VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belgaum, Karnataka-590 014

DATABASE MANAGEMENT SYSTEM

Subject Code: **BCS403**

(As per Visvesvaraya Technological University Syllabus)

B.E- 4rd Semester, Information Science and Engineering

Prepared By

Prof. Usha Kumari V
Assistant Professor

Mrs. Priyanka Ashok Tanawade
Lab Instructor

Reviewed By
Prof. MARY M DSOUZA
Assistant Professor

Approved By:
Dr. Kala Venugopal
Head of Department Information Science and Engineering

ACHARYA INSTITUTE OF TECHNOLOGY

(Affiliated to VTU, Belgaum, Approved by AICTE, New Delhi and Govt. of Karnataka),
Acharya Dr. Sarvepalli Radhakrishnan Road, Bangalore-560107.
Ph: 91-080-28396011, 23723466, 28376431
URL: www.acharya.ac.in

2023-24

Table of contents

Vision, Mission, Motto of Institute	I
Vision, Mission of Department	II
Laboratory Objectives	III
Program Specific Outcomes (PSOs)	III
Program outcomes (POs)	IV
Course Outcomes (COs)	VI

MOTTO

"Nurturing Aspirations Supporting Growth" VISION “Acharya Institute of Technology, committed to the cause of sustainable value-based education in all disciplines, envisions itself as a global fountainhead of innovative human enterprise, with inspirational initiatives for Academic Excellence”.

VISION OF THE INSTITUTE

Acharya Institute of Technology, committed to the cause of value-based education in all disciplines, envisions itself as fountainhead of innovative human enterprise, with inspirational initiatives for Academic Excellence.

MISSION OF INSTITUTE

“Acharya Institute of Technology strives to provide excellent academic ambiance to the students for achieving global standards of technical education, foster intellectual and personal development, meaningful research and ethical service to sustainable societal needs.”

VISION OF THE DEPARTMENT

“To be center of Academic and Research excellence in the field of Information Technology inculcating value based education for the development of quality Human Resource”

MISSION OF THE DEPARTMENT

“Equip students with fundamental concepts, practical knowledge and professional ethics through dedicated faculty for higher studies and professional career in various Scientific, Engineering and Technological streams leading to proficiency in the field of Information Technology”

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Able to apply knowledge of information management and communication systems to provide secured solutions for real time engineering applications.

PSO2: Apply best software engineering practices, modern tools and technologies to deliver quality products.

PROGRAM OUTCOMES (Pos)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects

and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

COURSE OUTCOMES

Course Outcomes-Program Outcomes mapping

CO1: Develop database applications for the given real-world problem**.**

COs	**Program Outcomes**												**Program Specific Outcomes**	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO-1	1				3									
CO-2	1				3									

Rubrics for assessing student's performance in Laboratory courses

The internals marks of lab for 2022 scheme is 15 Marks for Continuous Evaluation and 10 Marks for Lab Internals.

Continuous Evaluation for 2022 scheme:

Sl No	**Parameters**	**Mark**				
1.	Writing Program/Logic (present week's/previous week's)	5	The student is able to write the program without any logical and syntactical error and proper indentation is followed.	The student is able to write the program with minor logical error	The student has written incomplete program with major logical and syntactical error	The student is not attempted to write program.
	Parameters	5	5	4	3	0
2.	Execution of program	3	Student is able to execute, debug, and test the program for all possible inputs/test cases.	Student is able to execute the program, but fails to debug, and test the program for all possible inputs/test cases.	Student is executed the program partially (fails to meet desired output)	The student has not executed the program.
	Parameters	3	3	3	0	
3.	Record	5	Student submits the record on time and, neatly documented	Student fails to submit the record on	The student submit the record with the incorrect program	The student submit the record with the incorrect program

			with all possible input/output samples.			
	Parameters	5	5	4	2	0
4.	Viva	2	Student answers for at least 80% of questions	Student answers for at least 60% of questions	Student answers for at least 40% of questions	Student fails to answer any question
5		**50 Marks**				
		Marks	10	8	5	0
	Internal Assessment (50 Marks)	***Writing Program(10 Marks)***	The student is able to write the program without any logical and syntactical error and proper indentation is followed.	The student is able to write the program with minor logical error	The student has written incomplete program with major logical and syntactical error	The student is not attempted to write program.
		Executing program with different inputs(30 Marks)	Student is able to execute, debug, and test the program for all possible inputs/test cases.	Student is able to execute the program, but fails to debug, and test the program for all possible inputs/test cases.	Student is executed the program partially(fails to meet desired output)	The student has not executed the program.
		Marks	30	20		
		Viva(10 Marks)	Student answers for at least 80% of questions	Student answers for at least 50% of questions	Student fails to answer any question	

Programming Assignments

SL. NO	Name of Program	Page No
1	Create a table called Employee & execute the following. **Employee(EMPNO,ENAME,JOB, MANAGER_NO, SAL, COMMISSION)** 1. Create a user and grant all permissions to the user. 2. Insert the any three records in the employee table contains attributes EMPNO,ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback. Check the result. 3. Add primary key constraint and not null constraint to the employee table. 4. Insert null values to the employee table and verify the result.	
2	Create a table called Employee that contain attributes EMPNO,ENAME,JOB, MGR,SAL & execute the following. 1. Add a column commission with domain to the Employee table. 2. Insert any five records into the table. 3. Update the column details of job 4. Rename the column of Employ table using alter command. 5. Delete the employee whose Empno is 105.	
3	Queries using aggregate functions(COUNT,AVG,MIN,MAX,SUM),Group by,Orderby. **Employee(E_id, E_name, Age, Salary)** 1. Create Employee table containing all Records E_id, E_name, Age, Salary. 2. Count number of employee names from employeetable 3. Find the Maximum age from employee table. 4. Find the Minimum age from employeetable. 5. Find salaries of employee in Ascending Order. 6. Find grouped salaries of employees.	
4	Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old & new Salary. **CUSTOMERS(ID,NAME,AGE,ADDRESS,SALARY)**	
5	Create cursor for Employee table & extract the values from the table. Declare the variables,Open the cursor & extrct the values from the cursor. Close the cursor. **Employee(E_id, E_name, Age, Salary)**	
6	Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped.	
7	Install an Open Source NoSQL Data base MangoDB & perform basic CRUD(Create, Read, Update & Delete) operations. Execute MangoDB basic Queries using CRUD operations.	

LABORATORY PROGRAM

EXPERIMENT NO.1

1. Create a table called Employee & execute the following.

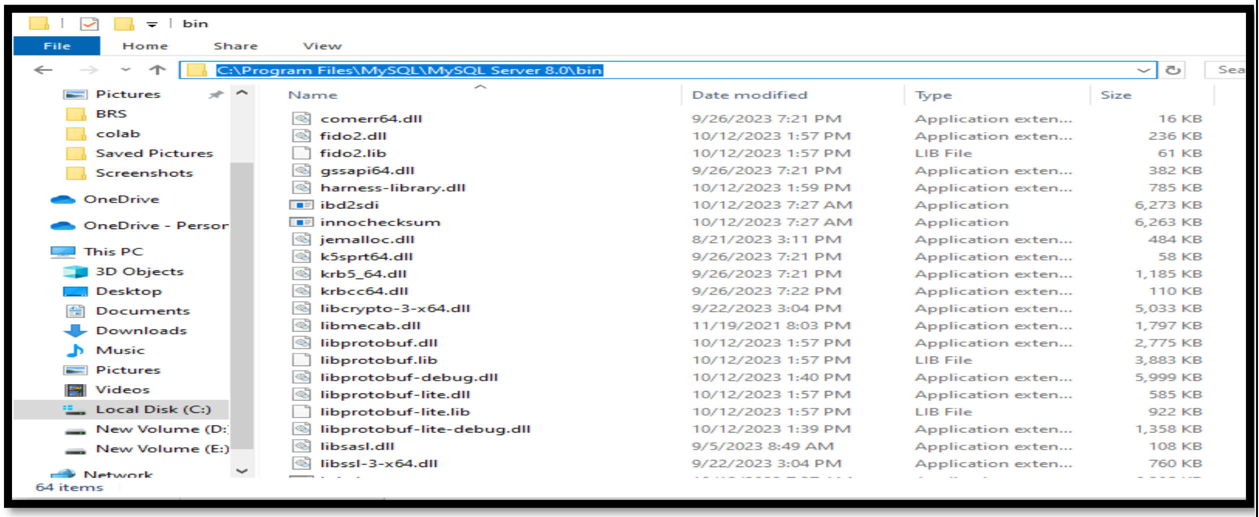
   Employee(EMPNO,ENAME,JOB, MANAGER_NO, SAL, COMMISSION)

   1. Create a user and grant all permissions to the user.
   2. Insert the any three records in the employee table contains attributes EMPNO,ENAME JOB, MANAGER_NO, SAL, COMMISSION and use rollback. Check the result.
   3. Add primary key constraint and not null constraint to the employee table. Insert null values to the employee table and verify the result.

