

EXPERIMENT~1

Aim: Write SQL queries to create for various databases using DDL commands (i.e. CREATE, ALTER, DROP, TRUNCATE).

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>cd CSE-528  
C:\Users\Heman\Desktop\CSE-528>
```

CREATE TABLE:

Creates a table with specified constraints.

Syntax:

```
CREATE TABLE tablename (  
Column2 data_type[constraint],  
PRIMARY KEY (column1[, column2]),  
FOREIGN KEY (column1[, column2]) REFERENCES tablename] [, CONSTRAINT constraint]);
```

```
SQL> CREATE TABLE NEW (  
2 ID VARCHAR2(5),  
3 NAME VARCHAR2(20) NOT NULL,  
4 DEPT VARCHAR2(15));  
  
Table created.
```

```
SQL> SET LIN100  
SQL> DESC NEW;  
Name Null? Type  
----  
ID          VARCHAR2(5)  
NAME        NOT NULL VARCHAR2(20)  
DEPT        VARCHAR2(15)
```

ALTER TABLE:

Used to add or modify table details like column names and data types, column constraints.

```
SQL> ALTER TABLE NEW  
  2 ADD SALARY NUMERIC(8,2);  
  
Table altered.
```

DROP TABLE:

Deletes the specified table.

```
SQL> DROP TABLE NEW;  
  
Table dropped.
```

Syntax:

```
DROP TABLE table_name;
```

TRUNCATE TABLE:

To remove all rows in a specified table.

Syntax:

```
TRUNCATE TABLE table_name;
```

```
SQL> TRUNCATE TABLE PERSONS;  
  
Table truncated.
```

Conclusion:

In this lab, we have practiced CREATE, ALTER, DROP, and TRUNCATE commands for the user created table.

EXPERIMENT~2

AIM: To write SQL queries to MANIPULATE TABLES for various databases using DML commands (i.e. INSERT, SELECT, UPDATE, DELETE).

CREATING TABLE:

```
SQL> CREATE TABLE persons(
 2      person_id NUMBER PRIMARY KEY,
 3      First_name VARCHAR2(50) NOT NULL,
 4      Last_name VARCHAR2(50) NOT NULL,
 5      phone_no NUMBER(10) NOT NULL
 6  );
```

Table created.

```
SQL> |
```

INSERT COMMAND:

It is used to add values to a table.

Syntax:

```
INSERT INTO table_name
```

```
VALUES (value1, value2, ...., valueN);
```

```
INSERT INTO table_name (column1, column2, ...., columnN)
```

```
VALUES (value1, value2, ...., valueN);
```

```
SQL> INSERT INTO persons(person_id,first_name,last_name,phone_no) VALUES(1,'Suresh','Krishna',7382790163);
1 row created.

SQL> INSERT INTO persons(person_id,first_name,last_name,phone_no) VALUES(2,'JHON','Cristofer',8548585678);
1 row created.

SQL> INSERT INTO persons(person_id,first_name,last_name,phone_no) VALUES(3,'RAM','SHANKAR',8548585678);
1 row created.
```

SELECT COMMAND:

The SELECT command is used to list the contents of a table.

Syntax:

```
SELECT * FROM table_name;  
SELECT column_name FROM table_name;
```

SQL> SELECT * FROM persons;			
PERSON_ID	FIRST_NAME	LAST_NAME	PHONE_NO
1	Suresh	Krishna	7382790163
2	JHON	Cristofer	8548585678
3	RAM	SHANKAR	8548585678

UPDATE COMMAND:

The UPDATE command is used to modify the contents of specified table.

Syntax:

```
UPDATE table_name  
SET column_name = value [,  
Column_name = value]  
[WHERE condition_list];
```

```
SQL> update persons set phone_no = 7993636609 where person_id = 1;  
1 row updated.  
SQL>
```

DELETE COMMAND:

To delete all rows or specified rows in a table.

Syntax:

```
DELETE FROM table_name [ WHERE condition_list];
```

S

```
SQL> delete from persons where person_id = 1;  
1 row deleted.  
SQL>
```

CONCLUSION:

In this lab, we have practiced INSERT, SELECT, UPDATE, and DELETE commands for user created table.

To perform create view we want to create a table

```
SQL> CREATE TABLE person(
 2 pid NUMBER NOT NULL,
 3 pname VARCHAR2(20),
 4 pcity VARCHAR2(20)
 5 );
```

Table created.

INSERT VALUES INTO THE TABLE

```
SQL> INSERT INTO person
 2 VALUES(1,'ABC','ATP');
```

1 row created.

```
SQL> INSERT INTO person
 2 VALUES(1,'ABC','AP');
```

1 row created.

```
SQL> INSERT INTO person
 2 VALUES(3,'AC','AP');
```

1 row created.

```
SQL> INSERT INTO person
 2 VALUES(4,'SDC','CSDP');
```

1 row created.

CREATE VIEWS FOR THE ABOVE TABLE USING BELOW SYNTAX

SYNTAX:

CREATE VIEW VIEW_NAME AS SELECT COLUMNS FROM TABLE;

EX:

```
SQL> CREATE VIEW employee AS SELECT pid, pname FROM person;  
View created.
```

To see the view we use the following keyword

SELECT*FROM employee

```
SQL> SELECT * FROM employee  
2 ;  
  
 PID PNAME  
-----  
 1 ABC  
 1 ABC  
 3 AC  
 4 SDC
```

To add views to the table we use the following syntax:

INSERT INTO VIEW_NAME(COL1,COL2)VALUES(value_list);

EX:

```
SQL> insert into employee(pid,pname)  
2 VALUES(5,'AC');  
  
1 row created.
```

```
SQL> SELECT*FROM employee  
2 ;
```

PID	PNAME
1	ABC
1	ABC
3	AC
4	SDC
5	AC

```
-----  
1 ABC  
1 ABC  
3 AC  
4 SDC  
5 AC
```

We observe that above table is updated.

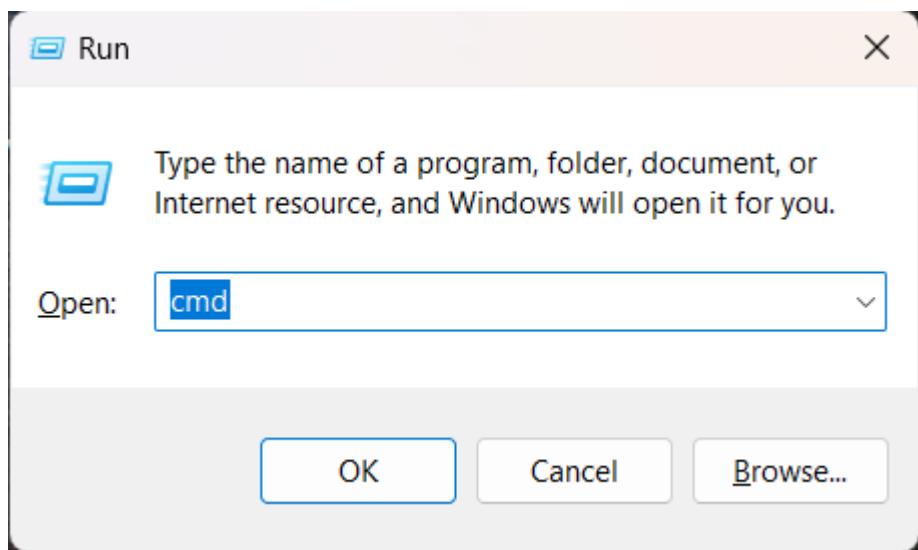
To delete the table

```
SQL> DELETE person;  
  
5 rows deleted.
```

When you delete the table the views and the table is also deleted.

```
SQL> select*from person  
2 ;  
  
no rows selected  
  
SQL> select*from employee  
2 ;  
  
no rows selected
```

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>
```

3. Now create a Directory using **mkdir** or **md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>
```

4. Now, move into the directory by using **cd** command show below.

```
C:\Users\Heman>cd Desktop

C:\Users\Heman\Desktop>mkdir CSE-528

C:\Users\Heman\Desktop>cd CSE-528

C:\Users\Heman\Desktop\CSE-528>
```

5. To Login, Type sqlplus command enter username and password when system is prompted.

Now create a table `instructor` with attributes `id, name, dept_name, salary`.

```
SQL> CREATE TABLE instructor (
  2  id NUMBER(10) PRIMARY KEY,
  3  name VARCHAR2(25) NOT NULL,
  4  dept_name VARCHAR2(25) NOT NULL,
  5  salary NUMBER(10,2) NOT NULL
  6  );
```

Table created.

6. Insert values into table `instructor`.

```
SQL> INSERT INTO instructor VALUES('528', 'Hemanth', 'cse', '100000');

1 row created.

SQL> INSERT INTO instructor VALUES('526', 'Harsha', 'cse', '90000');

1 row created.

