Lab Record
of

ADVANCED RDBMS (IB401)

BACHELOR OF TECHNOLOGY
In

COMPUTER SCIENCE AND ENGINEERING

Session 2022-23

Submitted to: -

Ms. Srabanti Maji
Assistant Professor,
School of Computing

Submitted by: -

Name: Shashant Nichantak
Roll No.: 190111003
Sap ID: 1000012451
Class/Section: B.Tech CSE-CSF

SCHOOL OF COMPUTING
DIT UNIVERSITY, DEHRADUN
(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)
Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.
Aug-Dec.2022

INDEX

S.NO.	NAME OF THE EXPERIMENTS	DATE OF CONDUCTION	DATE OF SUBMISSION	REMARKS
1	**USE OF DDL COMMANDS IN ORACLE**			
2	**USE OF DML COMMANDS IN ORACLE**			
3	GRADED LAB 1			
4	**Use of view in oracle.**			
5	GRADED LAB 2			
6	**Use of PL/SQL variables, reserved words, identifiers and anchored data types.**			
7	**Using clause in oracle**			
8	GRADED LAB 3			
9	**Use of Control Statement, Case and Loop in Pl/SQL**			
10	**Use of Procedure and Function in Pl/SQL**			
11	**Use of Trigger in Pl/SQL**			
12	**Exception handling and join in PL/SQL**			

Experiment – 1

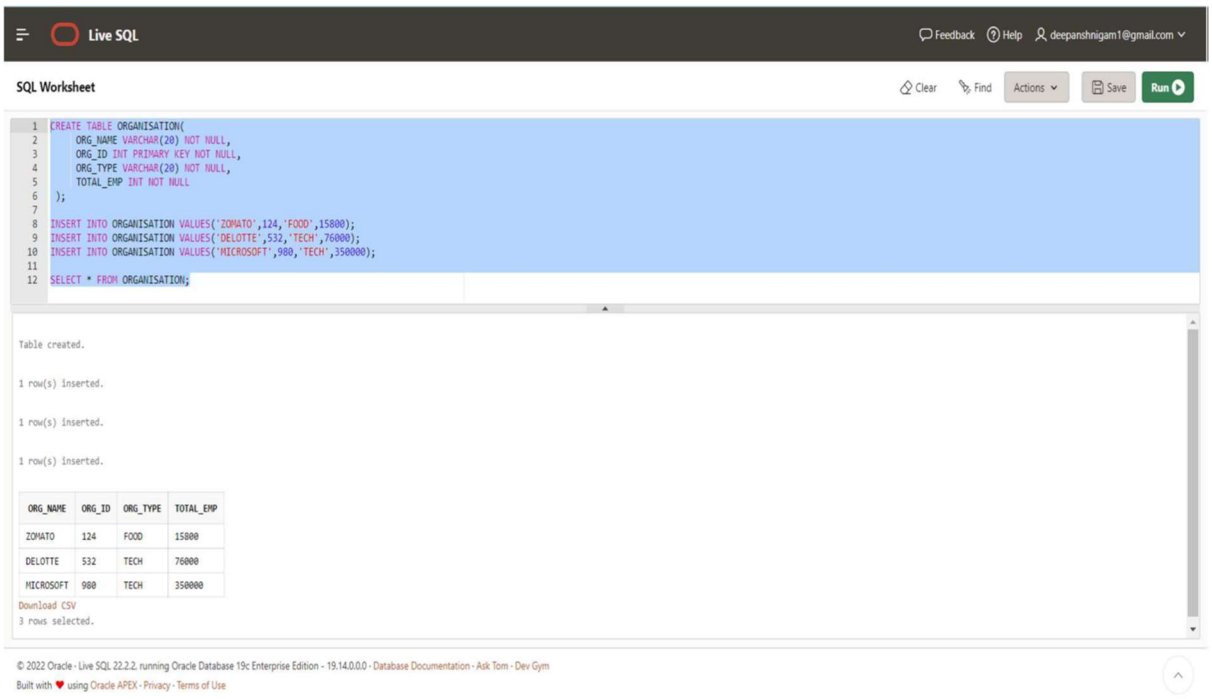
Objective - USE OF DDL COMMANDS IN ORACLE

i. Create table Organisation, insert records and display its content.

Queries –

```
CREATE TABLE ORGANISATION(
 ORG_NAME VARCHAR(20) NOT NULL,
 ORG_ID INT PRIMARY KEY NOT NULL,
 ORG_TYPE VARCHAR(20) NOT NULL,
 TOTAL_EMP INT NOT NULL
 );
INSERT INTO ORGANISATION VALUES('ZOMATO',124,'FOOD',15800);
INSERT INTO ORGANISATION VALUES('DELOTTE',532,'TECH',76000);
INSERT INTO ORGANISATION VALUES('MICROSOFT',980,'TECH',350000);
SELECT * FROM ORGANISATION;
```

Screenshot ->



ii. Create table Customer and utilise various DDL commands in it.

Queries ->

```
CREATE TABLE Customer (
PersonID int,
Firstname varchar(255),
Lastname varchar(255),
Address varchar(255),
Country varchar (255)
);
ALTER TABLE Customer
ADD ItemsPurchased int;
INSERT INTO Customer VALUES (1, 'Rachel', 'Adams', 'Munich', 'Germany', 9);
INSERT INTO Customer VALUES (2, 'Ram', 'Patel', 'Delhi', 'India', 3);
INSERT INTO Customer VALUES (3, 'Michael', 'Scott', 'Sydney', 'Australia', 5);
INSERT INTO Customer VALUES (4, 'Jim', 'Halpert', 'California', 'America', 7);
SELECT * FROM Customer;
```

Screenshot ->

SQL Worksheet

```
Table created.

Table altered.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

PERSONID	FIRSTNAME	LASTNAME	ADDRESS	COUNTRY	ITEMSPURCHASED
1	Rachel	Adams	Munich	Germany	9
2	Ram	Patel	Delhi	India	3
3	Michael	Scott	Sydney	Australia	5
4	Jim	Halpert	California	America	7

```
Download CSV
4 rows selected.
```

Experiment-2

Objective : USE OF DML COMMANDS IN ORACLE

(i) Create table Student (Roll, Name, Age, Marks, City), insert minimum 10

records and display its content,

Answer-

Query

