EXPERIMENT – 1

```
Microsoft Windows (Version 18.8.22621.2715)
(c) Microsoft Corporation. All rights reserved.

C:\Users\dandus\sqlplus

SQL-Plus: Release 21.8.8.8.8.9 - Production on Sun Dec 17 19:32:28 2823

Version 21.3.8.8.8

Copyright (c) 1982, 2821, Oracle. All rights reserved.

Enter user-name: system
Enter password:
Last Successful login time: Sun Dec 17 2823 19:86:34 +85:38

Connected to:
Gracle Database 21.Express Edition Release 21.8.8.8 - Production

Version 21.3.8.8 8

SQL-> CREATE TABLE customers1 (
2 customer_name VaRCHAPO NOT NULL,
4 city VARCHAR2(58) NOT NULL,
4 city VARCHAR2(58) NOT NULL,
5 quantity NUMBER NOT NULL,
6 quantity NUMBER NOT NULL,
7 production NULL,
8 production NULL,
9 quantity NUMBER NOT NULL,
10 quantity NUMBER NOT NULL,
11 quantity NUMBER NOT NULL,
12 quantity NUMBER NOT NULL,
13 quantity NUMBER NOT NULL,
14 quantity NUMBER NOT NULL,
15 quantity NUMBER NOT NULL,
16 quantity NUMBER NOT NULL,
17 purple quantity NUMBER NOT NULL,
18 quantity NUMBER NOT NULL,
19 purple ROT NULL,
20 purple ROT NULL,
31 quantity NUMBER NOT NULL,
41 quantity NUMBER NOT NULL,
42 quantity NUMBER NOT NULL,
43 quantity NUMBER NOT NULL,
44 quantity NUMBER NOT NULL,
55 quantity NUMBER NOT NULL,
56 quantity NUMBER NOT NULL,
57 quantity NUMBER NOT NULL,
58 quantity NUMBER NOT NULL,
59 quantity NUMBER NOT NULL,
50 quantity NUMBER NOT NULL,
51 quantity NUMBER NOT NULL,
52 quantity NUMBER NOT NULL,
53 quantity NUMBER NOT NULL,
54 quantity NUMBER NOT NULL,
55 quantity NUMBER NOT NULL,
56 quantity NUMBER NOT NULL,
57 quantity NUMBER NOT NULL,
58 quantity NUMBER NOT NULL,
59 quantity NUMBER NOT NULL,
50 quantity NUMBER NOT NULL,
51 quantity NUMBER NOT NULL,
54 quantity NUMBER NOT NULL,
56 quantity NUMBER NOT NULL,
57 quantity NUMBER NOT NULL,
58 quantity NUMBER NOT NULL,
59 quantity NUMBER NOT NULL,
50 qua
```

```
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
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;
                                          Null? Type
 Name
 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
```

```
Table created.

$QL> INSERT ALL
2 INTO fruits2(fruit_name,color)
3 VALUSE(faple:, Red')
4 INTO fruits2(fruit_name,color)
5 VATO fruits2(fruit_name,color)
6 INTO fruits2(fruit_name,color)
7 VALUSE(Banama:, 'vellow')
8 SELECT 1 FROM dual;
8 Tows created.

$QL> SELECT **FROM fruits2;
FROUT_NAME

COLOR

Apple
Red

Orange
Orange
Orange
Banana
Yellow

$QL> CREATE TABLE parts2(
2 part_id NUMBER() NOT NULL,
4 Lead_time NUMBER() NOT NULL,
5 cost NUMBER() ON NULL,
6 status NUMBER() ON NULL,
6 status NUMBER() ON NULL,
7 parlank NET(part_id)
8 );
7 Table created.
```

```
| Count | Coun
```

Step – 1: create student table

Step - 2: Insert few rows into student table

Step-3: Check whether rows are inserted or not

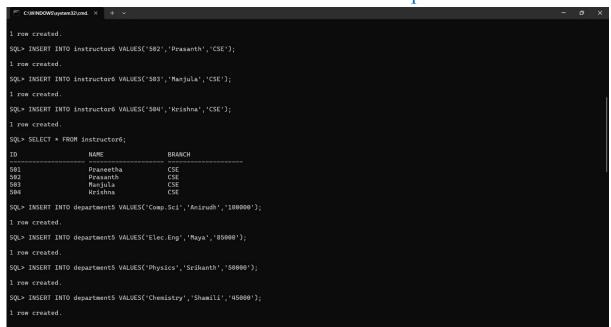
Step-4: Create view of name teacher with name, roll number constraints and check whether rows are inserted or not

```
| Commonwealthman | Commonweal
```

END

STEP-1: Create Instructor table and department table

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

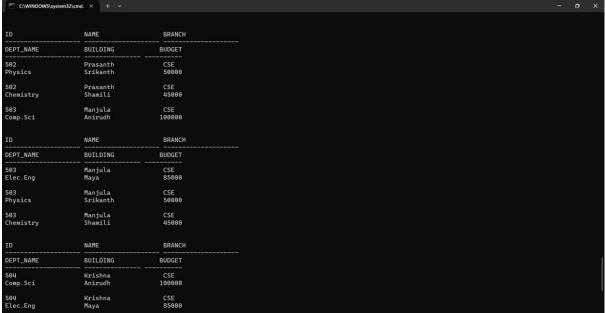


STEP-3: Perform RELATIONAL SET Operations

C\WINDOWS\system	n32\cmd. × + v	
Krishna		
SQL> SELECT * FF		
2 CROSS JOIN		PRANCU
ID	NAME	BRANCH
DEPT_NAME	BUILDING	BUDGET
501 Comp.Sci	Praneetha Anirudh	CSE 100000
501	Praneetha	CSE
Elec.Eng	Maya	85000
501 Physics	Praneetha Srikanth	CSE 50000
Physics	SPIKANTN	50000
ID	NAME	BRANCH
DEPT_NAME	BUILDING	BUDGET
501 Chemistry	Praneetha Shamili	CSE 45000
502 Comp.Sci	Prasanth Anirudh	CSE 100000
502	Prasanth	CSE
Elec.Eng	Maya	85000
ID	NAME	BRANCH
DEPT_NAME	BUILDING	BUDGET
502	Prasanth	CSE
Physics	Srikanth	50000

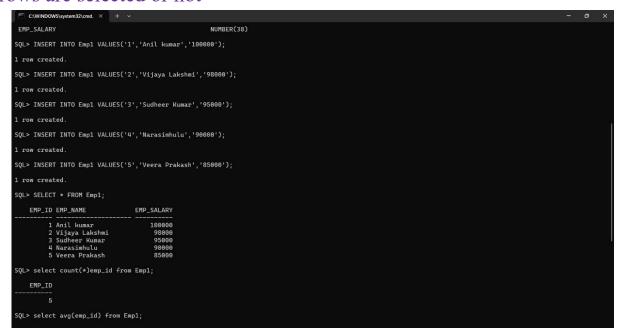
C:\WINDOWS\syste	m32\cmd. × + v	
ID	NAME	BRANCH
DEPT_NAME	BUILDING	BUDGET
502	Prasanth	CSE
Physics	Srikanth	50000
502 Chemistry	Prasanth Shamili	CSE 45000
503 Comp.Sci	Manjula Anirudh	CSE 100000
Comp. SCI	Alliudii	100000
ID	NAME	BRANCH
DEPT_NAME	BUILDING	BUDGET
503 Elec.Eng	Manjula	CSE 85000
	Maya	
503 Physics	Manjula Srikanth	CSE 50000
503	Manjula	CSE
Chemistry	Shamili	45000
ID	NAME	BRANCH
DEPT_NAME	BUILDING	BUDGET
504	Krishna	CSE
Comp.Sci	Anirudh	100000
504 Elec.Eng	Krishna Maya	CSE 85000
504	Krishna	CSE
Physics	Srikanth	50000





Step-1: Create employee table

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



Step-3: Implement 5 aggregate operations

Step-1: Create student table and blocks table

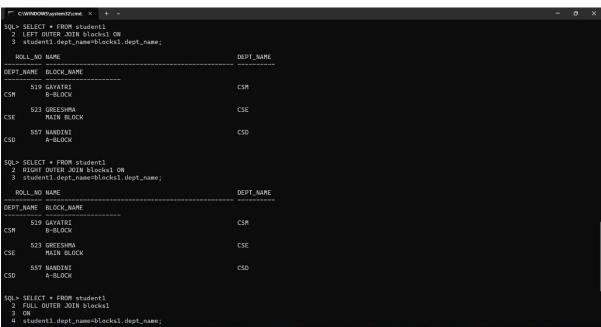
```
### CWWNDOWPROPRIED | Page | P
```

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



Step-3: Perform JOIN OPERATIONS

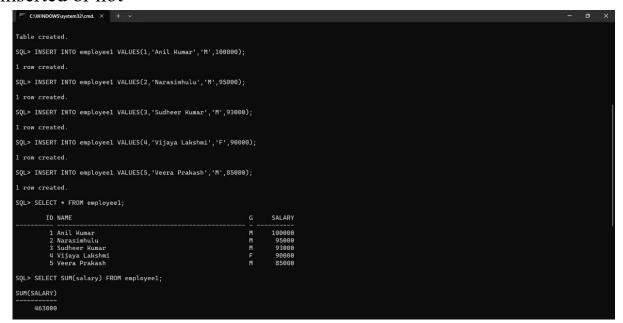






Step-1:Create Employee Table

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



Step-3: Perform AGGREGATE OPERATIONS

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

```
Microsoft Windows [Version 18.0.22621.2861]
(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 18:18:52.405:30

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('Harsha', 'Vardhan');
1 row created.

SQL- INSERT INTO names VALUES('Harsha', 'Vardhan');
1 row created.

SQL- INSERT INTO names VALUES('Harsha', 'Vardhan');
1 row created.

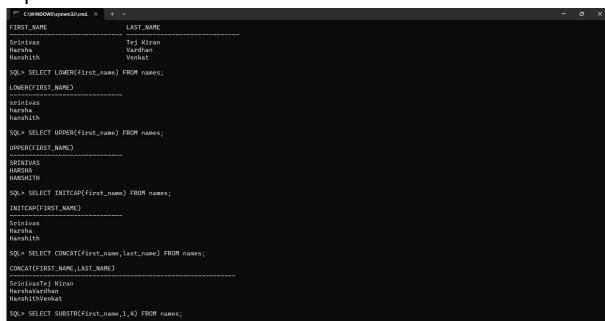
SQL- INSERT INTO names VALUES('Harsha', 'Vardhan');
1 row created.

SQL- INSERT INTO names VALUES('Harsha', 'Vardhan');
1 row created.

SQL- INSERT INTO names VALUES('Harsha', 'Vardhan');
1 row created.

SQL- SELECT * FROM names;
FIRST_NAME LAST_NAME
```

Step-2: Check whether rows are inserted or not



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

```
FIRST_NAME
                                       LAST_NAME
Srinivas
Harsha
Hanshith
                                       Tej Kiran
Vardhan
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
CONCAT(FIRST_NAME,LAST_NAME)
SrinivasTej Kiran
HarshaVardhan
HanshithVenkat
SQL> SELECT SUBSTR(first_name,1,4) FROM names;
```

EXPERIMENT-9

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

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

PL/SQL Program for calculating the factorial of given number

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

PL/SQL Program for displaying the Fibonacci series up to an integer

END

PL/SQL Program to implement Stored Procedure on table.

```
SQL> DECLARE

2 co NUMBER;

3 BEGIN

4 insertuser(12, 'Anvitha');

5 SELECT COUNTY(*) INTO co FROM sailor2;

6 DBMS_OUDPUT_PUT_LINE(co||' Record is inserted successfully');

7 END;

8 /

Record inserted successfully

PL/SQL procedure successfully completed.

SQL> |
```

END

PL/SQL Program to implement Stored Function on table

```
| Counted Minder PRIMARY Nety | Course name VARCHARZ(20) NOT NULL, | Strength NUMBER NOT NULL, | Strength NUMBER NOT NULL, | Strength NUMBER NOT NULL | Strength NUMBER (1, 'CSE', 50) | STREET ALL | STREET NUMBER (2, 'CSM', 60) | STREET NUMBER (3, 'ECE', 75) | SELECT * FROM dual; | STREET NUMBER (3, 'ECE', 75) | SELECT * FROM Dual; | STREET NUMBER NOT NUMBER NUMBER: 0, STREET NUMBER; | STREET NUMBER; |
```

PL/SQL Program to implement Trigger on table

```
Microsoft Windows (Version 10.0.2021.2861)
(c) Microsoft Corporation. All rights reserved.
(c) Microsoft Corporation. All rights reserved.
(c) Wiseraldandu-sqlplus

SQL*Plus: Release 21.0.0.0.0.0 - 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 21:84:27 +85:30

Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0 - Production
Version 21.3.0.0.0

SQL> GREATE TABLE instructor?(
2 id NUMBER PRIMARY MEY,
3 name VARCHARZ(26) NOT NULL,
4 dept.name VARCHARZ(26) NOT NULL,
5 salary NUMBER(10,2) CHECK(salary>10000)
6 );

Table created.

SQL> INSERT ALL
2 INTO instructor? VALUES (1, 'Anirudh', 'CSE', 50000)
3 INTO instructor? VALUES (1, 'Anirudh', 'CSE', 50000)
3 INTO instructor? VALUES (1, 'Anirudh', 'CSE', 50000)
4 INTO instructor? VALUES (3, 'Anirudh', 'CSE', 50000)
4 INTO instructor? VALUES (3, 'Anirudh', 'CSE', 50000)
5 INTO instructor? VALUES (3, 'Anirudh', 'CSE', 50000)
```

PL/SQL Program to implement Cursor on table

```
UDDATE customer6 SET salary=salary+1.5;

If solwnotfound THEN

Debts_OUTPUT.PUT_LINE(No customers updated');

Editors : salary=salary+1.5;

Str. sols : salary=salary=salary+1.5;

Ext. The sols : salary=salary+1.5;

PL/SQL procedure successfully completed.

SQL> INSERT ALL

2 INTO customer6 VALUESC1, 'Arun Neelakandan', 22,69890)

3 INTO customer6 VALUESC2, 'Darshana', 33,78980)

STINO customer6 VALUESC2, 'Marbya', 23,68890)

SELECT * FROM dual;

4 rows created.

SQL> DECLARE

2 c_id customer6 VALUES(4, 'Maya', 25,6890)

SCLECT * FROM dual;

4 rows created.

SQL> DECLARE

2 c_id customer6.namettype;

4 c_age customer6.namettype;

9 c_age customer6.agettype;

10 CUBSOR c_customers

10 POPEN c_customers;

10 POPEN c_customers;

11 EXIT WHEN c_customers;

12 DBMS_OUTPUT.PUT_LINE(c_id||' '||c_age|);

13 END LOOP;

14 CLOSE c_customers;

15 END;

16 IO)

PL/SQL procedure successfully completed.
```

END