**DATABASE MANAGEMENT SYSTEM LABORATORY**

**COURSE CODE: BCSE2073**

Lab Manual

*for*

BACHELOR OF

Engineering & Technology



**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**GALGOTIAS UNIVERSITY, GREATER NOIDA**

**UTTAR PRADESH**

**NAME-Abhinav Kumar Choudhary SUBMITTED TO-**

**SEC-21 Dr. SWATI SHARMA 21SCSE1011615**

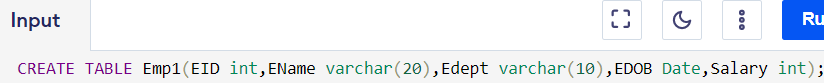
|  |  |
| --- | --- |
| Sr. No. | Title of Lab Experiments |
| 1. | Implement Data Definition language Statements. |
| 2. | Implement Data Manipulation Statements. |
| 3. | Implement SELECT command with different clauses. |
| 4. | Implement various type of Integrity Constraints on database. |
| 5. | Implement SINGLE ROW functions (Character, Numeric, Date functions) and GROUP functions (avg, count, max, min, sum). |
| 6. | Implement various type of SET OPERATORS (Union, Intersect, Minus) and JOINS. |
| 7. | Implement the concept of grouping of Data and Subqueries. |
| 8. | Implement the concept of Data Control Language (DCL), Transaction Control Language (TCL). |
| 9. | Implement Simple and Complex View. |
| 10. | Write a PL/SQL block to satisfy some conditions by accepting input from the user. |
| 11. | Write a PL/SQL block for greatest of three numbers using IF AND ELSEIF |
| 12. | Write a PL/SQL block for summation of odd numbers using for LOOP |
| 13. | Write a PL/SQL Procedure for GCD Numbers |
| 14. | Write a PL/SQL Procedure for cursor implementation |
| 15. | Write a PL/SQL block to implementation of factorial using function |

**EXPERIMENT 1**

**AIM**: Implement Data Definition language statements.

1. **CREATE TABLE**

Source code:



Output:



1. **ALTER TABLE**

Source code:

CREATE TABLE Emp1(EID int,EName varchar(20),Edept varchar(10),EDOB Date,Salary int);

ALTER TABLE Emp1 ADD Address varchar(10);

ALTER TABLE Emp1 MODIFY EID varchar(10);

ALTER TABLE Emp1 DROP column Edept ;

ALTER TABLE Emp1 RENAME COLUMN EID TO Roll\_no;

desc Emp1

Output:

Table altered.

TABLE Emp1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Null?** | | **Type** | |
| ROLL\_NO | | - | | VARCHAR2(10) |
| ENAME | | - | | VARCHAR2(20) |
| EDOB | | - | | DATE |
| SALARY | | - | | NUMBER |
| ADDRESS | | - | | VARCHAR2(10) |

[Download CSV](https://livesql.oracle.com/apex/f?p=590:1:15906384772268::NO:93::&success_msg=MSBkYXRhYmFzZSBvYmplY3RzIGhhdmUgYmVlbiBkcm9wcGVkLiAgU2Vzc2lvbiBo.,aXN0b3J5IGhhcyBiZWVuIHJlbW92ZWQuIE5MUyBzZXR0aW5ncyByZXNldC4~%2FomEyemN9TQt8zO17_1M-uYbfGYvB_iVqBYXr8yxu31znpXKyt8a2xf0Z1G6xjkVWj4Yrph4V47K1HB31f58g7Q)  
5 rows selected.

1. **DROP TABLE**

Source code:

DROP TABLE Emp1;

desc Emp1

Output:

Table dropped. ORA-20001: object EMP1 does not exist

1. **RENAME TABLE**

Source code:

RENAME Emp1 TO Employee;

desc Employee

Output:

TABLE EMPLOYEE

|  |  |  |
| --- | --- | --- |
| **Column** | **Null?** | **Type** |
| ROLL\_NO | - | VARCHAR2(10) |
| ENAME | - | VARCHAR2(20) |
| EDOB | - | DATE |
| SALARY | - | NUMBER |
| ADDRESS | - | VARCHAR2(10) |
|  |  |  |

1. **TRUNCATE TABLE**

Source code:

CREATE TABLE Emp1(EID int,EName varchar(20),Edept varchar(10),EDOB Date,Salary int);

TRUNCATE TABLE Emp1;

desc Emp1

Output: The data will be deleted but the structure will remain in the memory for further operations.