SQL> INSERT INTO instructor VALUES('525', 'Nivas', 'cse', '50000');

1 row created.
```

7. Using the command “`SELECT * FROM instructor;`” display table `instructor`.

ID	NAME	DEPT_NAME	SALARY
528	Hemanth	cse	100000
526	Harsha	cse	90000
525	Nivas	cse	50000

8. Create a table `departments` with attributes `dept_id, dept_name, building, budget`

```
SQL> CREATE TABLE departments (
  2  dept_id VARCHAR2(10) PRIMARY KEY,
  3  dept_name VARCHAR2(10) NOT NULL,
  4  building VARCHAR2(20),
  5  budget NUMBER(10,2) NOT NULL
  6  );
```

Table created.

9. Insert values into departments table

```
SQL> INSERT INTO departments VALUES('101','CSE','Block-A','100000');
1 row created.

SQL> INSERT INTO departments VALUES('102','ECE','Block-B','80000');
1 row created.

SQL> INSERT INTO departments VALUES('103','EEE','Block-C','85000');
1 row created.

SQL> INSERT INTO departments VALUES('104','cse','Block-D','500000');
1 row created.
```

10. [UNION](#): The attributes `dept_name` from `instructor` and `departments` are joined using command `UNION`

```
SQL> SELECT dept_name FROM instructor
  2 UNION
  3 SELECT dept_name FROM departments;
```

DEPT_NAME

```
-----
cse
CSE
ECE
EEE
```

11. [UNION ALL](#): The attributes `dept_name` from `instructor` and `department` are joined along with the duplicates using command `UNION ALL`.

```
SQL> SELECT dept_name FROM instructor
  2 UNION ALL
  3 SELECT dept_name FROM departments;

DEPT_NAME
-----
cse
cse
cse
CSE
ECE
EEE
cse

7 rows selected.
```

12. **INTERSECT**: Displays similar values in two or more attributes from department and instructor using command **INTERSECT**.

```
SQL> SELECT dept_name FROM instructor
  2 INTERSECT
  3 SELECT dept_name FROM departments;

DEPT_NAME
-----
cse
```

13. **MINUS**: It eliminates the same values of second column from the first column and represents the remaining values using command **MINUS**.

```
SQL> SELECT dept_name FROM departments
  2 MINUS
  3 SELECT dept_name FROM instructor;

DEPT_NAME
-----
CSE
ECE
EEE
```

14. CROSS JOIN: It cross products the all the attributes using command **CROSS JOIN**.

```
SQL> SELECT i.name,d.dept_name,d.budget
  2 FROM instructor i,department d;
```

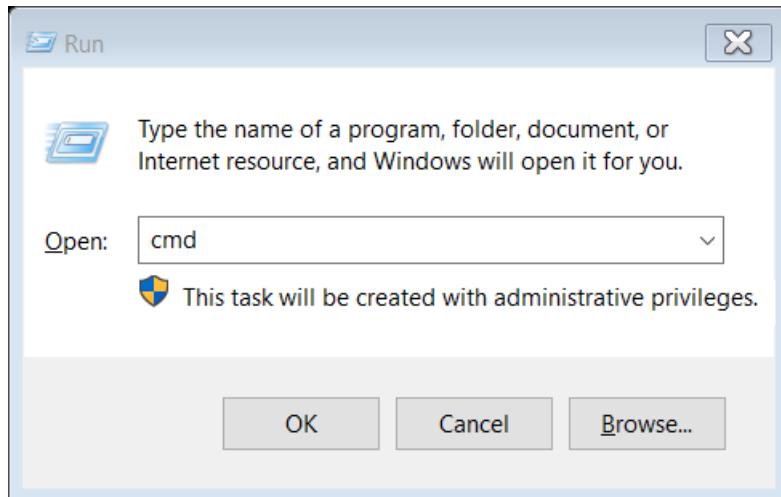
NAME	DEPT_NAME	BUDGET
Hemanth	CSE	50000
Harsha	CSE	50000
Nivas	CSE	50000
Hemanth	CIVIL	90000
Harsha	CIVIL	90000
Nivas	CIVIL	90000
Hemanth	Mech	50000
Harsha	Mech	50000
Nivas	Mech	50000

9 rows selected.

15. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	9
2.	Number of tables creation specified in observation	2
3.	Number of tables you created in the lab	2
4.	Number of Select Statements specified in the observation	2
5.	Number of Select statements you practised in lab	2
6.	Number of Insert Statements specified in observation	6
7	Number of Insert Statements you practiced in Lab	6
8	Total number of Statements specified in lab	20
9	Total number of statements practiced in lab	20
10	Number of any addition statements practiced by you.	3

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

```
C:\Windows\system32\cmd.exe + - Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd Desktop

C:\Users\Windows\Desktop>
```

A screenshot of a Windows Command Prompt window. The title bar shows "C:\Windows\system32\cmd.exe". The window displays the command "cd Desktop" being run, changing the current directory to "C:\Users\Windows\Desktop". The prompt then changes to "C:\Users\Windows\Desktop>".

3. Now create a Directory using **mkdir or md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>
```

- Now, move into the directory by using `cd` command show below

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>cd CSE-528  
C:\Users\Heman\Desktop\CSE-528>
```

- To Login, Type `sqlplus` command enter username and password when system is prompted.

Now create a table `student` with attributes `id, name, dept_name, salary`.

```
SQL> CREATE TABLE STUDENT (  
 2  ID VARCHAR(5),  
 3  NAME VARCHAR2(20) NOT NULL,  
 4  DEPT VARCHAR2(15),  
 5  SALARY NUMERIC(8,2)  
 6  );
```

```
Table created.
```

- Insert values into table `STUDENT`.

```

SQL> INSERT INTO student VALUES('528','HEMU','CSE','100000');

1 row created.

SQL> INSERT INTO student VALUES('529','RAJA','CSM','200000');

1 row created.

SQL> INSERT INTO student VALUES('530','RANI','CSD','300000');

1 row created.

SQL> INSERT INTO student VALUES('531','NANI','','');

1 row created.

SQL> INSERT INTO student VALUES('532','HEMANTH','','');

1 row created.

```

7. Using the command “[SELECT * FROM STUDENT;](#)” display table instructor.

ID	NAME	DEPT	SALARY
528	HEMU	CSE	100000
529	RAJA	CSM	200000
530	RANI	CSD	300000
531	NANI		
532	HEMANTH		

8. [IS NULL:](#) It is used to check null values and display null attributes.
It displays [attributes](#) that have [null values](#).

```

SQL> SELECT NAME FROM STUDENT
  2 WHERE DEPT IS NULL;

NAME
-----
NANI
HEMANTH

```

9. [BETWEEN:](#) This is used to check range of values.
By the following command it displays all the attributes [between 100000 and 200000](#).

```
SQL> SELECT ID FROM STUDENT
2 WHERE SALARY BETWEEN 100000 AND 200000;
```

ID
528
529

10. LIKE: This is used to check given string is present or not.
It displays all the attributes that start with character 'H'.

```
SQL> SELECT DEPT FROM STUDENT
2 WHERE NAME LIKE 'H%';
```

DEPT
CSE

11. IN: This is used to check a member is in a set or not.

```
SQL> SELECT NAME FROM STUDENT
2 WHERE ID IN('528','529');
```

NAME
HEMU
RAJA

12. EXITS: This is used to check whether given set is empty or not.
It displays null attributes that are null according to the given condition.

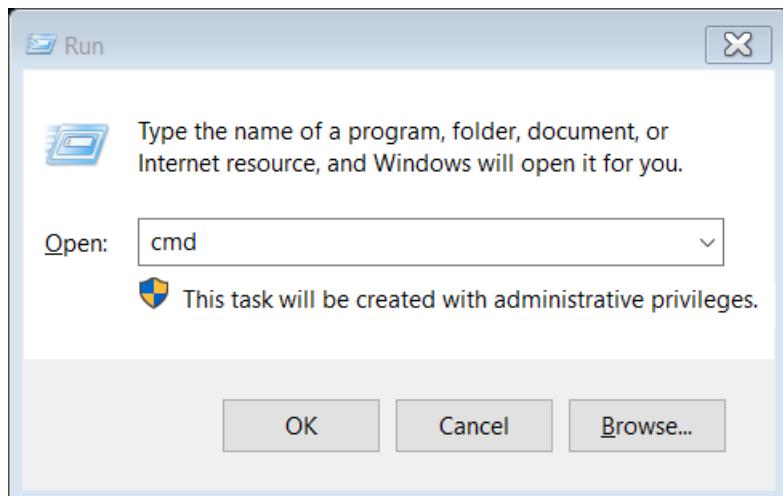
```
SQL> SELECT * FROM STUDENT WHERE EXISTS
  2 (SELECT * FROM STUDENT WHERE DEPT IS NULL);
```

ID	NAME	DEPT	SALARY
528	HEMU	CSE	100000
529	RAJA	CSM	200000
530	RANI	CSD	300000
531	NANI		
532	HEMANTH		

13. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	8
2.	Number of tables creation specified in observation	1
3.	Number of tables you created in the lab	1
4.	Number of Select Statements specified in the observation	7
5.	Number of Select statements you practised in lab	7
6.	Number of Insert Statements specified in observation	5
7	Number of Insert Statements you practiced in Lab	5
8	Total number of Statements specified in lab	20
9	Total number of statements practiced in lab	20

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

```
C:\Windows\system32\cmd.exe + - Microsoft Windows [Version 10.0.22621.2428] (c) Microsoft Corporation. All rights reserved. C:\Users\Windows>cd Desktop C:\Users\Windows\Desktop>
```

A screenshot of a Windows Command Prompt window. The title bar shows "C:\Windows\system32\cmd.exe". The window displays the command "cd Desktop" being run and the resulting directory change to "C:\Users\Windows\Desktop".

3. Now create a Directory using **mkdir** or **md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**.

