Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

DEPARTMENT OF INFORMATION TECHNOLOGY

Academic Year: 2023 - 24

COURSE CODE: DJS22ITL303 **CLASS: S. Y. B. Tech. Sem III (I1)**

COURSE NAME: Database Management Systems **SAP ID:60003220045**

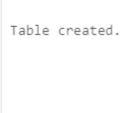
NAME: Anish Sharma **DATE: 3rd December 2023**

EXPERIMENT NO:7
To implement Triggers

Q1

i)

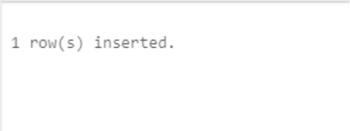
```
CREATE TABLE employee
(
   empno integer,
   fname varchar2(20),
   lname varchar2(20),
   salary integer,
   address varchar2(30)
);
```



```
Table created.
```

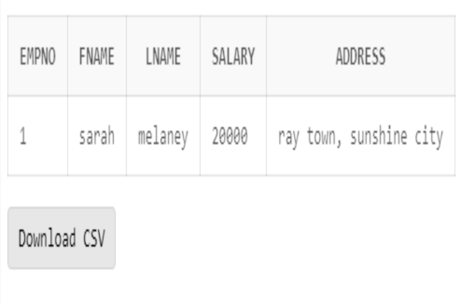
ii)

```
INSERT into employee
values(1,'sarah','melaney',20000,'ray town, sunshine
city');
```



```
1 row(s) inserted.
```

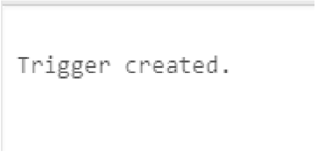
```
SELECT *
FROM employee
```



EMPNO	FNAME	LNAME	SALARY	ADDRESS
1	sarah	melaney	20000	ray town, sunshine city

iii)

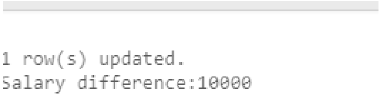
```
CREATE OR REPLACE TRIGGER pcd1
AFTER
UPDATE ON employee
FOR EACH ROW
DECLARE sal_diff number;
BEGIN
sal_diff:= :NEW.salary- :OLD.salary;
dbms_output.put_line('Salary difference:'||sal_diff);
END;
```



```
Trigger created.
```

iv) && v)

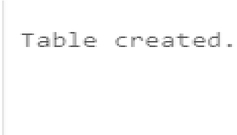
```
UPDATE  employee
set salary='30000';
```



```
1 row(s) updated.
Salary difference:10000
```

Q2

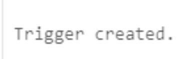
```
CREATE TABLE GRADES
(
   SENO number,
   M1 number,
   M2 number,
   M3 number,
   Avg_M number
);
```



```
Table created.
```

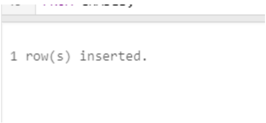
a)

```
CREATE OR REPLACE TRIGGER GRADES_TRG
AFTER
INSERT OR UPDATE ON GRADES
FOR EACH ROW
BEGIN
:NEW.Avg_M:= (:NEW.M1+:NEW.M2+:NEW.M3)/3;
END;
```



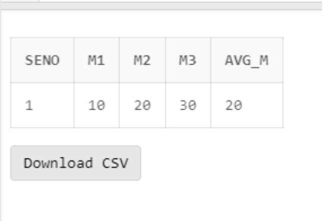
```
Trigger created.
```

```
INSERT into GRADES(SENO,M1,M2,M3) values(1,10,20,30);
```



```
1 row(s) inserted.
```