Table truncated. TABLE EMP1

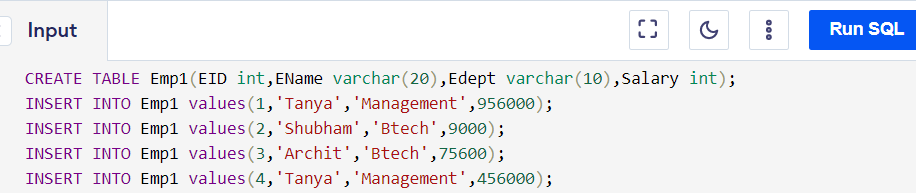
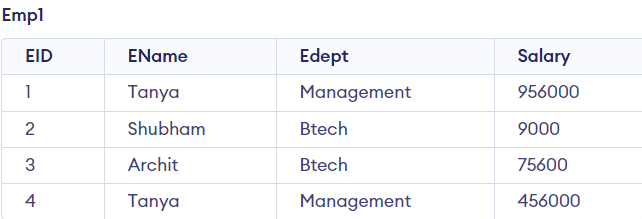
|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

EXPERIMENT 2

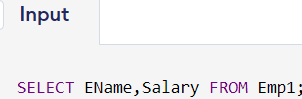
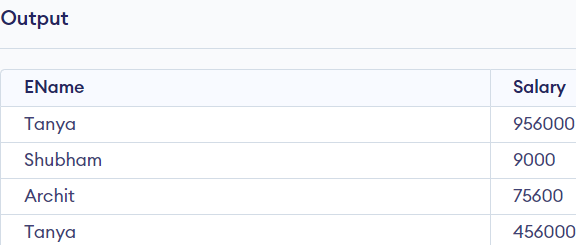
**AIM**: Implement Data Manipulation language statements (DML).

1. **INSERT**

Source Code: Output:

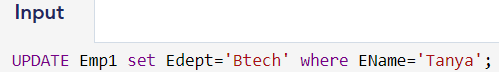
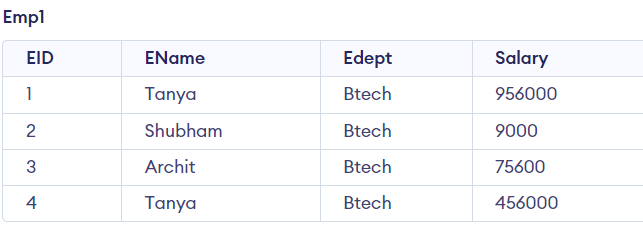
 

1. **SELECT**

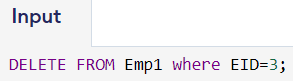
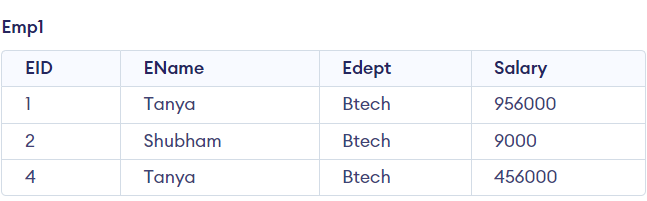
 

1. **UPDATE**

Output:

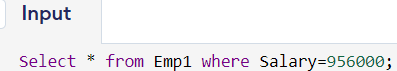
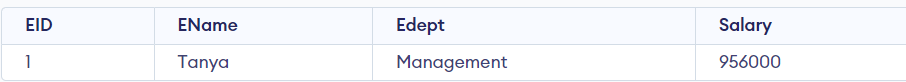
1. **DELETE**