```
--student table
CREATE TABLE Students
(
Roll INT,
Name VARCHAR(20),
Age INT,
Marks INT,
City VARCHAR(20),
PRIMARY KEY (Roll)
);
--insert query
INSERT INTO Students VALUES(1,'Deepansh',21,99,'Mumbai');
INSERT INTO Students VALUES(2,'Harshit',21,46,'Agra');
INSERT INTO Students VALUES(3,'Ashu',21,84,'Chennai');
INSERT INTO Students VALUES(4,'Shashant',21,65,'Dehradun');
INSERT INTO Students VALUES(5,'Devesh',21,87,'Meerut');
INSERT INTO Students VALUES(6,'Dhruv',21,98,'Mumbai');
INSERT INTO Students VALUES(7,'Yatharth',21,56,'Mumbai');
INSERT INTO Students VALUES(8,'Kunal',21,53,'Mumbai');
```

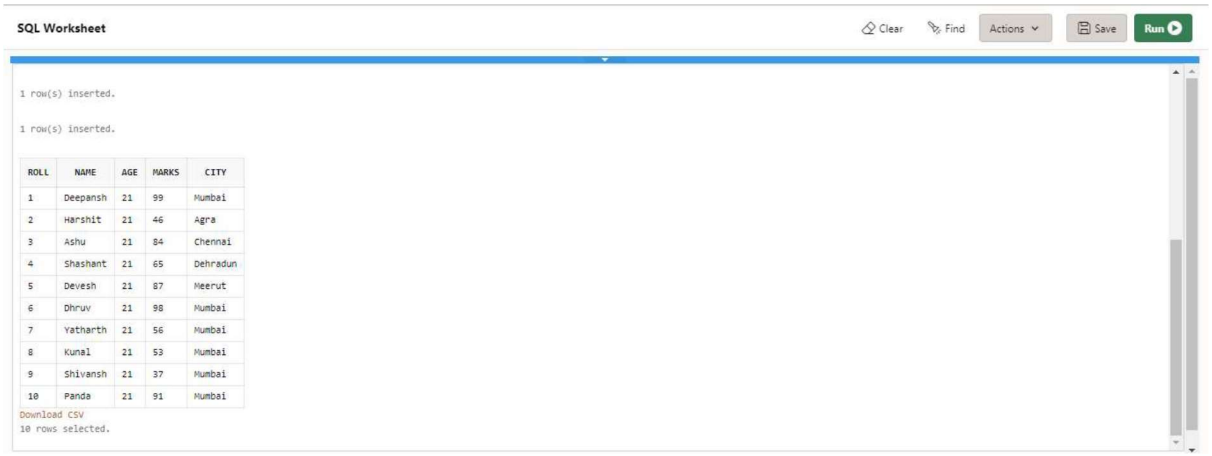
INSERT INTO Students VALUES(9,'Shivansh',21,37,'Mumbai');

INSERT INTO Students VALUES(10,'Panda',21,91,'Mumbai');

--result query

SELECT * from Students ORDER BY Roll ASC;

Screenshot



(ii) Perform DML operation in the Student table to change the marks of specific

Student and display the result.

Answer –

Query

SELECT * from Students ORDER BY Roll ASC;

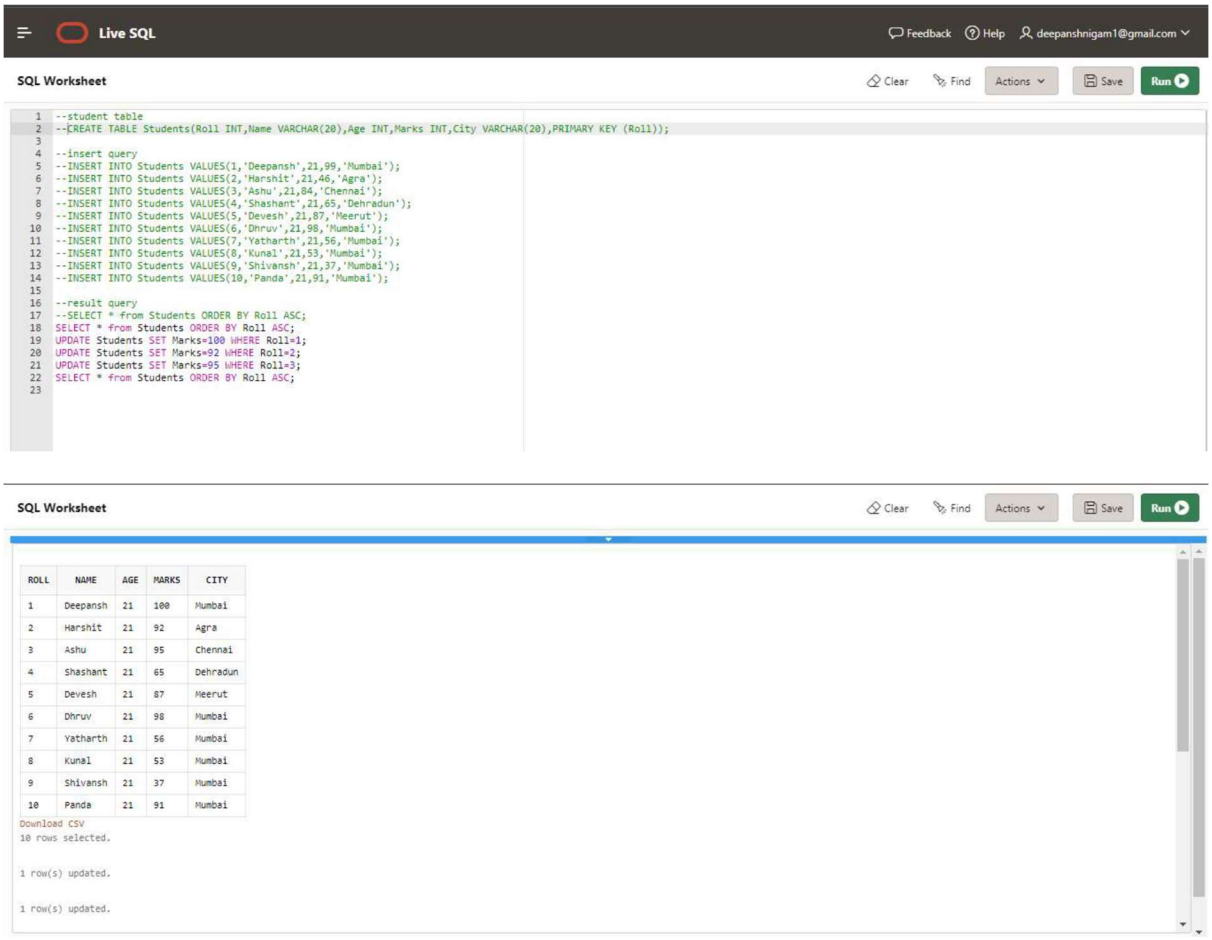
UPDATE Students SET Marks=100 WHERE Roll=1;

UPDATE Students SET Marks=92 WHERE Roll=2;

UPDATE Students SET Marks=95 WHERE Roll=3;

SELECT * from Students ORDER BY Roll ASC;

Screenshot



IB401 GRADED LAB1

1. Create a table Person_Info(Person_ID,PName,Page,Paddress),

Create another table Order_Info(Order_number,Person_ID,City,Quantity).Here Person_ID is

primary key in Person_Info Table and Foreign key in Order_Info table,Order_number is

primary key in Order_Info table Insert minimum 5 records in each table and display the results.

Answer –

Queries

```
-- person info table

CREATE TABLE Person_info(Person_ID INT,PName VARCHAR(20),Page
INT,Paddress

VARCHAR(20),PRIMARY

KEY(Person_ID));

INSERT INTO Person_info VALUES(1,'Deepansh',1,'Delhi');

INSERT INTO Person_info VALUES(2,'Harhsit',1,'Meerut');

INSERT INTO Person_info VALUES(3,'Ashu',1,'Jamshedpur');

INSERT INTO Person_info VALUES(4,'Shashant',1,'Bhuvneshwar');

INSERT INTO Person_info VALUES(5,'Devesh',1,'Haldwani');

-- order table

CREATE TABLE Order_info(Order_number INT,Person_ID INT,City
VARCHAR(20),Quantity

INT,PRIMARY KEY

(Order_number),FOREIGN KEY(Person_ID) REFERENCES Person_info(Person_ID));

INSERT INTO Order_info VALUES(1,1,'Delhi',3);