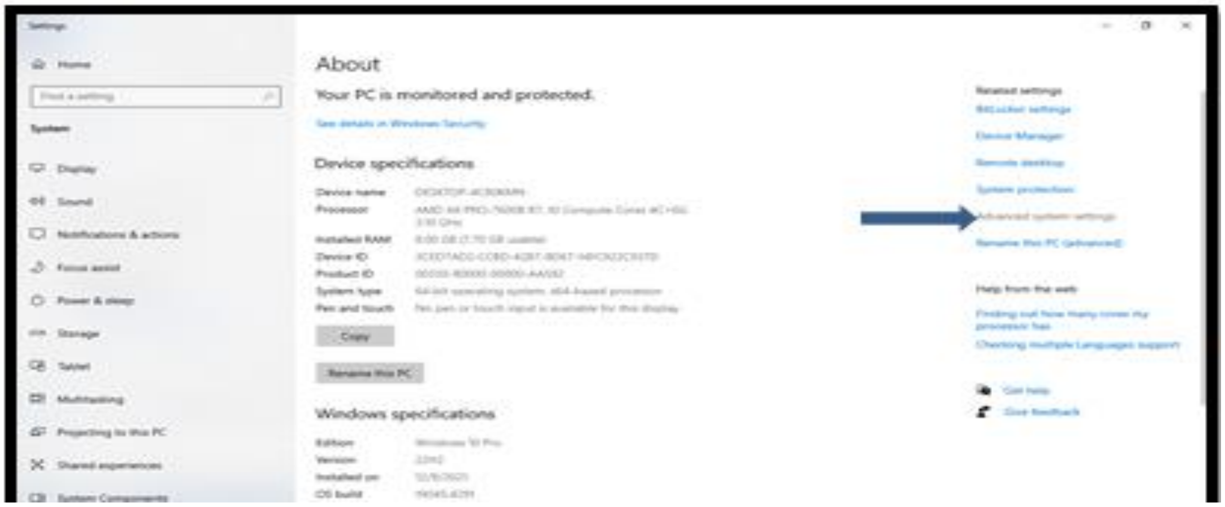
1. Create a user and grant all permissions to the user.

   Step 1: Copy the path of mysql server .

   C:\Program Files\MySQL\MySQL Server 8.0\bin



```
Name                         Date modified        Type                  Size
comerr64.dll                 9/26/2023 7:21 PM    Application exten...  16 KB
fido2.dll                    10/12/2023 1:57 PM   Application exten...  236 KB
fido2.lib                    10/12/2023 1:57 PM   LIB File              61 KB
gssapi64.dll                 9/26/2023 7:21 PM    Application exten...  382 KB
harness-library.dll          10/12/2023 1:59 PM   Application exten...  785 KB
ibd2sdi                      10/12/2023 7:27 AM   Application           6,273 KB
innochecksum                 10/12/2023 7:27 AM   Application           6,263 KB
jemalloc.dll                 8/21/2023 3:11 PM    Application exten...  484 KB
k5sprt64.dll                 9/26/2023 7:21 PM    Application exten...  58 KB
krb5_64.dll                  9/26/2023 7:21 PM    Application exten...  1,185 KB
krbcc64.dll                  9/26/2023 7:22 PM    Application exten...  110 KB
libcrypto-3-x64.dll          9/22/2023 3:04 PM    Application exten...  5,033 KB
libmecab.dll                 11/19/2021 8:03 PM   Application exten...  1,797 KB
libprotobuf.dll              10/12/2023 1:57 PM   Application exten...  2,775 KB
libprotobuf.lib              10/12/2023 1:57 PM   LIB File              3,883 KB
libprotobuf-debug.dll        10/12/2023 1:40 PM   Application exten...  5,999 KB
libprotobuf-lite.dll         10/12/2023 1:57 PM   Application exten...  585 KB
libprotobuf-lite.lib         10/12/2023 1:57 PM   LIB File              922 KB
libprotobuf-lite-debug.dll   10/12/2023 1:39 PM   Application exten...  1,358 KB
libsasl.dll                  9/5/2023 8:49 AM     Application exten...  108 KB
libssl-3-x64.dll             9/22/2023 3:04 PM    Application exten...  760 KB
```



About

Your PC is monitored and protected.

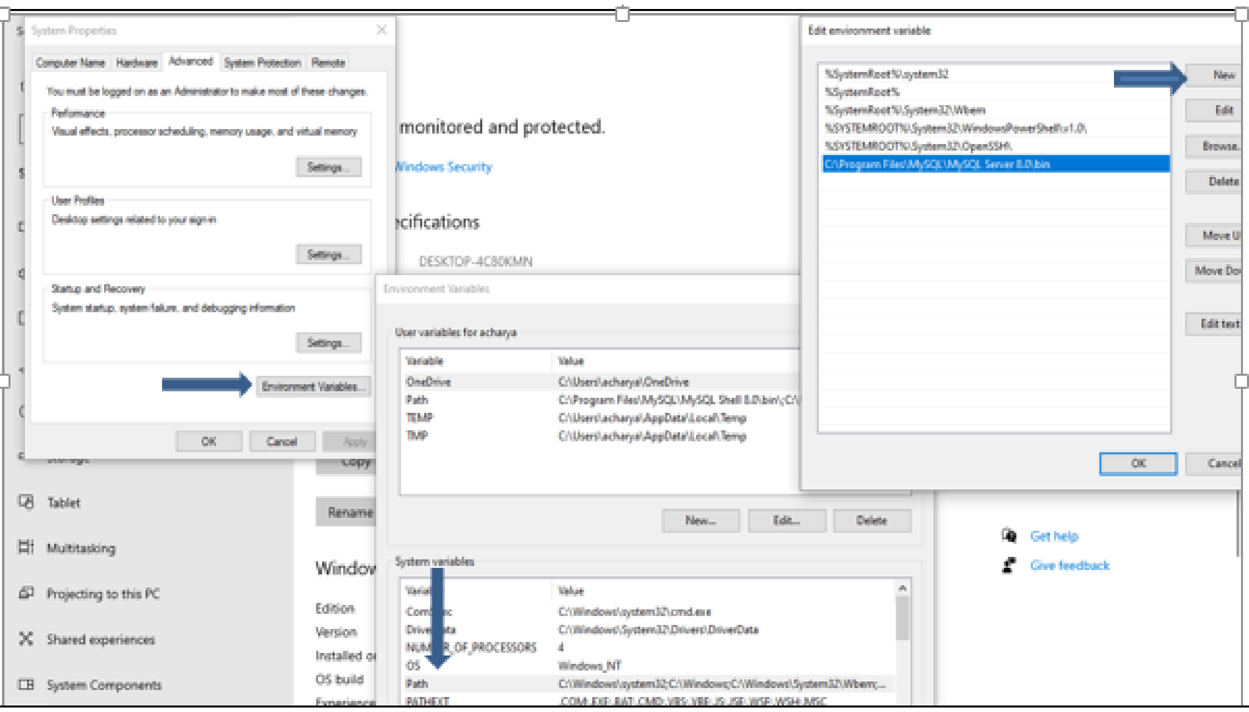
Device specifications

Device name [illegible]
Processor [illegible]
Installed RAM [illegible]
Device ID [illegible]
Product ID [illegible]
System type [illegible]
Pen and touch [illegible]