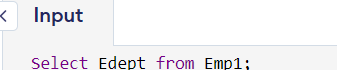
**EXPERIMENT 3**

**AIM** : Implement select command with different classes.

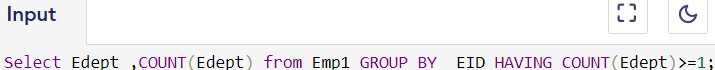
1. **WHERE**

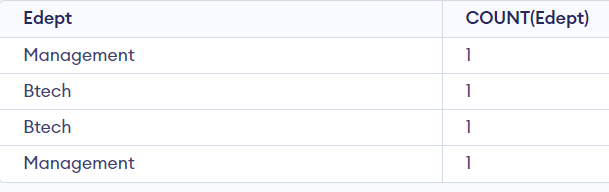
1. **GROUP BY**

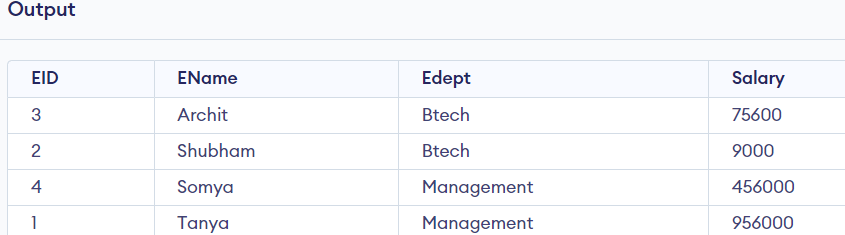
1. **HAVING**



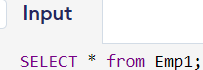
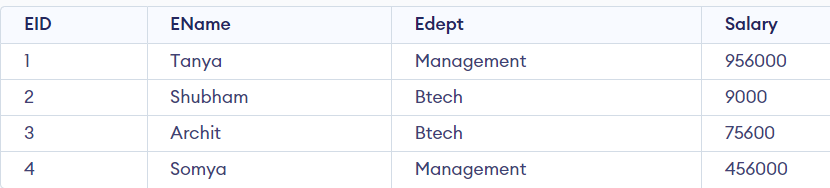
Output:



1. **ORDER BY**

1. **SELECT**

**EXPERIMENT 4**

**AIM** : Implement various types of integrity constraints on database.

1. **NOT NULL AND PRIMARY KEY**

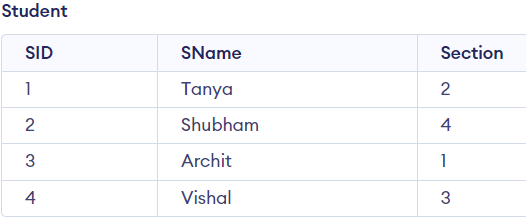
CREATE TABLE Student(SID int PRIMARY KEY,SName varchar(30) NOT NULL,Section int NOT NULL);

INSERT INTO Student values(1,'Tanya',2);

INSERT INTO Student values(2,'Shubham',4);

INSERT INTO Student values(3,'Archit',1);

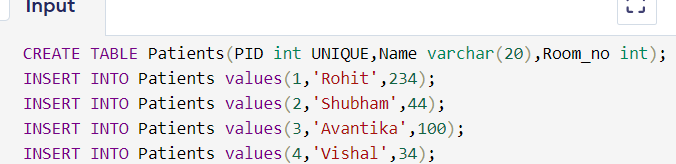
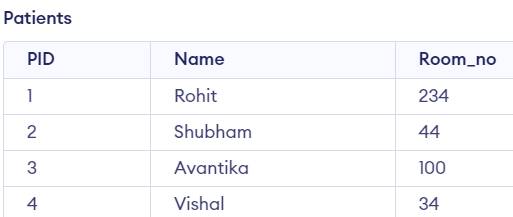
INSERT INTO Student values(4,'Vishal',3);



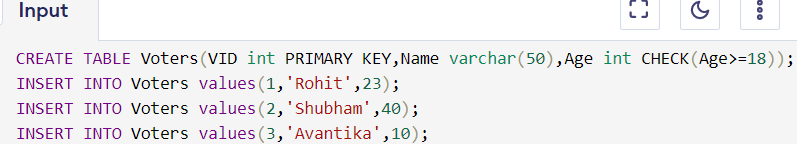
|  |  |  |
| --- | --- | --- |
|  |  |  |

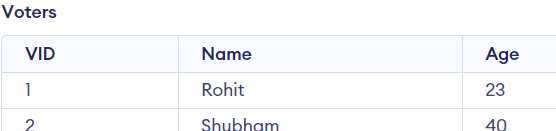
1. **UNIQUE**

Source code: Output:

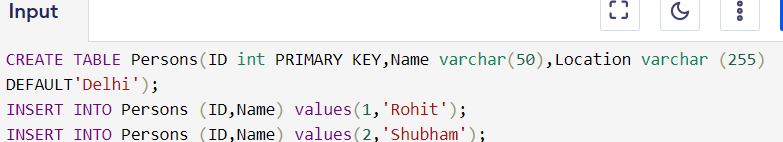
 

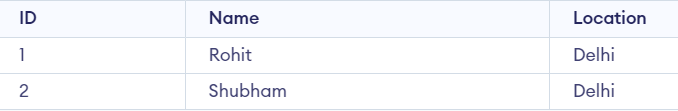
1. **CHECK**





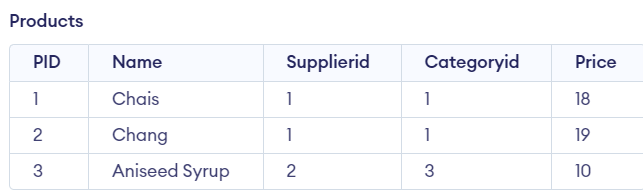
1. **DEFAULT**



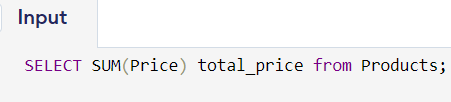
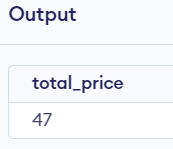


**EXPERIMENT 5**

**AIM**: Implement single row functions and group functions.

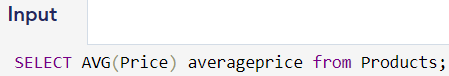
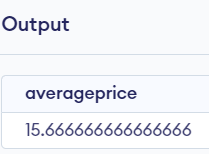


1. **SUM**

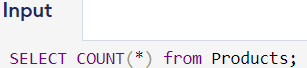
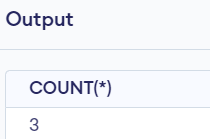
 