```
C:\Users\Heman>cd Desktop C:\Users\Heman\Desktop>mkdir CSE-528 C:\Users\Heman\Desktop>
```

A screenshot of a Windows Command Prompt window. The title bar shows "C:\Users\Heman>cd Desktop". The window displays the command "mkdir CSE-528" being run, which creates a new directory named "CSE-528" in the current directory.

- Now, move into the directory by using `cd` command show below.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>cd CSE-528  
C:\Users\Heman\Desktop\CSE-528>
```

- To Login, Type `sqlplus` command enter username and password when system is prompted.

Now create a table `student` with attributes `name, rollno, branch`.

```
SQL> CREATE TABLE student (  
 2  ROLLNO NUMBER,  
 3  NAME VARCHAR2(20),  
 4  BRANCH VARCHAR2(20)  
 5  );  
  
Table created.  
  
SQL>
```

- Insert values into table `student`.

```
SQL> INSERT INTO student VALUES('1','Hemu','cse');

1 row created.

SQL> INSERT INTO student VALUES('2','Raja','csm');

1 row created.

SQL> INSERT INTO student VALUES('3','Rani','csd');

1 row created.

SQL> INSERT INTO student VALUES('4','Nani','civil');

1 row created.
```

7. Using the command “`SELECT * FROM student;`” display table student.

```
SQL> SELECT * FROM student;
```

ROLLNO	NAME	BRANCH
1	Hemu	cse
2	Raja	csm
3	Rani	csd
4	Nani	civil

8. Create a table `library` with attributes `rollno, boom`.

```
SQL> CREATE TABLE library (
  2  ROLLNO NUMBER,
  3  BOOM VARCHAR2(20)
  4  );
```

```
Table created.
```

9. Insert values into library table.

```
SQL> INSERT INTO library VALUES('2','DBMS');

1 row created.

SQL> INSERT INTO library VALUES('3','Java');

1 row created.

SQL> INSERT INTO library VALUES('4','Maths');

1 row created.
```

10. Display table library using [SELECT * FROM library](#) command.

```
SQL> SELECT * FROM library;

ROLLNO BOOK
-----
11 DBMS
12 Java
13 Maths
```

11. [CONDITIONAL JOIN](#): It helps in retrieving the desired data and performing complex queries.

```
SQL> SELECT *
  2  FROM student
  3  JOIN library ON student.ROLLNO = library.ROLLNO;

ROLLNO NAME          BRANCH      ROLLNO
----- -----
BOOM
-----
2 Raja           csm        2
DBMS
-----
3 Rani           csd        3
Java
-----
4 Nani           civil      4
Maths
```

12. [EQUI JOIN](#): It helps in retrieving related information from different tables [by](#) matching corresponding values.

```
SQL> SELECT * FROM student JOIN library USING(ROLLNO);
```

ROLLNO	NAME	BRANCH	BOOM
2	Raja	csm	DBMS
3	Rani	csd	Java
4	Nani	civil	Maths

13. LEFT OUTER JOIN: It combines data from two or more tables based on the matching values in specified columns, but it also includes unmatched rows from the left table.

```
SQL> SELECT * FROM student NATURAL LEFT OUTER JOIN library;
```

ROLLNO	NAME	BRANCH	BOOM
2	Raja	csm	DBMS
3	Rani	csd	Java
4	Nani	civil	Maths
1	Hemu	cse	

14. RIGHT OUTER JOIN: It combines data from two or more tables based on the matching values in specified columns, but it also includes unmatched rows from the right table.

```
SQL> SELECT * FROM student NATURAL RIGHT OUTER JOIN library;
```

ROLLNO	NAME	BRANCH	BOOM
2	Raja	csm	DBMS
3	Rani	csd	Java
4	Nani	civil	Maths

15. FULL OUTERJOIN: It includes all the rows from both the left and right tables, even if there is no match.

```
SQL> SELECT * FROM student NATURAL FULL OUTER JOIN library;
```

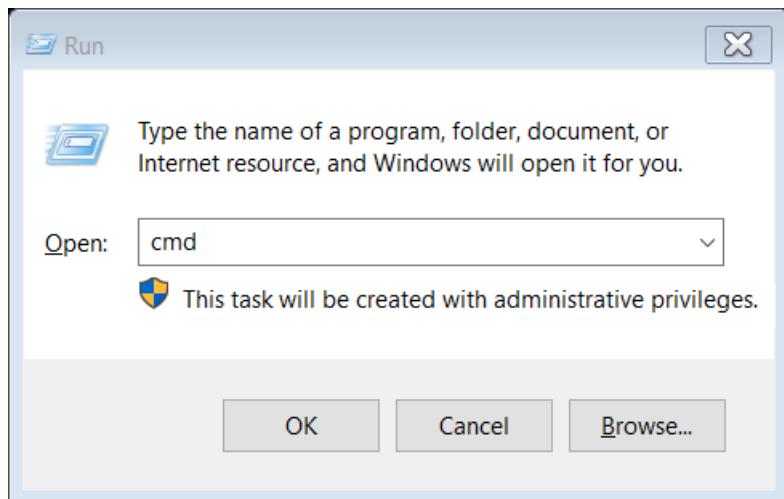
ROLLNO	NAME	BRANCH	BOOM
1	Hemu	cse	
2	Raja	csm	DBMS
3	Rani	csd	Java
4	Nani	civil	Maths

16. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	11
2.	Number of tables creation specified in observation	2
3.	Number of tables you created in the lab	2

4.	Number of Select Statements specified in the observation	7
5.	Number of Select statements you practised in lab	7
6	Number of Insert Statements specified in observation	7
7	Number of Insert Statements you practiced in Lab	7
8	Total number of Statements specified in lab	20
9	Total number of statements practiced in lab	20

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

```
C:\Windows\system32\cmd.exe + - ×
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd Desktop

C:\Users\Windows\Desktop>
```

A screenshot of a Windows Command Prompt window. The title bar shows "C:\Windows\system32\cmd.exe". The window displays the command "cd Desktop" being run, changing the current directory to "C:\Users\Windows\Desktop". The prompt then changes to "C:\Users\Windows\Desktop>".

3. Now create a Directory using **mkdir or md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**.

```
C:\Users\Heman>cd Desktop  
  
C:\Users\Heman\Desktop>mkdir CSE-528  
  
C:\Users\Heman\Desktop>
```

- Now, move into the directory by using `cd` command show below.

```
C:\Users\Heman>cd Desktop  
  
C:\Users\Heman\Desktop>mkdir CSE-528  
  
C:\Users\Heman\Desktop>cd CSE-528  
  
C:\Users\Heman\Desktop\CSE-528>
```

- To Login, Type `sqlplus` command enter username and password when system is prompted.