Windows specifications

Edition [illegible]
Version [illegible]
Installed on [illegible]
OS build [illegible]

Step 2:Paste the path of mysql server in ‘**Edit environment variable’**

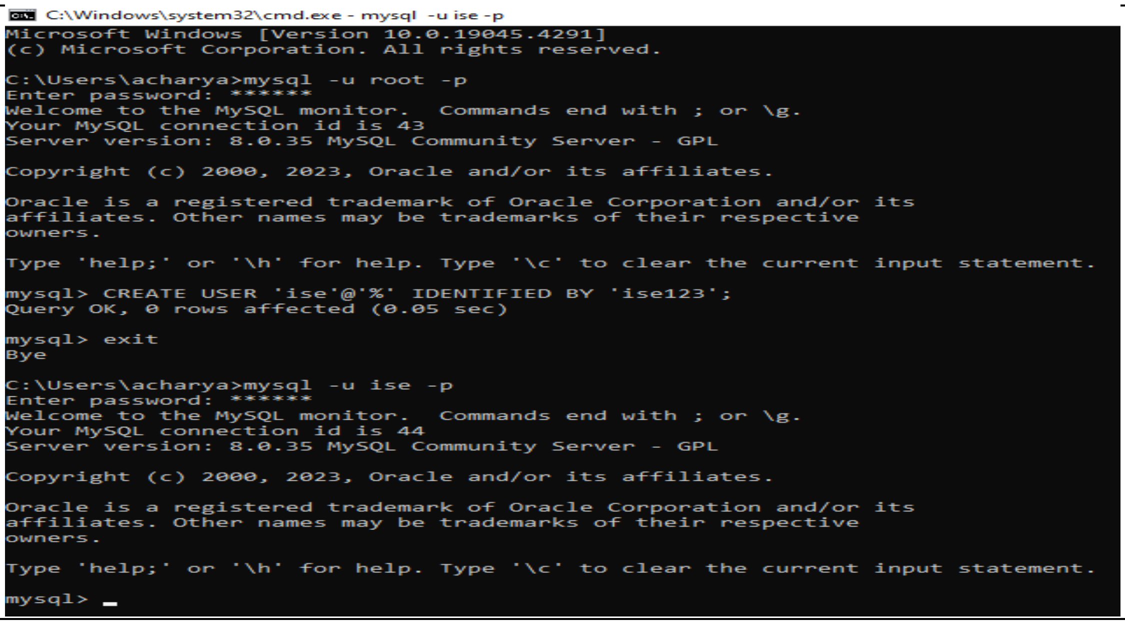


Variable	Value
OneDrive	C:\Users\acharya\OneDrive
Path	C:\Program Files\MySQL\MySQL Shell 8.0\bin\;C:\
TEMP	C:\Users\acharya\AppData\Local\Temp
TMP	C:\Users\acharya\AppData\Local\Temp

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC

```
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\MySQL\MySQL Server 8.0\bin
```

Step 3:Create user in Mysql:
mysql> **CREATE USER 'ise'@'%' IDENTIFIED BY 'ise123';**
Query OK, 0 rows affected (0.03 sec)
Step 4: Granting all permission to the user.
GRANT ALL PRIVILEGES ON *.* TO 'ise' @ '%' ;
Step 5: To display the users.
SELECT USER FROM MYSQL.USER;
Step 6: To Login to the user created from command prompt
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql -u username -p
Enter Password.

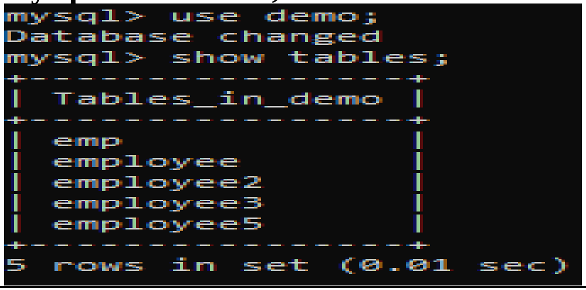

```
C:\Windows\system32\cmd.exe - mysql -u ise -p
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\acharya>mysql -u root -p
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 8.0.35 MySQL Community Server - GPL

Copyright (c) 2000, 2023, 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> CREATE USER 'ise'@'%' IDENTIFIED BY 'ise123';
Query OK, 0 rows affected (0.05 sec)

mysql> exit
Bye

C:\Users\acharya>mysql -u ise -p
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 8.0.35 MySQL Community Server - GPL

Copyright (c) 2000, 2023, 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> _
```

To change the password of user:
mysql> alter user 'ise'@'%' identified by '123';
Query OK, 0 rows affected (0.12 sec)
mysql> exit
Bye.

To Display the Database
mysql> show databases;

CREATE TABLE EMPLOYEE:
mysql> create table employee(empno int(6), ename varchar(10), job varchar(10),manager_no int(9),sal int(20), commission int(20));
Query OK, 0 rows affected, 4 warnings (2.48 sec)

mysql> show tables;

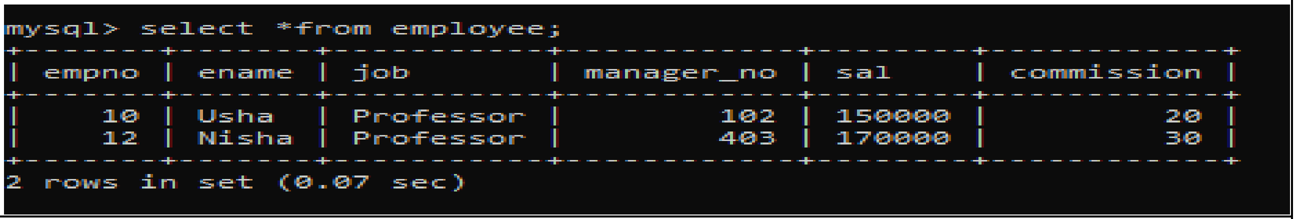


```
mysql> use demo;
Database changed
mysql> show tables;
+-----------------+
| Tables_in_demo  |
+-----------------+
| emp             |
| employee        |
| employee2       |
| employee3       |
| employee5       |
+-----------------+
5 rows in set (0.01 sec)
```

2.Insert Records:

mysql> insert into employee values(10,'Usha','Professor', 102, 150000, 20);
Query OK, 1 row affected (0.10 sec)

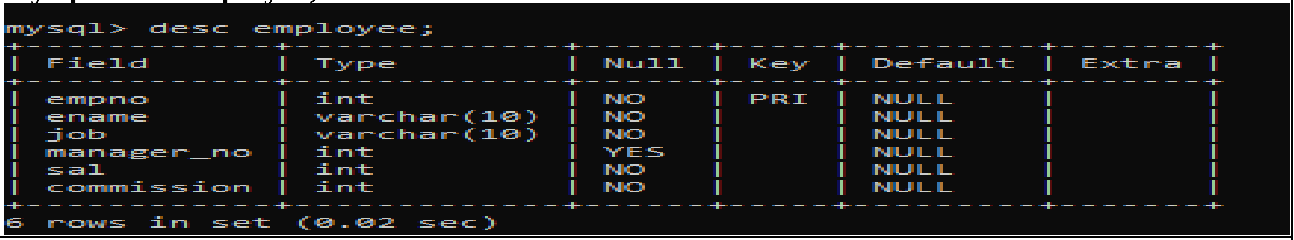
mysql> select *from employee;



```
mysql> select *from employee;
+-------+-------+-----------+------------+--------+------------+
| empno | ename | job       | manager_no | sal    | commission |
+-------+-------+-----------+------------+--------+------------+
|    10 | Usha  | Professor |        102 | 150000 |         20 |
|    12 | Nisha | Professor |        403 | 170000 |         30 |
+-------+-------+-----------+------------+--------+------------+
2 rows in set (0.07 sec)
```

3.ADD Primary Key
mysql> alter table employee add constraint pk_emp PRIMARY KEY(empno);
Query OK, 0 rows affected (2.27 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc employee;