|  |  |  |
| --- | --- | --- |
|  |  |  |

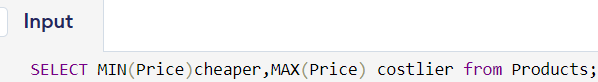
1. **AVERAGE**

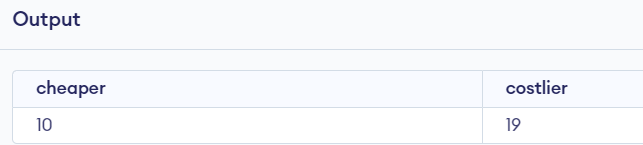
 

1. **COUNT**

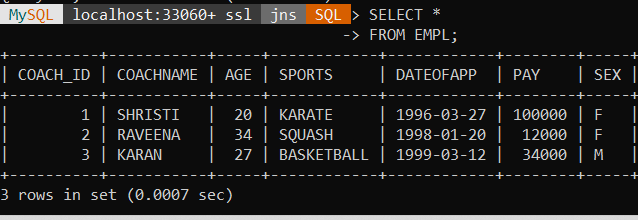
1. **MIN AND MAX**

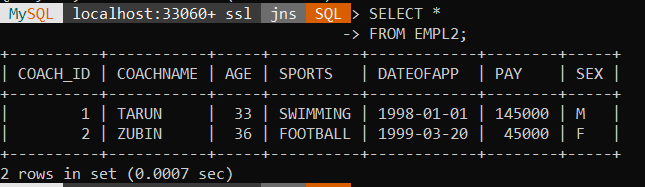




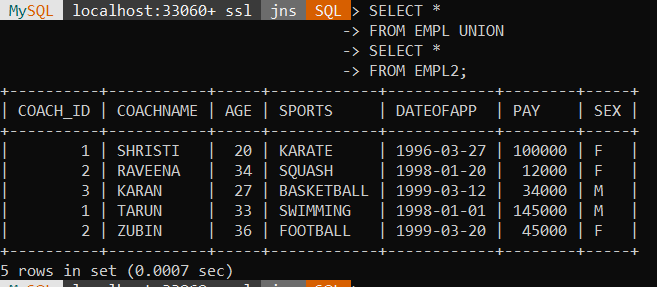
**EXPERIMENT 6**

**AIM**-Study of various type of SET OPERATORS





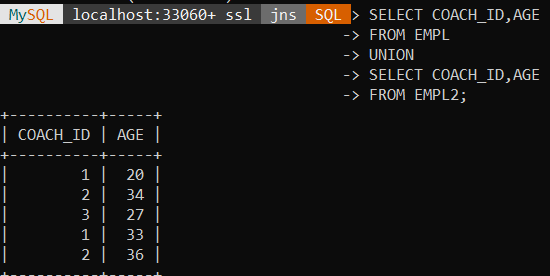
**UNION**



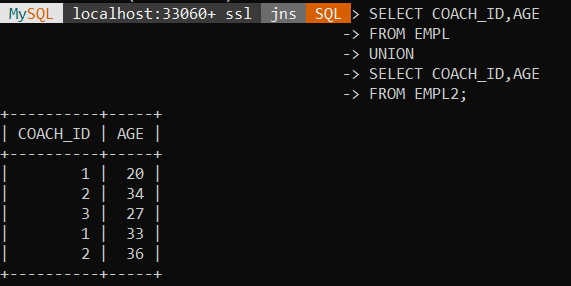
**EXPERIMENT 7**

**AIM**-Study and implement the concept of sub queries

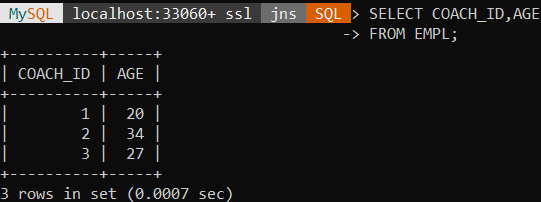
**UNION CLAUSE**



**INTERSECTION CLAUSE**

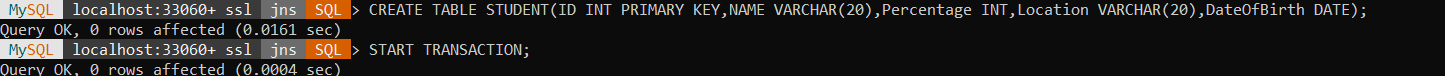
****

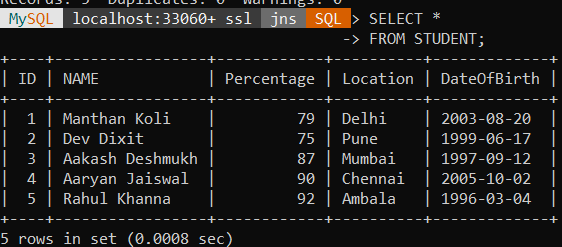
**MINUS CLAUSE**



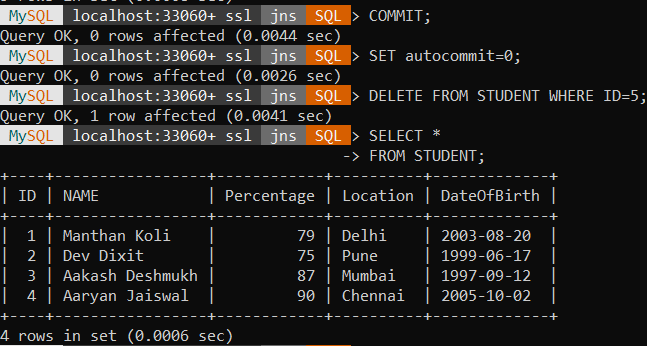
**EXPERIMENT 8**

**AIM**-Study and implement the concept of data control language(DCL),Transaction control language (TCL).

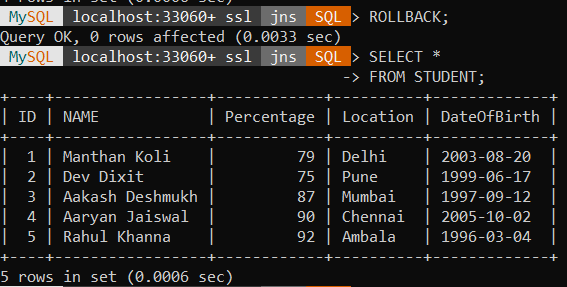




**COMMIT**

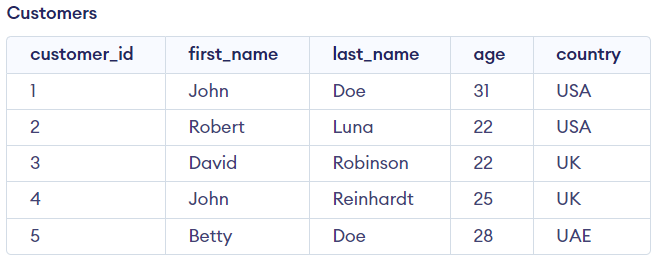


**ROLLBACK**

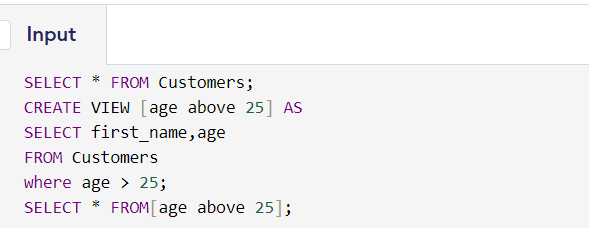


**EXPERIMENT 9**

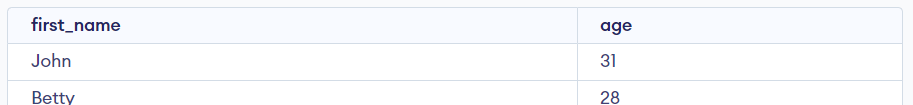
**AIM**-Study of Simple and Complex View



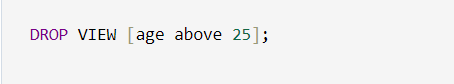
**CREATE VIEW**



OUTPUT



**DROP VIEW**

****

The drop view commands deletes a view.

The above sql command drops the “age above 25” view.

**EXPERIMENT 10:-**

**Aim:-** Write a PL/SQL block to satisfy some conditions by accepting

input from the user.

**Source Code:-** SQL> DECLARE

<VARIABLE DECLARATION>; BEGIN

<EXECUTABLE STATEMENT >;

END;

PL/SQL CODING FOR ADDITION OF TWO NUMBERS

SQL>Set serveroutput on SQL> declare

a number; b number; c number; begin a:=&a;

b:=&b; c:=a+b;

dbms\_output.put\_line(‘sum of’||a||’and’||b||’is’||c); end;

/

**INPUT: Enter value for a: 23 old 6: a:=&a;**

**new 6: a:=23; Enter value for b: 12 old 7: b:=&b;**

**new 7: b:=12;**

# OUTPUT:

sum of23and12is35

PL/SQL procedure successfully comple

# EXPERIMENT -11

**Aim:-** Write a PL/SQL block for greatest of three numbers using IF AND ELSEIF

**Source Code**:- SQL>Set serveroutput on SQL> declare

a number; b number; c number; d number; begin a:=&a;

b:=&b;

c:=&b; if(a>b)and(a>c) then

dbms\_output.put\_line('A is maximum'); elsif(b>a)and(b>c)then

dbms\_output.put\_line('B is maximum'); else

dbms\_output.put\_line('C is maximum'); end if;

end;

INPUT:

Enter value for a: 21 old 7: a:=&a; new 7: a:=21; Enter value for b: 12 old 8: b:=&b; new 8: b:=12; Enter value for b: 45 old 9: c:=&b; new 9: c:=45;

**Output:-** C is maximum

PL/SQL procedure successfully completed.

**EXPERIMENT 12:-**

**Aim:-** Write a PL/SQL block for summation of odd numbers using for LOOP

# Source Code:-

DECLARE

-- declare variable num num NUMBER(3) := 1; sum1 NUMBER(4) := 0;

BEGIN

WHILE num <= 5 LOOP

-- display odd number dbms\_output.Put\_line(num);

-- the sum of all odd numbers sum1 := sum1 + num;

--next odd number num := num + 2;

-- end loop END LOOP;

dbms\_output.Put\_line('Sum of all odd numbers is '|| sum1); END;



# Output:

# EXPERIMENT 13:-

**Aim:-** Write a PL/SQL Procedure for GCD Numbers

**Source Code**:- DECLARE num1 INTEGER;

num2 INTEGER; t INTEGER;