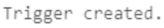
```
SELECT *
FROM GRADES;
```



SENO	M1	M2	M3	AVG_M
1	10	20	30	20

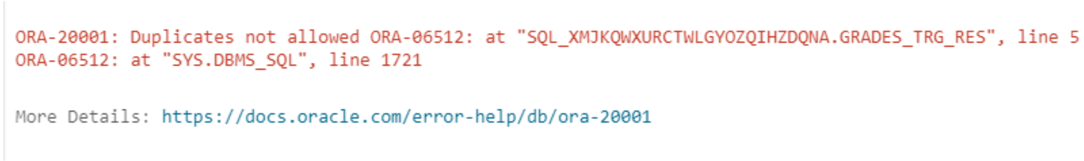
b)

```
CREATE OR REPLACE TRIGGER GRADES_TRG_RES
BEFORE
INSERT OR UPDATE ON GRADES
FOR EACH ROW
DECLARE duplicate_count number;
BEGIN
        SELECT COUNT(*) INTO duplicate_count FROM GRADES WHERE SENO=:NEW.SENO;
        IF duplicate_count>0 THEN
    raise_application_error(-20001,'Duplicates not allowed');
  END IF;
END;
```



```
Trigger created.
```

```
INSERT into GRADES(SENO,M1,M2,M3) values(1,20,20,20);
```



```
ORA-20001: Duplicates not allowed ORA-06512: at "SQL_XMJKQWXURCTWLGYOZQIHZDQNA.GRADES_TRG_RES", line 5
ORA-06512: at "SYS.DBMS_SQL", line 1721

More Details: https://docs.oracle.com/error-help/db/ora-20001
```

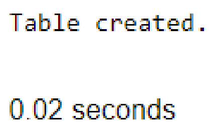
EXPERIMENT NO. 8

To Study and Implement TCL Commands

Implement the SQL statements for the following questions

1)

```
create table Employeee
(
first_name varchar(20),
last_name  varchar(20),
salary integer,
dept varchar(20),
dob Date,
city varchar(20)
);
```



```
Table created.

0.02 seconds
```

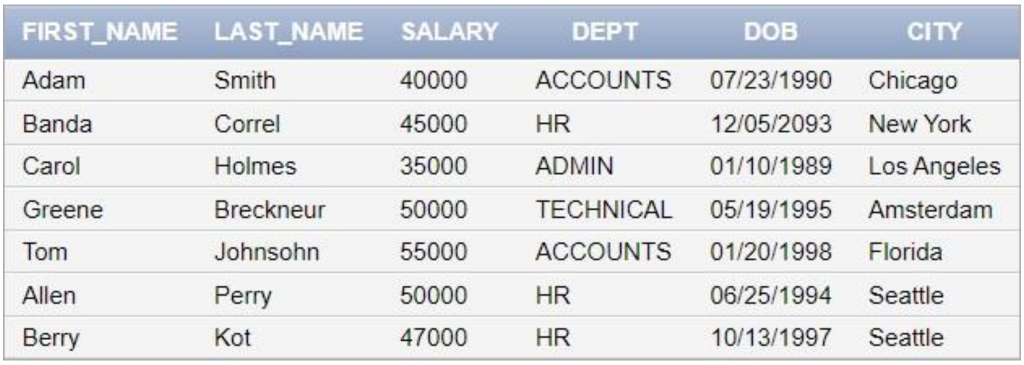
2)

```
insert into Employeee values('Adam', 'Smith', 40000, 'ACCOUNTS', 'jul-23-1990' ,'Chicago');
insert into Employeee values('Banda', 'Correl', 45000, 'HR', 'dec-05-2093' ,'New York');
insert into Employeee values('Carol', 'Holmes', 35000, 'ADMIN', 'jan-10-1989' ,'Los Angeles');
insert into Employeee values('Greene', 'Breckneur', 50000, 'TECHNICAL', 'may-19-1995' ,'Amsterdam');
insert into Employeee values('Tom', 'Johnsohn', 55000, 'ACCOUNTS', 'jan-20-1998' ,'Florida');
```

```
1 row(s) inserted.
```

3)