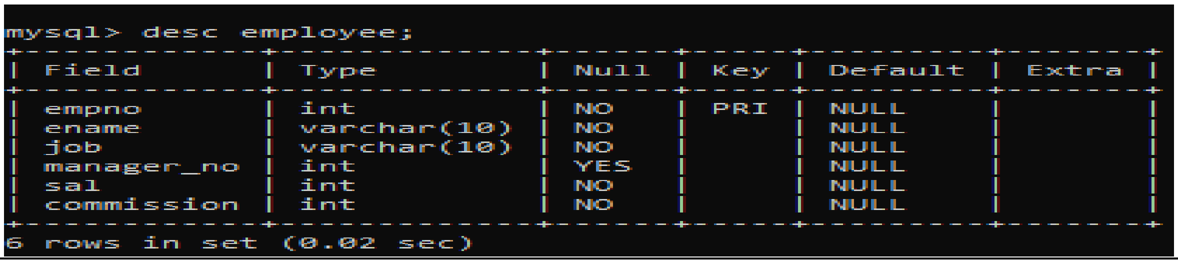


```
mysql> desc employee;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| empno      | int         | NO   | PRI | NULL    |       |
| ename      | varchar(10) | NO   |     | NULL    |       |
| job        | varchar(10) | NO   |     | NULL    |       |
| manager_no | int         | YES  |     | NULL    |       |
| sal        | int         | NO   |     | NULL    |       |
| commission | int         | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.02 sec)
```

4.set NOT NULL to columns and verify
mysql> alter table employee modify ename varchar(10) NOT NULL;

```
Query OK, 0 rows affected (1.37 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

mysql> desc employee;



```
mysql> desc employee;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| empno       | int         | NO   | PRI | NULL    |       |
| ename       | varchar(10) | NO   |     | NULL    |       |
| job         | varchar(10) | NO   |     | NULL    |       |
| manager_no  | int         | YES  |     | NULL    |       |
| sal         | int         | NO   |     | NULL    |       |
| commission  | int         | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
6 rows in set (0.02 sec)
```

```
mysql> alter table employee  modify commission int NOT NULL;
Query OK, 0 rows affected (1.60 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> alter table employee  modify sal int(20) NOT NULL;
Query OK, 0 rows affected, 1 warning (1.19 sec)
Records: 0  Duplicates: 0  Warnings: 1

mysql> alter table employee  modify job varchar(10) NOT NULL;
Query OK, 0 rows affected (0.83 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> desc employee;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| empno      | int         | NO   | PRI | NULL    |       |
| ename      | varchar(10) | NO   |     | NULL    |       |
| job        | varchar(10) | NO   |     | NULL    |       |
| manager_no | int         | YES  |     | NULL    |       |
| sal        | int         | NO   |     | NULL    |       |
| commission | int         | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.07 sec)
```

```
mysql> insert into employee VALUES(12,'Nisha','Professor',403,170000,30);
Query OK, 1 row affected (0.06 sec)

mysql> insert into employee VALUES(13,'isha','Professor',403,null,null);
ERROR 1048 (23000): Column 'sal' cannot be null
```

EXPERIMENT NO.2

Create a table called Employee that contain attributes EMPNO,ENAME,JOB, MGR,SAL & execute the following.

1. Add a column commission with domain to the Employee table.
2. Insert any five records into the table.
3. Update the column details of job
4. Rename the column of Employ table using alter command.
5. Delete the employee whose Empno is 105.

```
mysql> use demo;
Database changed
```

Create table Employee

```
mysql> create table Employee2(EMPNO INTEGER NOT NULL, ENAME
VARCHAR(20) NOT NULL,JOB VARCHAR(15),MGR CHAR(10),SAL INTEGER NOT
NULL,PRIMARY KEY(EMPNO) );
Query OK, 0 rows affected (1.84 sec)
```

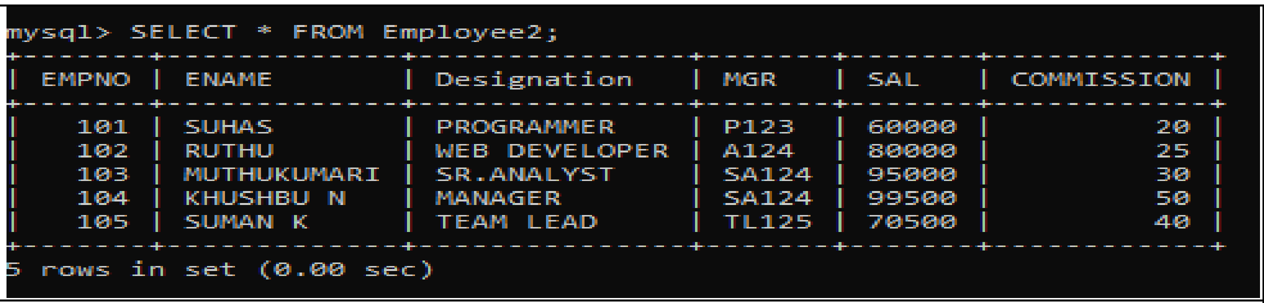
1. ADD COMMISSION column

```
mysql> ALTER TABLE Employee2 ADD COMMISSION INTEGER;
Query OK, 0 rows affected (0.61 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> DESC Employee2;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| EMPNO      | int         | NO   | PRI | NULL    |       |
| ENAME      | varchar(20) | NO   |     | NULL    |       |
| JOB        | varchar(15) | YES  |     | NULL    |       |
| MGR        | char(10)    | YES  |     | NULL    |       |
| SAL        | int         | NO   |     | NULL    |       |
| COMMISSION | int         | YES  |     | NULL    |       |
6 rows in set (0.34 sec)
```

2.Insert five Records:

```
mysql> INSERT INTO Employee2
VALUES(101,'SUHAS','PROGRAMMER','P123',60000,20);
Query OK, 1 row affected (0.15 sec)
mysql> INSERT INTO Employee2 VALUES(102,'RUTHU','ANALYST','A124',80000,25);
Query OK, 1 row affected (0.13 sec)
mysql> INSERT INTO Employee2 VALUES
(103,'MUTHUKUMARI','SR.ANALYST','SA124',95000,30);
mysql> INSERT INTO Employee2 VALUES
                              (104,'KHUSHBU N','MANAGER','SA124',99500,50);
mysql> INSERT INTO Employee2 VALUES
                              (105,'SUMAN K,'TEAM LEAD','TL125',70500,40);
mysql> SELECT * FROM Employee2;
```



```
mysql> SELECT * FROM Employee2;
+-------+-------------+---------------+-------+-------+------------+
| EMPNO | ENAME       | Designation   | MGR   | SAL   | COMMISSION |
+-------+-------------+---------------+-------+-------+------------+
|   101 | SUHAS       | PROGRAMMER    | P123  | 60000 |         20 |
|   102 | RUTHU       | WEB DEVELOPER | A124  | 80000 |         25 |
|   103 | MUTHUKUMARI | SR.ANALYST    | SA124 | 95000 |         30 |
|   104 | KHUSHBU N   | MANAGER       | SA124 | 99500 |         50 |
|   105 | SUMAN K     | TEAM LEAD     | TL125 | 70500 |         40 |
+-------+-------------+---------------+-------+-------+------------+
5 rows in set (0.00 sec)
```

3. Update

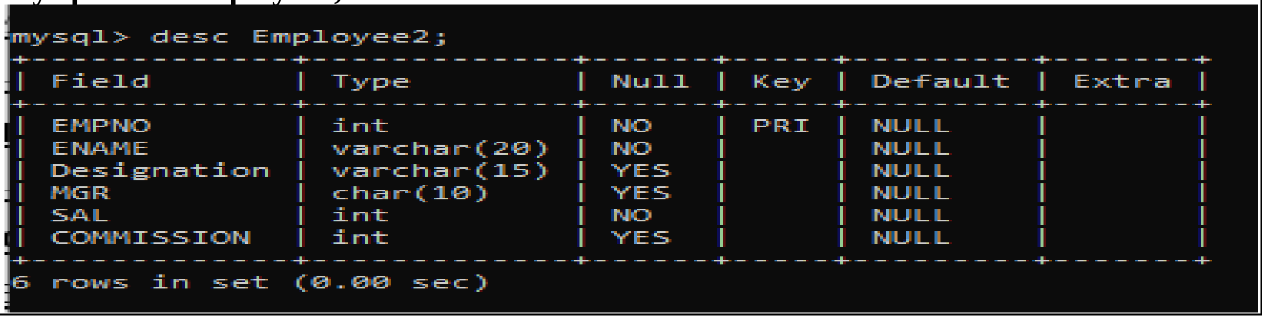
```
mysql> UPDATE Employee2 set JOB='WEB DEVELOPER' WHERE EMPNO=102 AND
COMMISSION=25;
Query OK, 1 row affected (0.13 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> SELECT * FROM Employee2;
+-------+-------------+---------------+-------+-------+------------+
| EMPNO | ENAME       | JOB           | MGR   | SAL   | COMMISSION |
+-------+-------------+---------------+-------+-------+------------+
|   101 | SUHAS       | PROGRAMMER    | P123  | 60000 |         20 |
|   102 | RUTHU       | WEB DEVELOPER | A124  | 80000 |         25 |
|   103 | MUTHUKUMARI | SR.ANALYST    | SA124 | 95000 |         30 |
|   104 | KHUSHBU N   | MANAGER       | SA124 | 99500 |         50 |
|   105 | SUMAN K     | TEAM LEAD     | TL125 | 70500 |         40 |
+-------+-------------+---------------+-------+-------+------------+
5 rows in set (0.00 sec)
```

4. Rename Column Name