INSERT INTO Order_info VALUES(2,2,'Meerut',4);
```

INSERT INTO Order_info VALUES(3,3,'Jamshedpur',3);

INSERT INTO Order_info VALUES(4,4,'Bhuvneshwar',1);

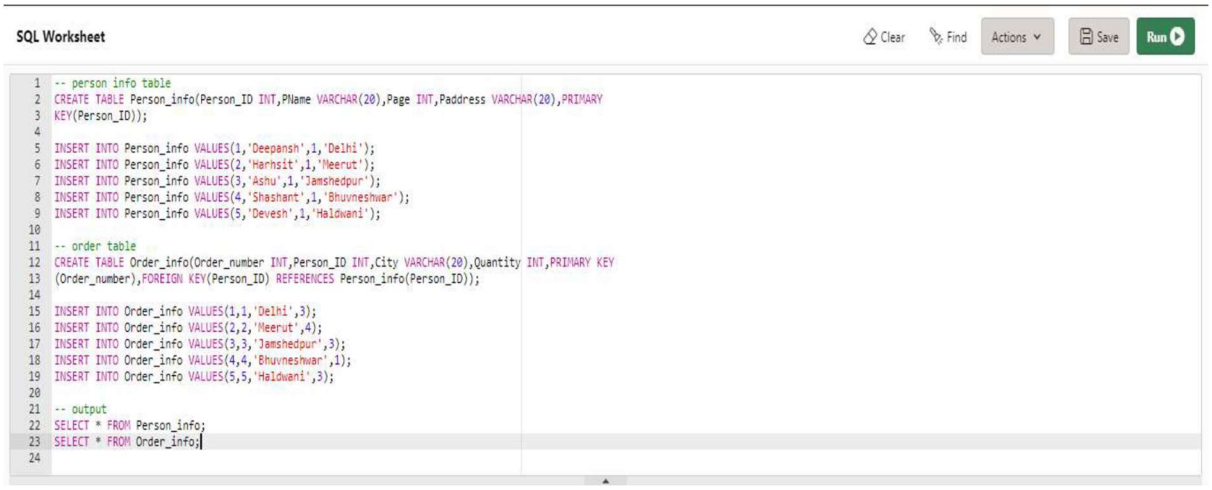
INSERT INTO Order_info VALUES(5,5,'Haldwani',3);

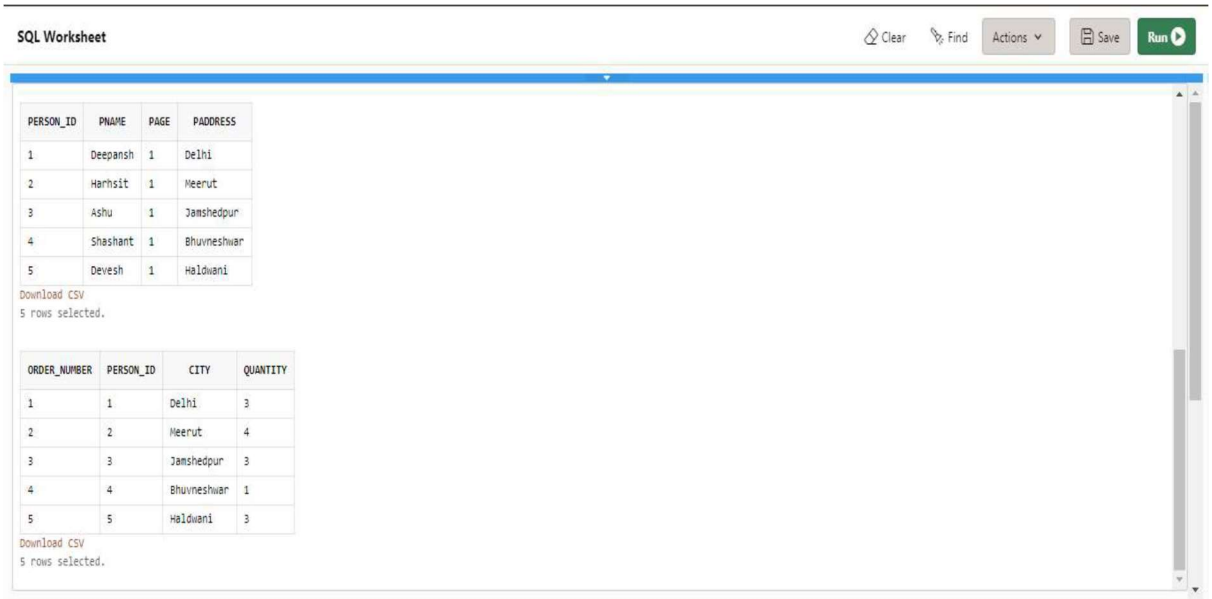
-- output

SELECT * FROM Person_info;

SELECT * FROM Order_info;

Screenshot





Experiment-3

Objective : Use of view in oracle.

Create two Table Employee (EID, Fname, Lname, Age, City, Country),

Order(OID,EID,Qty,Price).Insert minimum 6 records in each table. Create the view to display

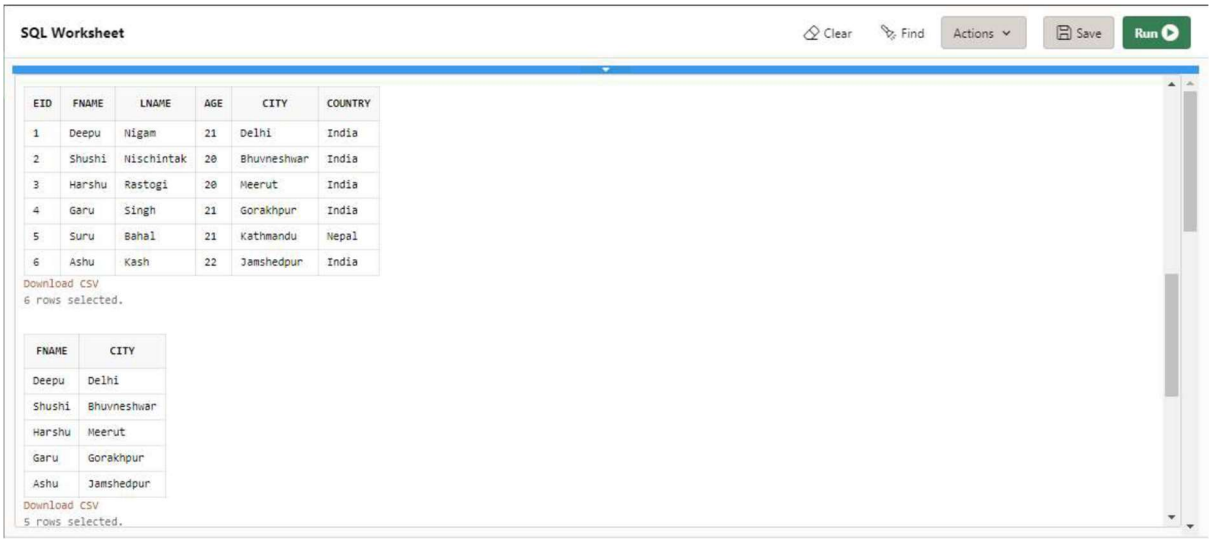
the result.

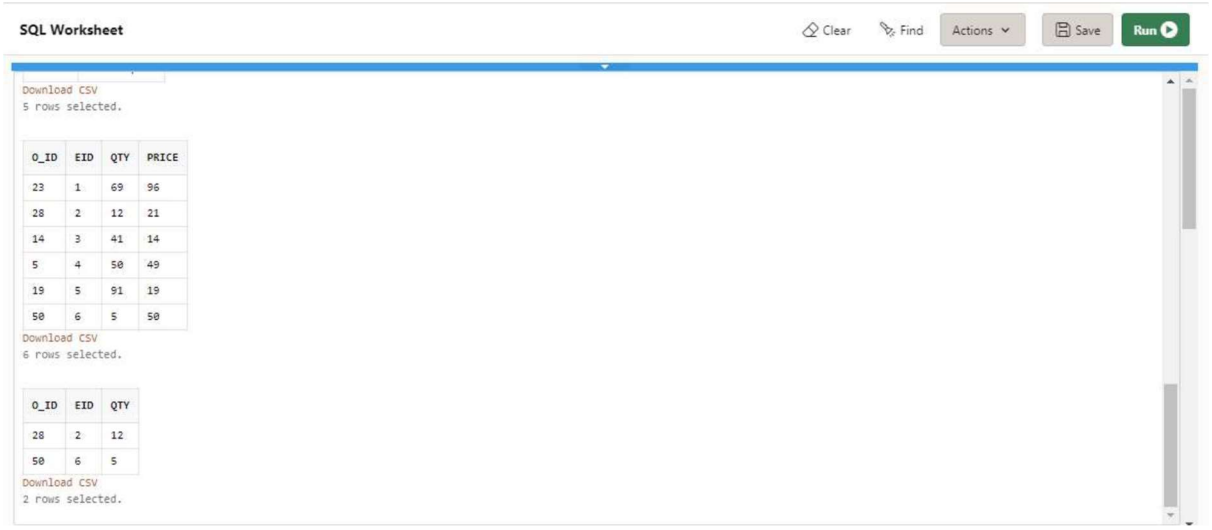
Query:

```
--create table employee

CREATE TABLE Employee (

EID int,

FName varchar2(20),

LName varchar2(20),

Age int,

City varchar(255),

Country varchar2(20)

);

--insert in table employee

INSERT INTO Employee VALUES ('1', 'Deepu', 'Nigam', '21', 'Delhi', 'India');

INSERT INTO Employee VALUES ('2', 'Shushi', 'Nischintak', '20', 'Bhuvneshwar',
'India');

INSERT INTO Employee VALUES ('3', 'Harshu', 'Rastogi', '20', 'Meerut', 'India');

INSERT INTO Employee VALUES ('4', 'Garu', 'Singh', '21', 'Gorakhpur', 'India');

INSERT INTO Employee VALUES ('5', 'Suru', 'Bahal', '21', 'Kathmandu', 'Nepal');

INSERT INTO Employee VALUES ('6', 'Ashu', 'Kash', '22', 'Jamshedpur', 'India');

--create table order

CREATE TABLE Orderr (

O_id int,
```

```
EID int,
Qty int,
Price int
);
--insert in table orderr
INSERT INTO Orderr VALUES ('23', '1', '69', '96');
INSERT INTO Orderr VALUES ('28', '2', '12', '21');
INSERT INTO Orderr VALUES ('14', '3', '41', '14');
INSERT INTO Orderr VALUES ('05', '4', '50', '49');
INSERT INTO Orderr VALUES ('19', '5', '91', '19');
INSERT INTO Orderr VALUES ('50', '6', '05', '50');
--create view indianemployee
CREATE VIEW IndianEmployee AS
SELECT FName, City
FROM Employee
WHERE Country = 'India';
--create view smallquantity
CREATE VIEW SmallQuantity AS
SELECT O_id,EID,Qty
FROM Orderr
WHERE Qty < 40;
--display results
SELECT * From Employee;
SELECT * From IndianEmployee;
SELECT * From Orderr;
SELECT * From SmallQuantity;
```

Screenshot:


SQL Worksheet
Clear
Find
Actions
Save
Run
EID FNAME LNAME AGE CITY COUNTRY
1 Deepu Nigam 21 Delhi India
2 Shushi Nischintak 20 Bhuvneshwar India
3 Harshu Rastogi 20 Meerut India
4 Garu Singh 21 Gorakhpur India
5 Suru Bahal 21 Kathmandu Nepal
6 Ashu Kash 22 Jamshedpur India
Download CSV
6 rows selected.
FNAME CITY
Deepu Delhi
Shushi Bhuvneshwar
Harshu Meerut
Garu Gorakhpur
Ashu Jamshedpur
Download CSV
5 rows selected.


SQL Worksheet
Clear
Find
Actions
Save
Run
Download CSV
5 rows selected.
O_ID EID QTY PRICE
23 1 69 96
28 2 12 21
14 3 41 14
5 4 50 49
19 5 91 19
50 6 5 50
Download CSV
6 rows selected.
O_ID EID QTY
28 2 12
50 6 5
Download CSV
2 rows selected.

IB401 GRADED LAB2

Questions: Create a table Organization (OrgID,PrName,Quantity), Create a procedure to

insert records in this table call the procedure by PLSQL program.Table must have

minimum 10 records. Then display the content of the table.

CODE:

```
CREATE TABLE Organizationdetail(ORG_ID NUMBER (5), PRNAME VARCHAR2 (20),

QUANTITY NUMBER (10));

CREATE OR REPLACE PROCEDURE insertOrganizationdetail(

ORG_ID in NUMBER,

PRNAME in VARCHAR2,

QUANTITY in NUMBER

)

IS

begin

INSERT INTO Organizationdetail VALUES(ORG_ID, PRNAME, QUANTITY);

COMMIT;

end;

/

BEGIN

insertOrganizationdetail(1001,'Garima singh', 10);

insertOrganizationdetail(1002,'Ashutosh kashyap', 5);

insertOrganizationdetail(1003,'Shardul', 15);

insertOrganizationdetail(1004,'Harsh', 8);

insertOrganizationdetail(1005,'Mudit', 12);
```

insertOrganizationdetail(1006,'Ayushi', 9);

insertOrganizationdetail(1007,'Kritika', 10);

insertOrganizationdetail(1008,'Avhiya', 15);

insertOrganizationdetail(1009,'Prassidha', 6);

insertOrganizationdetail(1010,'Prathu', 7);

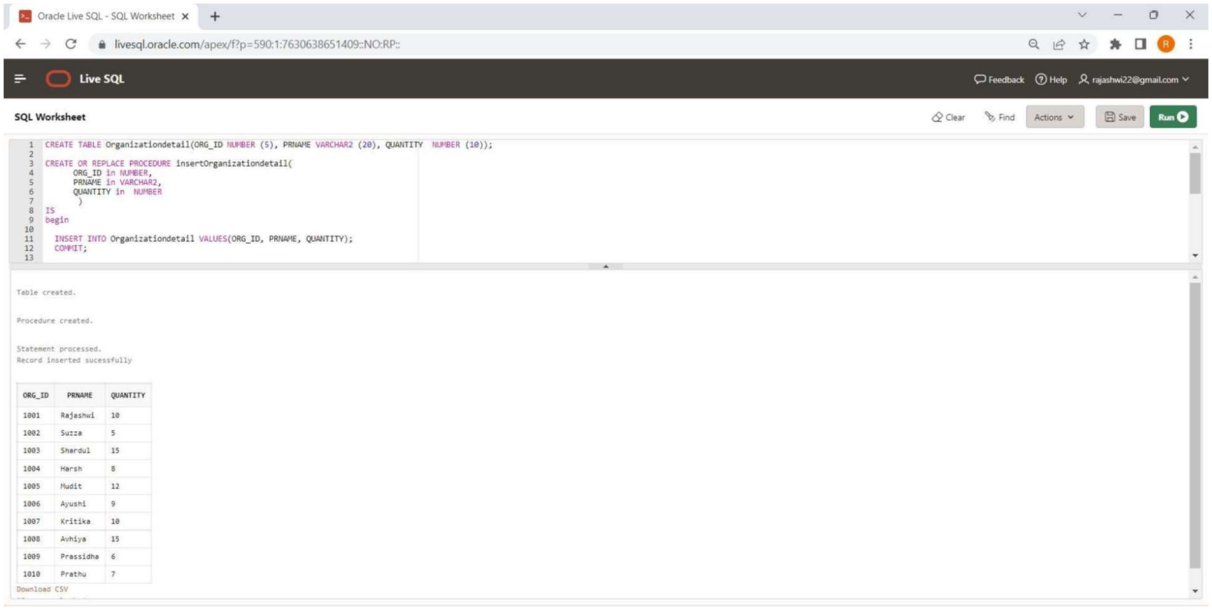
dbms_output.put_line('Record inserted sucessfully');

END;

/

select * from Organizationdetail;

SCREENSHOTS:



Questions: Create another table Student (Roll,Name,Age,Class,City,State,Country).Insert

minimum 12 records in this table. Create a procedure by using IN and OUT Parameter to

display the records of specific student.

CODE:

create table Student(Roll int,Name varchar(255),Age int, CLass varchar(255),City

varchar(255),State varchar(255),Country varchar(255));

insert into Student values (1001 , 'Garima singh', 18, 'Biology', 'india', 'Gorakhpur', ‘Uttar

pradesh');

insert into Student values (1002 , 'Tejashwi', 25, 'Information mangement', 'Kathmandu',

'Bagmati', 'Nepal');

insert into Student values (1003 , 'Ashutosh', 28, 'Maths', 'Jharkhand', 'Gomti nagar', 'India');

insert into Student values (1004 , 'Harsh', 18, 'Medicine', 'Dehradun', 'Uttarakhand', 'India');

insert into Student values (1005 , 'Suzza', 28, 'Environment', 'Biratnagar', 'Dhangadi', 'Nepal');

insert into Student values (1006 , 'Ayushi', 25, 'Chemistry', 'Biratnagar', 'Dhangadi', 'Nepal');

insert into Student values (1007 , 'Nitin', 31, 'Social Studies', 'Hariyana', 'Hariyana', 'India');

insert into Student values (1008 , 'Chandan', 8, 'English', 'Kolkata', 'West Bengal', 'India');

insert into Student values (1009 , 'Kritika', 21, 'Maths', 'Kthamndu', 'Bagmati', 'Nepal');

insert into Student values (1010 , 'Parthu', 18, 'Medicine', 'Dehradun', 'Uttarakhand', 'India');

insert into Student values (1011 , 'Ayeshree', 28, 'Nepali', 'Biratnagar', 'Dhangadi', 'Nepal');

insert into Student values (1012 , 'Mudit', 18, 'Physical Education', 'Dehradun', 'Uttarakhand',

'India');

SELECT * from Student;

create or replace procedure Information(Roll_no out Student.Roll%type,

Fname out Student.Name%type

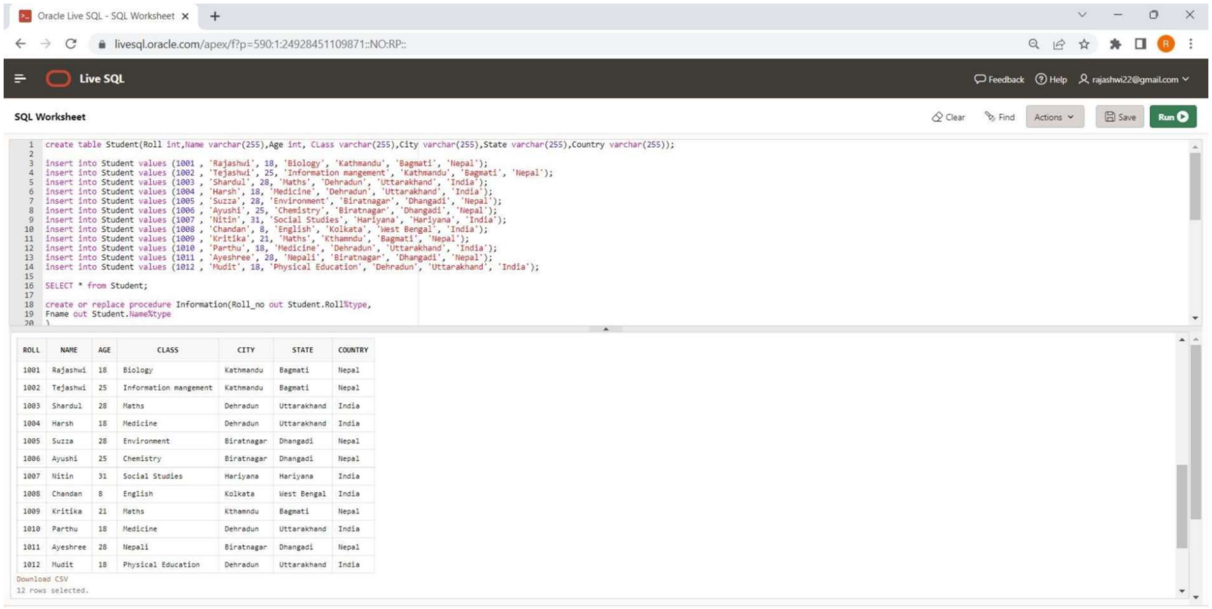
)

IS

Begin

```
select Roll,Name into Roll_no,Fname
from Student
WHERE Roll=1001;
exception
when others then
dbms_output.put_line(sqlcode);
dbms_output.put_line(sqlerrm);
END;
/
declare
Roll_no Student.Roll%type;
Fname Student.Name%type;
Begin
Information(Roll_no,Fname);
dbms_output.put_line(Roll_no);
dbms_output.put_line(Fname);
end;
/
Select * from Information;
```

SCREENSHOTS:



Experiment-4

Objective : Use of PL/SQL variables, reserved words, identifiers and anchored data types.

(I) WAP in pl/sql to declare variable and perform addition, subtraction and multiplication operation and display the result.

(II) WAP in pl/sql to show the variable scope

(III) WAP using pl/sql constant and display the result on the screen

(IV) Define reserved word, write a program by using reserved word.

(V) Using %TYPE to Declare Variables of the Types of Other Variables

Program:

I)