Now create a table `Instructor` with attributes `id, name, salary`.

```
SQL> CREATE TABLE instructor (  
  2  ID VARCHAR2(5),  
  3  NAME VARCHAR2(20) NOT NULL,  
  4  DEPT_NAME VARCHAR2(20),  
  5  SALARY NUMERIC(8,2)  
  6  );  
  
Table created.  
  
SQL> |
```

- Insert values into table `instructor`.

```
SQL> INSERT INTO instructor VALUES('101', 'Hemanth', 'cse', '65000');

1 row created.

SQL> INSERT INTO instructor VALUES('102', 'Kumar', 'finance', '90000');

1 row created.

SQL> INSERT INTO instructor VALUES('103', 'Raja', 'music', '40000');

1 row created.

SQL> INSERT INTO instructor VALUES('104', 'Rani', 'physics', '95000');

1 row created.

SQL> INSERT INTO instructor VALUES('105', 'King', 'cse', '63000');

1 row created.
```

7. Using the command “[SELECT * FROM instructor;](#)” display table instructor.

```
SQL> SELECT * FROM instructor;

ID      NAME          DEPT_NAME        SALARY
-----  -----
101    Hemanth        cse            65000
102    Kumar          finance        90000
103    Raja           music          40000
104    Rani           physics        95000
105    King            cse            63000
```

8. [Count:](#) It displays the count of members present in instructor.

```
SQL> SELECT COUNT(*) FROM instructor;

COUNT(*)
-----
5

SQL>
```

9. [AVERAGE\(AVG\):](#) It displays average salary of each department.

```
SQL> SELECT DEPT_NAME,avg(salary) as avg_salary
  2  FROM instructor
  3  GROUP by DEPT_NAME;
```

DEPT_NAME	AVG_SALARY
cse	64000
finance	90000
music	40000
physics	95000

10. Create a table `student` with attributes `id, name, marks`.

```
SQL> CREATE TABLE student (
  2  ID VARCHAR2(5),
  3  NAME VARCHAR2(20),
  4  MARKS NUMBER
  5  );
```

Table created.

```
SQL> |
```

11. Insert values into student table.

```
SQL> INSERT INTO student VALUES('528','Hemu','20');

1 row created.

SQL> INSERT INTO student VALUES('529','Nani','30');

1 row created.

SQL> INSERT INTO student VALUES('530','Rani','40');

1 row created.
```

12. Display table student using `SELECT * FROM student` command.

```
SQL> SELECT * FROM student;
```

ID	NAME	MARKS
528	Hemu	20
529	Nani	30
530	Rani	40

13. SUM: It displays sum of all the marks from the table.

```
SQL> SELECT SUM(marks) FROM student;
```

```
SUM(MARKS)
```

```
-----  
90
```

```
SQL> |
```

14. MIN: It displays the minimum marks from the table.

```
SQL> SELECT MIN(marks) FROM student;
```

```
MIN(MARKS)
```

```
-----  
20
```

```
SQL> |
```

15. MAX: It displays the maximum marks from the table.

```
SQL> SELECT MAX(marks) FROM student;
```

```
MAX(MARKS)
```

```
-----  
40
```

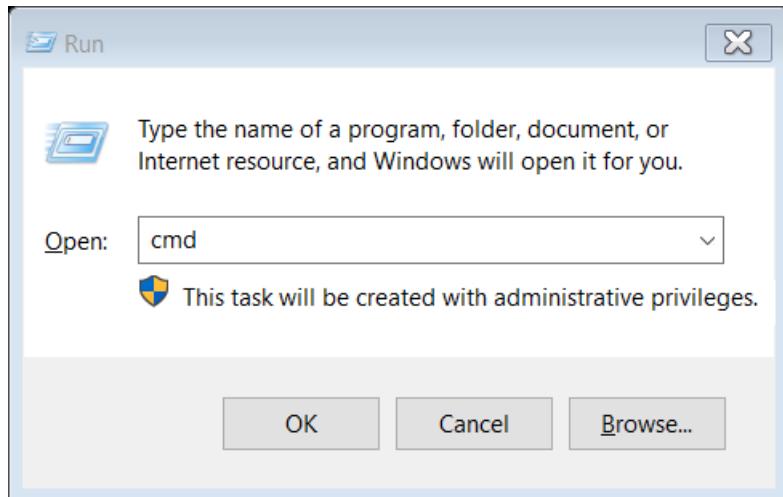
```
SQL> |
```

16. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	11

2.	Number of tables creation specified in observation	2
3.	Number of tables you created in the lab	2
4.	Number of Select Statements specified in the observation	7
5.	Number of Select statements you practised in lab	7
6.	Number of Insert Statements specified in observation	8
7.	Number of Insert Statements you practiced in Lab	8
8.	Total number of Statements specified in lab	20
9.	Total number of statements practiced in lab	20

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

A screenshot of a Windows Command Prompt window. The title bar shows "C:\Windows\system32\cmd.exe". The window content shows:

```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd Desktop

C:\Users\Windows\Desktop>
```

The command "cd Desktop" is entered and executed, changing the current directory to "Desktop". The prompt then changes to "C:\Users\Windows\Desktop>".

3. Now create a Directory using **mkdir or md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**.

```
C:\Users\Heman>cd Desktop

C:\Users\Heman\Desktop>mkdir CSE-528

C:\Users\Heman\Desktop>
```

- Now, move into the directory by using `cd` command show below.

```
C:\Users\Heman>cd Desktop

C:\Users\Heman\Desktop>mkdir CSE-528

C:\Users\Heman\Desktop>cd CSE-528

C:\Users\Heman\Desktop\CSE-528>
```

- To Login, Type `sqlplus` command enter username and password when system is prompted.

[LOWER\(\)](#): It converts a string to lowercase.

```
SQL> SELECT LOWER('HELLO WORLD') FROM DUAL;

LOWER('HELL
-----
hello world

SQL> |
```

- [UPPER\(\)](#): It converts a string to uppercase.

```
SQL> SELECT UPPER('hello world') FROM DUAL;

UPPER('HELL
-----
HELLO WORLD

SQL> |
```

7. [INITCAP\(\)](#): It returns the capitals of selected string.

```
INITCAP('HE
-----
Hello World

SQL> |
```

8. [CONCAT\(\)](#): It adds two or more expressions together.

```
SQL> SELECT CONCAT('HELLO', 'WORLD') FROM DUAL;

CONCAT('HE
-----
HELLOWORLD

SQL> |
```

9. [SUBSTR\(\)](#): It extracts a substring from a string.

```
SQL> SELECT SUBSTR('HELLO WORLD', 1, 5) FROM DUAL;

SUBST
-----
HELLO

SQL> |
```

10. [LENGTH\(\)](#): It returns the length of the given string.

```
SQL> SELECT LENGTH('HELLO WORLD') FROM DUAL;

LENGTH('HELLOWORLD')
-----
11

SQL> |
```

11. [INSTR\(\)](#): It returns the position or the first occurrence of a string in another string.

```
SQL> SELECT INSTR('HELLO WORLD', 'HELLO') FROM DUAL;

INSTR('HELLOWORLD', 'HELLO')
-----
1

SQL> |
```

12. [TRIM\(\)](#): It removes the selected one from string.

```
SQL> SELECT TRIM('H' FROM 'HELLO WORLD') FROM DUAL;

TRIM('H'FR
-----
ELLO WORLD

SQL> |
```

13. [Round\(\)](#): It returns the specified values.

```
SQL> SELECT ROUND(45.626,2) FROM DUAL;

ROUND(45.626,2)
-----
45.63

SQL> |
```

14. [TRUNCATE\(\)](#): It removes the decimal values which are specified.

```
SQL> SELECT TRUNC(45.626,0) FROM DUAL;

TRUNC(45.626,0)
-----
45

SQL> |
```

15. [MOD\(\)](#): It returns the remainder.

```
SQL> SELECT MOD(1600,99) FROM DUAL;  
  
MOD(1600,99)  
-----  
16
```

```
SQL> |
```

16. SYSDATE ():

```
SQL> SELECT SYSDATE FROM DUAL;
```

```
SYSDATE
```

```
-----  
03-DEC-23
```

```
SQL> |
```

17. MONTHS_BETWEEN ():

```
SQL> SELECT MONTHS_BETWEEN(SYSDATE, '17-NOV-23') FROM DUAL;
```

```
MONTHS_BETWEEN(SYSDATE, '17-NOV-23')
```

```
-----  
.562598193
```

```
SQL> |
```

18. ADD_MONTHS ():

```
SQL> SELECT ADD_MONTHS(SYSDATE,5) FROM DUAL;
```

```
ADD_MONTH
```

```
-----
```

```
03-MAY-24
```

```
SQL> |
```

19. NEXT_DAY ():

```
SQL> SELECT NEXT_DAY(SYSDATE, 'WEDNESDAY') FROM DUAL;

NEXT_DAY(
-----
06-DEC-23

SQL> |
```

20. LAST_DAY():

```
SQL> SELECT LAST_DAY(SYSDATE) FROM DUAL;

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

SQL> |
```

21. TRUNC():

```
SQL> SELECT TRUNC(SYSDATE, 'DAY') FROM DUAL;

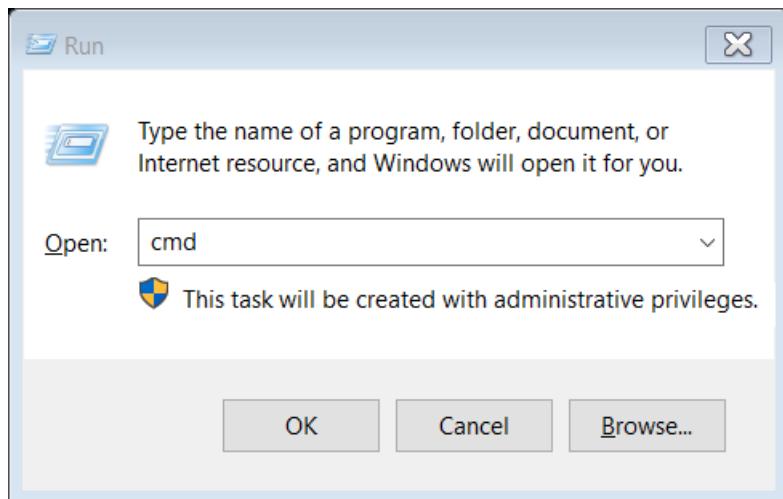
TRUNC(SYS
-----
03-DEC-23

SQL> |
```

22. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	17
4.	Number of Select Statements specified in the observation	17
5.	Number of Select statements you practised in lab	17
10	Total number of Statements specified in lab	21
11	Total number of statements practiced in lab	21
12	Number of any addition statements practiced by you.	3

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

```
C:\Windows\system32\cmd.exe + - ×
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd Desktop

C:\Users\Windows\Desktop>
```

A screenshot of a Windows Command Prompt window. The title bar shows "C:\Windows\system32\cmd.exe". The window displays the command "cd Desktop" being run, changing the current directory to "C:\Users\Windows\Desktop". The prompt then changes to "C:\Users\Windows\Desktop>".