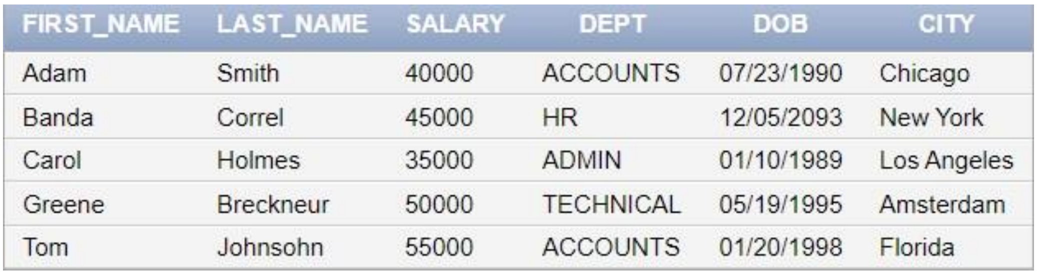
```
Begin
savepoint A;
insert into Employeee values('Allen', 'Perry', 50000, 'HR', 'june-25-1994' ,'Seattle');
End;

Begin
savepoint B;
insert into Employeee values('Berry', 'Kot', 47000, 'HR', 'oct-13-1997' ,'Seattle');
End;
select *
from Employeee;
```



FIRST_NAME	LAST_NAME	SALARY	DEPT	DOB	CITY
Adam	Smith	40000	ACCOUNTS	07/23/1990	Chicago
Banda	Correl	45000	HR	12/05/2093	New York
Carol	Holmes	35000	ADMIN	01/10/1989	Los Angeles
Greene	Breckneur	50000	TECHNICAL	05/19/1995	Amsterdam
Tom	Johnsohn	55000	ACCOUNTS	01/20/1998	Florida
Allen	Perry	50000	HR	06/25/1994	Seattle
Berry	Kot	47000	HR	10/13/1997	Seattle

4)

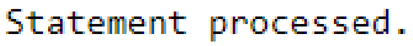
```
Begin
Rollback to A;
End;
select *
from Employeee;
```



FIRST_NAME	LAST_NAME	SALARY	DEPT	DOB	CITY
Adam	Smith	40000	ACCOUNTS	07/23/1990	Chicago
Banda	Correl	45000	HR	12/05/2093	New York
Carol	Holmes	35000	ADMIN	01/10/1989	Los Angeles
Greene	Breckneur	50000	TECHNICAL	05/19/1995	Amsterdam
Tom	Johnsohn	55000	ACCOUNTS	01/20/1998	Florida

5)

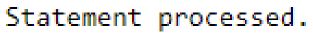
```
commit;
```



```
Statement processed.
```

6)

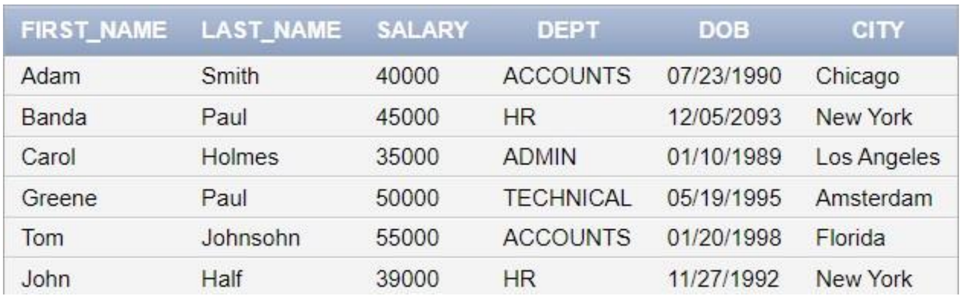
```
Begin
savepoint P;
insert into Employeee values('John', 'Half', 39000, 'HR', 'nov-27-1992' ,'New York');
End;
```



```
Statement processed.
```