```
mysql> ALTER TABLE Employee2 rename column JOB to Designation;
Query OK, 0 rows affected (0.63 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> desc Employee2;
```

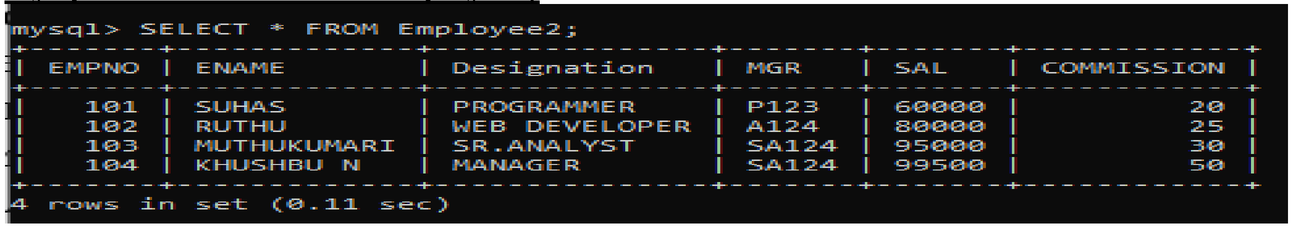


```
mysql> desc Employee2;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| EMPNO       | int         | NO   | PRI | NULL    |       |
| ENAME       | varchar(20) | NO   |     | NULL    |       |
| Designation | varchar(15) | YES  |     | NULL    |       |
| MGR         | char(10)    | YES  |     | NULL    |       |
| SAL         | int         | NO   |     | NULL    |       |
| COMMISSION  | int         | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

5. Delete Employee whose EMPNO=105

```
mysql> DELETE FROM Employee2 where EMPNO=105;
Query OK, 1 row affected (0.13 sec)
```

mysql> SELECT * FROM Employee2;



```
mysql> SELECT * FROM Employee2;
+-------+----------------+----------------+-------+-------+------------+
| EMPNO | ENAME          | Designation    | MGR   | SAL   | COMMISSION |
+-------+----------------+----------------+-------+-------+------------+
|   101 | SUHAS          | PROGRAMMER     | P123  | 60000 |         20 |
|   102 | RUTHU          | WEB DEVELOPER  | A124  | 80000 |         25 |
|   103 | MUTHUKUMARI    | SR.ANALYST     | SA124 | 95000 |         30 |
|   104 | KHUSHBU N      | MANAGER        | SA124 | 99500 |         50 |
+-------+----------------+----------------+-------+-------+------------+
4 rows in set (0.11 sec)
```

EXPERIMENT NO.3

Queries using aggregate functions(COUNT,AVG,MIN,MAX,SUM),Group by,Orderby.
Employee(E_id, E_name, Age, Salary)

1. Create Employee table containing all Records E_id, E_name, Age, Salary.
2. Count number of employee names from employeetable
3. Find the Maximum age from employee table.
4. Find the Minimum age from employeetable.
5. Find salaries of employee in Ascending Order.
6. Find grouped salaries of employees.

1.Create Employee Containing all Records:

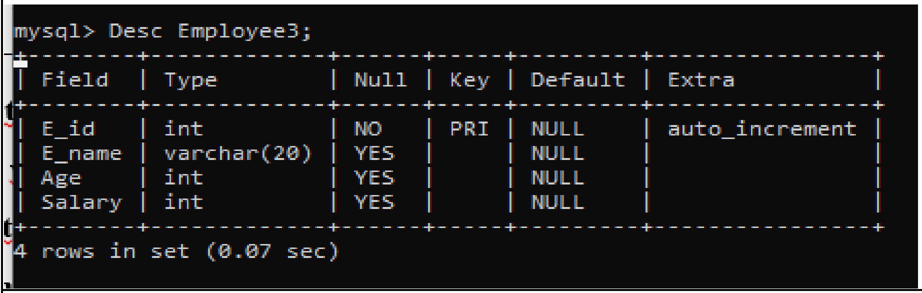
a. mysql> use demo;
Database changed

b. mysql> create table Employee3(E_id int auto_increment,E_name varchar(20),Age integer,Salary int,primary key(E_id));

Query OK, 0 rows affected (0.94 sec)

insert into Employee3 values(101,'Sharada',24,50000);
insert into Employee3 values(102,'Ronald',22,55000);
insert into Employee3 values(103,'Mihir',26,59000);
insert into Employee3 values(104,'Aayushi',23,57000);
insert into Employee3 values(105,'Ribika',24,60000);

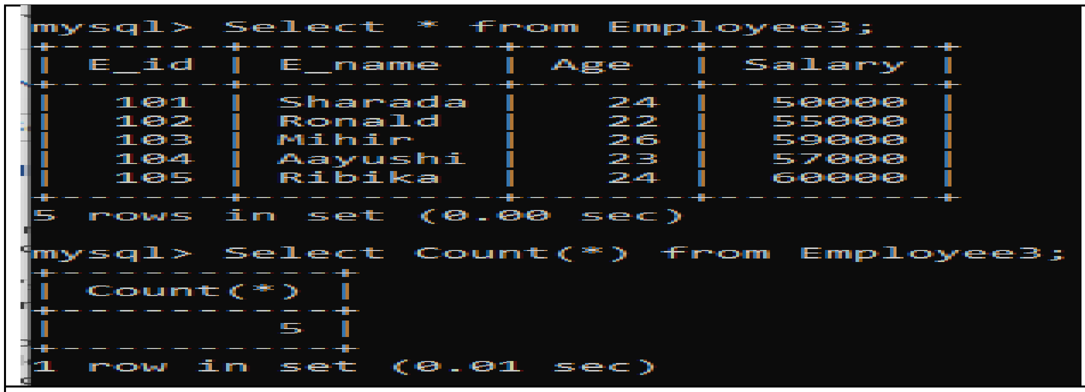
c. mysql> Desc Employee3;



```
mysql> Desc Employee3;
+--------+-------------+------+-----+---------+----------------+
| Field  | Type        | Null | Key | Default | Extra          |
+--------+-------------+------+-----+---------+----------------+
| E_id   | int         | NO   | PRI | NULL    | auto_increment |
| E_name | varchar(20) | YES  |     | NULL    |                |
| Age    | int         | YES  |     | NULL    |                |
| Salary | int         | YES  |     | NULL    |                |
+--------+-------------+------+-----+---------+----------------+
4 rows in set (0.07 sec)
```

2.Count number of Employee:

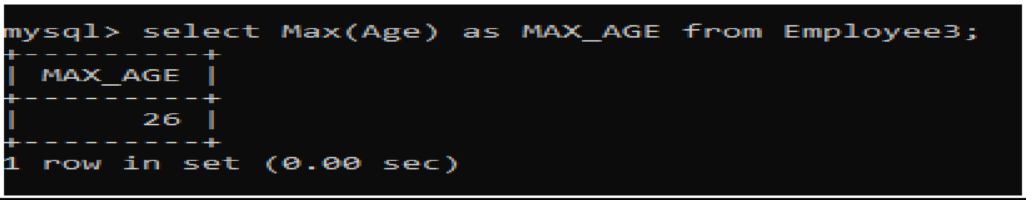
mysql> Select Count(*) from Employee3;



```
mysql> Select * from Employee3;
+-------+----------+-------+----------+
| E_id  | E_name   | Age   | Salary   |
+-------+----------+-------+----------+
|   101 | Sharada  |    24 |   50000  |
|   102 | Ronald   |    22 |   55000  |
|   103 | Mihir    |    26 |   59000  |
|   104 | Aayushi  |    23 |   57000  |
|   105 | Ribika   |    24 |   60000  |
+-------+----------+-------+----------+
5 rows in set (0.00 sec)

mysql> Select Count(*) from Employee3;
+----------+
| Count(*) |
+----------+
|        5 |
+----------+
1 row in set (0.01 sec)
```

3.Find Maximum Age from Employee Table:

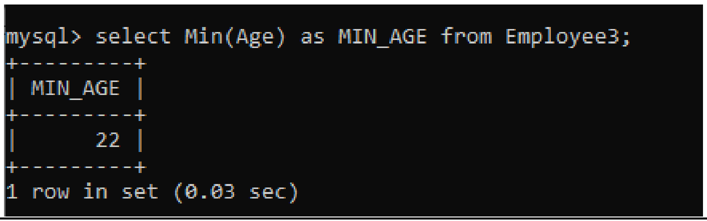
mysql> select Max(Age) as MAX_AGE from Employee3;



```
mysql> select Max(Age) as MAX_AGE from Employee3;
+---------+
| MAX_AGE |
+---------+
|      26 |
+---------+
1 row in set (0.00 sec)
```

4.Find Minimum Age from Employee Table:

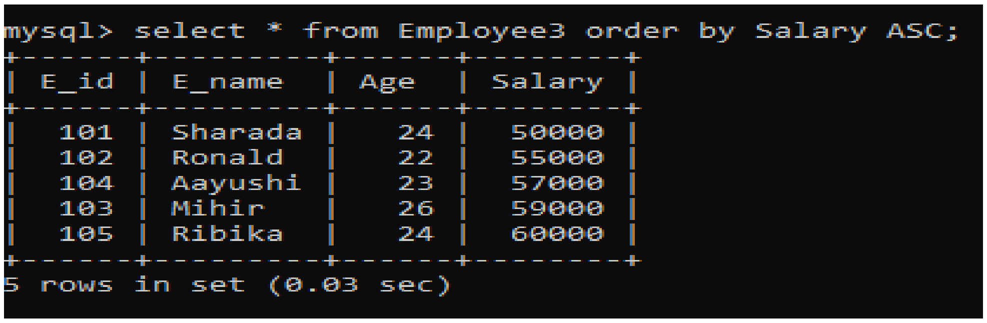
mysql> select Min(Age) as MIN_AGE from Employee3;



```
mysql> select Min(Age) as MIN_AGE from Employee3;
+---------+
| MIN_AGE |
+---------+
|      22 |
+---------+
1 row in set (0.03 sec)
```

5.Find Salary of employee in Ascending order :

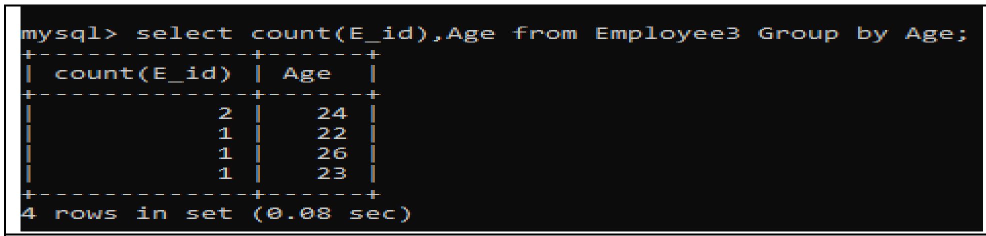
mysql> select * from Employee3 order by Salary ASC;



```
mysql> select * from Employee3 order by Salary ASC;
+------+---------+------+--------+
| E_id | E_name  | Age  | Salary |
+------+---------+------+--------+
|  101 | Sharada |   24 |  50000 |
|  102 | Ronald  |   22 |  55000 |
|  104 | Aayushi |   23 |  57000 |
|  103 | Mihir   |   26 |  59000 |
|  105 | Ribika  |   24 |  60000 |
+------+---------+------+--------+
5 rows in set (0.03 sec)
```

6. Find grouped salaries of employees:

mysql> Select count(E_id),Age from Employee3 Group by Age;



```
mysql> select count(E_id),Age from Employee3 Group by Age;
+-------------+------+
| count(E_id) | Age  |
+-------------+------+
|           2 |   24 |
|           1 |   22 |
|           1 |   26 |
|           1 |   23 |
+-------------+------+
4 rows in set (0.08 sec)
```

EXPERIMENT NO.4

Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old & new Salary.

CUSTOMERS(ID,NAME,AGE,ADDRESS,SALARY)

1.UPDATE Trigger :

```
delimiter $$
CREATE TRIGGER display_salary
AFTER UPDATE ON customer
FOR EACH ROW
BEGIN
  DECLARE saldiff INT;
IF(NEW.ID > 0)THEN
  set @saldiff = NEW.salary-OLD.salary;
  insert into sal_diff values(new.id,new.name,@saldiff);
  update salary_budget set total=total+@saldiff;
END IF;
END$$
delimiter ;
```

2.INSERT Trigger:

```
DELIMITER $$
CREATE TRIGGER newcustomer
BEFORE INSERT ON customer
FOR EACH ROW
BEGIN
   if(new.id>0)then
   insert into sal_diff values(new.id,new.name,0);
   UPDATE salary_budget SET total = total+new.salary;
end if;
END$$
DELIMITER ;
```

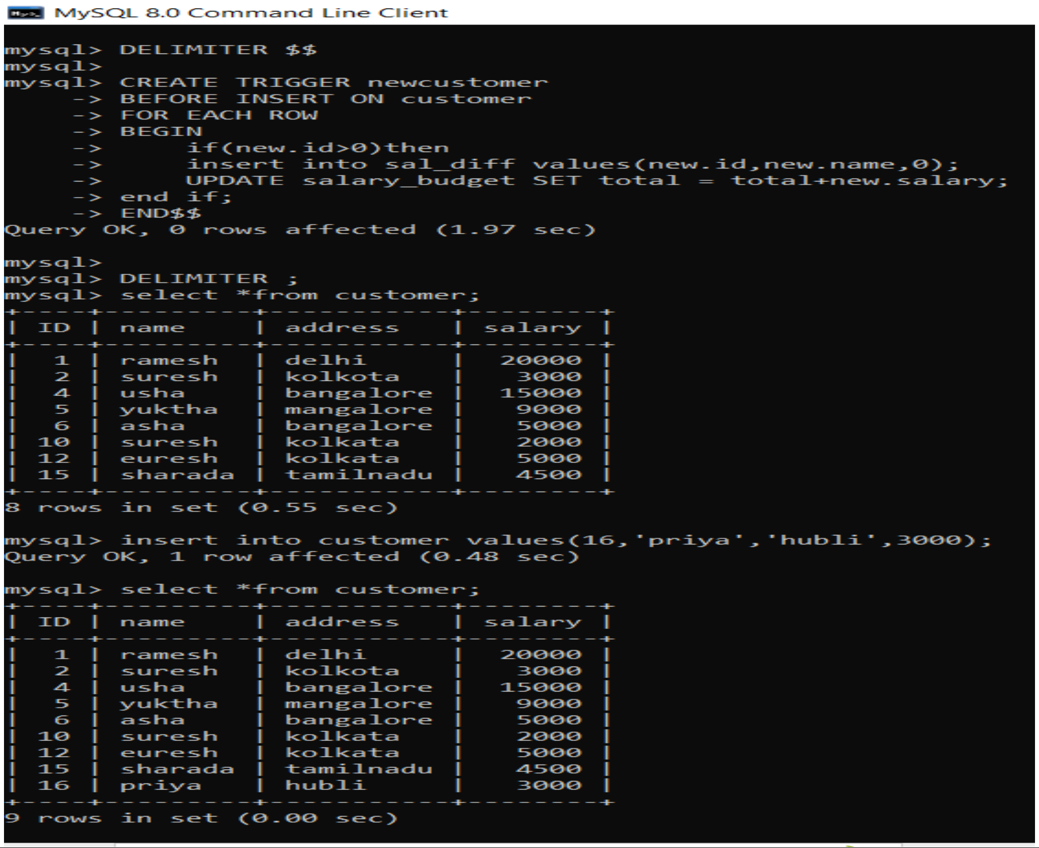
3.DELETE Trigger:

```
Delimiter  $$
CREATE TRIGGER after_sal_delete
AFTER DELETE
ON customer
FOR EACH ROW
UPDATE salary_budget
SET total = total - old.salary;
end$$
Delimiter ;
```