```
DECLARE
X number(10);
Y number (10);
SUMM number(12);
SUB number(12);
MUL number(20);
DIV number(10);
BEGIN
X:=10;
Y:=12;
SUMM:=X+Y;
SUB:= Y-X;
MUL:= X*Y;
DIV:=Y/X;
```

dbms_output.put_line('Sum is equal to '||SUMM);

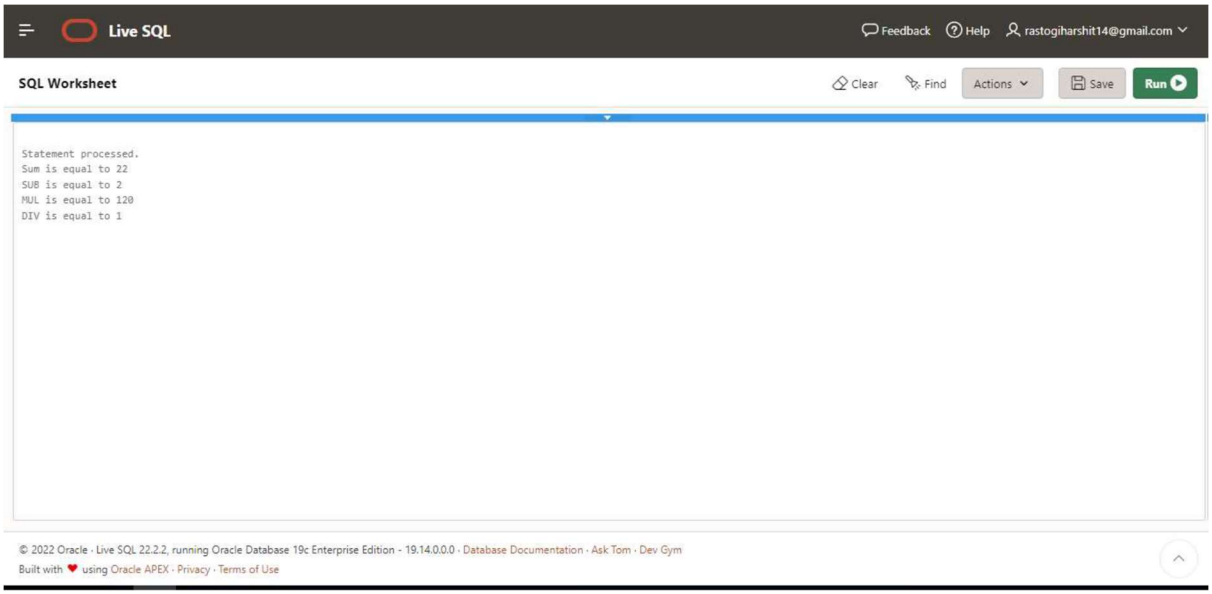
dbms_output.put_line('SUB is equal to '||SUB);

dbms_output.put_line('MUL is equal to '||MUL);

dbms_output.put_line('DIV is equal to '||DIV);

END;

Output:



II)

DECLARE

-- Global variables

num1 number := 22;

num2 number := 24;

BEGIN

dbms_output.put_line('Outer Variable num1: ' || num1);

dbms_output.put_line('Outer Variable num2: ' || num2);

DECLARE

-- Local variables

num1 number := 05;

num2 number := 14;

BEGIN

dbms_output.put_line('Inner Variable num1: ' || num1);

dbms_output.put_line('Inner Variable num2: ' || num2);

END;

END;

Output:

III)

DECLARE

CONST_NUM CONSTANT NUMBER := 21;

BEGIN

dbms_output.put_line(CONST_NUM);

END;

Output:

IV)

DECLARE

"DECLARE" varchar2(25) := 'This is UPPERCASE';

"Declare" varchar2(25) := 'This is Proper Case';

"declare" varchar2(25) := 'This is lowercase';

BEGIN

DBMS_Output.Put_Line("DECLARE");

DBMS_Output.Put_Line("Declare");

DBMS_Output.Put_Line("declare");

END;

Output:

```
Statement processed.
This is UPPERCASE
This is Proper Case
This is lowercase
```

V)

CREATE TABLE USERR(

User_ID NUMBER,

User_Name VARCHAR2(100)

);

DECLARE

P_ID USERR.User_ID%TYPE;

P_NAME USERR.User_Name%TYPE;

BEGIN

P_ID := 12;

P_NAME := 'RAM';

DBMS_Output.Put_Line(P_ID);

DBMS_Output.Put_Line(P_NAME);

END;

Output:


Table created.
Statement processed.
12
RAM
© 2022 Oracle · Live SQL 22.2.2, running Oracle Database 19c Enterprise Edition - 19.14.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ❤ using Oracle APEX · Privacy · Terms of Use

Experiment-5

Objective : Using Oracle Clause

1 Create a table CustomerInfo(NAME,AGE,SALARY,STATE,PIN).Insert minimum 10 records in it.Then use Distinct Clause, From clause and Order By clause and display their results.

2. Create another table SALESDEPARTMENT(ITEM,SALES,BILLING_ADDRESS,COST).Insert minimum 7 records in it and run Aggregate function and Group by clause in this table to display the result.

A-1

Program:

```
create table CustomerInfo(name varchar(20), age int, salary number, state varchar(10), pin number);

insert into CustomerInfo values('Sid',21,1000000,'UK',248001);

insert into CustomerInfo values('Ekans',21,2000000,'KAT',248069);

insert into CustomerInfo values('Adim',20,3000000,'UK',248007);

insert into CustomerInfo values('Spars',23,4000000,'UK',250001);

insert into CustomerInfo values('Ansul',21,5000000,'HP',348001);

insert into CustomerInfo values('Cheta',20,6000000,'UK',248006);

insert into CustomerInfo values('Mumdi',21,7000000,'WB',248002);

insert into CustomerInfo values('Rajas',21,10,'KAT',248571);

select distinct state from CustomerInfo where salary > 3000000; select name, age, salary from CustomerInfo order by salary;
```

Output:

STATE
WB
HP
UK

Download CSV
3 rows selected.

NAME	AGE	SALARY
Rajas	21	10
Sid	21	1000000
Ekans	21	2000000
Adim	20	3000000
Spars	23	4000000
Ansul	21	5000000
Cheta	20	6000000
Mumdi	21	7000000

Download CSV
8 rows selected.

A-2

Program:

```
create table SalesDepartment(item varchar(10), sales number, address varchar(200), cost number);

insert into SalesDepartment values('Iphone',2,'Dehra',100000);

insert into SalesDepartment values('Iphone',4,'Guldarpur',17000);

insert into SalesDepartment values('POCO',5,'Sanskar Garden',20000);

insert into SalesDepartment values('Lava',10,'Sela',1000);

insert into SalesDepartment values('Lava',3,'IEL Bul',10000);

insert into SalesDepartment values('Jio',1,'Cat',1000);

select * from SalesDepartment;

select item, sum(cost) from SalesDepartment group by item;
```

Output:

ITEM	SALES	ADDRESS	COST
Iphone	2	Dehra	100000
Iphone	4	Guldarpur	17000
POCO	5	Sanskar Garden	20000
Lava	10	Sela	1000
Lava	3	IEL Bul	10000
Jio	1	Cat	1000

Download CSV
6 rows selected.

ITEM	SUM(COST)
Iphone	117000
Lava	11000
Jio	1000
POCO	20000

Download CSV
4 rows selected.

IB401 GRADED LAB3

1. Create a table Employee(EID, EName, Age, Salary, Address),

Write a PL/SQL Program to display the utilization of Internal Cursor.

Create another program of External Cursor in PL/SQL Then display the content of the table.

A1.

Implicit Cursor:

```
Create table Employee(
    EID number,
    EName varchar(20),
    Age number,
    Salary number,
    Address varchar(10)
    );

insert into Employee values(190, 'Aditya', 20, 90000, 'XYZ');
insert into Employee values(200, 'Chetan', 20, 110000, 'ABC');
insert into Employee values(210, 'Siddhant', 21, 100000, 'EFG');
insert into Employee values(220, 'Ekansh', 21, 6000000, 'GHI');
insert into Employee values(230, 'Shardul', 22, 3000000, 'HIJ');

DECLARE
    totalrows number;
BEGIN
```

```
BEGIN
    update Employee
    set Salary = Salary + 5000;
    if sql%notfound then
        dbms_output.put_line('no customers');
    elsif sql%found then
        totalrows := sql%rowcount;
        dbms_output.put_line('Customers affected: ' || totalrows);
    end if;
END;
```

```
Statement processed.
Customers affected: 5
```

Explicit Cursor:

```
Create table Employee(
    EID number,
    EName varchar(20),
    Age number,
    Salary number,
    Address varchar(10)
    );

insert into Employee values(190, 'Aditya', 20, 90000, 'XYZ');
insert into Employee values(200, 'Chetan', 20, 110000, 'ABC');
insert into Employee values(210, 'Siddhant', 21, 100000, 'EFG');
insert into Employee values(220, 'Ekansh', 21, 6000000, 'GHI');
insert into Employee values(230, 'Shardul', 22, 3000000, 'HIJ');
```

```
DECLARE
   eid Employee.EID%type;
   ename Employee.EName%type;
   eaddr Employee.Address%type;
   CURSOR employees is
      SELECT EID, EName, Address FROM Employee;
BEGIN
   OPEN employees;
   LOOP
   FETCH employees into eid, ename, eaddr;
      EXIT WHEN employees%notfound;
      dbms_output.put_line(eid || ' ' || ename || ' ' || eaddr);
   END LOOP;
   CLOSE employees;
END;
```

```
Statement processed.
190 Aditya XYZ
200 Chetan ABC
210 Siddhant EFG
220 Ekansh GHI
230 Shardul HIJ
```

Experiment – 6

Objective: Using Control Statement, Case and Loop in Pl/SQL

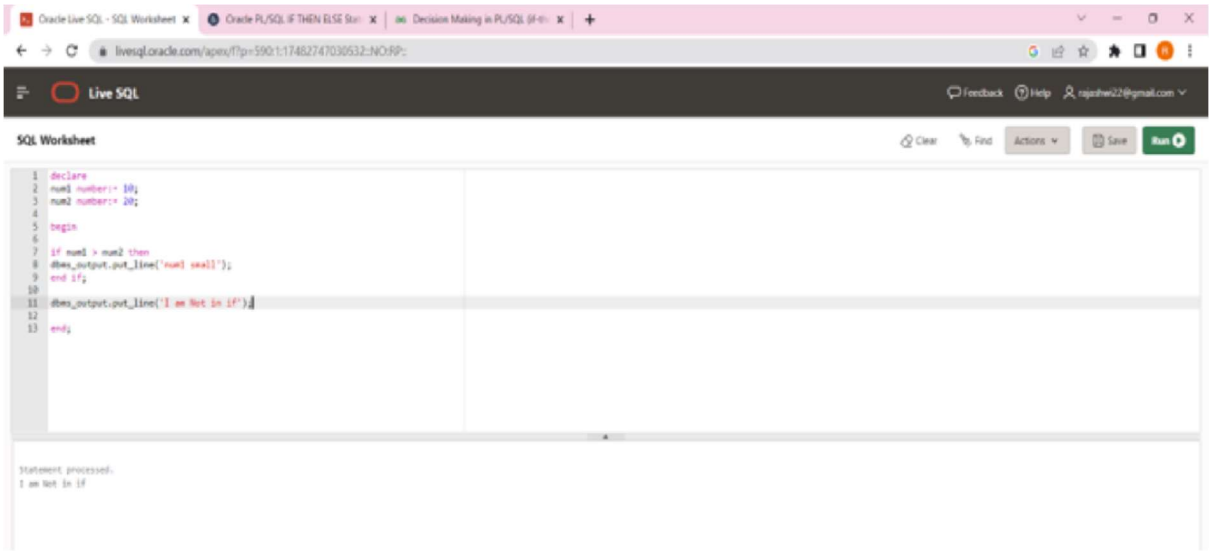
(i) WAP in Pl/SQL using if elseif endif statement.

If statement:

Code:

```
Declare
num1 number:= 10;
num2 number:= 20;
begin
if num1 > num2 then dbms_output.put_line('num1 small');
end if;
dbms_output.put_line('I am Not in if');
end;
```

Screenshot:



If elseif endif

Code:

```
declare
```

num1 number:= 10;

num2 number:= 20;

begin

if num1 < num2 then

dbms_output.put_line('i am in if block');

ELSE

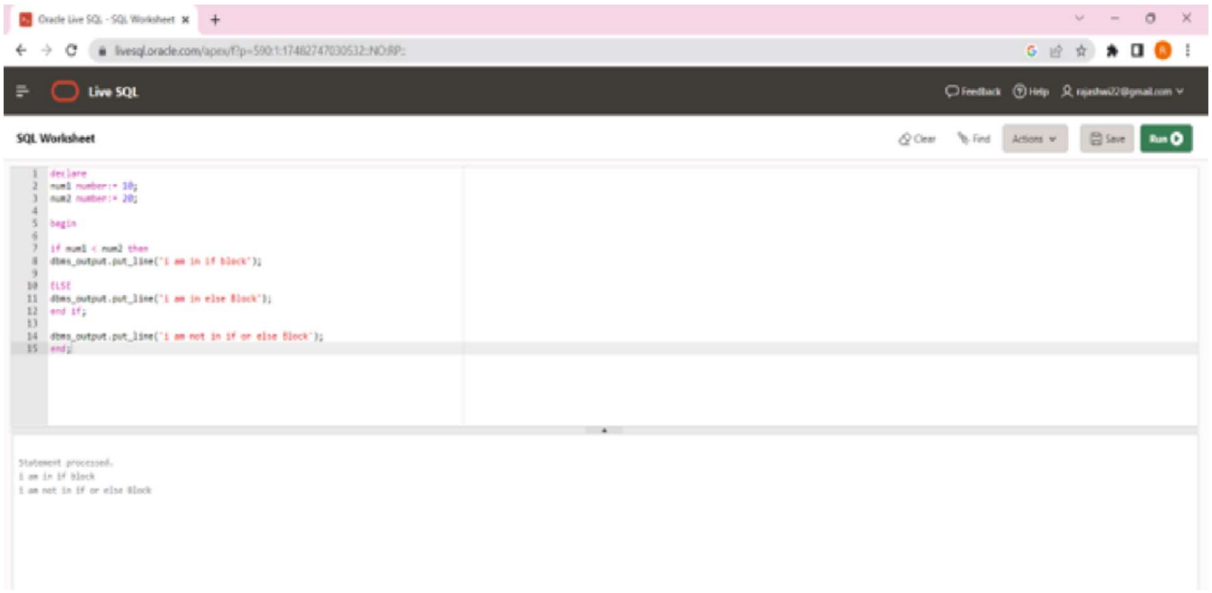
dbms_output.put_line('i am in else Block');

end if;

dbms_output.put_line('i am not in if or else Block');

end;

Screenshot:



(ii) WAP in Pl/SQL using case statement to display the performance of the user

CODE:

DECLARE

userperformance char(1) := 'A';

BEGIN

CASE userperformance

when 'A' then dbms_output.put_line('Excellent');

when 'B' then dbms_output.put_line('Very good');

when 'C' then dbms_output.put_line('Well done');

when 'D' then dbms_output.put_line('Fair');

when 'F' then dbms_output.put_line('Better try again');

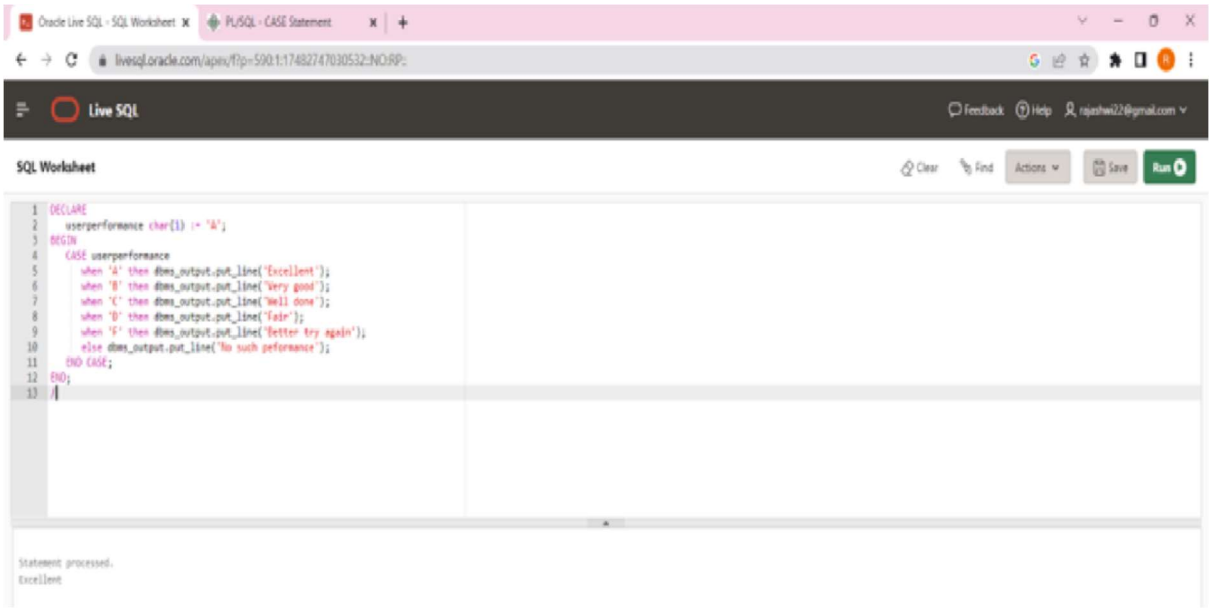
else dbms_output.put_line('No such peformance');

END CASE;

END;

/

SCREENSHOTS:



(iii) WAP in Pl/SQL to display the utilization of for loop with example

CODE:

DECLARE

VAR1 NUMBER;

BEGIN

VAR1:=10;

FOR VAR2 IN 1..10

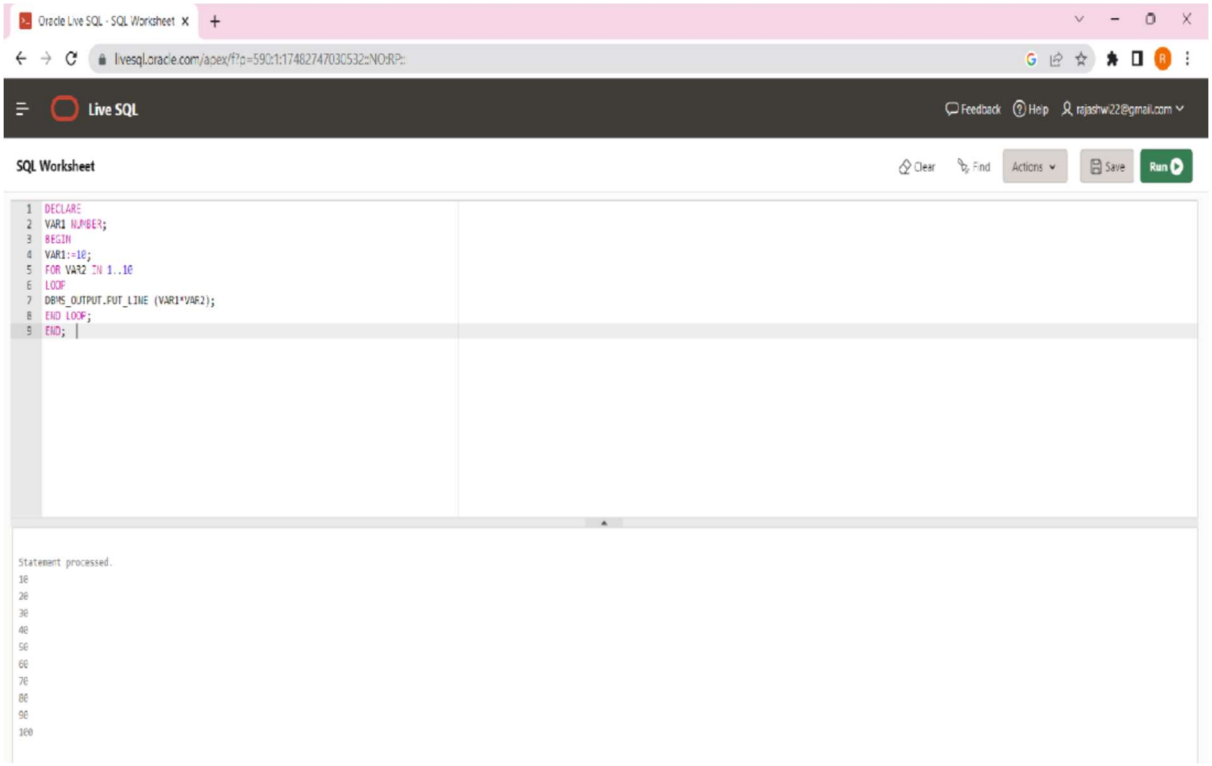
LOOP

DBMS_OUTPUT.PUT_LINE (VAR1*VAR2);

END LOOP;

END;

SCREENSHOTS:



Experiment-7

Objective : Use of Procedure and Function in Pl/SQL

i. Create table PERSON. Insert minimum 6 records in it. Display utilisation of

Procedure and call it and display.

Code-

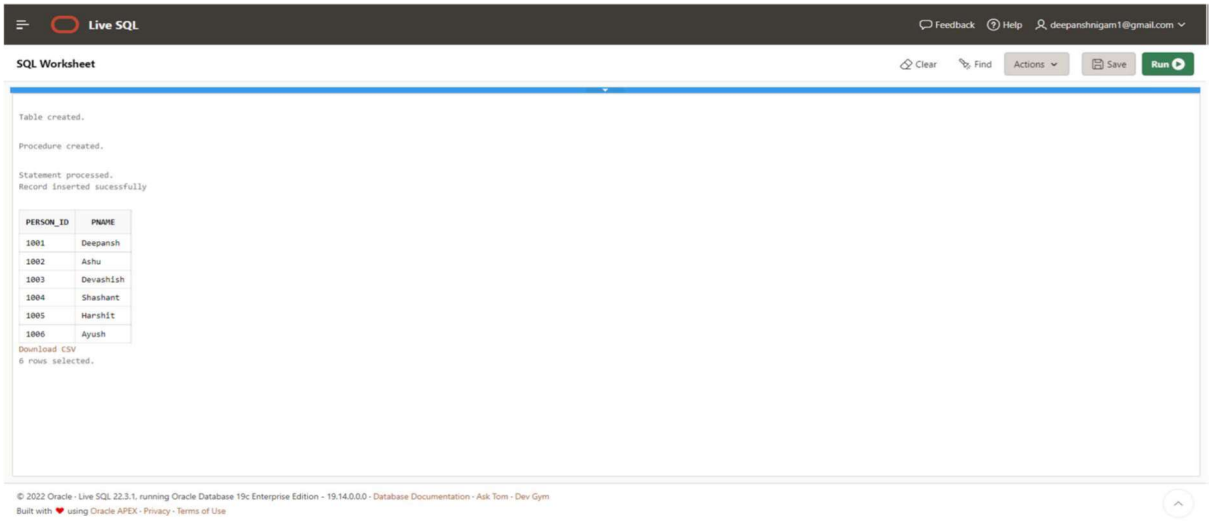
```
CREATE TABLE persondetail(person_ID NUMBER (5), PNAME VARCHAR2
(20));
CREATE OR REPLACE PROCEDURE insertPersondetail(
person_ID in NUMBER,
PNAME in VARCHAR2
)
IS
begin
INSERT INTO persondetail VALUES(person_ID, PNAME);
COMMIT;
end;
/
BEGIN
insertPersondetail(1001,'Deepansh');
insertPersondetail(1002,'Ashu');
insertPersondetail(1003,'Devashish');
insertPersondetail(1004,'Shashant');
insertPersondetail(1005,'Harshit');
insertPersondetail(1006,'Ayush');
dbms_output.put_line('Record inserted sucessfully');
```

END;

/

select * from persondetail;

Screenshot



ii. In PLSQL perform various operations using function and call this function to display result.

Code-

CREATE TABLE Customer (

CustomerID int,

CName varchar(255),

CAge int,

Address varchar(255)

);

INSERT INTO Customer VALUES(101, 'Deepansh Nigam', 21, 'Delhi');

INSERT INTO Customer VALUES(102, 'Harshit Rastogi', 22, 'Meerut');

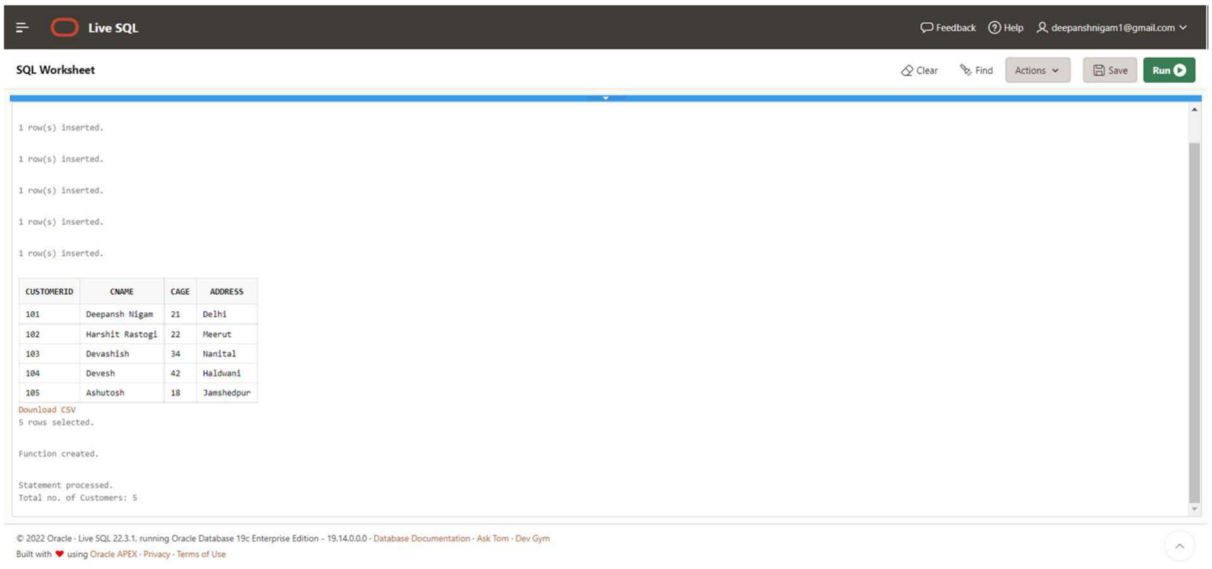
INSERT INTO Customer VALUES(103, 'Devashish', 34, 'Nanital');

INSERT INTO Customer VALUES(104, 'Devesh', 42, 'Haldwani');

INSERT INTO Customer VALUES(105, 'Ashutosh', 18, 'Jamshedpur');