7)

```
Begin
savepoint C;
update Employeee
set last_name= 'Paul'
where first_name = 'Banda';
End;

Begin
savepoint D;
update Employeee
set last_name= 'Paul'
where first_name = 'Greene';
End;

select *
from Employeee;
```



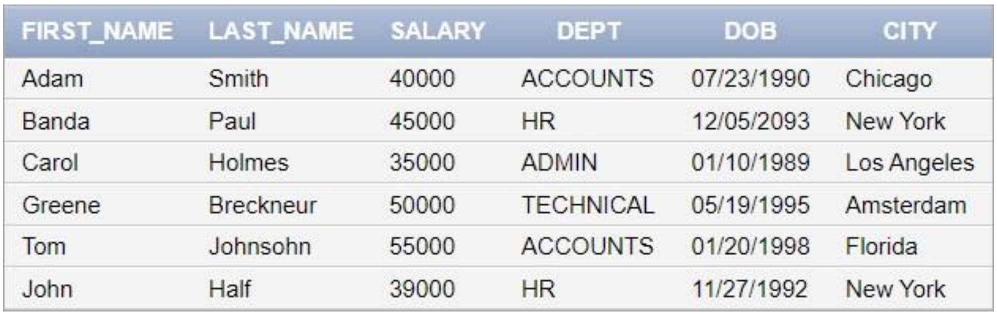
FIRST_NAME	LAST_NAME	SALARY	DEPT	DOB	CITY
Adam	Smith	40000	ACCOUNTS	07/23/1990	Chicago
Banda	Paul	45000	HR	12/05/2093	New York
Carol	Holmes	35000	ADMIN	01/10/1989	Los Angeles
Greene	Paul	50000	TECHNICAL	05/19/1995	Amsterdam
Tom	Johnsohn	55000	ACCOUNTS	01/20/1998	Florida
John	Half	39000	HR	11/27/1992	New York

8)

```
Begin
Rollback to D;
End;
```

```
Statement processed.
```

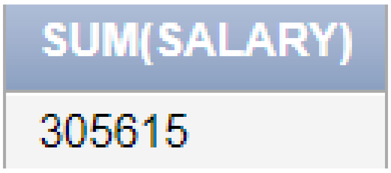
```
select *
from Employeee;
```



FIRST_NAME	LAST_NAME	SALARY	DEPT	DOB	CITY
Adam	Smith	40000	ACCOUNTS	07/23/1990	Chicago
Banda	Paul	45000	HR	12/05/2093	New York
Carol	Holmes	35000	ADMIN	01/10/1989	Los Angeles
Greene	Breckneur	50000	TECHNICAL	05/19/1995	Amsterdam
Tom	Johnsohn	55000	ACCOUNTS	01/20/1998	Florida
John	Half	39000	HR	11/27/1992	New York

9)

```
Begin
savepoint E;
update Employeee
set salary=salary*1.05;
End;

Begin savepoint F;
update Employeee
set salary=salary*1.05;
End;

Begin savepoint G;
update Employeee
set salary=salary*1.05;
End;
Select SUM(salary) from Employeee;
```

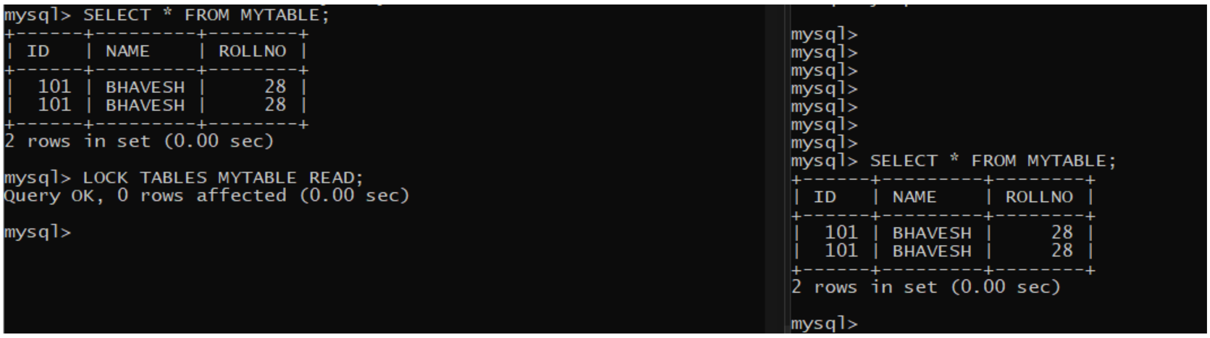


SUM(SALARY)
305615

EXPERIMENT NO. 9

1.

When user 1 locks in read mode, user 2 can read without any conflict but cannot write as read – write is a conflict.



```
mysql> SELECT * FROM MYTABLE;
+------+---------+--------+
| ID   | NAME    | ROLLNO |
+------+---------+--------+
|  101 | BHAVESH |     28 |
|  101 | BHAVESH |     28 |
+------+---------+--------+
2 rows in set (0.00 sec)

mysql> LOCK TABLES MYTABLE READ;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

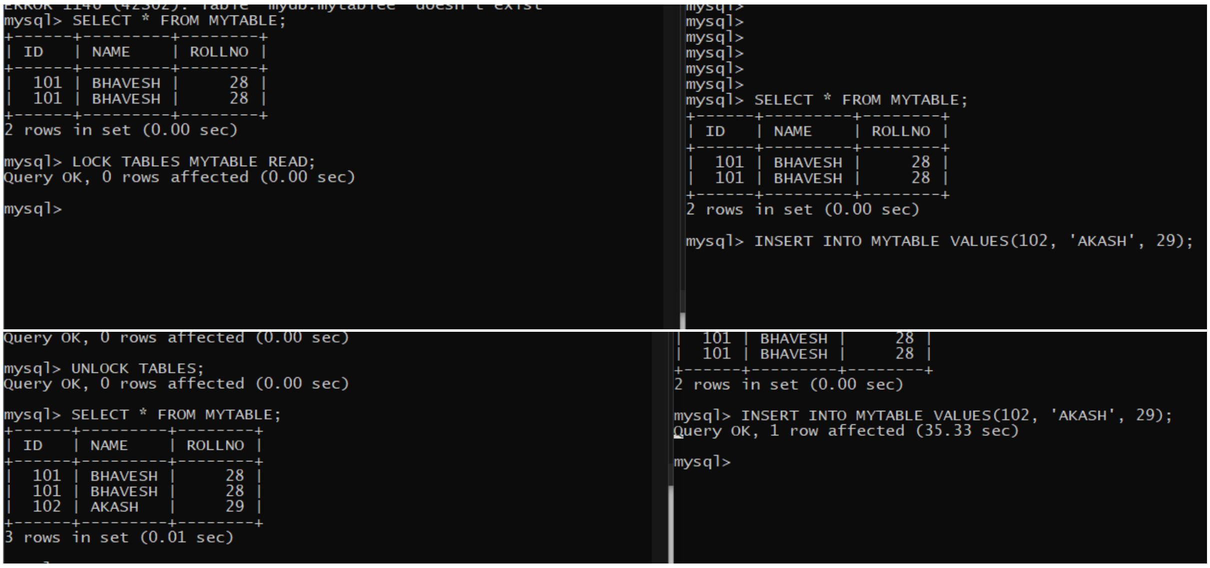
```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> SELECT * FROM MYTABLE;
+------+---------+--------+
| ID   | NAME    | ROLLNO |
+------+---------+--------+
|  101 | BHAVESH |     28 |
|  101 | BHAVESH |     28 |
+------+---------+--------+
2 rows in set (0.00 sec)

mysql>
```

User 2 can read safely.

2.

Read – write conflict

When user 1 is locked in read mode, user 2 cannot perform write operations.