Trigger Execution

To see running Triggers:
mysql> show triggers;

Insert Trigger



```
mysql> DELIMITER $$
mysql>
mysql> CREATE TRIGGER newcustomer
    -> BEFORE INSERT ON customer
    -> FOR EACH ROW
    -> BEGIN
    ->     if(new.id>0)then
    ->     insert into sal_diff values(new.id,new.name,0);
    ->     UPDATE salary_budget SET total = total+new.salary;
    -> end if;
    -> END$$
Query OK, 0 rows affected (1.97 sec)

mysql>
mysql> DELIMITER ;
mysql> select *from customer;
+----+---------+-----------+--------+
| ID | name    | address   | salary |
+----+---------+-----------+--------+
|  1 | ramesh  | delhi     |  20000 |
|  2 | suresh  | kolkota   |   3000 |
|  4 | usha    | bangalore |  15000 |
|  5 | yuktha  | mangalore |   9000 |
|  6 | asha    | bangalore |   5000 |
| 10 | suresh  | kolkata   |   2000 |
| 12 | euresh  | kolkata   |   5000 |
| 15 | sharada | tamilnadu |   4500 |
+----+---------+-----------+--------+
8 rows in set (0.55 sec)

mysql> insert into customer values(16,'priya','hubli',3000);
Query OK, 1 row affected (0.48 sec)

mysql> select *from customer;
+----+---------+-----------+--------+
| ID | name    | address   | salary |
+----+---------+-----------+--------+
|  1 | ramesh  | delhi     |  20000 |
|  2 | suresh  | kolkota   |   3000 |
|  4 | usha    | bangalore |  15000 |
|  5 | yuktha  | mangalore |   9000 |
|  6 | asha    | bangalore |   5000 |
| 10 | suresh  | kolkata   |   2000 |
| 12 | euresh  | kolkata   |   5000 |
| 15 | sharada | tamilnadu |   4500 |
| 16 | priya   | hubli     |   3000 |
+----+---------+-----------+--------+
9 rows in set (0.00 sec)
```

DROP the Trigger:

mysql> drop trigger display_salary;

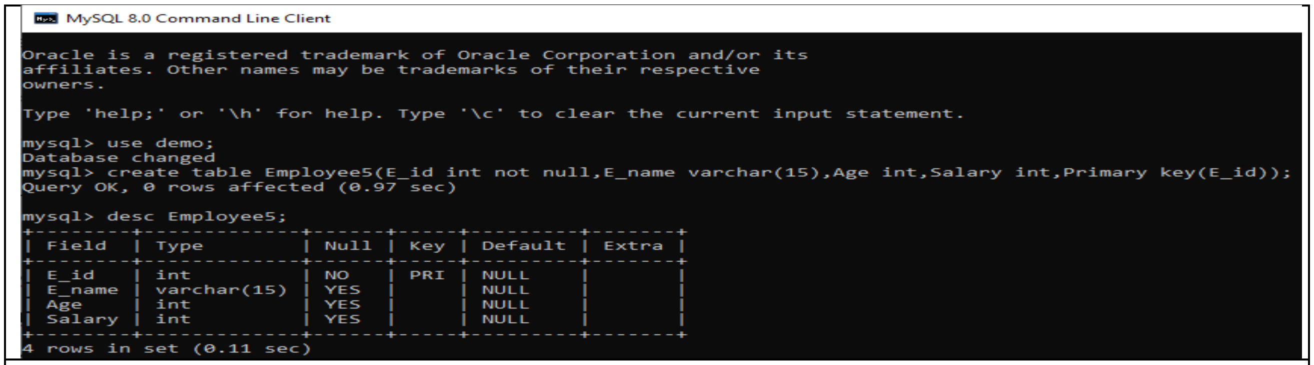
Query OK, 0 rows affected (0.11 sec)

EXPERIMENT NO.5

Create cursor for Employee table & extract the values from the table. Declare the variables,Open the cursor & extrct the values from the cursor. Close the cursor.
Employee(E_id, E_name, Age, Salary)

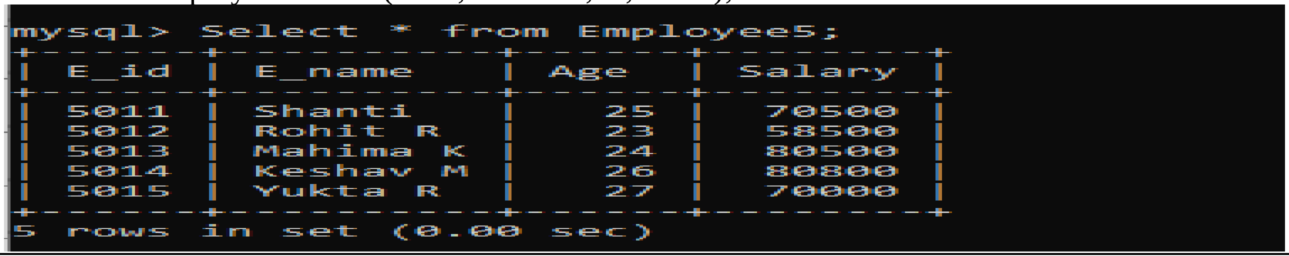
mysql> use demo;
Database changed

mysql> create table Employee5(E_id int not null,E_name varchar(15),Age int,Salary int,Primary key(E_id));



```
MySQL 8.0 Command Line Client

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 demo;
Database changed
mysql> create table Employee5(E_id int not null,E_name varchar(15),Age int,Salary int,Primary key(E_id));
Query OK, 0 rows affected (0.97 sec)

mysql> desc Employee5;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| E_id   | int         | NO   | PRI | NULL    |       |
| E_name | varchar(15) | YES  |     | NULL    |       |
| Age    | int         | YES  |     | NULL    |       |
| Salary | int         | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
4 rows in set (0.11 sec)
```

insert into Employee5 values(5011,'Shanti',25,70500);
insert into Employee5 values(5012,'Rohit R',23,58500);
insert into Employee5 values(5013,'Mahima K',24,80500);
insert into Employee5 values(5014,'Keshav M',26,80800);
insert into Employee5 values(5015,'Yukta R',27,70000);



```
mysql> Select * from Employee5;
+------+-----------+------+--------+
| E_id | E_name    | Age  | Salary |
+------+-----------+------+--------+
| 5011 | Shanti    |   25 |  70500 |
| 5012 | Rohit R   |   23 |  58500 |
| 5013 | Mahima K  |   24 |  80500 |
| 5014 | Keshav M  |   26 |  80800 |
| 5015 | Yukta R   |   27 |  70000 |
+------+-----------+------+--------+
5 rows in set (0.00 sec)
```

An explicit cursor has four basic operations:

- Open: The Cursor is opened, and the result set is populated.
- Fetch: The Cursor retrieves a single row from the result set and makes it available for processing.
- Close: The Cursor is closed, and the result set is no longer available for processing.
- Deallocate: The Cursor is deallocated, and the memory it uses is freed up.