3. Now create a Directory using **mkdir or md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**.

```
C:\Users\Heman>cd Desktop  
  
C:\Users\Heman\Desktop>mkdir CSE-528  
  
C:\Users\Heman\Desktop>
```

- Now, move into the directory by using `cd` command show below.

```
C:\Users\Heman>cd Desktop  
  
C:\Users\Heman\Desktop>mkdir CSE-528  
  
C:\Users\Heman\Desktop>cd CSE-528  
  
C:\Users\Heman\Desktop\CSE-528>
```

- To Login, Type `sqlplus` command enter username and password when system is prompted.

PRIMARY KEY: A primary key is a field which can uniquely identify each row in table and this constraint is used to specify a field as primary key.

```
SQL> CREATE TABLE student (  
 2 ID NUMBER,  
 3 NAME VARCHAR(10),  
 4 ADDRESS VARCHAR(20)  
 5 );
```

```
Table created.
```

```
SQL> |
```

- FOREIGN KEY:** A foreign key is a field which can uniquely identify each row in another table.

```
SQL> CREATE TABLE orders (  
 2 O_ID NUMBER NOT NULL,  
 3 C_ID NUMBER,  
 4 PRIMARY KEY(O_ID),  
 5 FOREIGN KEY(C_ID)REFERENCES customer(C_ID)  
 6 );
```

```
Table created.
```

7. UNIQUE: This constraint when specified with a column, tells that the values in the column must be unique i.e, the values in any row of a column must not be repeated.

```
SQL> CREATE TABLE student (
  2  ID NUMBER UNIQUE,
  3  NAME VARCHAR(10),
  4  ADDRESS VARCHAR(20)
  5 );
```

```
Table created.
```

```
SQL> |
```

8. NOT NULL: This constraint tells that we cannot store a null value in a column.

```
SQL> CREATE TABLE student (
  2  ID NUMBER,
  3  NAME VARCHAR(10) NOT NULL,
  4  ADDRESS VARCHAR(20)
  5 );
```

```
Table created.
```

```
SQL> |
```

9. DEFAULT: This constraint specifies a default value for the column when no value is specified by the user.

```
SQL> CREATE TABLE student (
  2  ID NUMBER,
  3  NAME VARCHAR(10) NOT NULL,
  4  AGE NUMBER DEFAULT 18
  5 );
```

```
Table created.
```

```
SQL> |
```

10. CHECK: This constraint helps to validate the value for the column to meet a particular condition i.e it helps to ensure that the value stored in a column meets a specific condition.

```
SQL> CREATE TABLE student (
 2  ID NUMBER NOT NULL,
 3  NAME VARCHAR(10) NOT NULL,
 4  AGE NUMBER NOT NULL CHECK(AGE>=18)
 5  );
```

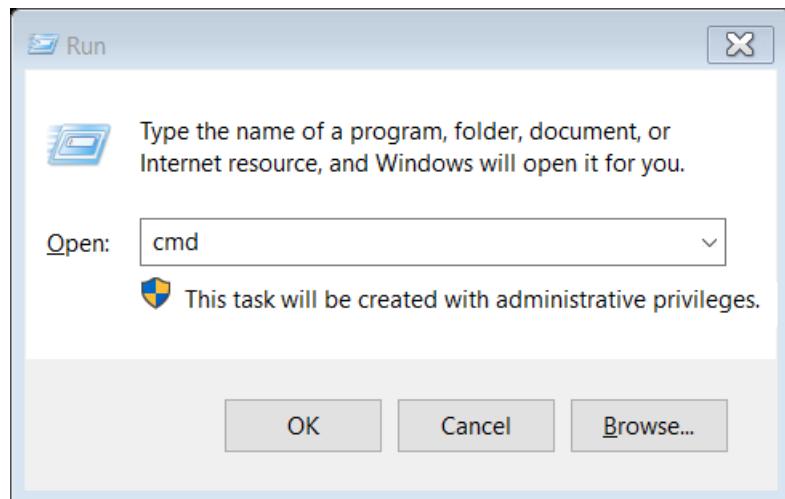
Table created.

```
SQL> |
```

11. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	6
2.	Number of tables creation specified in observation	6
3.	Number of tables you created in the lab	6
4.	Total number of Statements specified in lab	20
5.	Total number of statements practiced in lab	20
6.	Number of any addition statements practiced by you.	3

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

```
C:\Windows\system32\cmd.exe + - ×
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd Desktop

C:\Users\Windows\Desktop>
```

A screenshot of a Windows Command Prompt window. The title bar shows the path "C:\Windows\system32\cmd.exe". The window displays the standard Windows welcome message. In the command line, the user types "cd Desktop" and presses Enter. The cursor then moves to the end of the command line, indicating the operation is complete.

3. Now create a Directory using **mkdir or md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**.

```
C:\Users\Heman>cd Desktop

C:\Users\Heman\Desktop>mkdir CSE-528

C:\Users\Heman\Desktop>
```

4. Now, move into the directory by using `cd` command show below.

```
C:\Users\Heman>cd Desktop

C:\Users\Heman\Desktop>mkdir CSE-528

C:\Users\Heman\Desktop>cd CSE-528

C:\Users\Heman\Desktop\CSE-528>
```

5. To Login, Type `sqlplus` command enter username and password when system is prompted.

Now write the code to find factorial of given number using WHILE LOOP. We use `'/'` to end and execute the program.

```
C:\Windows\system32\cmd.exe: X + ▾

SQL> DECLARE
  2   fact NUMBER:=1;
  3   n NUMBER;
  4   n1 NUMBER;
  5   BEGIN
  6   n:=&n;
  7   n1:=n;
  8   WHILE n>0 LOOP
  9   fact:=n*fact;
 10  n:=n-1;
 11  END LOOP;
 12  DBMS_OUTPUT.PUT_LINE('The Factorial of'||n1||' is'||fact);
 13  END;
 14 /
```

6. After the execution the following displays.

```
Enter value for n: 7
old   6: n:=&n;
new   6: n:=7;

PL/SQL procedure successfully completed.
```

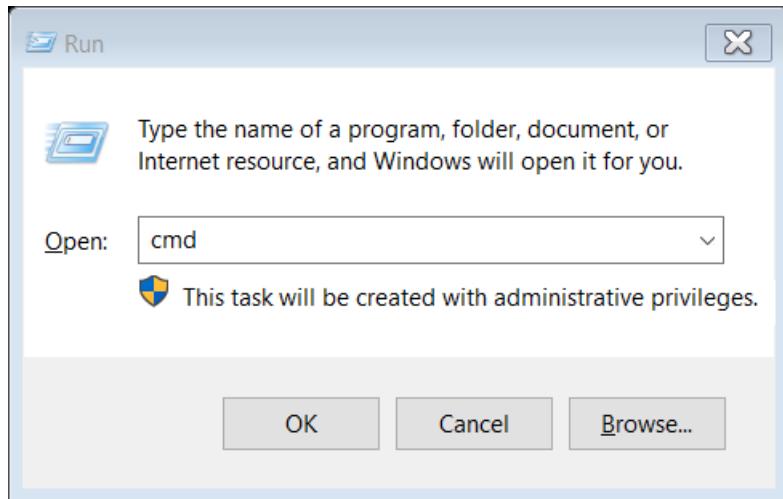
7. To display the output, we use SET SERVEROUT ON. The following output is displayed giving the factorial of the given number.

```
PL/SQL procedure successfully completed.  
  
SQL> SET SERVEROUT ON  
SQL> /  
Enter value for n: 7  
old   6: n:=&n;  
new   6: n:=7;  
The Factorial of 7 is 5040  
  
PL/SQL procedure successfully completed.  
  
SQL> |
```

8. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	3
2.	Number of tables creation specified in observation	3
3.	Number of Select statements you practised in lab	5
4.	Total number of programs specified in lab	1
5.	Total number of programs practiced in lab	1

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

```
C:\Windows\system32\cmd.exe + - Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd Desktop

C:\Users\Windows\Desktop>
```

A screenshot of a Windows Command Prompt window. The title bar shows "C:\Windows\system32\cmd.exe". The window displays the command "cd Desktop" being run, changing the current directory to "C:\Users\Windows\Desktop". The prompt then changes to "C:\Users\Windows\Desktop>".

3. Now create a Directory using **mkdir or md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>
```

4. Now, move into the directory by using `cd` command show below.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>cd CSE-528  
C:\Users\Heman\Desktop\CSE-528>
```

5. To Login, Type `sqlplus` command enter username and password when system is prompted.

Now write the code to find given number is prime or not. We use `'/'` to end and execute the program.

```
C:\Windows\system32\cmd.e: + ^

1  DECLARE
2  n NUMBER;
3  n1 NUMBER;
4  i NUMBER;
5  temp NUMBER;
6  BEGIN
7  n :=&n;
8  n1:=n;
9  i := 2;
10 temp := 1;
11 FOR i IN 2..n/2
12 LOOP
13 IF MOD(n, i) = 0
14 THEN
15 temp := 0;
16 EXIT;
17 END IF;
18 END LOOP;
19 IF temp = 1
20 THEN
21 DBMS_OUTPUT.PUT_LINE(n||' is a prime number');
22 ELSE
23 DBMS_OUTPUT.PUT_LINE(n||' is not a prime number');
24 END IF;
25* END;
SQL> /
```

6. After the execution the following displays.

```
Enter value for n: 528
old    7: n := &n;
new    7: n := 528;
528 is not a prime number

PL/SQL procedure successfully completed.
```

7. To display the output, we use SET SERVEROUT ON. The following output is displayed that given number is prime or not.

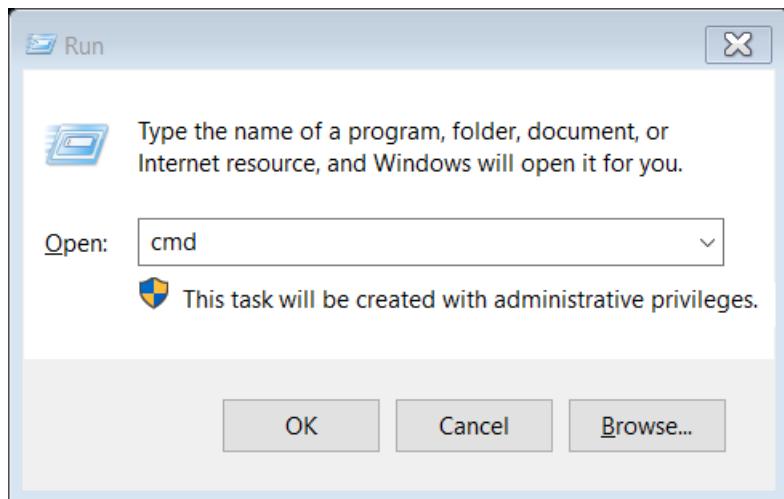
```
SQL> SET SERVEROUT ON
SQL> /
Enter value for n: 528
old    7: n := &n;
new    7: n := 528;
528 is not a prime number

PL/SQL procedure successfully completed.
```

8. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	3
2.	Number of tables creation specified in observation	3
3.	Number of Select statements you practised in lab	5
4.	Total number of programs specified in lab	1
5.	Total number of programs practiced in lab	1

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

A screenshot of a Windows Command Prompt window. The title bar shows the path "C:\Windows\system32\cmd.exe". The window content shows the following text:

```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd Desktop

C:\Users\Windows\Desktop>
```

The command "cd Desktop" is typed at the prompt, and the cursor is positioned after the command.

3. Now create a Directory using **mkdir or md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**.

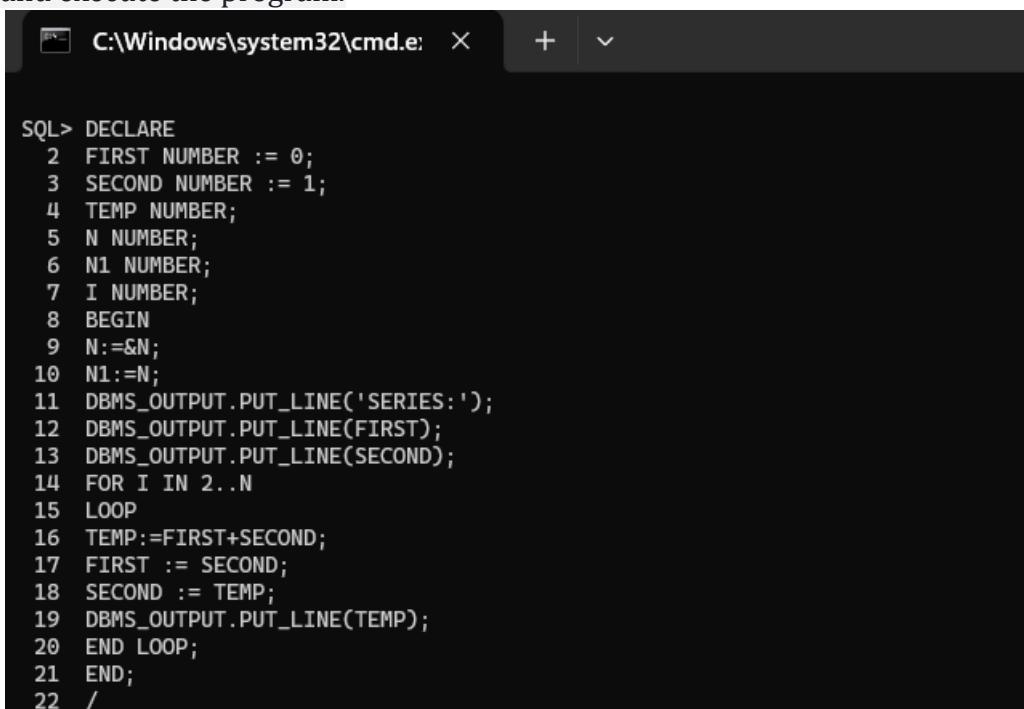
```
C:\Users\Heman>cd Desktop  
  
C:\Users\Heman\Desktop>mkdir CSE-528  
  
C:\Users\Heman\Desktop>
```

4. Now, move into the directory by using `cd` command show below.

```
C:\Users\Heman>cd Desktop  
  
C:\Users\Heman\Desktop>mkdir CSE-528  
  
C:\Users\Heman\Desktop>cd CSE-528  
  
C:\Users\Heman\Desktop\CSE-528>
```

5. To Login, Type `sqlplus` command enter username and password when system is prompted.

Now write the code to display Fibonacci series until the given number. We use `'/'` to end and execute the program.



The screenshot shows a Windows Command Prompt window titled "C:\Windows\system32\cmd.e...". The window contains the following PL/SQL code:

```
SQL> DECLARE
 2  FIRST NUMBER := 0;
 3  SECOND NUMBER := 1;
 4  TEMP NUMBER;
 5  N NUMBER;
 6  N1 NUMBER;
 7  I NUMBER;
 8  BEGIN
 9  N:=&N;
10  N1:=N;
11  DBMS_OUTPUT.PUT_LINE('SERIES:');
12  DBMS_OUTPUT.PUT_LINE(FIRST);
13  DBMS_OUTPUT.PUT_LINE(SECOND);
14  FOR I IN 2..N
15  LOOP
16  TEMP:=FIRST+SECOND;
17  FIRST := SECOND;
18  SECOND := TEMP;
19  DBMS_OUTPUT.PUT_LINE(TEMP);
20  END LOOP;
21  END;
22  /
```

6. After the execution the following displays.

```
Enter value for n: 9
old   9: N:=&N;
new   9: N:=9;

PL/SQL procedure successfully completed.
```

7. To display the output, we use SET SERVEROUT ON. The following output is displaying the Fibonacci series until the given number.

```
SQL> SET SERVEROUT ON
SQL> /
Enter value for n: 9
old   9: N:=&N;
new   9: N:=9;
SERIES:
0
1
1
2
3
5
8
13
21
34

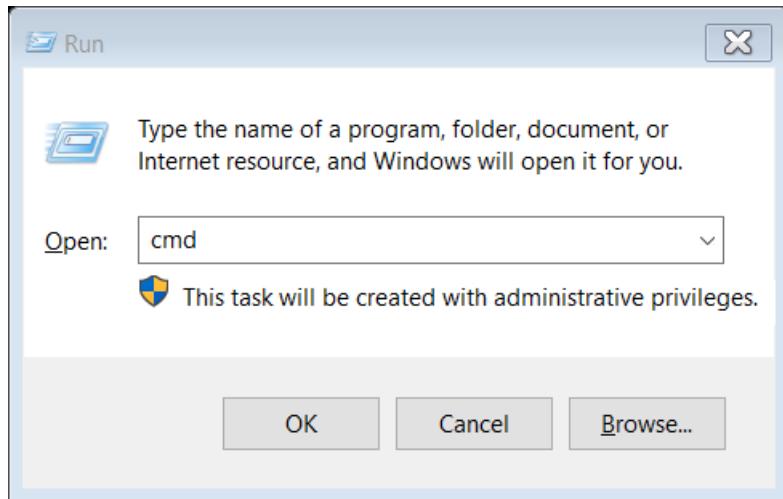
PL/SQL procedure successfully completed.

SQL> |
```

8. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	3
2.	Number of tables creation specified in observation	3
3.	Number of Select statements you practised in lab	5
4.	Total number of programs specified in lab	1
5.	Total number of programs practiced in lab	1

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

```
C:\Windows\system32\cmd.exe + - Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd Desktop

C:\Users\Windows\Desktop>
```

A screenshot of a Windows Command Prompt window. The title bar shows "C:\Windows\system32\cmd.exe". The window displays the command "cd Desktop" being run and the resulting directory change to "C:\Users\Windows\Desktop". The command prompt prompt is ">".

3. Now create a Directory using **mkdir or md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>
```

- Now, move into the directory by using `cd` command show below.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>cd CSE-528  
C:\Users\Heman\Desktop\CSE-528>
```

- To Login, Type `sqlplus` command enter username and password when system is prompted.

Now create a table `sailor1` with attributes `id, name`.

```
SQL> CREATE TABLE sailor1(  
 2      id NUMBER PRIMARY KEY,  
 3      name VARCHAR2(50) NOT NULL  
 4  );
```

```
Table created.
```

```
SQL>
```

- Create procedure.

```
SQL> CREATE OR REPLACE PROCEDURE insertuser(id IN NUMBER, name IN VARCHAR2)
  2 AS
  3 BEGIN
  4 INSERT INTO sailor1 VALUES(id, name);
  5 DBMS_OUTPUT.PUT_LINE('Record inserted successfully');
  6 END;
  7 /
```

Procedure created.

```
SQL>
```

7. Insert a user into the table.

```
SQL> DECLARE
  2 co NUMBER;
  3 BEGIN
  4 insertuser(11, 'RANI');
  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>
```

8. Insert two records into the table.

```
SQL> DECLARE
  2 co NUMBER;
  3 BEGIN
  4 insertuser(43, 'Madhuri');
  5 SELECT COUNT(*) INTO co FROM sailor1;
  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.

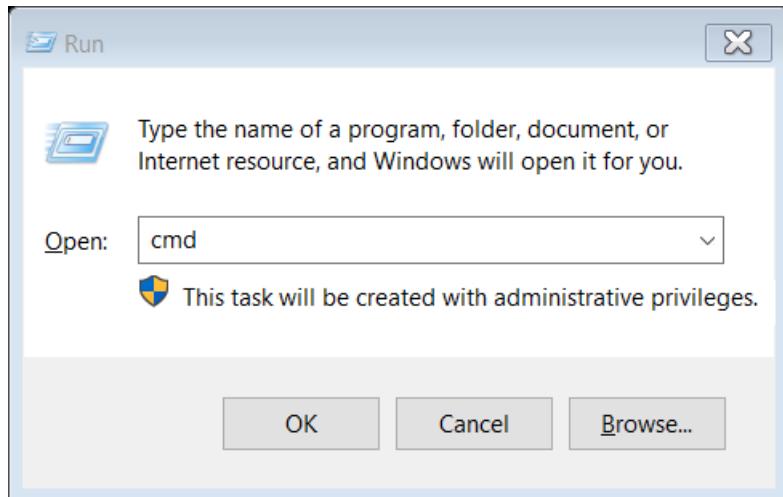
```
SQL> |
```

9. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	4
2.	Number of tables creation specified in observation	1

3.	Number of tables you created in the lab	1
4.	Number of Insert Statements specified in observation	2
5.	Number of Insert Statements you practiced in Lab	2
6.	Total number of Statements specified in lab	5
7.	Total number of statements practiced in lab	5

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

A screenshot of a Windows Command Prompt window. The title bar shows the path "C:\Windows\system32\cmd.exe". The window content shows the following text:

```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd Desktop

C:\Users\Windows\Desktop>
```

The command "cd Desktop" is typed at the prompt, and the window shows the current directory as "C:\Users\Windows\Desktop".

3. Now create a Directory using **mkdir or md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>
```

4. Now, move into the directory by using `cd` command show below.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>cd CSE-528  
C:\Users\Heman\Desktop\CSE-528>
```

5. To Login, Type `sqlplus` command enter username and password when system is prompted.
6. Create a function.

```
SQL> CREATE OR REPLACE FUNCTION ADDER(N1 IN NUMBER, N2 IN NUMBER)  
  2  RETURN NUMBER  
  3  IS  
  4  N3 NUMBER(8);  
  5  BEGIN  
  6  N3 :=N1+N2;  
  7  RETURN N3;  
  8  END;  
  9 /  
  
Function created.
```

7. Displaying the OUTPUT.

```
SQL> DECLARE
  2   N3 NUMBER(2);
  3   BEGIN
  4     N3 := ADDER(11,22);
  5     DBMS_OUTPUT.PUT_LINE('ADDITION IS: ' || N3);
  6   END;
  7 /

PL/SQL procedure successfully completed.

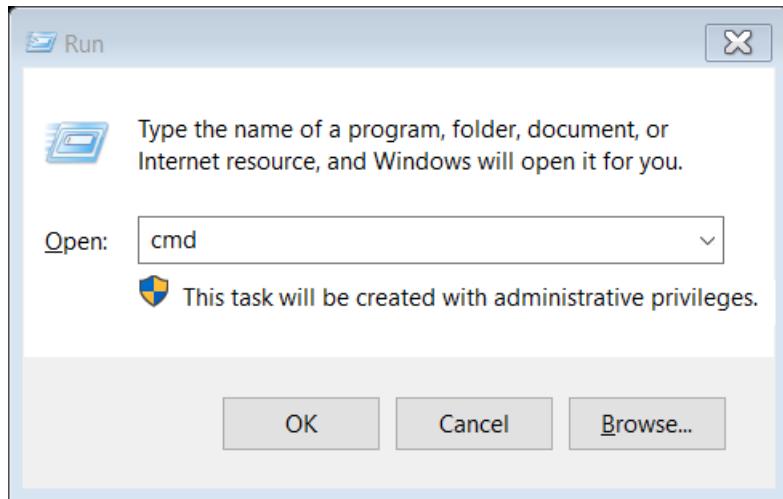
SQL> SET SERVEROUT ON
SQL> /
ADDITION IS: 33

PL/SQL procedure successfully completed.
```

8. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	2
2.	Total number of Statements specified in lab	3
3.	Total number of statements practiced in lab	3

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

```
C:\Windows\system32\cmd.exe + - Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd Desktop

C:\Users\Windows\Desktop>
```

A screenshot of a Windows Command Prompt window. The title bar shows "C:\Windows\system32\cmd.exe". The window displays the command "cd Desktop" being run and the resulting directory change to "C:\Users\Windows\Desktop". The command prompt prompt is visible at the end of the line.

3. Now create a Directory using **mkdir or md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>
```

4. Now, move into the directory by using `cd` command show below.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>cd CSE-528  
C:\Users\Heman\Desktop\CSE-528>
```

5. To Login, Type `sqlplus` command enter username and password when system is prompted.

Now create a table `instructor`.

```
SQL> CREATE TABLE INSTRUCTOR (  
  2   ID VARCHAR2(5),  
  3   NAME VARCHAR2(20) NOT NULL,  
  4   DEPT_NAME VARCHAR2(20),  
  5   SALARY NUMERIC(8,2) CHECK (SALARY > 29000),  
  6   PRIMARY KEY (ID),  
  7   FOREIGN KEY (DEPT_NAME) REFERENCES DEPARTMENT(DEPT_NAME) ON DELETE SET NULL  
  8 );  
  
Table created.
```

6. Insert values into table `instructor`.

```

SQL> INSERT ALL
  2  INTO instruc VALUES (1,'Abhi','CSE',50000)
  3  INTO instruc VALUES (2,'Narsimha','CSM',75000)
  4  INTO instruc VALUES (3,'Balaji','CSE',80000)
  5  INTO instruc VALUES (4,'Rani','CSD',47000)
  6  SELECT * FROM dual;

4 rows created.

SQL> |

```

7. Create a Trigger.

```

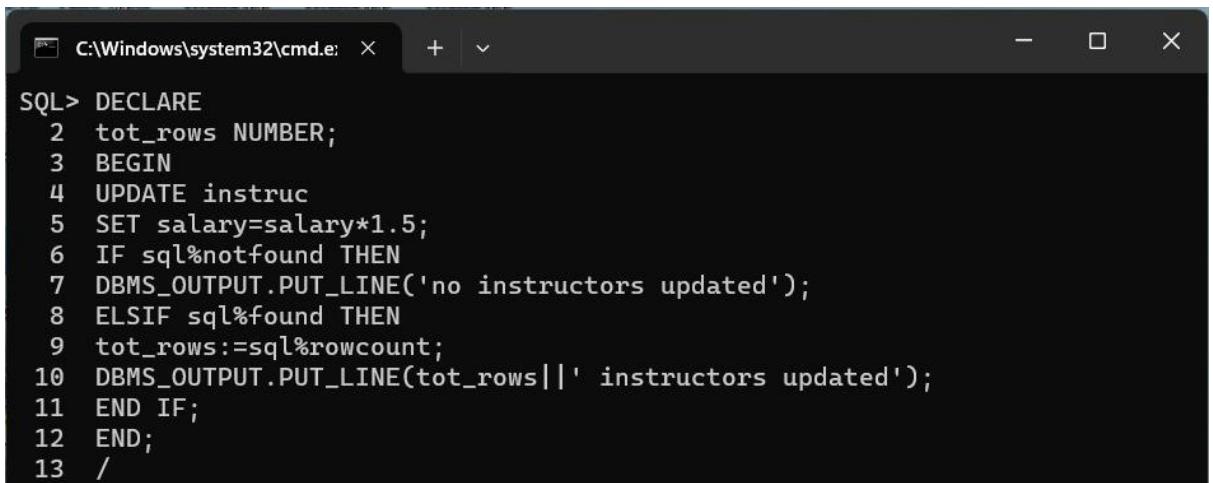
SQL> CREATE OR REPLACE TRIGGER display_changes
  2  BEFORE UPDATE ON instruc
  3  FOR EACH ROW
  4  WHEN (NEW.ID = OLD.ID)
  5  DECLARE
  6    sal_diff number;
  7  BEGIN
  8    sal_diff := :NEW.salary - :OLD.salary;
  9    dbms_output.put_line('Old salary: ' || :OLD.salary);
10    dbms_output.put_line('New salary: ' || :NEW.salary);
11    dbms_output.put_line('Salary difference: ' || sal_diff);
12  END;
13  /

```

Trigger created.

SQL>

8. Updating the values of the rows by the given condition.



```

SQL> DECLARE
  2  tot_rows NUMBER;
  3  BEGIN
  4  UPDATE instruc
  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  /

```

9. The values are updated.

```
Old salary: 50000
New salary: 75000
Salary difference: 25000
Old salary: 75000
New salary: 112500
Salary difference: 37500
Old salary: 80000
New salary: 120000
Salary difference: 40000
Old salary: 47000
New salary: 70500
Salary difference: 23500
4 instructors updated

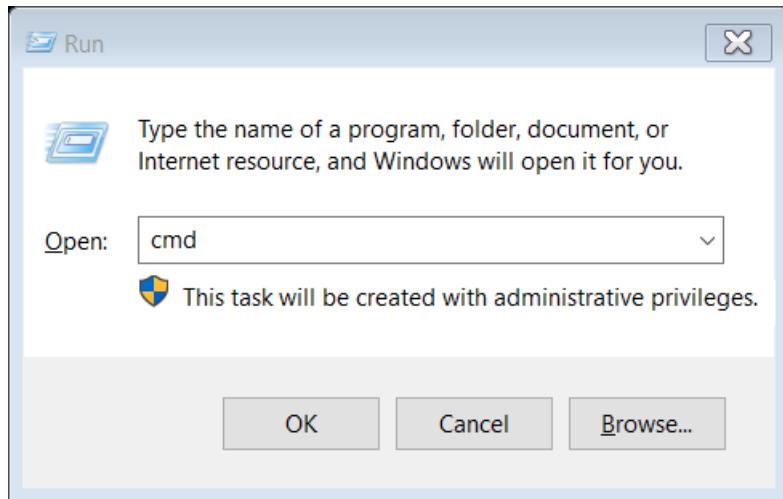
PL/SQL procedure successfully completed.

SQL> |
```

10. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	5
2.	Number of tables creation specified in observation	1
3.	Number of tables you created in the lab	1
4.	Number of Insert Statements specified in observation	2
5.	Number of Insert Statements you practiced in Lab	2
6.	Total number of Statements specified in lab	5
7.	Total number of statements practiced in lab	5

1. Open the command prompt Press **WIN+R** type cmd.



2. Once cmd prompt open go to **DESKTOP** using cd Desktop.

A screenshot of a Windows Command Prompt window. The title bar shows the path "C:\Windows\system32\cmd.exe". The window content shows the command line:

```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Windows>cd Desktop

C:\Users\Windows\Desktop>
```

The command "cd Desktop" is entered and the prompt changes to show the new directory path.

3. Now create a Directory using **mkdir or md** command using your branch abbreviation and last 3 digits hall ticket number like **md cse-528**.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>
```

- Now, move into the directory by using `cd` command show below.

```
C:\Users\Heman>cd Desktop  
C:\Users\Heman\Desktop>mkdir CSE-528  
C:\Users\Heman\Desktop>cd CSE-528  
C:\Users\Heman\Desktop\CSE-528>
```

- To Login, Type `sqlplus` command enter username and password when system is prompted.

Now create a table `customers` with attributes `id, name, age, salary`.

```
SQL> CREATE TABLE customers(  
 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> |
```

- Updating the rows.

```
SQL> DECLARE
  2  tot_rows NUMBER;
  3  BEGIN
  4  UPDATE customers 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 /
No customers updated

PL/SQL procedure successfully completed.

SQL> |
```

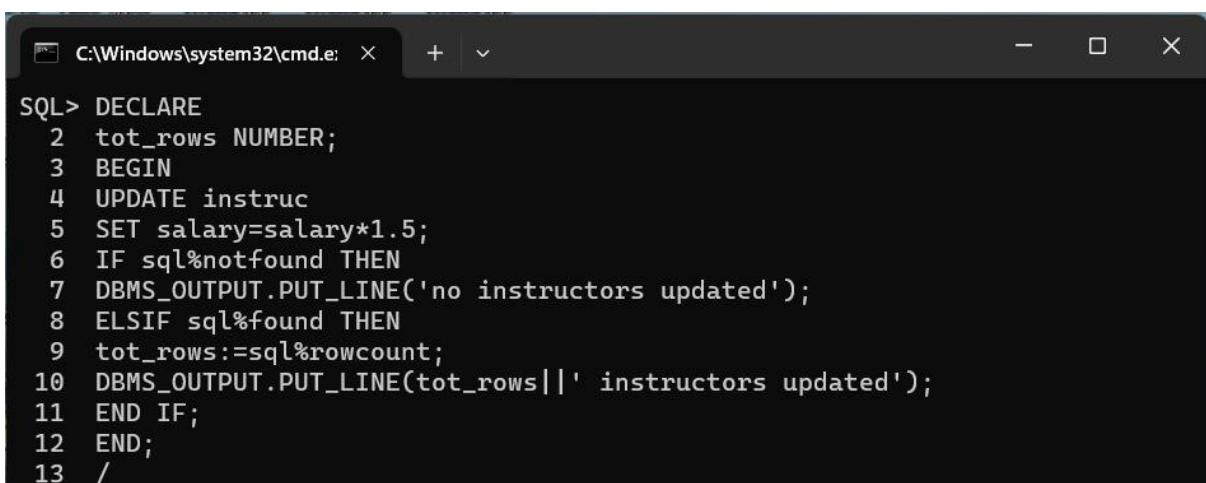
7. Inserting values into customer table.

```
SQL> INSERT ALL
  2  INTO customers VALUES (501,'Ramu',22,60000)
  3  INTO customers VALUES (502,'Ramesh',33,70000)
  4  INTO customers VALUES (503,'Suresh',23,65000)
  5  INTO customers VALUES (504,'Ravi',25,60000)
  6  SELECT * FROM dual;

4 rows created.

SQL> |
```

8. Updating the values of the rows by the given condition.



A screenshot of a Windows Command Prompt window titled 'C:\Windows\system32\cmd.exe'. The window contains the following PL/SQL code:

```
SQL> DECLARE
  2  tot_rows NUMBER;
  3  BEGIN
  4  UPDATE instruc
  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 /
```

9. Program using explicit cursor.

S

```

SQL> DECLARE
 2  c_id customers.id%type;
 3  c_name customers.name%type;
 4  c_age customers.age%type;
 5  CURSOR c_customers IS
 6  SELECT id,name,age FROM customers;
 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 /
501 Ramu 22
502 Ramesh 33
503 Suresh 23
504 Ravi 25

PL/SQL procedure successfully completed.

SQL> |

```

10. Summary of the Lab Report.

SNO	Summary Information	Total
1.	Number of Screen Shorts taken from Step 5	5
2.	Number of tables creation specified in observation	1
3.	Number of tables you created in the lab	1
4.	Number of Insert Statements specified in observation	2
5.	Number of Insert Statements you practiced in Lab	2
6.	Total number of Statements specified in lab	5
7.	Total number of statements practiced in lab	5