```
ERROR 1146 (42S02): Table 'mydb.mytablee' doesn't exist
mysql> SELECT * FROM MYTABLE;
+------+---------+--------+
| ID   | NAME    | ROLLNO |
+------+---------+--------+
|  101 | BHAVESH |     28 |
|  101 | BHAVESH |     28 |
+------+---------+--------+
2 rows in set (0.00 sec)

mysql> LOCK TABLES MYTABLE READ;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

```
mysql>
mysql>
mysql>
mysql>
mysql>
mysql>
mysql> SELECT * FROM MYTABLE;
+------+---------+--------+
| ID   | NAME    | ROLLNO |
+------+---------+--------+
|  101 | BHAVESH |     28 |
|  101 | BHAVESH |     28 |
+------+---------+--------+
2 rows in set (0.00 sec)

mysql> INSERT INTO MYTABLE VALUES(102, 'AKASH', 29);
```

```
Query OK, 0 rows affected (0.00 sec)

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT * FROM MYTABLE;
+------+---------+--------+
| ID   | NAME    | ROLLNO |
+------+---------+--------+
|  101 | BHAVESH |     28 |
|  101 | BHAVESH |     28 |
|  102 | AKASH   |     29 |
+------+---------+--------+
3 rows in set (0.01 sec)
```

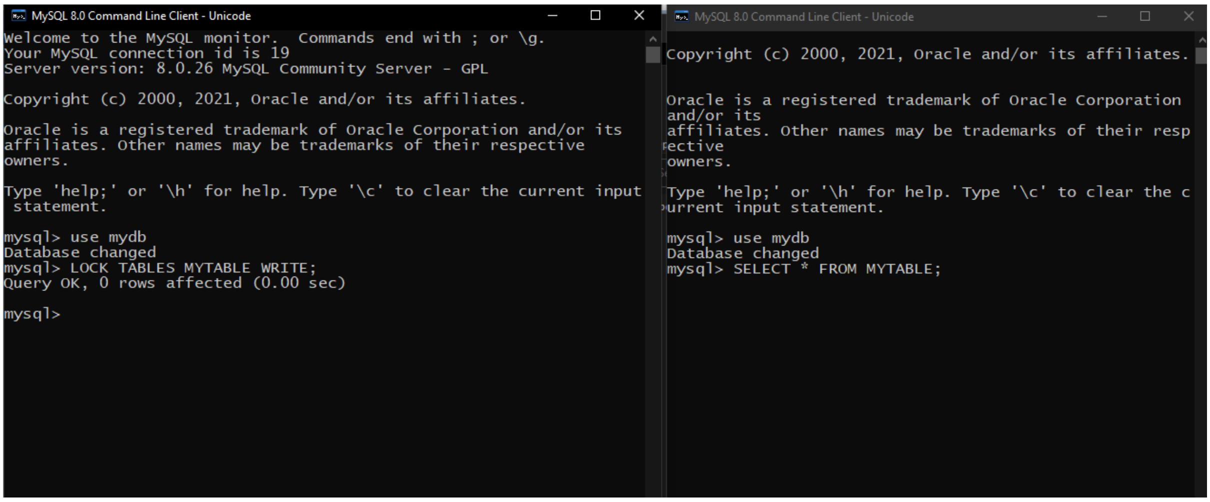
```
|  101 | BHAVESH |     28 |
|  101 | BHAVESH |     28 |
+------+---------+--------+
2 rows in set (0.00 sec)

mysql> INSERT INTO MYTABLE VALUES(102, 'AKASH', 29);
Query OK, 1 row affected (35.33 sec)

mysql>
```

User 2 waits until the lock is removed by user 1.

Then user 2 successfully performs write operation.

Similarly, if user 1 is locked in write mode, user 2 cannot perform read operations.

User 1 in write mode.

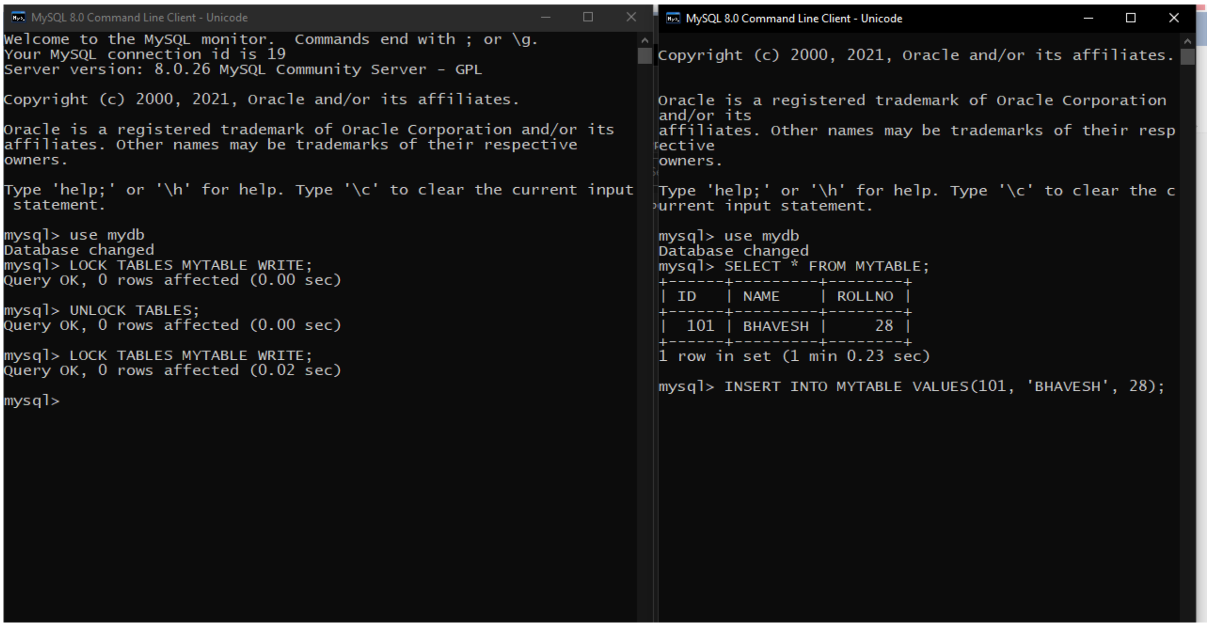

MySQL 8.0 Command Line Client - Unicode
MySQL 8.0 Command Line Client - Unicode

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
 statement.

mysql> use mydb
Database changed
mysql> LOCK TABLES MYTABLE WRITE;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

```
Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation
and/or its
affiliates. Other names may be trademarks of their resp
ective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the c
urrent input statement.

mysql> use mydb
Database changed
mysql> SELECT * FROM MYTABLE;
```

user 2 trying to write.
But it needs to wait for lock to be removed.

3.
Write – Write conflict
Both users cannot write on the same resource at the same time as it is a conflict.


MySQL 8.0 Command Line Client - Unicode
MySQL 8.0 Command Line Client - Unicode

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.26 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
 statement.

mysql> use mydb
Database changed
mysql> LOCK TABLES MYTABLE WRITE;
Query OK, 0 rows affected (0.00 sec)

mysql> UNLOCK TABLES;
Query OK, 0 rows affected (0.00 sec)

mysql> LOCK TABLES MYTABLE WRITE;
Query OK, 0 rows affected (0.02 sec)

mysql>
```

```
Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation
and/or its
affiliates. Other names may be trademarks of their resp
ective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the c
urrent input statement.

mysql> use mydb
Database changed
mysql> SELECT * FROM MYTABLE;
+------+---------+--------+
| ID   | NAME    | ROLLNO |
+------+---------+--------+
|  101 | BHAVESH |     28 |
+------+---------+--------+
1 row in set (1 min 0.23 sec)

mysql> INSERT INTO MYTABLE VALUES(101, 'BHAVESH', 28);
```