BEGIN

num1 := 8;

num2 := 48;

WHILE MOD(num2, num1) != 0 LOOP

t := MOD(num2, num1); num2 := num1;

num1 := t;

END LOOP; dbms\_output.Put\_line('GCD of '

||num1

||' and '

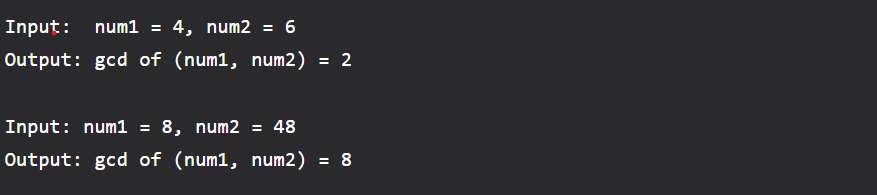
||num2

||' is '

||num1);

END;

-- Program End



# Output:-



# EXPERIMENT 14:-

**Aim:-** Write a PL/SQL Procedure for cursor implementation

# Source Code:-

DECLARE

total\_rows number(2); BEGIN

UPDATE customers

SET salary = salary + 500;

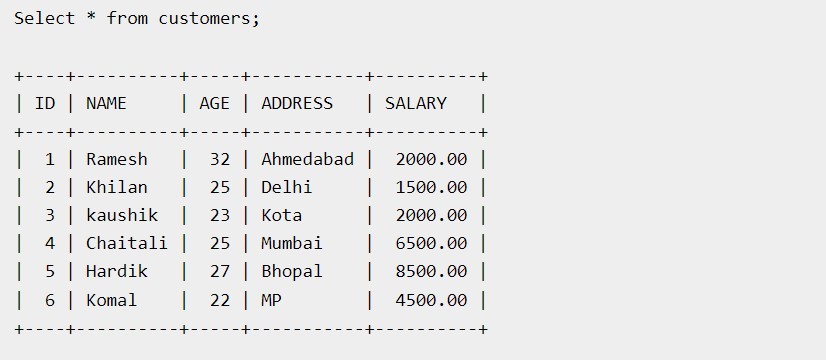
IF sql%notfound THEN dbms\_output.put\_line('no customers selected');

ELSIF sql%found THEN total\_rows := sql%rowcount;

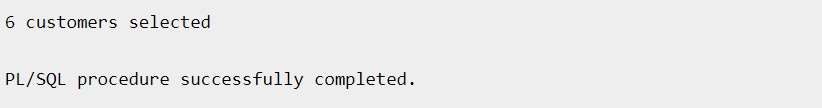
dbms\_output.put\_line( total\_rows || ' customers selected '); END IF;

END;

/



**Output:-**



DECLARE

c\_id customers.id%type; c\_name customers.name%type; c\_addr customers.address%type; CURSOR c\_customers is

SELECT id, name, address FROM customers; BEGIN

OPEN c\_customers; LOOP

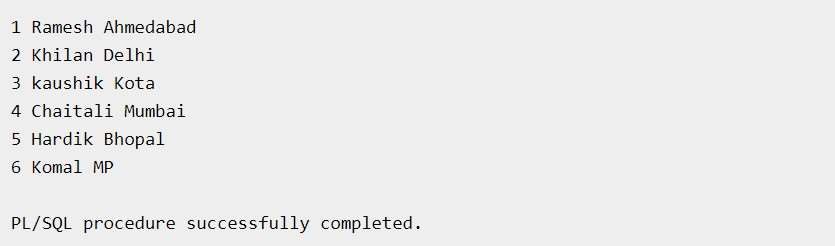
FETCH c\_customers into c\_id, c\_name, c\_addr; EXIT WHEN c\_customers%notfound;

dbms\_output.put\_line(c\_id || ' ' || c\_name || ' ' || c\_addr); END LOOP;

CLOSE c\_customers; END;

/

**Output:-**



**EXPERIMENT 15:-**

**Aim:-** Write a PL/SQL block to implementation of factorial using

function.

# Source Code:-

declare

fac number :=1;

n number := &1;

begin

while n > 0 loop fac:=n\*fac;

n:=n-1;

end loop; dbms\_output.put\_line(fac);

end;



**Output:-**

