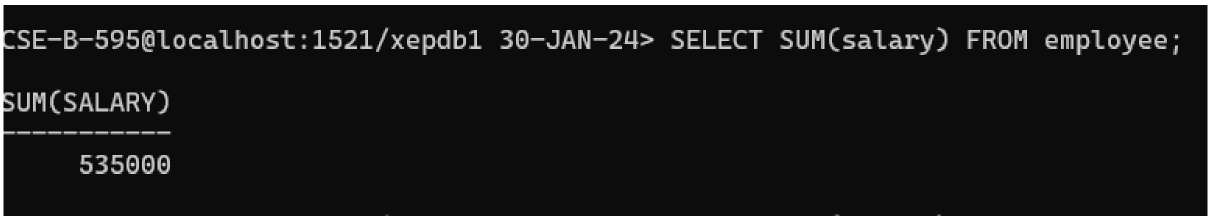
Viewing the data



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT * FROM employee;

        ID NAME                                               G     SALARY
---------- -------------------------------------------------- - ----------
         1 RAJU                                               M      90000
         2 Balaji                                             M      95000
         3 Aravind                                            M      80000
         4 Abhilash                                           M     100000
         5 Rani                                               F      85000
         6 Pinky                                              F      85000

6 rows selected.
```

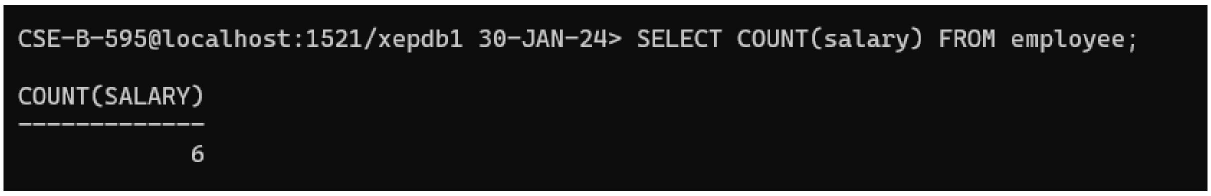
SUM

To find salary (sum of salaries):



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT SUM(salary) FROM employee;

SUM(SALARY)
-----------
     535000
```

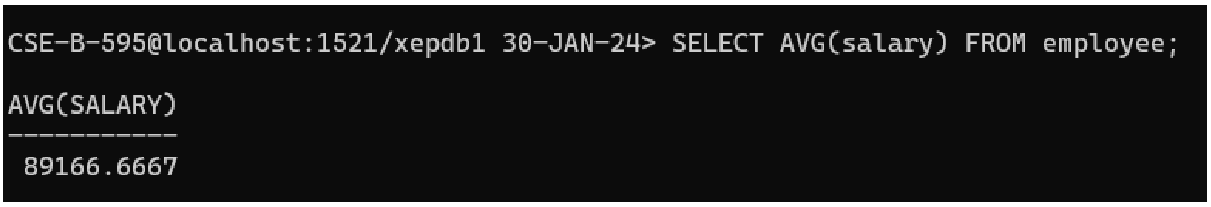
COUNT



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT COUNT(salary) FROM employee;

COUNT(SALARY)
-------------
            6
```

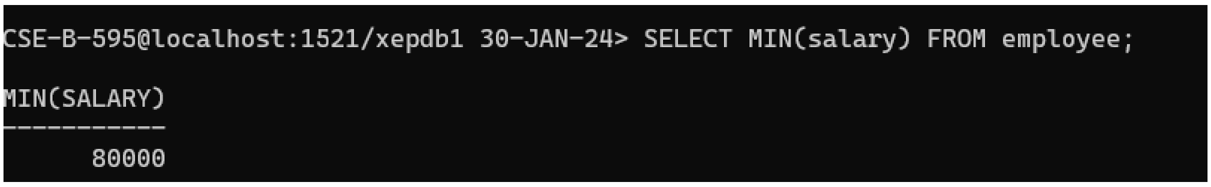
AVERAGE

To find average:



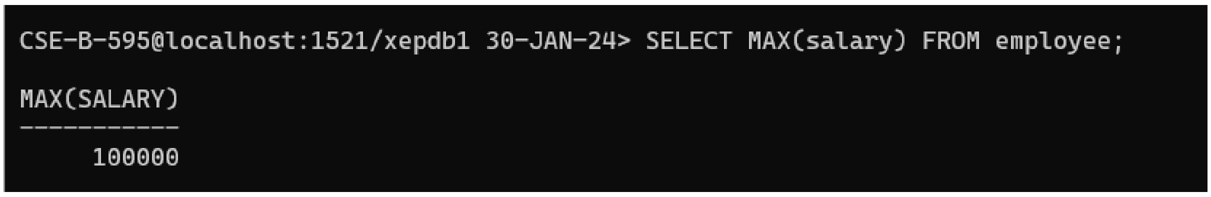
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT AVG(salary) FROM employee;

AVG(SALARY)
-----------
 89166.6667
```

MIN



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT MIN(salary) FROM employee;

MIN(SALARY)
-----------
      80000
```

MAX



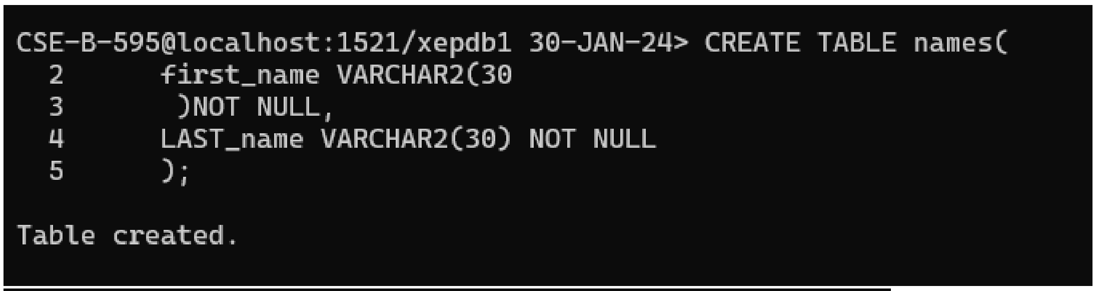
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT MAX(salary) FROM employee;

MAX(SALARY)
-----------
     100000
```

CONCLUSION: THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.

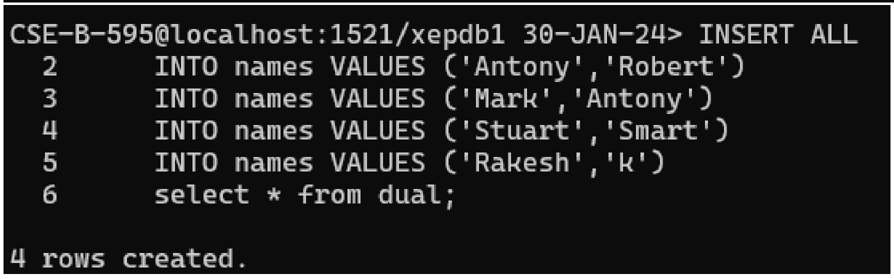
EXPERIMENT-8

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

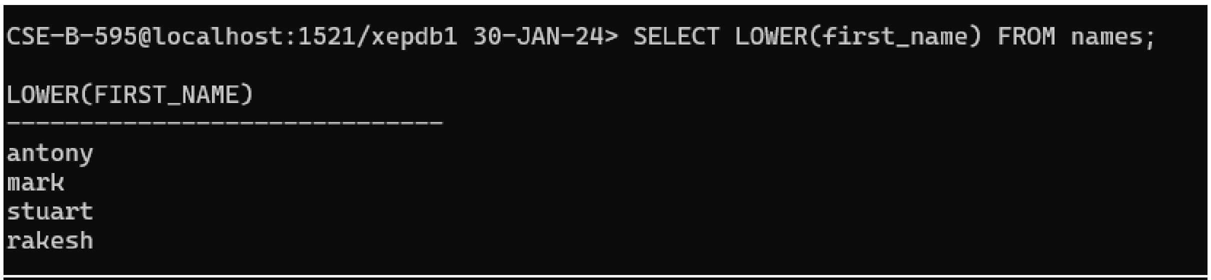
DATE FUNCTIONS



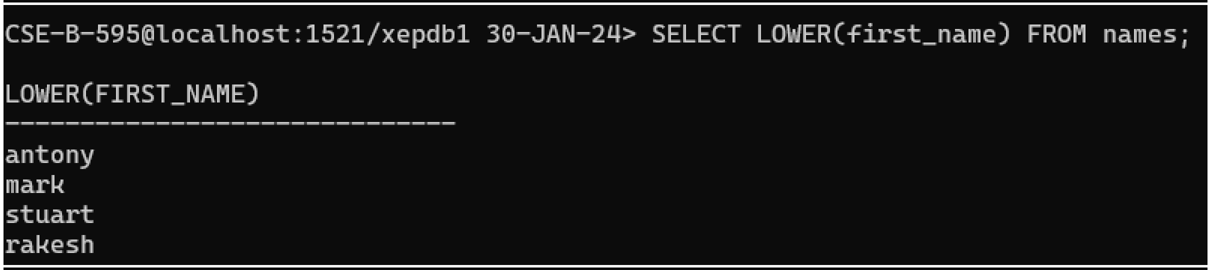
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> CREATE TABLE names(
  2       first_name VARCHAR2(30
  3        )NOT NULL,
  4       LAST_name VARCHAR2(30) NOT NULL
  5       );

Table created.
```



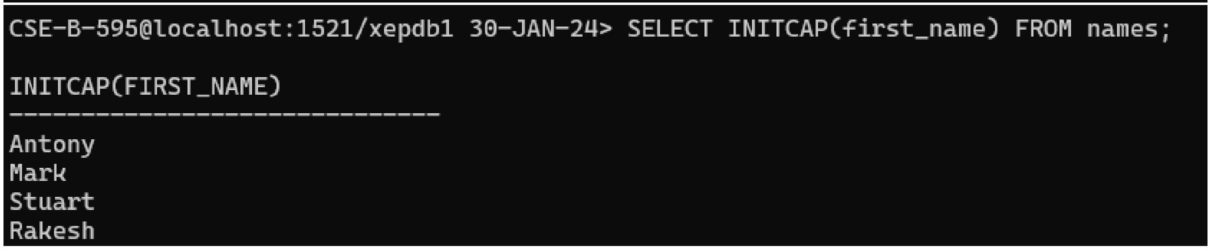
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> INSERT ALL
  2       INTO names VALUES ('Antony','Robert')
  3       INTO names VALUES ('Mark','Antony')
  4       INTO names VALUES ('Stuart','Smart')
  5       INTO names VALUES ('Rakesh','k')
  6       select * from dual;

4 rows created.
```



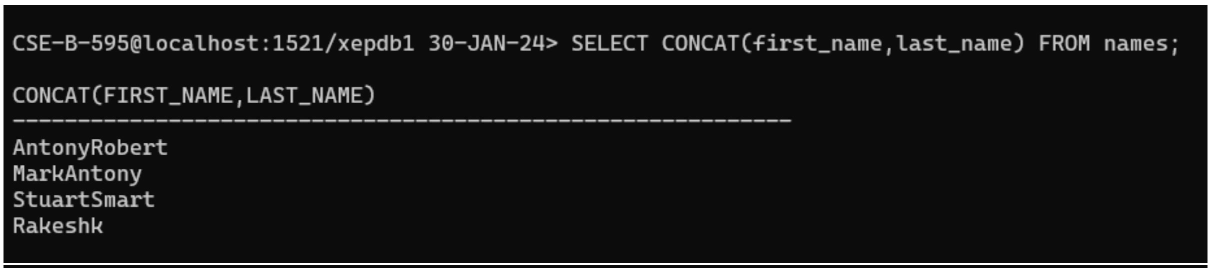
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT LOWER(first_name) FROM names;

LOWER(FIRST_NAME)
------------------------------
antony
mark
stuart
rakesh
```



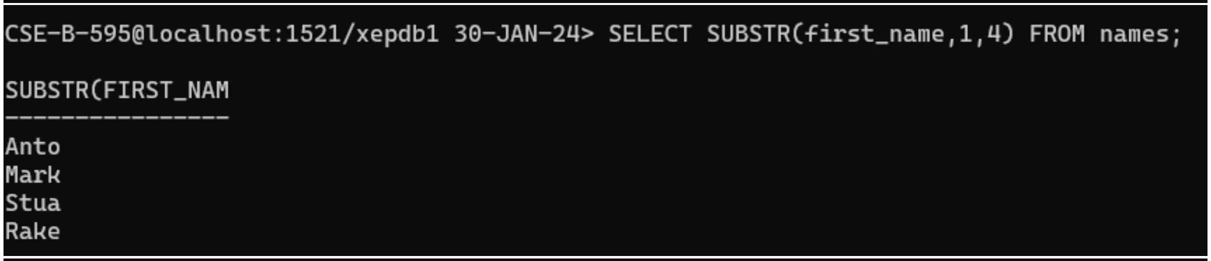
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT LOWER(first_name) FROM names;

LOWER(FIRST_NAME)
------------------------------
antony
mark
stuart
rakesh
```



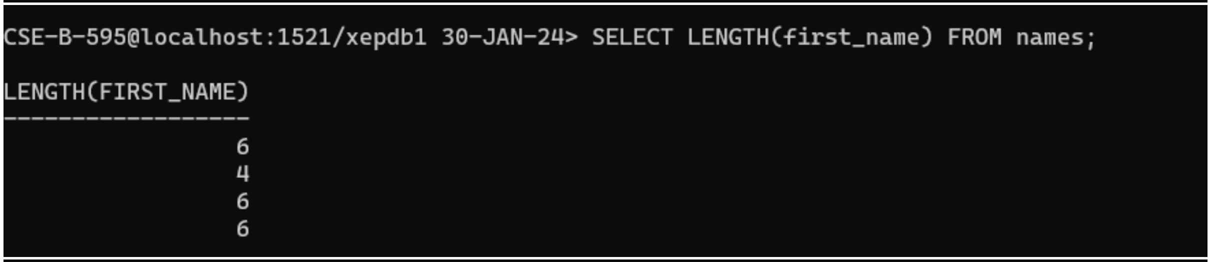
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT INITCAP(first_name) FROM names;

INITCAP(FIRST_NAME)
------------------------------
Antony
Mark
Stuart
Rakesh
```



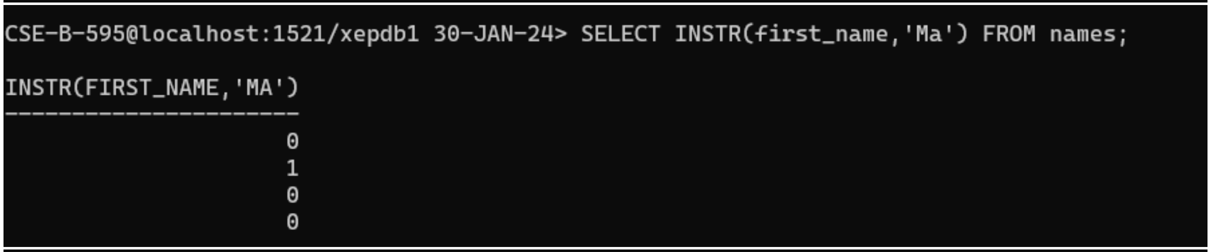
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT CONCAT(first_name,last_name) FROM names;

CONCAT(FIRST_NAME,LAST_NAME)
------------------------------------------------------------
AntonyRobert
MarkAntony
StuartSmart
Rakeshk
```



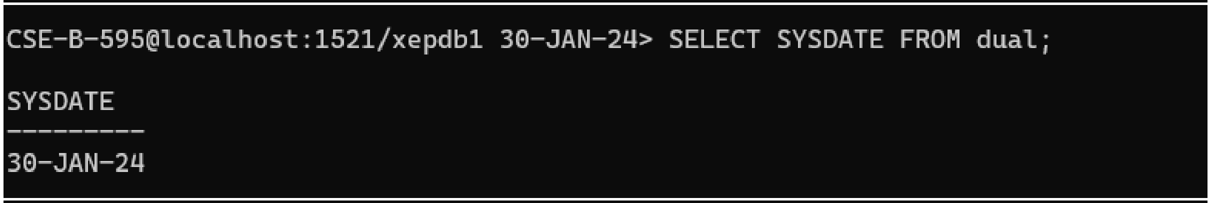
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT SUBSTR(first_name,1,4) FROM names;

SUBSTR(FIRST_NAM
----------------
Anto
Mark
Stua
Rake
```



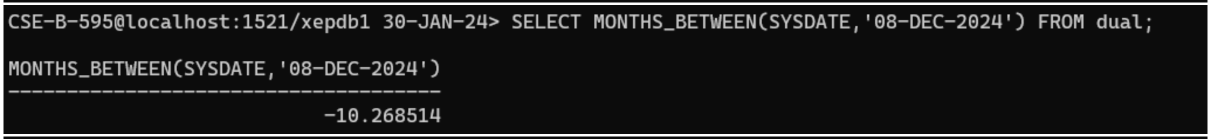
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT LENGTH(first_name) FROM names;

LENGTH(FIRST_NAME)
------------------
                 6
                 4
                 6
                 6
```



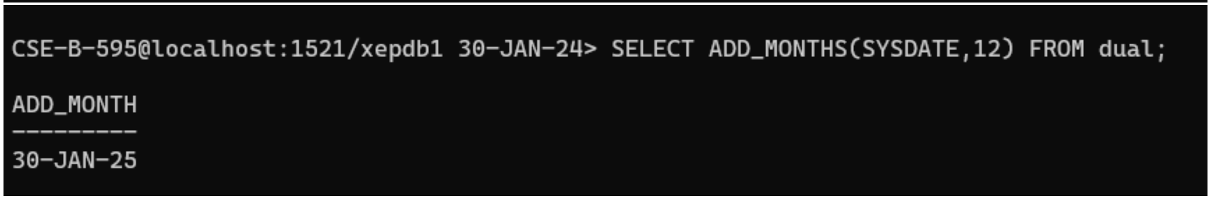
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT INSTR(first_name,'Ma') FROM names;

INSTR(FIRST_NAME,'MA')
----------------------
                     0
                     1
                     0
                     0
```



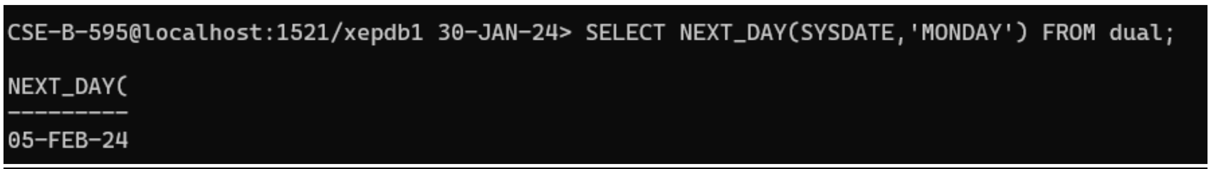
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT SYSDATE FROM dual;

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



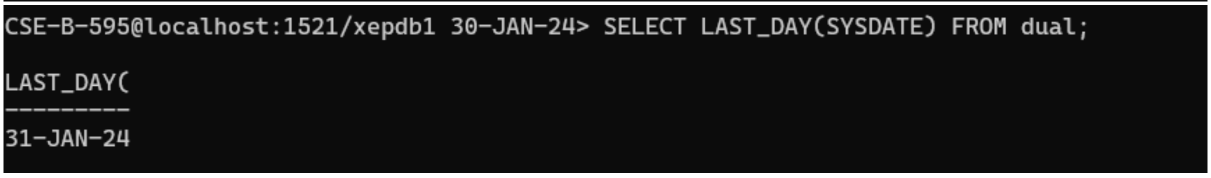
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT MONTHS_BETWEEN(SYSDATE,'08-DEC-2024') FROM dual;

MONTHS_BETWEEN(SYSDATE,'08-DEC-2024')
-------------------------------------
                           -10.268514
```



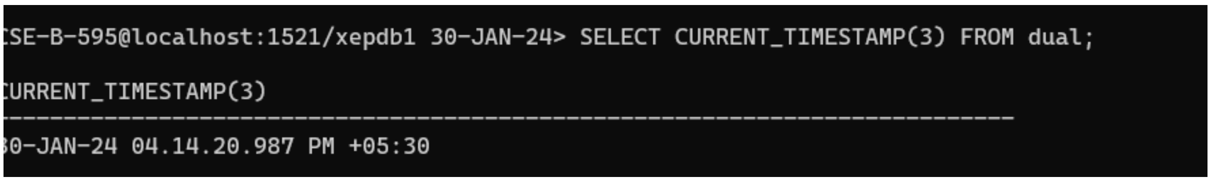
```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT ADD_MONTHS(SYSDATE,12) FROM dual;

ADD_MONTH
---------
30-JAN-25
```



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT NEXT_DAY(SYSDATE,'MONDAY') FROM dual;

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



```
CSE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT LAST_DAY(SYSDATE) FROM dual;

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

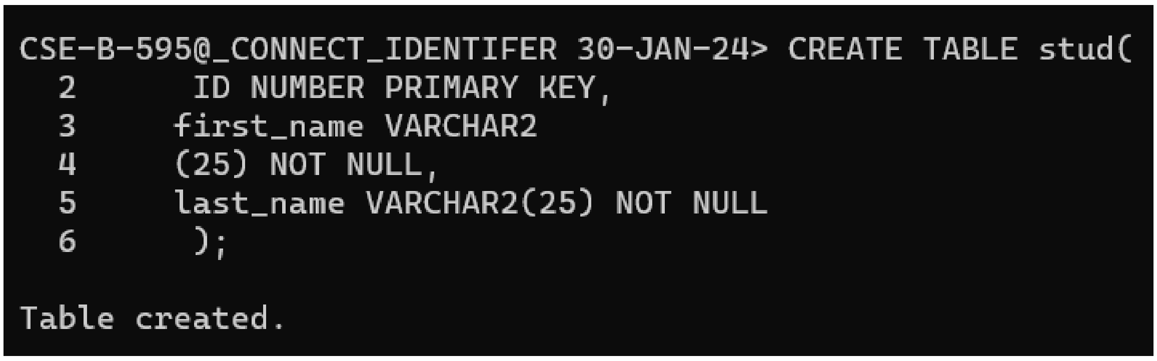


```
SE-B-595@localhost:1521/xepdb1 30-JAN-24> SELECT CURRENT_TIMESTAMP(3) FROM dual;

URRENT_TIMESTAMP(3)
---------------------------------------------------------------------------
0-JAN-24 04.14.20.987 PM +05:30
```

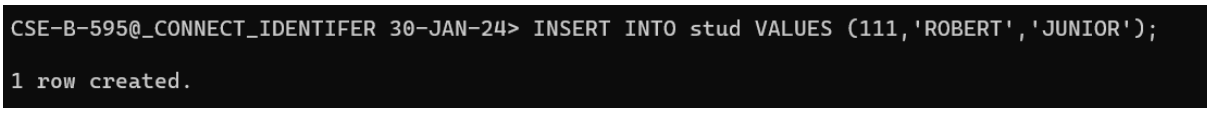
CONCLUSION: In this lab, we successfully practiced SQL Queries to perform ORACLE BUILT – IN Functions.

EXPERIMENT - 09

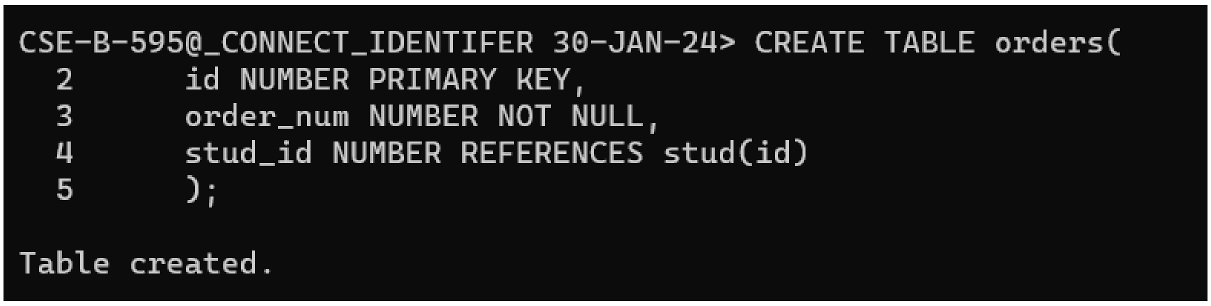
WRITE A SQL QUERIES TO PERFORM KEY CONSTRAINTS (PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, CHECK, DEFAULT).



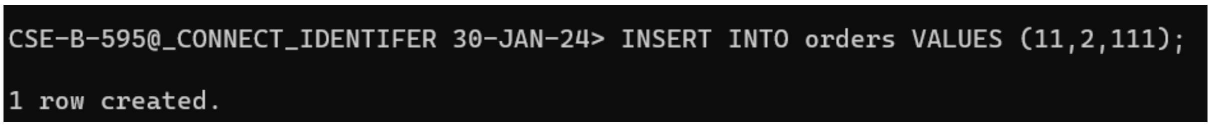
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> CREATE TABLE stud(
         ID NUMBER PRIMARY KEY,
        first_name VARCHAR2
        (25) NOT NULL,
        last_name VARCHAR2(25) NOT NULL
         );

Table created.
```



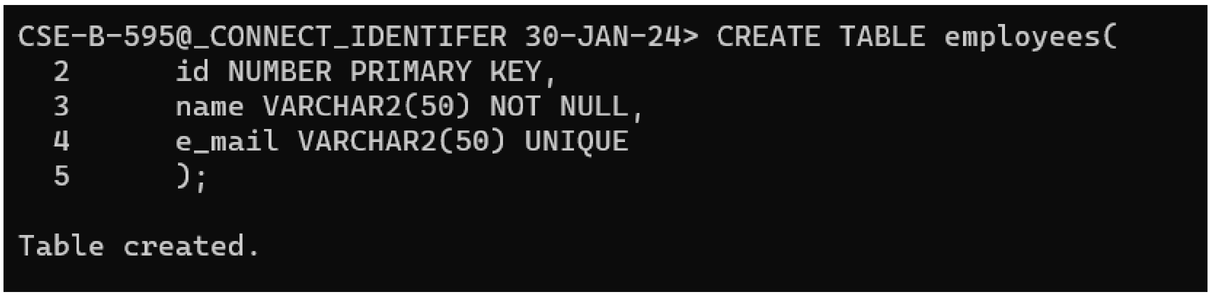
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> INSERT INTO stud VALUES (111,'ROBERT','JUNIOR');

1 row created.
```



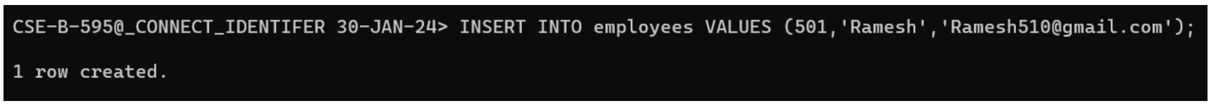
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> CREATE TABLE orders(
  2      id NUMBER PRIMARY KEY,
  3      order_num NUMBER NOT NULL,
  4      stud_id NUMBER REFERENCES stud(id)
  5      );

Table created.
```



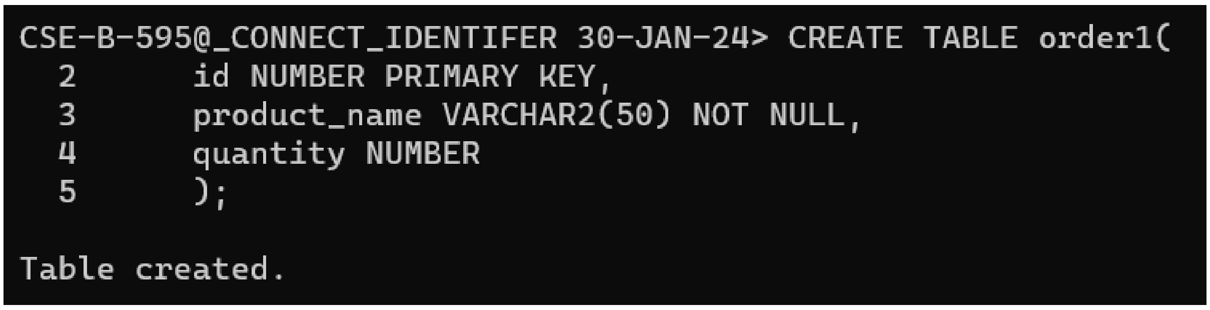
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> INSERT INTO orders VALUES (11,2,111);

1 row created.
```



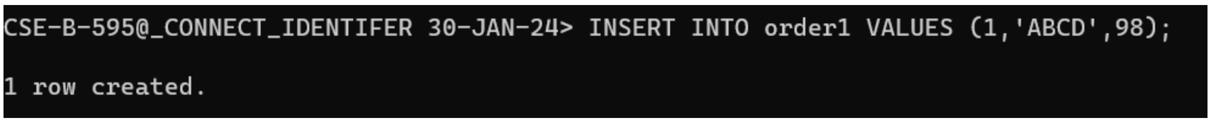
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> CREATE TABLE employees(
  2      id NUMBER PRIMARY KEY,
  3      name VARCHAR2(50) NOT NULL,
  4      e_mail VARCHAR2(50) UNIQUE
  5      );

Table created.
```



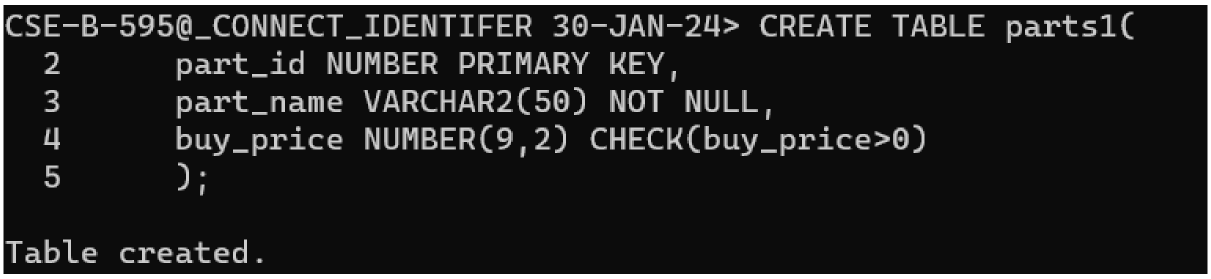
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> INSERT INTO employees VALUES (501,'Ramesh','Ramesh510@gmail.com');

1 row created.
```



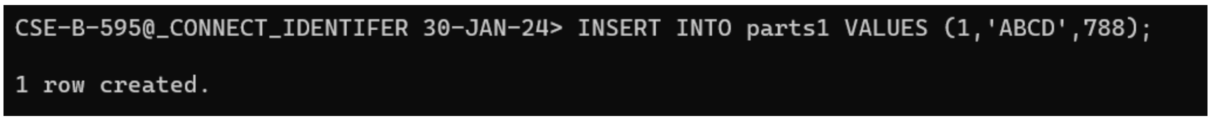
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> CREATE TABLE order1(
  2      id NUMBER PRIMARY KEY,
  3      product_name VARCHAR2(50) NOT NULL,
  4      quantity NUMBER
  5      );

Table created.
```



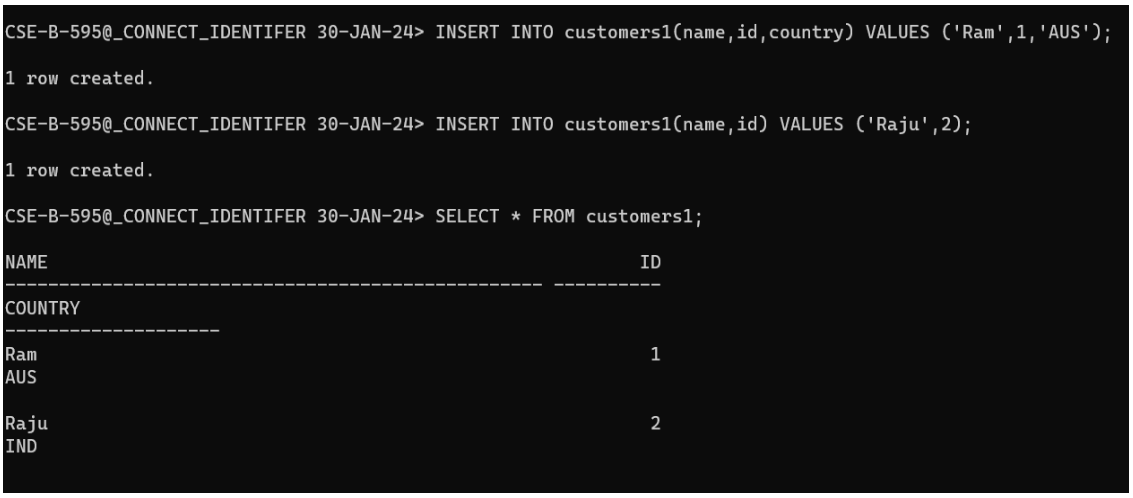
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> INSERT INTO order1 VALUES (1,'ABCD',98);

1 row created.
```



```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> CREATE TABLE parts1(
        part_id NUMBER PRIMARY KEY,
        part_name VARCHAR2(50) NOT NULL,
        buy_price NUMBER(9,2) CHECK(buy_price>0)
        );

Table created.
```



```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> INSERT INTO parts1 VALUES (1,'ABCD',788);

1 row created.
```



```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> INSERT INTO customers1(name,id,country) VALUES ('Ram',1,'AUS');

1 row created.

CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> INSERT INTO customers1(name,id) VALUES ('Raju',2);

1 row created.

CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> SELECT * FROM customers1;

NAME                                                       ID
-------------------------------------------------- ----------
COUNTRY
--------------------
Ram                                                         1
AUS

Raju                                                        2
IND
```

CONCLUSION: In this lab, we successfully executed SQL Queries to perform KEY CONSTRAINTS.

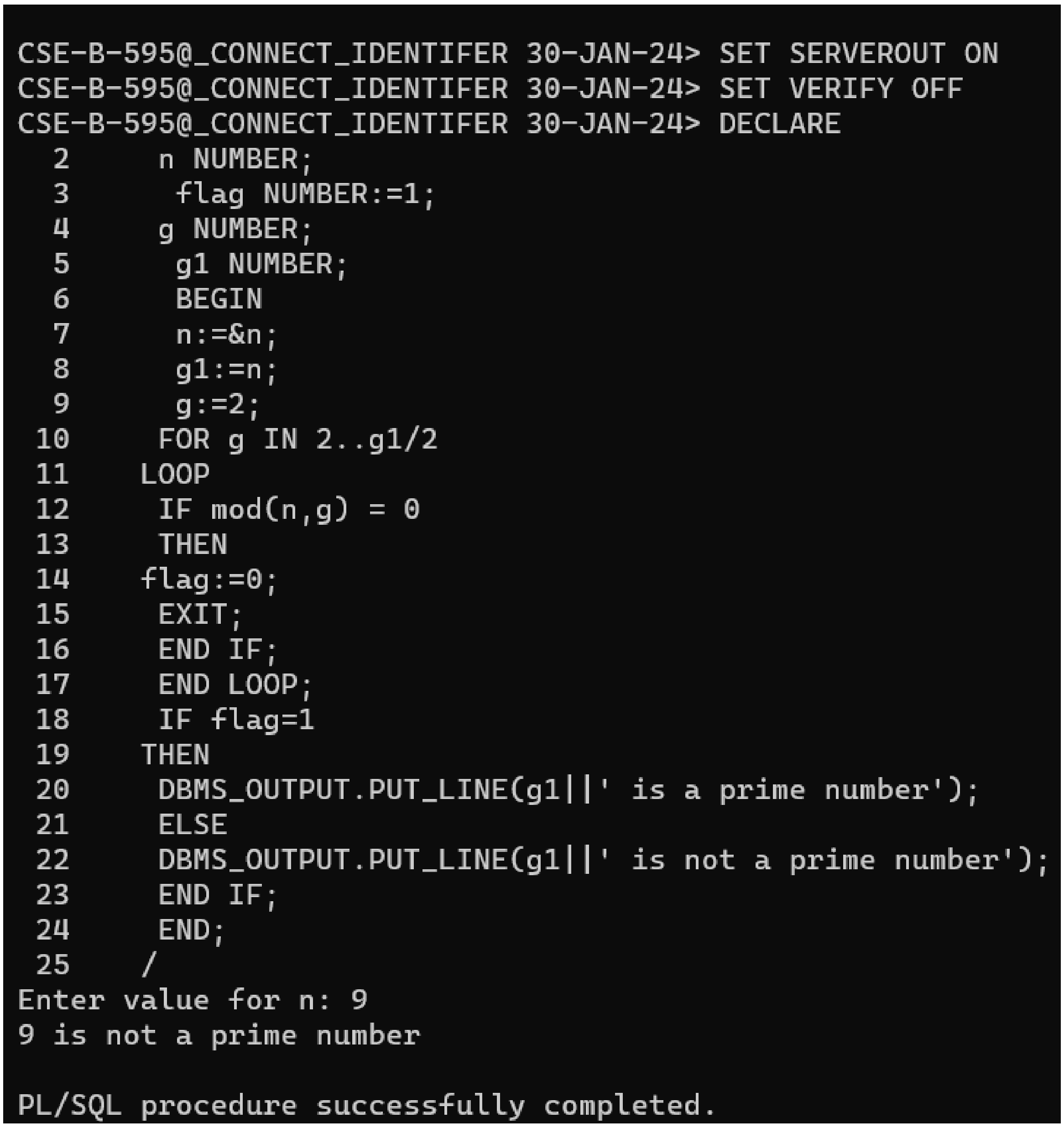
EXPERIMENT-10

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

EXPERIMENT-11

WRITE A PL/SQL PROGRAM FOR FINDING THE GIVEN NUMBER

IS PRIME OR NOT

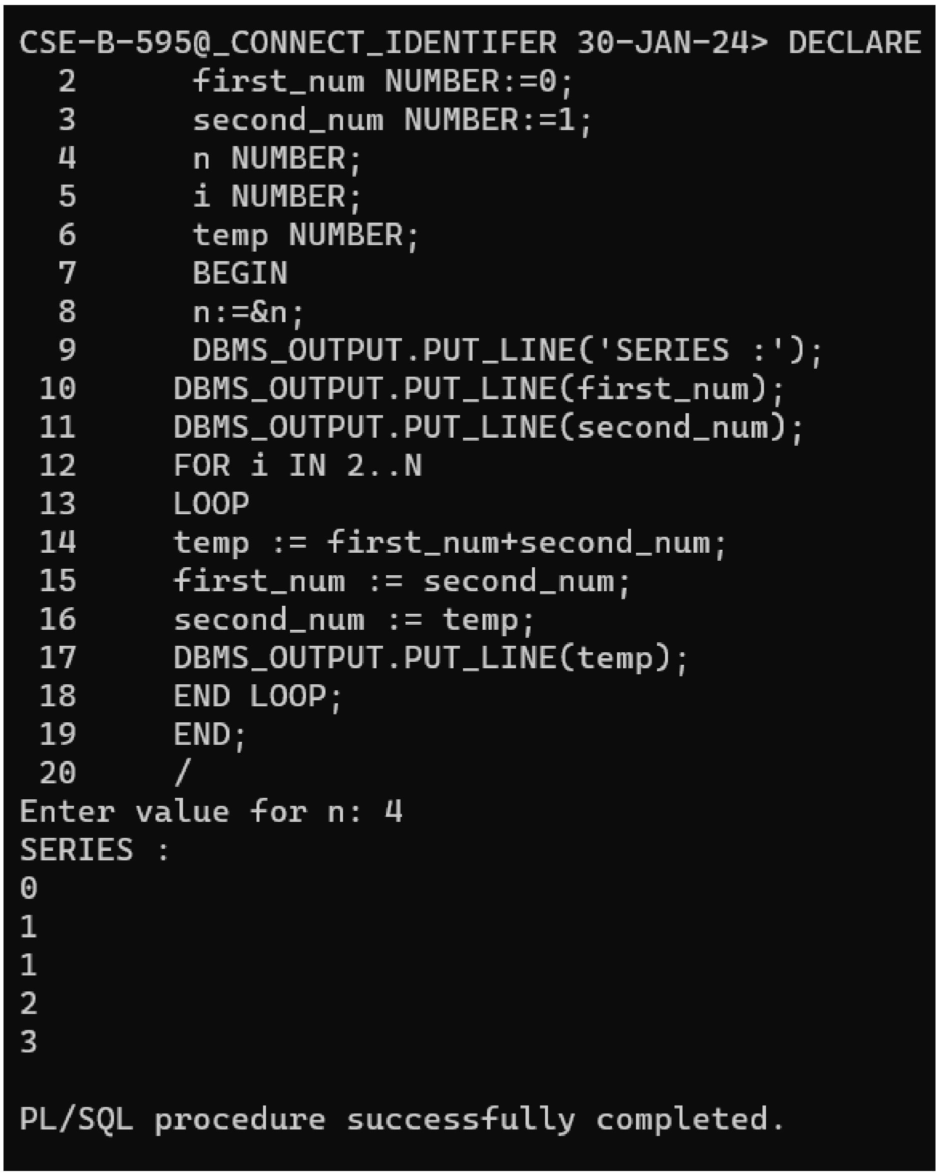


```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> SET SERVEROUT ON
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> SET VERIFY OFF
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> DECLARE
        n NUMBER;
         flag NUMBER:=1;
        g NUMBER;
         g1 NUMBER;
         BEGIN
         n:=&n;
         g1:=n;
         g:=2;
        FOR g IN 2..g1/2
       LOOP
        IF mod(n,g) = 0
        THEN
       flag:=0;
        EXIT;
        END IF;
        END LOOP;
        IF flag=1
       THEN
        DBMS_OUTPUT.PUT_LINE(g1||' is a prime number');
        ELSE
        DBMS_OUTPUT.PUT_LINE(g1||' is not a prime number');
        END IF;
        END;
       /
Enter value for n: 9
9 is not a prime number

PL/SQL procedure successfully completed.
```

CONCLUSION: THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.

EXPERIMENT-12

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

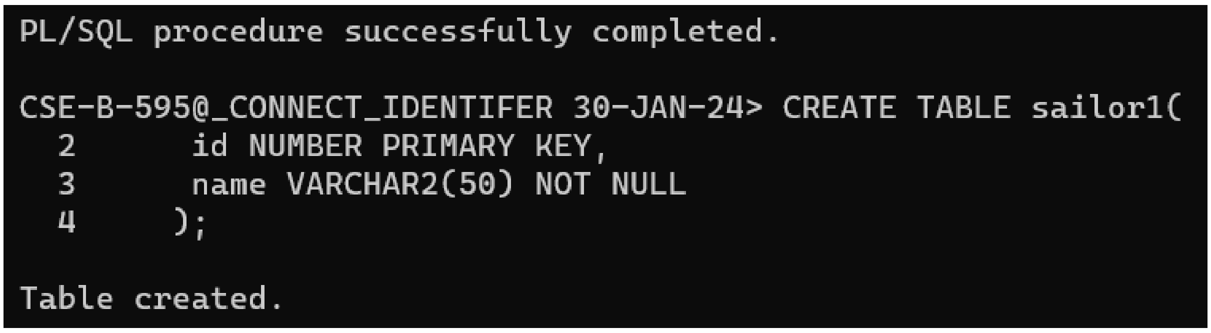


```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> DECLARE
         first_num NUMBER:=0;
         second_num NUMBER:=1;
         n NUMBER;
         i NUMBER;
         temp NUMBER;
         BEGIN
         n:=&n;
         DBMS_OUTPUT.PUT_LINE('SERIES :');
        DBMS_OUTPUT.PUT_LINE(first_num);
        DBMS_OUTPUT.PUT_LINE(second_num);
        FOR i IN 2..N
        LOOP
        temp := first_num+second_num;
        first_num := second_num;
        second_num := temp;
        DBMS_OUTPUT.PUT_LINE(temp);
        END LOOP;
        END;
        /
Enter value for n: 4
SERIES :
0
1
1
2
3

PL/SQL procedure successfully completed.
```

CONCLUSION: THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.

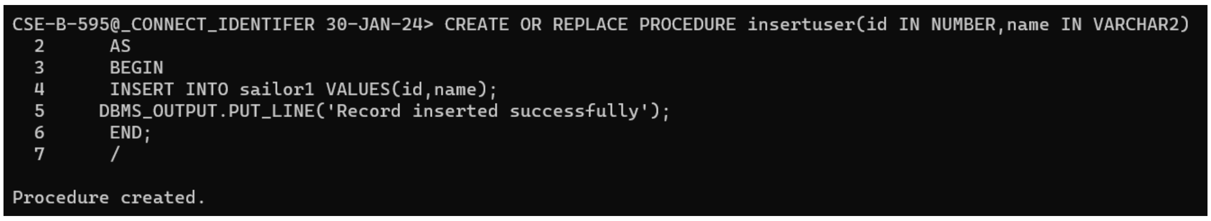
EXPERIMENT-13

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



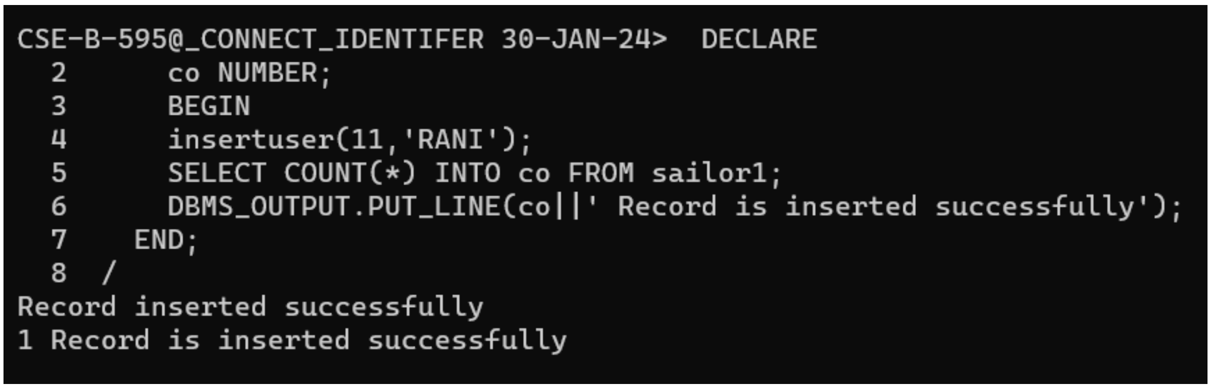
```
PL/SQL procedure successfully completed.

CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> CREATE TABLE sailor1(
         id NUMBER PRIMARY KEY,
         name VARCHAR2(50) NOT NULL
        );

Table created.
```

PROCEDURE CREATION



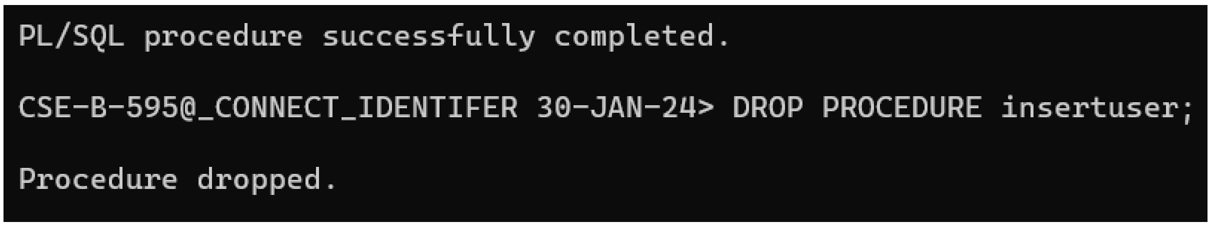
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> CREATE OR REPLACE PROCEDURE insertuser(id IN NUMBER,name IN VARCHAR2)
         AS
         BEGIN
         INSERT INTO sailor1 VALUES(id,name);
        DBMS_OUTPUT.PUT_LINE('Record inserted successfully');
         END;
         /

Procedure created.
```

EXECUTION PROCEDURE



```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24>  DECLARE
         co NUMBER;
         BEGIN
         insertuser(11,'RANI');
         SELECT COUNT(*) INTO co FROM sailor1;
         DBMS_OUTPUT.PUT_LINE(co||' Record is inserted successfully');
       END;
   /
Record inserted successfully
1 Record is inserted successfully
```

DROP PROCEDURE



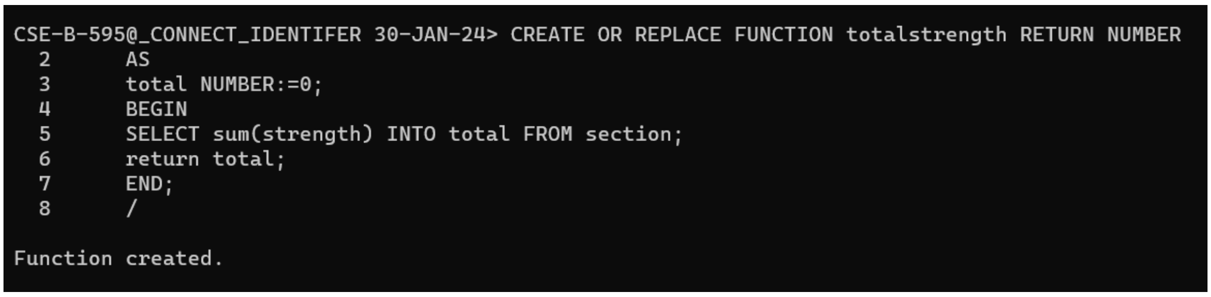
```
PL/SQL procedure successfully completed.

CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> DROP PROCEDURE insertuser;

Procedure dropped.
```

CONCLUSION: THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.

EXPERIMENT-14

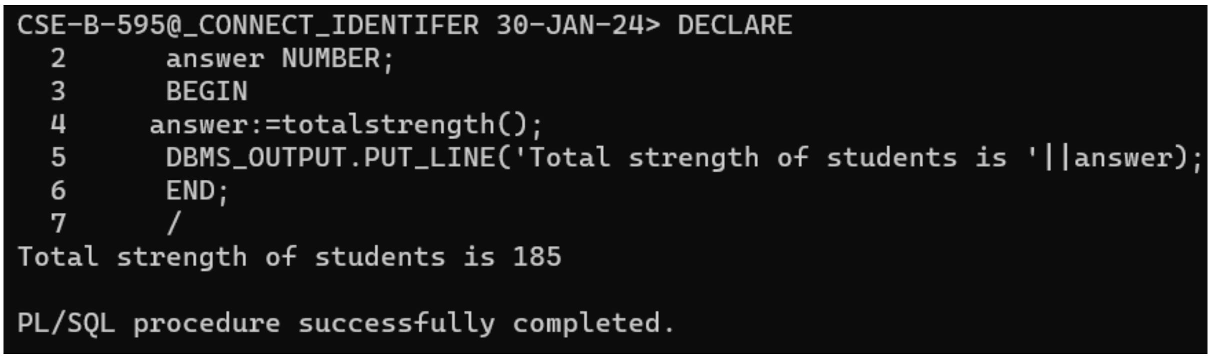
WRITE A PL/SQL PROGRAM TO IMPLEMENT STORED FUNCTION ON TABLE

FUNCTION CREATION



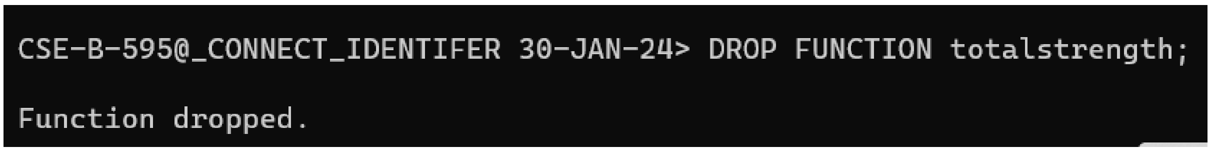
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> CREATE OR REPLACE FUNCTION totalstrength RETURN NUMBER
       AS
       total NUMBER:=0;
       BEGIN
       SELECT sum(strength) INTO total FROM section;
       return total;
       END;
       /

Function created.
```

EXECUTION PROCEDURE



```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> DECLARE
        answer NUMBER;
        BEGIN
       answer:=totalstrength();
        DBMS_OUTPUT.PUT_LINE('Total strength of students is '||answer);
        END;
        /
Total strength of students is 185

PL/SQL procedure successfully completed.
```

DROP FUNCTION



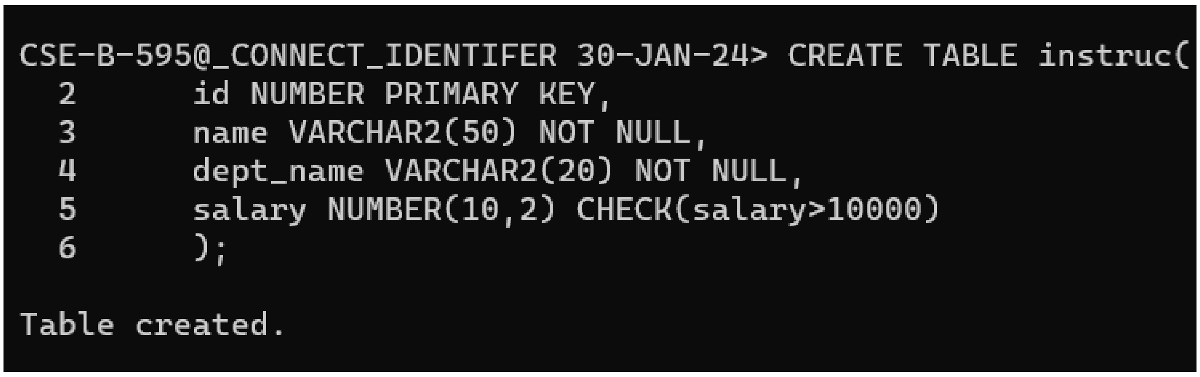
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> DROP FUNCTION totalstrength;

Function dropped.
```

CONCLUSION: THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.

EXPERIMENT-15

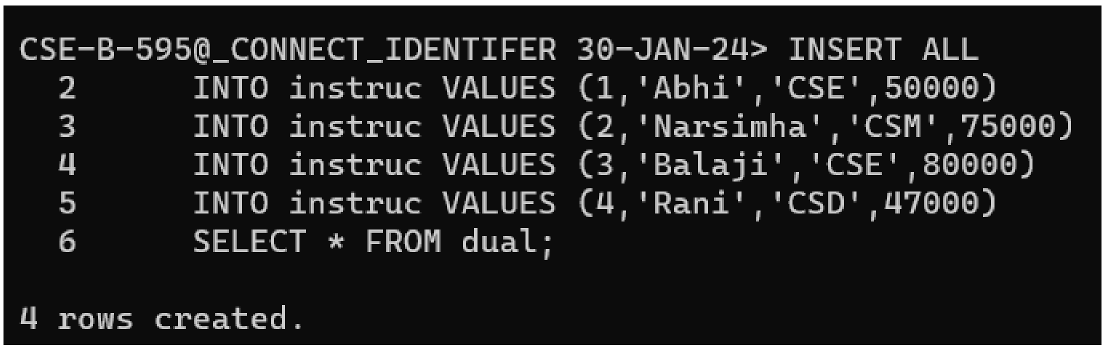
WRITE A PL/SQL PROGRAM TO IMPLEMENT TRIGGER ON TABLE

CREATING TABLES



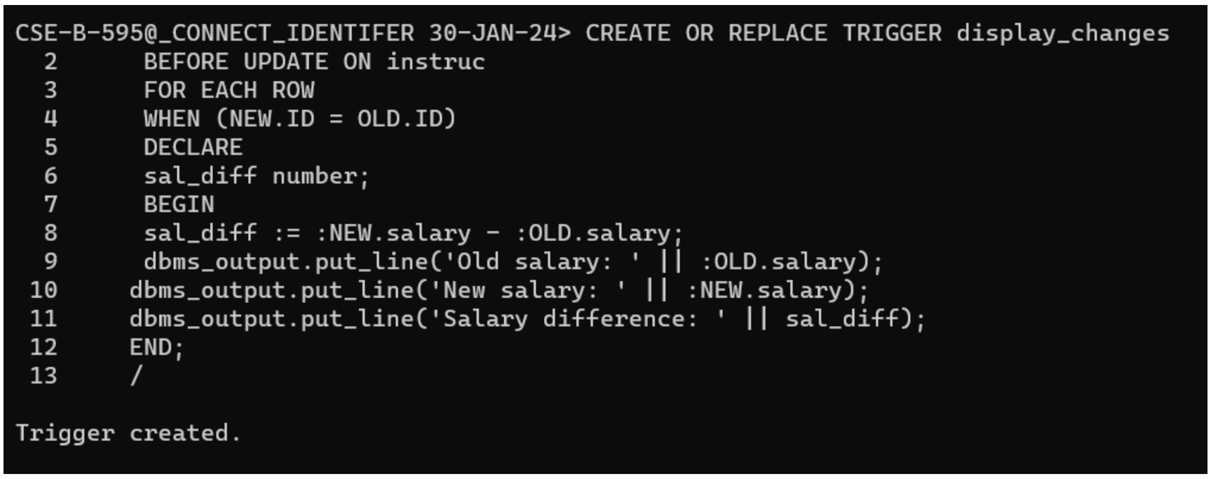
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> CREATE TABLE instruc(
        id NUMBER PRIMARY KEY,
        name VARCHAR2(50) NOT NULL,
        dept_name VARCHAR2(20) NOT NULL,
        salary NUMBER(10,2) CHECK(salary>10000)
        );

Table created.
```

INSERTING VALUES



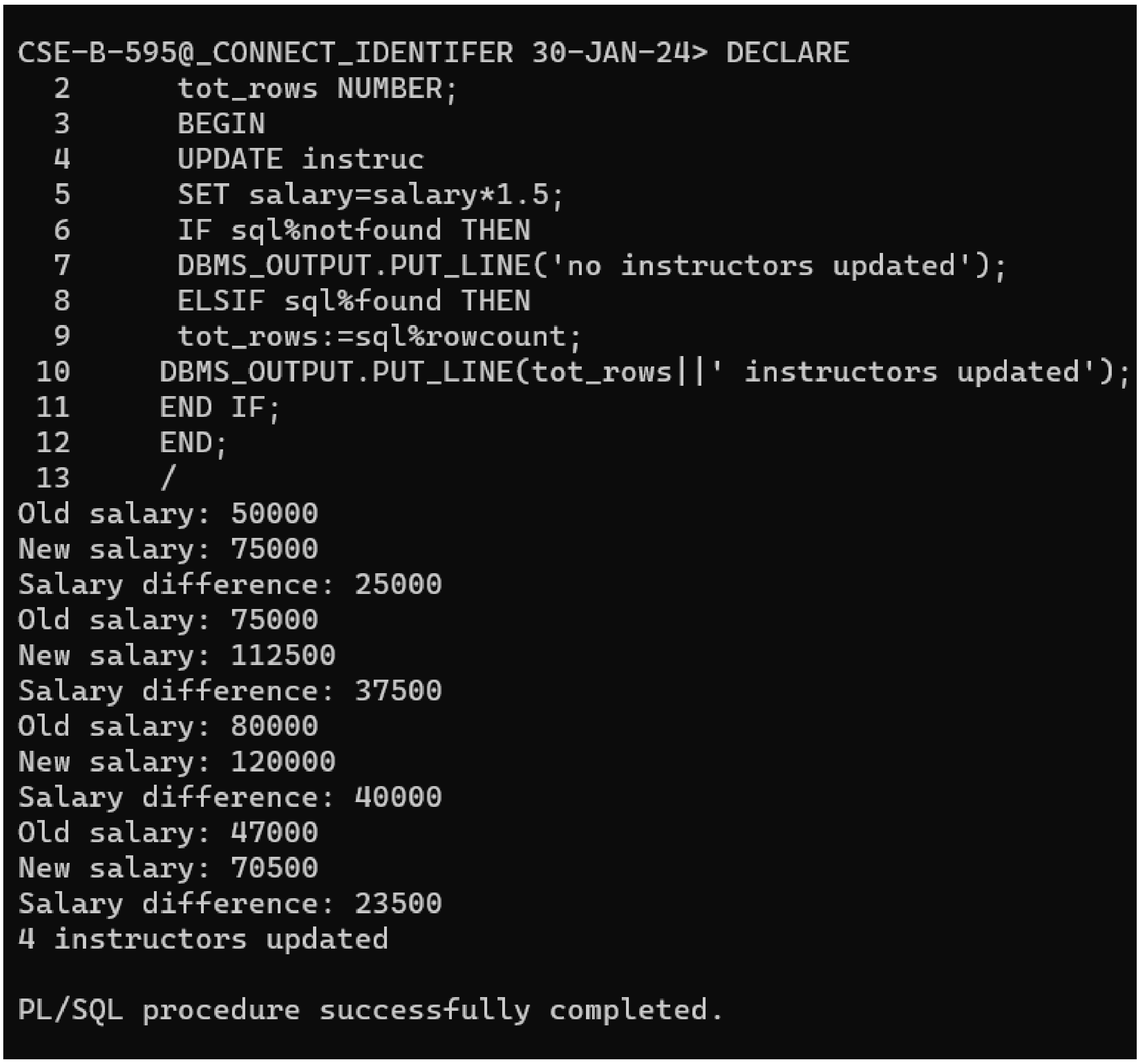
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> INSERT ALL
        INTO instruc VALUES (1,'Abhi','CSE',50000)
        INTO instruc VALUES (2,'Narsimha','CSM',75000)
        INTO instruc VALUES (3,'Balaji','CSE',80000)
        INTO instruc VALUES (4,'Rani','CSD',47000)
        SELECT * FROM dual;

4 rows created.
```

CREATION OF TRIGGER



```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> CREATE OR REPLACE TRIGGER display_changes
        BEFORE UPDATE ON instruc
        FOR EACH ROW
        WHEN (NEW.ID = OLD.ID)
        DECLARE
        sal_diff number;
        BEGIN
        sal_diff := :NEW.salary - :OLD.salary;
        dbms_output.put_line('Old salary: ' || :OLD.salary);
        dbms_output.put_line('New salary: ' || :NEW.salary);
        dbms_output.put_line('Salary difference: ' || sal_diff);
        END;
        /

Trigger created.
```

EXECUTION OF TRIGGER



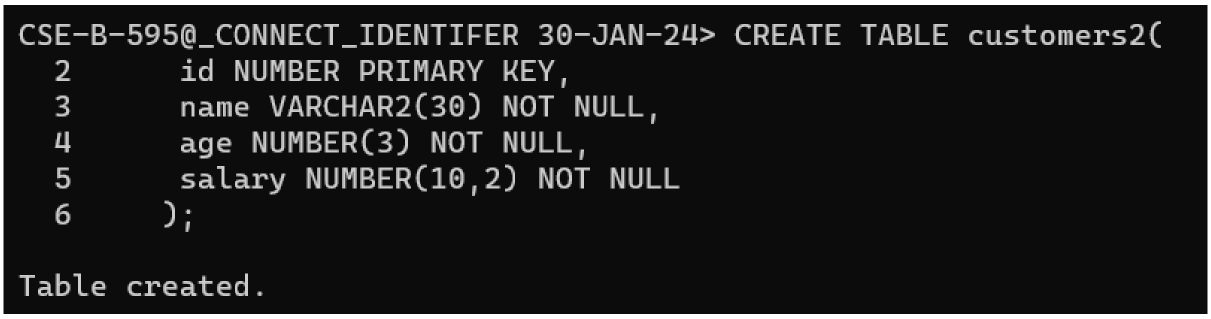
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> DECLARE
         tot_rows NUMBER;
         BEGIN
         UPDATE instruc
         SET salary=salary*1.5;
         IF sql%notfound THEN
         DBMS_OUTPUT.PUT_LINE('no instructors updated');
         ELSIF sql%found THEN
         tot_rows:=sql%rowcount;
        DBMS_OUTPUT.PUT_LINE(tot_rows||' instructors updated');
        END IF;
        END;
        /
Old salary: 50000
New salary: 75000
Salary difference: 25000
Old salary: 75000
New salary: 112500
Salary difference: 37500
Old salary: 80000
New salary: 120000
Salary difference: 40000
Old salary: 47000
New salary: 70500
Salary difference: 23500
4 instructors updated

PL/SQL procedure successfully completed.
```

CONCLUSION: THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.

EXPERIMENT-16

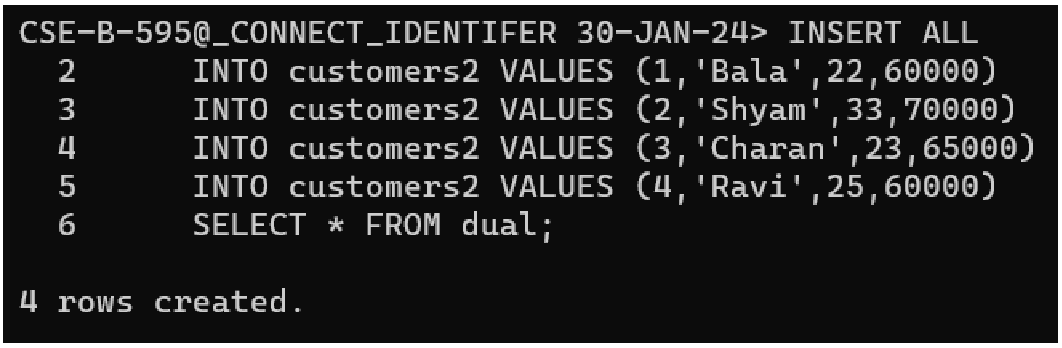
WRITE A PL/SQL PROGRAM TO IMPLEMENT CURSOR ON TABLE

CREATING TABLES



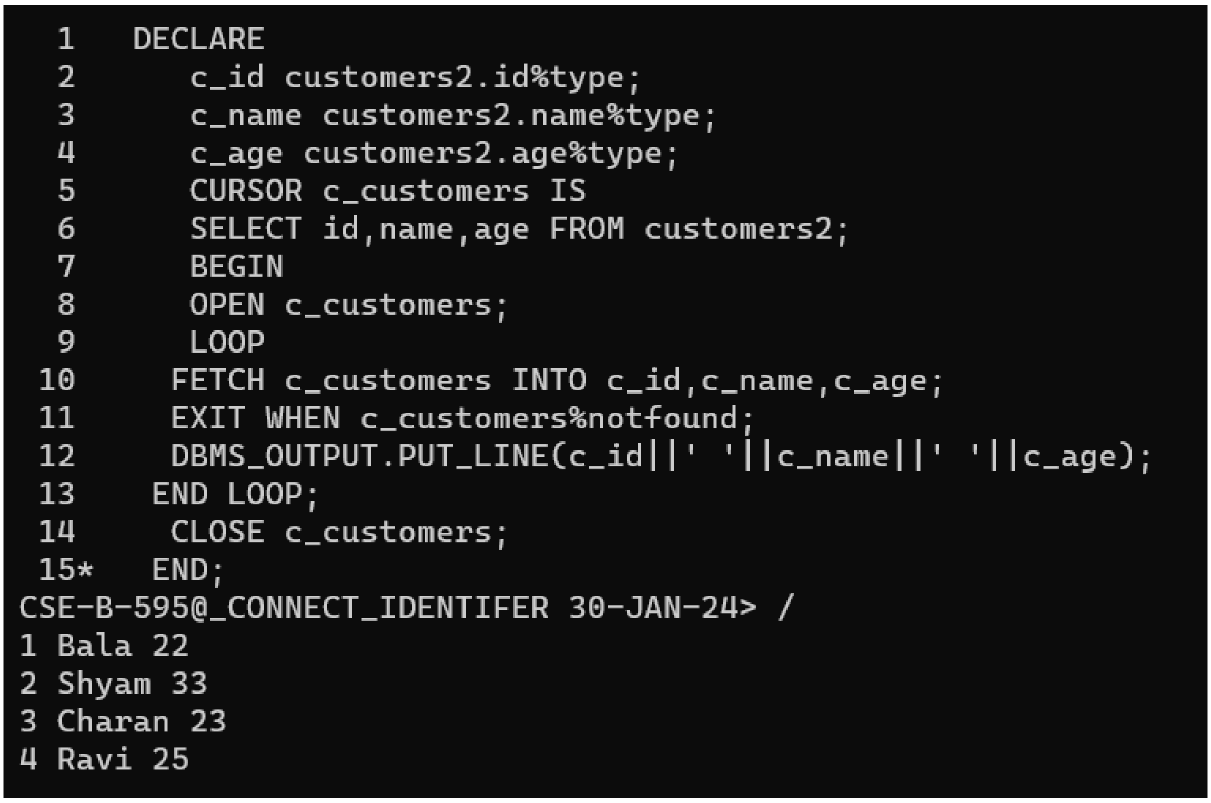
```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> CREATE TABLE customers2(
         id NUMBER PRIMARY KEY,
         name VARCHAR2(30) NOT NULL,
         age NUMBER(3) NOT NULL,
         salary NUMBER(10,2) NOT NULL
        );

Table created.
```

INSERTING VALUES



```
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> INSERT ALL
        INTO customers2 VALUES (1,'Bala',22,60000)
        INTO customers2 VALUES (2,'Shyam',33,70000)
        INTO customers2 VALUES (3,'Charan',23,65000)
        INTO customers2 VALUES (4,'Ravi',25,60000)
        SELECT * FROM dual;

4 rows created.
```

CREATE PROCEDURE



```
DECLARE
   c_id customers2.id%type;
   c_name customers2.name%type;
   c_age customers2.age%type;
   CURSOR c_customers IS
   SELECT id,name,age FROM customers2;
   BEGIN
   OPEN c_customers;
   LOOP
  FETCH c_customers INTO c_id,c_name,c_age;
  EXIT WHEN c_customers%notfound;
  DBMS_OUTPUT.PUT_LINE(c_id||' '||c_name||' '||c_age);
 END LOOP;
  CLOSE c_customers;
 END;
CSE-B-595@_CONNECT_IDENTIFER 30-JAN-24> /
1 Bala 22
2 Shyam 33
3 Charan 23
4 Ravi 25
```

CONCLUSION: THE PROGRAM USING CURSOR IS SUCCESSFULLY COMPLETED.