```
Select * from customer;

CREATE OR REPLACE FUNCTION totalCustomers

RETURN number IS

total number(2) := 0;

BEGIN

SELECT count(*) into total

FROM customer;

RETURN total;

END;

/

DECLARE

c number(2);

BEGIN

c := totalCustomers();

dbms_output.put_line('Total no. of Customers: ' || c);

END;
```

Screenshot



Experiment – 8

Objective: Use of Trigger in Pl/SQL

i.In PL/SQL Create table Employee. Insert minimum 8 records in it and display the

utilization of Trigger

Code:

```
CREATE TABLE Employee(EID number, EName varchar(255),EAge number, EAddress
varchar(255), salary number);
INSERT INTO Employee VALUES (101, 'Rajashwi', 22, 'Kathmandu', 7500);
INSERT INTO Employee VALUES (102, 'Mudit', 21, 'Dehradun', 8500);
INSERT INTO Employee VALUES (102, 'Chandan', 25, 'Kanpur', 4500);
INSERT INTO Employee VALUES (104, 'Nitin', 28, 'Hariyana', 1500);
INSERT INTO Employee VALUES (105, 'Surabhi', 22, 'Biratnagar', 3500);
Select * from Employee;
CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON Employee
FOR EACH ROW
WHEN (NEW.EID > 0)
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;
```

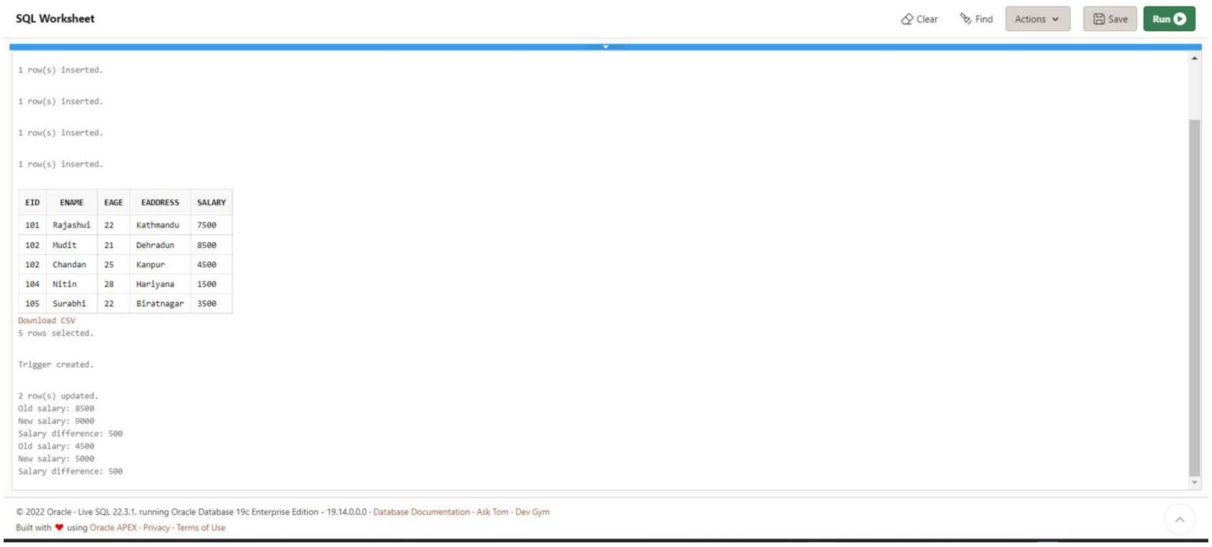
/

UPDATE Employee

SET salary = salary + 500

WHERE EID = 102;

Screenshot:



ii. Create a tableUser .Insert records in it. Perform Trigger operation in it, also use enable

and drop trigger operation in it.

CODE:

CREATE TABLE Userd(EID number, EName varchar(255),EAge number, EAddress

varchar(255), salary number);

INSERT INTO Userd VALUES (101, 'Rajashwi', 22, 'Kathmandu', 7500);

INSERT INTO Userd VALUES (102, 'Mudit', 21, 'Dehradun', 8500);

INSERT INTO Userd VALUES (102, 'Chandan', 25, 'Kanpur', 4500);

INSERT INTO Userd VALUES (104, 'Nitin', 28, 'Hariyana', 1500);

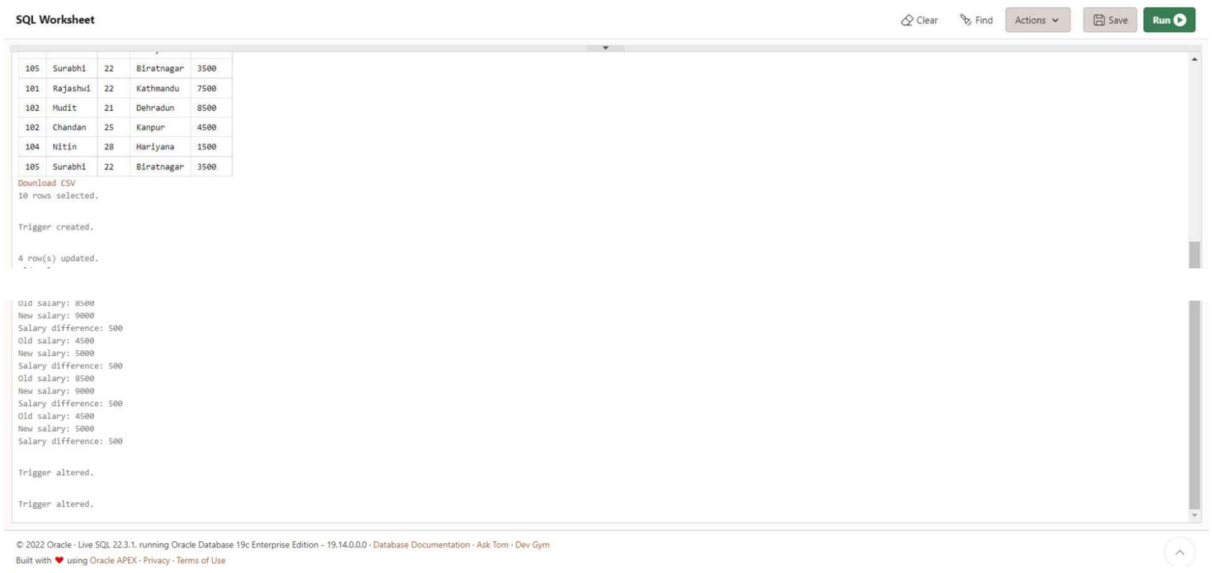
INSERT INTO Userd VALUES (105, 'Surabhi', 22, 'Biratnagar', 3500);

Select * from Userd;

CREATE OR REPLACE TRIGGER display_salary_changes

```
BEFORE DELETE OR INSERT OR UPDATE ON Userd

FOR EACH ROW

WHEN (NEW.EID > 0)

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;

/

UPDATE Userd

SET salary = salary + 500

WHERE EID = 102;

ALTER TRIGGER display_salary_changes DISABLE;

ALTER TRIGGER display_salary_changes ENABLE;
```

Screenshot:



Experiment – 9

Objective : Using Exception handling in PL/SQL

WAP in Pl/SQL to display the utilization of exception handling.

Code:

```
CREATE TABLE Customer(CID number, CName varchar(255),CAge number, Address
varchar(255), salary number);
INSERT INTO Customer VALUES (101, 'Rajashwi', 22, 'Kathmandu', 7500);
INSERT INTO Customer VALUES (102, 'Mudit', 21, 'Dehradun', 8500);
INSERT INTO Customer VALUES (102, 'Chandan', 25, 'Kanpur', 4500);
INSERT INTO Customer VALUES (104, 'Nitin', 28, 'Hariyana', 1500);
INSERT INTO Customer VALUES (105, 'Surabhi', 22, 'Biratnagar', 3500);
Select * from Customer;
DECLARE
c_id Customer.CID%type := 8;
c_name Customer.CName%type;
c_addr Customer.address%type;
BEGIN
SELECT CName, address INTO c_name, c_addr
FROM Customer
WHERE CID = c_id;
DBMS_OUTPUT.PUT_LINE ('Name: '|| c_name);
DBMS_OUTPUT.PUT_LINE ('Address: ' || c_addr);
EXCEPTION
WHEN no_data_found THEN
dbms_output.put_line('No such customer!');
WHEN others THEN
```

```
dbms_output.put_line('Error!');

END;

/
```

Screenshot: