

224G1A05B8_VASAVI AKANKSHA E

EXPERIMENT – 1

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Sun Dec 17 19:32:20 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Sun Dec 17 2023 19:06:34 +05:30

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

SQL> CREATE TABLE customers1 (
  2 customer_id number(10) NOT NULL,
  3 customer_name VARCHAR2(50) NOT NULL,
  4 city VARCHAR2(50)
  5 );

Table created.

SQL> CREATE TABLE purchase_order_items (
  2 po_nr NUMBER NOT NULL,
  3 item_nr NUMBER NOT NULL,
  4 product_id NUMBER NOT NULL,
  5 quantity NUMBER NOT NULL,
  6 purchase_unit NUMBER NOT NULL,
  7 buy_price NUMBER(9, 2) NOT NULL,
  8 delivery_date DATE,
  9 PRIMARY KEY(po_nr, item_nr)
  10 );

Table created.
```

```
C:\WINDOWS\system32\cmd. x + v

SQL> ALTER TABLE customers1
  2 ADD birthdate DATE NOT NULL;

Table altered.

SQL> DESC customers1;
Name                               Null?    Type
-----
CUSTOMER_ID                        NOT NULL NUMBER(10)
CUSTOMER_NAME                      NOT NULL VARCHAR2(50)
CITY                              VARCHAR2(50)
BIRTHDATE                         NOT NULL DATE

SQL> ALTER TABLE customers1
  2 ADD (
  3 phone VARCHAR(20),
  4 email VARCHAR(100)
  5 );

Table altered.

SQL> DESC customers1;
Name                               Null?    Type
-----
CUSTOMER_ID                        NOT NULL NUMBER(10)
CUSTOMER_NAME                      NOT NULL VARCHAR2(50)
CITY                              VARCHAR2(50)
BIRTHDATE                         NOT NULL DATE
PHONE                             VARCHAR2(20)
EMAIL                             VARCHAR2(100)

SQL> CREATE TABLE persons (
  2 person_id NUMBER,
  3 first_name VARCHAR2(50) NOT NULL,
  4 last_name VARCHAR2(50) NOT NULL,
  5 PRIMARY KEY (person_id)
  6 );
```

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v

SQL> DESC customers1;
Name                                     Null?   Type
-----
CUSTOMER_ID                             NOT NULL NUMBER(10)
CUSTOMER_NAME                           NOT NULL VARCHAR2(50)
CITY                                     VARCHAR2(50)
BIRTHDATE                               NOT NULL DATE
PHONE                                    VARCHAR2(20)
EMAIL                                    VARCHAR2(100)

SQL> CREATE TABLE persons (
2  person_id NUMBER,
3  first_name VARCHAR2(50) NOT NULL,
4  last_name VARCHAR2(50) NOT NULL,
5  PRIMARY KEY (person_id)
6  );

Table created.

SQL> DROP TABLE persons;

Table dropped.

SQL> CREATE TABLE customers_copy
2  AS
3  SELECT
4  *
5  FROM
6  customers;

Table created.

SQL> TRUNCATE TABLE customers_copy;

Table truncated.

SQL>
```

EXPERIMENT-2

```
C:\WINDOWS\system32\cmd. X + v

SQL> CREATE TABLE discounts4 (
2 discount_id NUMBER,
3 discount_name VARCHAR2(255) NOT NULL,
4 amount NUMBER(3, 1) NOT NULL,
5 start_date DATE NOT NULL,
6 expired_date DATE NOT NULL
7 );

Table created.

SQL> INSERT INTO discounts4(discount_id,discount_name,amount,start_date,expired_date)
2 VALUES(1,'Summer Promotion',9.5,DATE '2023-09-10',DATE '2023-12-26');

1 row created.

SQL> DESC discounts4;
Name                               Null?    Type
-----
DISCOUNT_ID                      NUMBER
DISCOUNT_NAME                    NOT NULL VARCHAR2(255)
AMOUNT                             NOT NULL NUMBER(3,1)
START_DATE                         NOT NULL DATE
EXPIRED_DATE                       NOT NULL DATE
```

```
C:\WINDOWS\system32\cmd. X + v

SQL> CREATE TABLE orders2 (
2 cid NUMBER PRIMARY KEY,
3 oid NUMBER,
4 ono NUMBER
5 );

Table created.

SQL> INSERT INTO orders2 VALUES(1,101,501);

1 row created.

SQL> INSERT INTO orders2 VALUES(2,201,601);

1 row created.

SQL> SELECT * FROM orders2;

  CID      OID      ONO
-----
    1      101      501
    2      201      601

SQL> CREATE TABLE fruits2 (
2 fruit_name VARCHAR2(100) PRIMARY KEY,
3 color VARCHAR2(100) NOT NULL
4 );

Table created.

SQL> INSERT ALL
2 INTO fruits2(fruit_name,color)
3 VALUES('Apple','Red')
4 INTO fruits2(fruit_name,color)
5 VALUES('Orange','Orange')
6 INTO fruits2(fruit_name,color)
7 VALUES('Banana','Yellow')
8 SELECT 1 FROM dual;
```

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. X + v
Table created.

SQL> INSERT ALL
  2 INTO fruits2(fruit_name,color)
  3 VALUES('Apple','Red')
  4 INTO fruits2(fruit_name,color)
  5 VALUES('Orange','Orange')
  6 INTO fruits2(fruit_name,color)
  7 VALUES('Banana','Yellow')
  8 SELECT 1 FROM dual;

3 rows created.

SQL> SELECT * FROM fruits2;

FRUIT_NAME
-----
COLOR
-----
Apple
Red

Orange
Orange

Banana
Yellow

SQL> CREATE TABLE parts2(
  2 part_id NUMBER,
  3 part_name VARCHAR2(50) NOT NULL,
  4 lead_time NUMBER(2,0) NOT NULL,
  5 cost NUMBER(9,2) NOT NULL,
  6 status NUMBER(1,0) NOT NULL,
  7 PRIMARY KEY(part_id)
  8 );

Table created.
```

```
C:\WINDOWS\system32\cmd. X + v

SQL> INSERT INTO parts2(part_id,part_name,lead_time,cost,status)
  2 VALUES(1,'Sed dictum',5,134,0);

1 row created.

SQL> INSERT INTO parts2(part_id,part_name,lead_time,cost,status)
  2 VALUES(2,'tristique neque',3,62,1);

1 row created.

SQL> INSERT INTO parts2(part_id,part_name,lead_time,cost,status)
  2 VALUES(3,'dolor quam',16,82,1);

1 row created.

SQL> SELECT * FROM parts2 ORDER BY part_name;

PART_ID PART_NAME LEAD_TIME
-----
COST STATUS
-----
1 Sed dictum 5
134 0
3 dolor quam 16
82 1
2 tristique neque 3
62 1

SQL> UPDATE parts2
  2 SET cost=130;

3 rows updated.

SQL> UPDATE parts2
  2 SET cost = 130
  3 WHERE part_id = 1;
```

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v

SQL> UPDATE parts2
2 SET cost = 130
3 WHERE part_id = 1;

1 row updated.

SQL> SELECT * FROM parts2 WHERE part_id = 1;

PART_ID PART_NAME LEAD_TIME
-----
COST STATUS
-----
1 Sed dictum 5
130 0

SQL> UPDATE parts2
2 SET lead_time=30,cost=120,status=1
3 WHERE part_id=5;

0 rows updated.

SQL> SELECT * FROM parts2 WHERE part_id=1;

PART_ID PART_NAME LEAD_TIME
-----
COST STATUS
-----
1 Sed dictum 5
130 0

SQL> UPDATE parts2
2 SET cost = cost*1.05;

3 rows updated.

SQL> SELECT * FROM parts2;
```

```
C:\WINDOWS\system32\cmd. x + v

PART_ID PART_NAME LEAD_TIME
-----
COST STATUS
-----
1 Sed dictum 5
136.5 0
2 tristique neque 3
136.5 1
3 dolor quam 16
136.5 1

SQL> DELETE FROM parts2 WHERE part_id=1;

1 row deleted.

SQL> SELECT * FROM parts2;

PART_ID PART_NAME LEAD_TIME
-----
COST STATUS
-----
2 tristique neque 3
136.5 1
3 dolor quam 16
136.5 1

SQL> DELETE FROM parts2 WHERE status=1;

2 rows deleted.

SQL> SELECT * FROM parts2;

no rows selected
```

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v
136.5      1
3 dolor quam      16
136.5      1

SQL> DELETE FROM parts2 WHERE part_id=1;
1 row deleted.

SQL> SELECT * FROM parts2;
PART_ID PART_NAME      LEAD_TIME
-----
COST      STATUS
-----
2 tristique neque      3
136.5      1
3 dolor quam      16
136.5      1

SQL> DELETE FROM parts2 WHERE status=1;
2 rows deleted.

SQL> SELECT * FROM parts2;
no rows selected

SQL> DELETE FROM parts2;
0 rows deleted.

SQL> SELECT * FROM parts2;
no rows selected

SQL> |
```

EXPERIMENT-3

Step – 1: create student table

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v
SQL> CREATE TABLE students1 (
  2 Name VARCHAR2(20),
  3 ROLLNO NUMBER,
  4 COURSE VARCHAR2(20)
  5 );
Table created.
SQL> INSERT INTO students1 VALUES('Greeshma',523,'CSE');
1 row created.
SQL> INSERT INTO students1 VALUES('Naveen',524,'CSE');
1 row created.
SQL> INSERT INTO students1 VALUES('Praneetha',521,'CSE');
1 row created.
SQL> select * from students1;
NAME                ROLLNO  COURSE
-----
Greeshma            523     CSE
Naveen              524     CSE
Praneetha           521     CSE
SQL> CREATE VIEW teacher as SELECT name,rollno FROM students1;
View created.
SQL> INSERT INTO teacher(name,rollno)VALUES('Manjula',548);
1 row created.
SQL> INSERT INTO teacher(name,rollno)VALUES('Krishna',555);
1 row created.
```

Step – 2 : Insert few rows into student table

```
C:\WINDOWS\system32\cmd. x + v
SQL> CREATE TABLE students1 (
  2 Name VARCHAR2(20),
  3 ROLLNO NUMBER,
  4 COURSE VARCHAR2(20)
  5 );
Table created.
SQL> INSERT INTO students1 VALUES('Greeshma',523,'CSE');
1 row created.
SQL> INSERT INTO students1 VALUES('Naveen',524,'CSE');
1 row created.
SQL> INSERT INTO students1 VALUES('Praneetha',521,'CSE');
1 row created.
SQL> select * from students1;
NAME                ROLLNO  COURSE
-----
Greeshma            523     CSE
Naveen              524     CSE
Praneetha           521     CSE
SQL> CREATE VIEW teacher as SELECT name,rollno FROM students1;
View created.
SQL> INSERT INTO teacher(name,rollno)VALUES('Manjula',548);
1 row created.
SQL> INSERT INTO teacher(name,rollno)VALUES('Krishna',555);
1 row created.
```

Step-3: Check whether rows are inserted or not

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v
SQL> CREATE TABLE students1 (
2 Name VARCHAR2(20),
3 ROLLNO NUMBER,
4 COURSE VARCHAR2(20)
5 );
Table created.
SQL> INSERT INTO students1 VALUES('Greeshma',523,'CSE');
1 row created.
SQL> INSERT INTO students1 VALUES('Naveen',524,'CSE');
1 row created.
SQL> INSERT INTO students1 VALUES('Praneetha',521,'CSE');
1 row created.
SQL> select * from students1;
NAME                ROLLNO  COURSE
-----
Greeshma             523     CSE
Naveen               524     CSE
Praneetha            521     CSE
SQL> CREATE VIEW teacher as SELECT name,rollno FROM students1;
View created.
SQL> INSERT INTO teacher(name,rollno)VALUES('Manjula',548);
1 row created.
SQL> INSERT INTO teacher(name,rollno)VALUES('Krishna',555);
1 row created.
```

Step-4 : Create view of name teacher with name, roll number constraints and check whether rows are inserted or not

```
C:\WINDOWS\system32\cmd. x + v
SQL> INSERT INTO students1 VALUES('Naveen',524,'CSE');
1 row created.
SQL> INSERT INTO students1 VALUES('Praneetha',521,'CSE');
1 row created.
SQL> select * from students1;
NAME                ROLLNO  COURSE
-----
Greeshma             523     CSE
Naveen               524     CSE
Praneetha            521     CSE
SQL> CREATE VIEW teacher as SELECT name,rollno FROM students1;
View created.
SQL> INSERT INTO teacher(name,rollno)VALUES('Manjula',548);
1 row created.
SQL> INSERT INTO teacher(name,rollno)VALUES('Krishna',555);
1 row created.
SQL> SELECT * FROM teacher;
NAME                ROLLNO
-----
Greeshma            523
Naveen              524
Praneetha           521
Manjula             548
Krishna             555
SQL>
```

END
EXPERIMENT-4

224G1A05B8_VASAVI AKANKSHA E

STEP-1: *Create Instructor table and department table*

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 19 06:42:02 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Tue Dec 19 2023 06:39:11 +05:30

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

SQL> CREATE TABLE instructor6(
  2 ID VARCHAR2(20),
  3 NAME VARCHAR2(20),
  4 BRANCH VARCHAR2(20)
  5 );

Table created.

SQL> CREATE TABLE department5(
  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.

SQL> INSERT INTO instructor6 VALUES('501','Praneetha','CSE');

1 row created.

SQL> INSERT INTO instructor6 VALUES('502','Prasanth','CSE');
```

STEP-2: Insert values into instructor table and department table

```
C:\WINDOWS\system32\cmd. x + v

1 row created.

SQL> INSERT INTO instructor6 VALUES('502','Prasanth','CSE');

1 row created.

SQL> INSERT INTO instructor6 VALUES('503','Manjula','CSE');

1 row created.

SQL> INSERT INTO instructor6 VALUES('504','Krishna','CSE');

1 row created.

SQL> SELECT * FROM instructor6;

ID          NAME          BRANCH
-----
501         Praneetha      CSE
502         Prasanth       CSE
503         Manjula        CSE
504         Krishna        CSE

SQL> INSERT INTO department5 VALUES('Comp.Sci','Anirudh','100000');

1 row created.

SQL> INSERT INTO department5 VALUES('Elec.Eng','Maya','85000');

1 row created.

SQL> INSERT INTO department5 VALUES('Physics','Srikanth','50000');

1 row created.

SQL> INSERT INTO department5 VALUES('Chemistry','Shamili','45000');

1 row created.
```

STEP-3: Perform RELATIONAL SET Operations

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v

SQL> SELECT * FROM department5;

DEPT_NAME          BUILDING          BUDGET
-----
Comp.Sci            Anirudh           100000
Elec.Eng            Maya              85000
Physics             Srikanth          50000
Chemistry           Shamili           45000

SQL> SELECT name FROM instructor6
  2 UNION
  3 (SELECT d_name FROM department5);
(SQL> SELECT d_name FROM department5)
      *
ERROR at line 3:
ORA-00904: "D_NAME": invalid identifier

SQL> SELECT NAME FROM instructor6
  2 UNION
  3 SELECT dept_name FROM department5;

NAME
-----
Praneetha
Prasanth
Manjula
Krishna
Comp.Sci
Elec.Eng
Physics
Chemistry

8 rows selected.

SQL> SELECT NAME FROM instructor6
  2 UNION ALL
  3 SELECT dept_name FROM department5;
```

```
C:\WINDOWS\system32\cmd. x + v

  2 UNION ALL
  3 SELECT dept_name FROM department5;

NAME
-----
Praneetha
Prasanth
Manjula
Krishna
Comp.Sci
Elec.Eng
Physics
Chemistry

8 rows selected.

SQL> SELECT NAME FROM instructor6
  2 INTERSECT
  3 SELECT dept_name FROM department5;

no rows selected

SQL> SELECT NAME FROM instructor6
  2 INTERSECT ALL
  3 SELECT dept_name FROM department5;

no rows selected

SQL> SELECT NAME FROM instructor6
  2 MINUS
  3 SELECT dept_name FROM department5;

NAME
-----
Praneetha
Prasanth
Manjula
Krishna

SQL> |
```

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v
Krishna
SQL> SELECT * FROM instructor6
2 CROSS JOIN department5;
```

ID	NAME	BRANCH
501	Praneetha	CSE
501	Anirudh	100000
501	Praneetha	CSE
501	Maya	85000
501	Praneetha	CSE
501	Srikanth	50000

ID	NAME	BRANCH
501	Praneetha	CSE
501	Shamili	45000
502	Prasanth	CSE
502	Anirudh	100000
502	Prasanth	CSE
502	Maya	85000

ID	NAME	BRANCH
502	Prasanth	CSE
502	Srikanth	50000

```
C:\WINDOWS\system32\cmd. x + v
```

ID	NAME	BRANCH
502	Prasanth	CSE
502	Srikanth	50000
502	Prasanth	CSE
502	Shamili	45000
503	Manjula	CSE
503	Anirudh	100000

ID	NAME	BRANCH
503	Manjula	CSE
503	Maya	85000
503	Manjula	CSE
503	Srikanth	50000
503	Manjula	CSE
503	Shamili	45000

ID	NAME	BRANCH
504	Krishna	CSE
504	Anirudh	100000
504	Krishna	CSE
504	Maya	85000
504	Krishna	CSE
504	Srikanth	50000

224G1A05B8_VASAVI AKANKSHA E

C:\WINDOWS\system32\cmd. X + v

ID	NAME	BRANCH
504	Krishna	CSE
Chemistry	Shamili	45000

16 rows selected.

SQL> SELECT * FROM instructor6
2 NATURAL JOIN department5;

ID	NAME	BRANCH
501	Praneetha	CSE
Comp.Sci	Anirudh	100000
501	Praneetha	CSE
Elec.Eng	Maya	85000
501	Praneetha	CSE
Physics	Srikanth	50000

ID	NAME	BRANCH
501	Praneetha	CSE
Chemistry	Shamili	45000
502	Prasanth	CSE
Comp.Sci	Anirudh	100000
502	Prasanth	CSE
Elec.Eng	Maya	85000

C:\WINDOWS\system32\cmd. X + v

ID	NAME	BRANCH
502	Prasanth	CSE
Physics	Srikanth	50000
502	Prasanth	CSE
Chemistry	Shamili	45000
503	Manjula	CSE
Comp.Sci	Anirudh	100000

ID	NAME	BRANCH
503	Manjula	CSE
Elec.Eng	Maya	85000
503	Manjula	CSE
Physics	Srikanth	50000
503	Manjula	CSE
Chemistry	Shamili	45000

ID	NAME	BRANCH
504	Krishna	CSE
Comp.Sci	Anirudh	100000
504	Krishna	CSE
Elec.Eng	Maya	85000

END

224G1A05B8_VASAVI AKANKSHA E

EXPERIMENT-5

Step-1: *Create employee table*

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Mon Dec 18 19:12:50 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Mon Dec 18 2023 18:49:56 +05:30

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

SQL> CREATE TABLE Emp1(
  2 emp_id int,
  3 emp_name VARCHAR(20),
  4 emp_salary int
  5 );

Table created.

SQL> DESC Emp1;
      Name                                         Null?     Type
-----
EMP_ID                                           NUMBER(38)
EMP_NAME                                         VARCHAR2(20)
EMP_SALARY                                       NUMBER(38)

SQL> INSERT INTO Emp1 VALUES('1','Anil kumar','100000');

1 row created.

SQL> INSERT INTO Emp1 VALUES('2','Vijaya Lakshmi','98000');

1 row created.
```

Step-2: *Insert few rows into the Employee table and check whether rows are selected or not*

```
C:\WINDOWS\system32\cmd. x + v
EMP_SALARY                                         NUMBER(38)

SQL> INSERT INTO Emp1 VALUES('1','Anil kumar','100000');

1 row created.

SQL> INSERT INTO Emp1 VALUES('2','Vijaya Lakshmi','98000');

1 row created.

SQL> INSERT INTO Emp1 VALUES('3','Sudheer Kumar','95000');

1 row created.

SQL> INSERT INTO Emp1 VALUES('4','Narasimhulu','90000');

1 row created.

SQL> INSERT INTO Emp1 VALUES('5','Veera Prakash','85000');

1 row created.

SQL> SELECT * FROM Emp1;

EMP_ID EMP_NAME EMP_SALARY
-----
1 Anil kumar 100000
2 Vijaya Lakshmi 98000
3 Sudheer Kumar 95000
4 Narasimhulu 90000
5 Veera Prakash 85000

SQL> select count(*)emp_id from Emp1;

EMP_ID
-----
5

SQL> select avg(emp_id) from Emp1;
```

Step-3: *Implement 5 aggregate operations*

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v
EMP_ID EMP_NAME EMP_SALARY
-----
1 Anil kumar 100000
2 Vijaya Lakshmi 90000
3 Sudheer Kumar 95000
4 Narasimhulu 90000
5 Veera Prakash 85000

SQL> select count(*) emp_id from Emp1;

EMP_ID
-----
5

SQL> select avg(emp_id) from Emp1;

AVG(EMP_ID)
-----
3

SQL> select min(emp_id) from Emp1;

MIN(EMP_ID)
-----
1

SQL> select max(emp_id) from Emp1;

MAX(EMP_ID)
-----
5

SQL> |
```

END

EXPERIMENT-6

Step-1: Create student table and blocks table

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 19 16:57:26 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Tue Dec 19 2023 08:06:11 +05:30

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

SQL> CREATE TABLE student1(
 2 roll_no NUMBER PRIMARY KEY,
 3 name VARCHAR2(50) NOT NULL,
 4 dept_name VARCHAR2(10) NOT NULL
 5 );

Table created.

SQL> CREATE TABLE blocks1(
 2 dept_name VARCHAR2(10) PRIMARY KEY,
 3 block_name VARCHAR2(20) NOT NULL
 4 );

Table created.

SQL> INSERT INTO student1 VALUES(519,'GAYATRI','CSM');

1 row created.

SQL> INSERT INTO student1 VALUES(523,'GREESHMA','CSE');

1 row created.
```

Step-2: Insert values into student and blocks table and check whether rows are inserted or not

```
C:\WINDOWS\system32\cmd. x + v

SQL> INSERT INTO student1 VALUES(519,'GAYATRI','CSM');

1 row created.

SQL> INSERT INTO student1 VALUES(523,'GREESHMA','CSE');

1 row created.

SQL> INSERT INTO student1 VALUES(557,'NANDINI','CSD');

1 row created.

SQL> SELECT * FROM student1;

  ROLL_NO NAME                                DEPT_NAME
-----
    519 GAYATRI                                CSM
    523 GREESHMA                               CSE
    557 NANDINI                                CSD

SQL> INSERT INTO blocks1 VALUES('CSM','B-BLOCK');

1 row created.

SQL> INSERT INTO blocks1 VALUES('CSE','MAIN BLOCK');

1 row created.

SQL> INSERT INTO blocks1 VALUES('CSD','A-BLOCK');

1 row created.

SQL> SELECT * FROM blocks1;

DEPT_NAME  BLOCK_NAME
-----
CSM        B-BLOCK
CSE        MAIN BLOCK
CSD        A-BLOCK
```

Step-3: Perform JOIN OPERATIONS

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v
SQL> SELECT * FROM student1
2 JOIN blocks1 ON
3 student1.dept_name=blocks1.dept_name;

ROLL_NO NAME                                DEPT_NAME
-----
DEPT_NAME BLOCK_NAME
-----
CSM      519 GAYATRI
          B-BLOCK                                CSM
CSE      523 GREESHMA
          MAIN BLOCK                             CSE
CSD      557 NANDINI
          A-BLOCK                                CSD

SQL> SELECT * FROM student1 JOIN blocks1
2 USING(dept_name);

DEPT_NAME ROLL_NO NAME
-----
BLOCK_NAME
-----
CSM      519 GAYATRI
B-BLOCK
CSE      523 GREESHMA
MAIN BLOCK
CSD      557 NANDINI
A-BLOCK

SQL> SELECT * FROM student1
2 LEFT OUTER JOIN blocks1 ON
3 student1.dept_name=blocks1.dept_name;
```

```
C:\WINDOWS\system32\cmd. x + v
SQL> SELECT * FROM student1
2 LEFT OUTER JOIN blocks1 ON
3 student1.dept_name=blocks1.dept_name;

ROLL_NO NAME                                DEPT_NAME
-----
DEPT_NAME BLOCK_NAME
-----
CSM      519 GAYATRI
          B-BLOCK                                CSM
CSE      523 GREESHMA
          MAIN BLOCK                             CSE
CSD      557 NANDINI
          A-BLOCK                                CSD

SQL> SELECT * FROM student1
2 RIGHT OUTER JOIN blocks1 ON
3 student1.dept_name=blocks1.dept_name;

ROLL_NO NAME                                DEPT_NAME
-----
DEPT_NAME BLOCK_NAME
-----
CSM      519 GAYATRI
          B-BLOCK                                CSM
CSE      523 GREESHMA
          MAIN BLOCK                             CSE
CSD      557 NANDINI
          A-BLOCK                                CSD

SQL> SELECT * FROM student1
2 FULL OUTER JOIN blocks1
3 ON
4 student1.dept_name=blocks1.dept_name;
```


224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v

SQL> SELECT * FROM student1
2 RIGHT OUTER JOIN blocks1 ON
3 student1.dept_name=blocks1.dept_name;

ROLL_NO NAME                                DEPT_NAME
-----
DEPT_NAME BLOCK_NAME
-----
CSM      519 GAYATRI
          B-BLOCK                                CSM
CSE      523 GREESHMA
          MAIN BLOCK                             CSE
CSD      557 NANDINI
          A-BLOCK                                CSD

SQL> SELECT * FROM student1
2 FULL OUTER JOIN blocks1
3 ON
4 student1.dept_name=blocks1.dept_name;

ROLL_NO NAME                                DEPT_NAME
-----
DEPT_NAME BLOCK_NAME
-----
CSM      519 GAYATRI
          B-BLOCK                                CSM
CSE      523 GREESHMA
          MAIN BLOCK                             CSE
CSD      557 NANDINI
          A-BLOCK                                CSD

SQL>
```

END

EXPERIMENT-7

Step-1:Create Employee Table

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 19 18:18:46 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Tue Dec 19 2023 16:57:35 +05:30

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

SQL> CREATE TABLE employee1(
  2 ID NUMBER PRIMARY KEY,
  3 name VARCHAR2(50) NOT NULL,
  4 gender CHAR NOT NULL,
  5 salary NUMBER(10,2) NOT NULL
  6 );

Table created.

SQL> INSERT INTO employee1 VALUES(1,'Anil Kumar','M',100000);

1 row created.

SQL> INSERT INTO employee1 VALUES(2,'Narasimhulu','M',95000);

1 row created.

SQL> INSERT INTO employee1 VALUES(3,'Sudheer Kumar','M',93000);

1 row created.

SQL> INSERT INTO employee1 VALUES(4,'Vijaya Lakshmi','F',90000);
```

Step-2: Insert values into Employee table and check whether rows are inserted or not

```
C:\WINDOWS\system32\cmd. x + v

Table created.

SQL> INSERT INTO employee1 VALUES(1,'Anil Kumar','M',100000);

1 row created.

SQL> INSERT INTO employee1 VALUES(2,'Narasimhulu','M',95000);

1 row created.

SQL> INSERT INTO employee1 VALUES(3,'Sudheer Kumar','M',93000);

1 row created.

SQL> INSERT INTO employee1 VALUES(4,'Vijaya Lakshmi','F',90000);

1 row created.

SQL> INSERT INTO employee1 VALUES(5,'Veera Prakash','M',85000);

1 row created.

SQL> SELECT * FROM employee1;

  ID NAME                G SALARY
-----
  1 Anil Kumar            M 100000
  2 Narasimhulu          M  95000
  3 Sudheer Kumar        M  93000
  4 Vijaya Lakshmi       F   90000
  5 Veera Prakash        M   85000

SQL> SELECT SUM(salary) FROM employee1;

SUM(SALARY)
-----
      463000
```

Step-3: Perform AGGREGATE OPERATIONS

224G1A05B8_VASAVI AKANKSHA E

```
SQL> SELECT AVG(salary) FROM employee1;

AVG(SALARY)
-----
    92600

SQL> SELECT COUNT(salary) FROM employee1;

COUNT(SALARY)
-----
              5

SQL> SELECT MIN(salary) FROM employee1;

MIN(SALARY)
-----
    85000

SQL> SELECT MAX(salary) FROM employee1;

MAX(SALARY)
-----
   100000

SQL> |
```

END

EXPERIMENT-8

Step-1: Create names table and insert values into names table

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 19 18:36:55 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Tue Dec 19 2023 18:18:52 +05:30

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

SQL> CREATE TABLE names(
  2 first_name VARCHAR2(30) NOT NULL,
  3 last_name VARCHAR2(30) NOT NULL
  4 );

Table created.

SQL> INSERT INTO names VALUES('Srinivas','Tej Kiran');

1 row created.

SQL> INSERT INTO names VALUES('Harsha','Vardhan');

1 row created.

SQL> INSERT INTO names VALUES('Hanshith','Venkat');

1 row created.

SQL> SELECT * FROM names;

FIRST_NAME      LAST_NAME
-----
Srinivas        Tej Kiran
Harsha          Vardhan
Hanshith        Venkat
```

Step-2: Check whether rows are inserted or not

```
C:\WINDOWS\system32\cmd. x + v

FIRST_NAME      LAST_NAME
-----
Srinivas        Tej Kiran
Harsha          Vardhan
Hanshith        Venkat

SQL> SELECT LOWER(first_name) FROM names;

LOWER(FIRST_NAME)
-----
srinivas
harsha
hanshith

SQL> SELECT UPPER(first_name) FROM names;

UPPER(FIRST_NAME)
-----
SRINIVAS
HARSHA
HANSHITH

SQL> SELECT INITCAP(first_name) FROM names;

INITCAP(FIRST_NAME)
-----
Srinivas
Harsha
Hanshith

SQL> SELECT CONCAT(first_name,last_name) FROM names;

CONCAT(FIRST_NAME, LAST_NAME)
-----
SrinivasTej Kiran
HarshaVardhan
HanshithVenkat

SQL> SELECT SUBSTR(first_name,1,4) FROM names;
```

224G1A05B8_VASAVI AKANKSHA E

Step-3: Perform ORACLE BUILT-IN FUNCTIONS (i.e. DATE, TIME)

```
C:\WINDOWS\system32\cmd. x + v
FIRST_NAME          LAST_NAME
-----
Srinivas            Tej Kiran
Harsha              Vardhan
Hanshith            Venkat

SQL> SELECT LOWER(first_name) FROM names;

LOWER(FIRST_NAME)
-----
srinivas
harsha
hanshith

SQL> SELECT UPPER(first_name) FROM names;

UPPER(FIRST_NAME)
-----
SRINIVAS
HARSHA
HANSHITH

SQL> SELECT INITCAP(first_name) FROM names;

INITCAP(FIRST_NAME)
-----
Srinivas
Harsha
Hanshith

SQL> SELECT CONCAT(first_name,last_name) FROM names;

CONCAT(FIRST_NAME, LAST_NAME)
-----
SrinivasTej Kiran
HarshaVardhan
HanshithVenkat

SQL> SELECT SUBSTR(first_name,1,4) FROM names;
```

```
C:\WINDOWS\system32\cmd. x + v
HanshithVenkat

SQL> SELECT SUBSTR(first_name,1,4) FROM names;

SUBSTR(FIRST_NAME)
-----
Srin
Hars
Hans

SQL> SELECT LENGTH(first_name) FROM names;

LENGTH(FIRST_NAME)
-----
8
6
8

SQL> SELECT INSTR(first_name,'Ma') FROM names;

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

SQL> SELECT TRIM(' ' FROM first_name) FROM names;

TRIM(' ' FROM FIRST_NAME)
-----
Srinivas
Harsha
Hanshith

SQL> SELECT ROUND(11.111,2) FROM dual;

ROUND(11.111,2)
-----
11.11
```

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v
ROUND(11.111,2)
-----
11.11

SQL> SELECT MOD(11,2) FROM dual;

MOD(11,2)
-----
1

SQL> SELECT SYSDATE FROM dual;

SYSDATE
-----
19-DEC-23

SQL> SELECT MONTHS_BETWEEN(SYSDATE,'19-DEC-2024') FROM dual;

MONTHS_BETWEEN(SYSDATE,'19-DEC-2024')
-----
-12

SQL> SELECT ADD_MONTHS(SYSDATE,12) FROM dual;

ADD_MONTH
-----
19-DEC-24

SQL> SELECT NEXT_DAY(SYSDATE,'TUESDAY') FROM dual;

NEXT_DAY(
-----
26-DEC-23

SQL> SELECT LAST_DAY(SYSDATE) FROM dual;

LAST_DAY(
-----
31-DEC-23
```

```
C:\WINDOWS\system32\cmd. x + v
-----
1

SQL> SELECT SYSDATE FROM dual;

SYSDATE
-----
19-DEC-23

SQL> SELECT MONTHS_BETWEEN(SYSDATE,'19-DEC-2024') FROM dual;

MONTHS_BETWEEN(SYSDATE,'19-DEC-2024')
-----
-12

SQL> SELECT ADD_MONTHS(SYSDATE,12) FROM dual;

ADD_MONTH
-----
19-DEC-24

SQL> SELECT NEXT_DAY(SYSDATE,'TUESDAY') FROM dual;

NEXT_DAY(
-----
26-DEC-23

SQL> SELECT LAST_DAY(SYSDATE) FROM dual;

LAST_DAY(
-----
31-DEC-23

SQL> SELECT CURRENT_TIMESTAMP(3) FROM dual;

CURRENT_TIMESTAMP(3)
-----
19-DEC-23 06:50:30.009 PM +05:30

SQL>
```

END

EXPERIMENT-9

Create some tables and perform KEY CONSTRAINTS (i.e.

224G1A05B8_VASAVI AKANKSHA E

PRIMARY KEY, FOREIGN KEY, UNIQUE, NOT NULL, CHECK, DEFAULT)

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 19 19:01:20 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Tue Dec 19 2023 18:37:02 +05:30

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

SQL> CREATE TABLE student2(
  2 ID NUMBER PRIMARY KEY,
  3 first_name VARCHAR2(25) NOT NULL,
  4 last_name VARCHAR2(25) NOT NULL
  5 );

Table created.

SQL> INSERT INTO student2 VALUES(523,'SIDHU','POLISHETTY');

1 row created.

SQL> INSERT INTO student2 VALUES(519,'ANVITHA','SHETTY');

1 row created.

SQL> SELECT * FROM student2;

   ID FIRST_NAME      LAST_NAME
-----
  523 SIDHU          POLISHETTY
  519 ANVITHA          SHETTY
```

```
C:\WINDOWS\system32\cmd. x + v

   ID FIRST_NAME      LAST_NAME
-----
  523 SIDHU          POLISHETTY
  519 ANVITHA          SHETTY

SQL> CREATE TABLE orders2(
  2 id NUMBER PRIMARY KEY,
  3 order_num NUMBER NOT NULL,
  4 stud_id NUMBER REFERENCES stud(id)
  5 );
CREATE TABLE orders2(
  *
ERROR at line 1:
ORA-00955: name is already used by an existing object

SQL> CREATE TABLE orders4(
  2 id NUMBER PRIMARY KEY,
  3 order_num NUMBER NOT NULL,
  4 student2_id NUMBER REFERENCES student2(id)
  5 );

Table created.

SQL> INSERT INTO orders4 VALUES(11,2,111);
INSERT INTO orders4 VALUES(11,2,111)
*
ERROR at line 1:
ORA-02291: integrity constraint (SYSTEM.SYS_C008408) violated - parent key not found

SQL> INSERT INTO orders4 VALUES(2011,7,112);
INSERT INTO orders4 VALUES(2011,7,112)
*
ERROR at line 1:
ORA-02291: integrity constraint (SYSTEM.SYS_C008408) violated - parent key not found
```

224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v

SQL> CREATE TABLE employees3(
  2 id NUMBER PRIMARY KEY,
  3 name VARCHAR2(50) NOT NULL,
  4 email VARCHAR2(50) UNIQUE
  5 );

Table created.

SQL> INSERT INTO employees3 VALUES(123,'Suresh','suresh123@gmail.com');

1 row created.

SQL> INSERT INTO employees3 VALUES(456,'Sunil','sunil456@gmail.com');

1 row created.

SQL> CREATE TABLE orders5(
  2 id NUMBER PRIMARY KEY,
  3 product_name VARCHAR2(50) NOT NULL,
  4 quantity NUMBER
  5 );

Table created.

SQL> INSERT INTO orders5 VALUES(1,'ABCD',98);

1 row created.

SQL> INSERT INTO orders5 VALUES(2,'UVWX',89);

1 row created.

SQL> CREATE TABLE parts2(
  2 part_id NUMBER PRIMARY KEY,
  3 part_name VARCHAR2(50) NOT NULL,
  4 buy_price NUMBER(9,2) CHECK(buy_price>0)
  5 );
CREATE TABLE parts2(
  *
```

```
C:\WINDOWS\system32\cmd. x + v

SQL> CREATE TABLE parts3(
  2 part_id NUMBER PRIMARY KEY,
  3 part_name VARCHAR2(50) NOT NULL,
  4 buy_price NUMBER(9,2) CHECK(buy_price > 0)
  5 );

Table created.

SQL> INSERT INTO parts3 VALUES(3,'NGL',523);

1 row created.

SQL> INSERT INTO parts3 VALUES(4,'CSK',519);

1 row created.

SQL> CREATE TABLE customers3(
  2 name VARCHAR2(50) NOT NULL,
  3 id NUMBER PRIMARY KEY,
  4 country VARCHAR2(20) DEFAULT 'IND'
  5 );

Table created.

SQL> INSERT INTO customers3(name,id,country) VALUES ('Naveen',1,'USA');

1 row created.

SQL> INSERT INTO customers3(name,id) VALUES('Greeshma',2);

1 row created.

SQL> SELECT * FROM customers3;

NAME                                ID
-----
COUNTRY
-----
```


224G1A05B8_VASAVI AKANKSHA E

```
C:\WINDOWS\system32\cmd. x + v
Table created.
SQL> INSERT INTO parts3 VALUES(3,'NGL',523);
1 row created.
SQL> INSERT INTO parts3 VALUES(4,'CSK',519);
1 row created.
SQL> CREATE TABLE customers3(
2 name VARCHAR2(50) NOT NULL,
3 id NUMBER PRIMARY KEY,
4 country VARCHAR2(20) DEFAULT 'IND'
5 );
Table created.
SQL> INSERT INTO customers3(name,id,country) VALUES ('Naveen',1,'USA');
1 row created.
SQL> INSERT INTO customers3(name,id) VALUES('Greeshma',2);
1 row created.
SQL> SELECT * FROM customers3;
NAME                                ID
-----
COUNTRY
-----
Naveen                                1
USA
Greeshma                             2
IND
SQL>
```

END

EXPERIMENT-10

PL/SQL Program for

calculating the factorial of given number

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 19 19:34:10 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Tue Dec 19 2023 19:01:26 +05:30

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

SQL> SET SERVEROUT ON
SQL> SET VERIFY OFF
SQL> DECLARE
  2  n NUMBER;
  3  fac NUMBER:=1;
  4  n1 NUMBER;
  5  BEGIN
  6  n:=&n;
  7  n1:=n;
  8  WHILE n1>0 LOOP
  9  fac := n1*fac;
 10  n1:=n1-1;
 11  END LOOP;
 12  DBMS_OUTPUT.PUT_LINE('The Factorial of '||n||' is '||fac);
 13  END;
 14  /
Enter value for n: 5
The Factorial of 5 is 120

PL/SQL procedure successfully completed.

SQL> /
```

```
C:\WINDOWS\system32\cmd. x + v
Last Successful login time: Tue Dec 19 2023 19:01:26 +05:30

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

SQL> SET SERVEROUT ON
SQL> SET VERIFY OFF
SQL> DECLARE
  2  n NUMBER;
  3  fac NUMBER:=1;
  4  n1 NUMBER;
  5  BEGIN
  6  n:=&n;
  7  n1:=n;
  8  WHILE n1>0 LOOP
  9  fac := n1*fac;
 10  n1:=n1-1;
 11  END LOOP;
 12  DBMS_OUTPUT.PUT_LINE('The Factorial of '||n||' is '||fac);
 13  END;
 14  /
Enter value for n: 5
The Factorial of 5 is 120

PL/SQL procedure successfully completed.

SQL> /
Enter value for n: 6
The Factorial of 6 is 720

PL/SQL procedure successfully completed.

SQL> /
Enter value for n: 99
The Factorial of 99 is ~

PL/SQL procedure successfully completed.

SQL>
```

END

EXPERIMENT-11

for

PL/SQL Program finding whether the given number is prime or not

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 19 20:05:16 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Tue Dec 19 2023 19:44:29 +05:30

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

SQL> SET SERVEROUT ON
SQL> SET VERIFY OFF
SQL> DECLARE
  2 n NUMBER;
  3 flag NUMBER:=1;
  4 g NUMBER;
  5 g1 NUMBER;
  6 BEGIN
  7 n:=&n;
  8 g1:=n;
  9 g:=2;
 10 FOR g IN 2..g1/2
 11 LOOP
 12 IF mod(n,g) = 0
 13 THEN
 14 flag:=0;
 15 EXIT;
 16 END IF;
 17 END LOOP;
 18 IF flag=1
 19 THEN
 20 DBMS_OUTPUT.PUT_LINE(g1||' is a prime number');
```

```
C:\WINDOWS\system32\cmd. x + v
 4 g NUMBER;
 5 g1 NUMBER;
 6 BEGIN
 7 n:=&n;
 8 g1:=n;
 9 g:=2;
10 FOR g IN 2..g1/2
11 LOOP
12 IF mod(n,g) = 0
13 THEN
14 flag:=0;
15 EXIT;
16 END IF;
17 END LOOP;
18 IF flag=1
19 THEN
20 DBMS_OUTPUT.PUT_LINE(g1||' is a prime number');
21 ELSE
22 DBMS_OUTPUT.PUT_LINE(g1||' is not a prime number');
23 END IF;
24 END;
25 /
Enter value for n: 9
9 is not a prime number

PL/SQL procedure successfully completed.

SQL> /
Enter value for n: 8
8 is not a prime number

PL/SQL procedure successfully completed.

SQL> /
Enter value for n: 7
7 is a prime number

PL/SQL procedure successfully completed.

SQL>
```

END

EXPERIMENT-12

PL/SQL Program for

displaying the Fibonacci series up to an integer

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 19 20:24:09 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Tue Dec 19 2023 20:17:31 +05:30

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

SQL> SET SERVEROUT ON
SQL> SET VERIFY OFF
SQL> DECLARE
  2 first_num NUMBER:=0;
  3 second_num NUMBER:=1;
  4 n NUMBER;
  5 i NUMBER;
  6 temp NUMBER;
  7 BEGIN
  8 n:=&n;
  9 DBMS_OUTPUT.PUT_LINE('SERIES :');
 10 DBMS_OUTPUT.PUT_LINE(first_num);
 11 DBMS_OUTPUT.PUT_LINE(second_num);
 12 FOR i IN 2..n
 13 LOOP
 14 temp := first_num+second_num;
 15 first_num := second_num;
 16 second_num := temp;
 17 DBMS_OUTPUT.PUT_LINE(temp);
 18 END LOOP;
 19 END;
 20 /
```

```
C:\WINDOWS\system32\cmd. x + v

 14 temp := first_num+second_num;
 15 first_num := second_num;
 16 second_num := temp;
 17 DBMS_OUTPUT.PUT_LINE(temp);
 18 END LOOP;
 19 END;
 20 /
Enter value for n: 4
SERIES :
0
1
1
2
3

PL/SQL procedure successfully completed.

SQL> /
Enter value for n: 3
SERIES :
0
1
1
2

PL/SQL procedure successfully completed.

SQL> /
Enter value for n: 5
SERIES :
0
1
1
2
3
5

PL/SQL procedure successfully completed.

SQL> |
```

EXPERIMENT-13

for

END

PL/SQL Program to implement Stored Procedure on table.

EXPERIMENT-14

```
C:\WINDOWS\system32\cmd. x + v
Version 21.3.0.0.0
Copyright (c) 1982, 2021, Oracle. All rights reserved.
Enter user-name: system
Enter password:
Last Successful login time: Tue Dec 19 2023 20:35:18 +05:30

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

SQL> CREATE TABLE sailor2(
  2 id NUMBER PRIMARY KEY,
  3 name VARCHAR2(50) NOT NULL
  4 );

Table created.

SQL> CREATE OR REPLACE PROCEDURE insertuser(id IN NUMBER,name IN VARCHAR2)
  2 AS
  3 BEGIN
  4 INSERT INTO sailor2 VALUES(id,name);
  5 DBMS_OUTPUT.PUT_LINE('Record inserted successfully');
  6 END;
  7 /

Procedure created.

SQL> DECLARE
  2 co NUMBER;
  3 BEGIN
  4 insertuser(23,'Greeshma Sai');
  5 SELECT COUNT(*) INTO co FROM sailor1;
  6 DBMS_OUTPUT.PUT_LINE(co||' Record is inserted successfully');
  7 END;
  8 /

PL/SQL procedure successfully completed.
```

```
SQL> DECLARE
  2 co NUMBER;
  3 BEGIN
  4 insertuser(12,'Anvitha');
  5 SELECT COUNT(*) INTO co FROM sailor2;
  6 DBMS_OUTPUT.PUT_LINE(co||' Record is inserted successfully');
  7 END;
  8 /
Record inserted successfully
2 Record is inserted successfully

PL/SQL procedure successfully completed.

SQL> |
```

END

Stored Function on table

EXPERIMENT-15

PL/SQL Program to implement

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 19 21:04:19 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Tue Dec 19 2023 20:43:43 +05:30

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

SQL> CREATE TABLE section1(
  2 id NUMBER PRIMARY KEY,
  3 course_name VARCHAR2(20) NOT NULL,
  4 strength NUMBER NOT NULL
  5 );

Table created.

SQL> INSERT ALL
  2 INTO section1 VALUES (1,'CSE',50)
  3 INTO section1 VALUES (2,'CSM',60)
  4 INTO section1 VALUES (3,'ECE',75)
  5 SELECT * FROM dual;

3 rows created.

SQL> SET SERVEROUT ON
SQL> SET VERIFY OFF
SQL> CREATE OR REPLACE FUNCTION totalstrength RETURN NUMBER
  2 AS
  3 total NUMBER:=0;
  4 BEGIN
```

```
C:\WINDOWS\system32\cmd. x + v
  2 id NUMBER PRIMARY KEY,
  3 course_name VARCHAR2(20) NOT NULL,
  4 strength NUMBER NOT NULL
  5 );

Table created.

SQL> INSERT ALL
  2 INTO section1 VALUES (1,'CSE',50)
  3 INTO section1 VALUES (2,'CSM',60)
  4 INTO section1 VALUES (3,'ECE',75)
  5 SELECT * FROM dual;

3 rows created.

SQL> SET SERVEROUT ON
SQL> SET VERIFY OFF
SQL> CREATE OR REPLACE FUNCTION totalstrength RETURN NUMBER
  2 AS
  3 total NUMBER:=0;
  4 BEGIN
  5 SELECT sum(strength) INTO total FROM section1;
  6 return total;
  7 END;
  8 /

Function created.

SQL> DECLARE
  2 answer NUMBER;
  3 BEGIN
  4 answer:=totalstrength();
  5 DBMS_OUTPUT.PUT_LINE('Total strength of students is '||answer);
  6 END;
  7 /

Total strength of students is 185

PL/SQL procedure successfully completed.

SQL>
```

EXPERIMENT-16

PL/SQL Program to implement

END

Trigger on table

EXPERIMENT-17

PL/SQL Program to implement

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 19 21:16:29 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Tue Dec 19 2023 21:04:27 +05:30

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

SQL> CREATE TABLE instructor7(
  2 id NUMBER PRIMARY KEY,
  3 name VARCHAR2(50) NOT NULL,
  4 dept_name VARCHAR2(20) NOT NULL,
  5 salary NUMBER(10,2) CHECK(salary>10000)
  6 );

Table created.

SQL> INSERT ALL
  2 INTO instructor7 VALUES
  3 ;

*
ERROR at line 3:
ORA-00936: missing expression

SQL> INSERT ALL
  2 INTO instructor7 VALUES(1,'Anirudh','CSE',50000)
  3 INTO instructor7 VALUES(2,'Maya','CSM',70000)
  4 INTO instructor7 VALUES(3,'Sidhu','ECE',75000)
```

```
C:\WINDOWS\system32\cmd. x + v

SQL> INSERT ALL
  2 INTO instructor7 VALUES(1,'Anirudh','CSE',50000)
  3 INTO instructor7 VALUES(2,'Maya','CSM',70000)
  4 INTO instructor7 VALUES(3,'Sidhu','ECE',75000)
  5 INTO instructor7 VALUES(4,'Anvitha','EEE',80000)
  6 SELECT * FROM dual;

4 rows created.

SQL> CREATE OR REPLACE TRIGGER display_changes
  2 BEFORE UPDATE ON instructor7
  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.

SQL> DECLARE
  2 tot_rows NUMBER;
  3 BEGIN
  4 UPDATE instructor7
  5 SET salary=salary*1.5;
  6 IF sql%notfound THEN
  7 DBMS_OUTPUT.PUT_LINE('no instructors updated');
  8 ELSIF sql%found THEN
  9 tot_rows:=sql%rowcount;
 10 DBMS_OUTPUT.PUT_LINE(tot_rows||' instructors updated');
 11 END IF;
 12 END;
 13 /

PL/SQL procedure successfully completed.
```

EXPERIMENT-18

PL/SQL Program to implement

END

Cursor on table

EXPERIMENT-19

PL/SQL Program to implement

```
C:\WINDOWS\system32\cmd. x + v
Microsoft Windows [Version 10.0.22621.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandu>sqlplus

SQL*Plus: Release 21.0.0.0.0 - Production on Tue Dec 19 21:36:03 2023
Version 21.3.0.0.0

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

Enter user-name: system
Enter password:
Last Successful login time: Tue Dec 19 2023 21:16:36 +05:30

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

SQL> CREATE TABLE customer6(
  2 id NUMBER PRIMARY KEY,
  3 name VARCHAR2(30) NOT NULL,
  4 age NUMBER(3) NOT NULL,
  5 salary NUMBER(10,2) NOT NULL
  6 );

Table created.

SQL> DECLARE
  2 tot_rows NUMBER;
  3 BEGIN
  4 UPDATE customer6 SET salary=salary*1.5;
  5 IF sql%notfound THEN
  6 DBMS_OUTPUT.PUT_LINE('No customers updated');
  7 ELSIF sql%found THEN
  8 tot_rows := sql%rowcount;
  9 DBMS_OUTPUT.PUT_LINE(tot_rows||' customers updated');
  10 END IF;
  11 END;
  12 /
```

```
C:\WINDOWS\system32\cmd. x + v
  4 UPDATE customer6 SET salary=salary*1.5;
  5 IF sql%notfound THEN
  6 DBMS_OUTPUT.PUT_LINE('No customers updated');
  7 ELSIF sql%found THEN
  8 tot_rows := sql%rowcount;
  9 DBMS_OUTPUT.PUT_LINE(tot_rows||' customers updated');
  10 END IF;
  11 END;
  12 /

PL/SQL procedure successfully completed.

SQL> INSERT ALL
  2 INTO customer6 VALUES(1,'Arun Neelakandan',22,60000)
  3 INTO customer6 VALUES(2,'Darshana',33,70000)
  4 INTO customer6 VALUES(3,'Nithya',23,65000)
  5 INTO customer6 VALUES(4,'Maya',25,60000)
  6 SELECT * FROM dual;

4 rows created.

SQL> DECLARE
  2 c_id customer6.id%type;
  3 c_name customer6.name%type;
  4 c_age customer6.age%type;
  5 CURSOR c_customers IS
  6 SELECT id,name,age FROM customer6;
  7 BEGIN
  8 OPEN c_customers;
  9 LOOP
  10 FETCH c_customers INTO c_id,c_name,c_age;
  11 EXIT WHEN c_customers%notfound;
  12 DBMS_OUTPUT.PUT_LINE(c_id||' '||c_name||' '||c_age);
  13 END LOOP;
  14 CLOSE c_customers;
  15 END;
  16 /

PL/SQL procedure successfully completed.
```

EXPERIMENT-20

PL/SQL Program to implement

END