1.Declare a cursor and retrieve all rows from a table:

DECLARE cur CURSOR FOR SELECT * FROM Employee5;

2. Open a cursor and fetch the first row:

OPEN cur;

FETCH NEXT FROM cur;

3. Loop through all rows in the Cursor:

```
WHILE @@FETCH_STATUS = 0
BEGIN
   FETCH NEXT FROM cur;
END
```

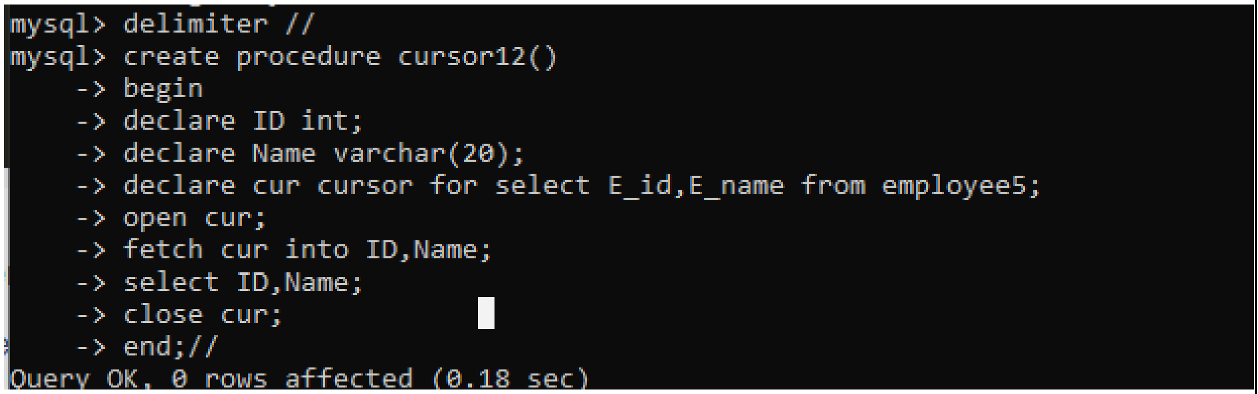
4. Close the Cursor and deallocate memory:

CLOSE cur;

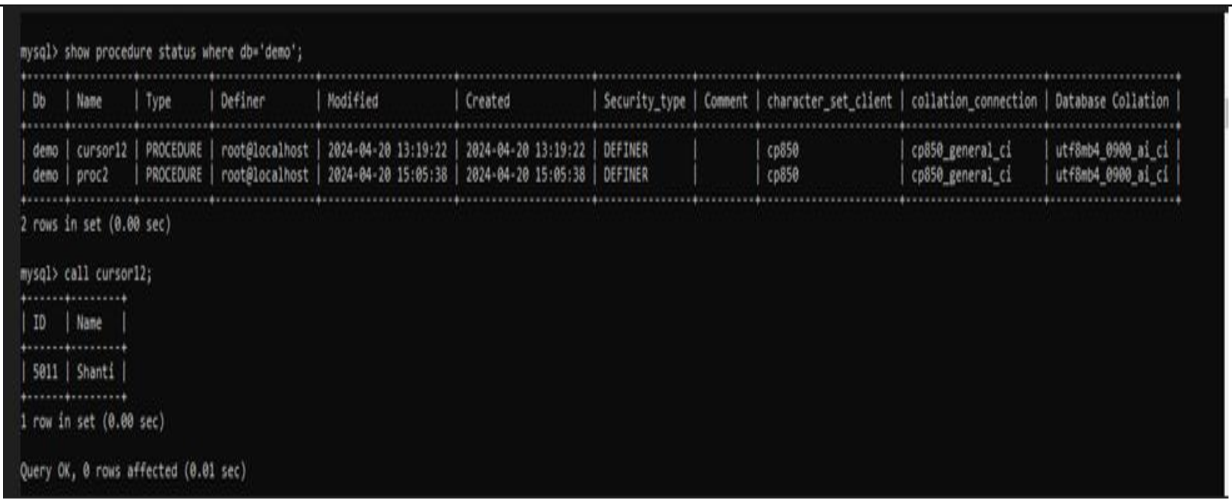
DEALLOCATE cur;

Procedure 1:

```
mysql> delimiter //
mysql> create procedure cursor12()
    -> begin
    -> declare ID int;
    -> declare Name varchar(20);
    -> declare cur cursor for select E_id,E_name from employee5;
    -> open cur;
-> fetch cur into ID,Name;
    -> select ID,Name;
    -> close cur;
    -> end;//
Query OK, 0 rows affected (0.18 sec)
```



```
mysql> delimiter //
mysql> create procedure cursor12()
    -> begin
    -> declare ID int;
    -> declare Name varchar(20);
    -> declare cur cursor for select E_id,E_name from employee5;
    -> open cur;
    -> fetch cur into ID,Name;
    -> select ID,Name;
    -> close cur;
    -> end;//
Query OK, 0 rows affected (0.18 sec)
```

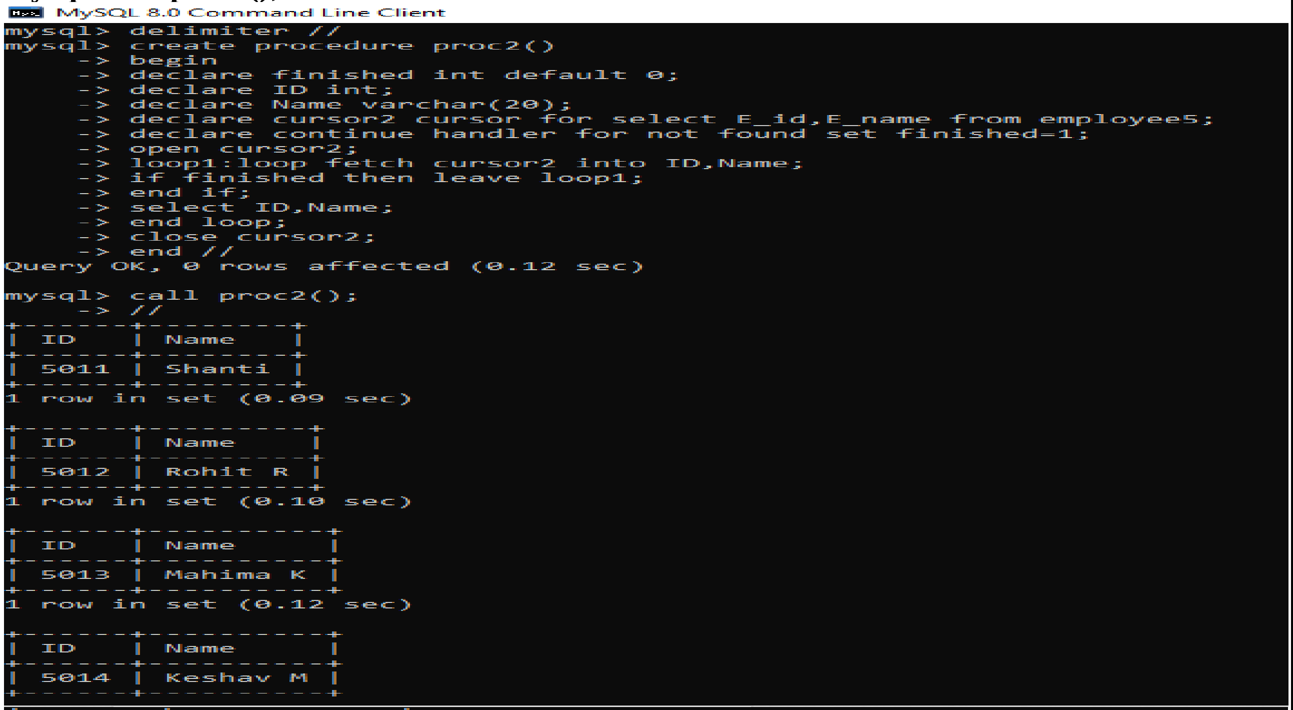


```
mysql> show procedure status where db='demo';
+------+----------+-----------+----------------+---------------------+---------------------+---------------+---------+----------------------+----------------------+--------------------+
| Db   | Name     | Type      | Definer        | Modified            | Created             | Security_type | Comment | character_set_client | collation_connection | Database Collation |
+------+----------+-----------+----------------+---------------------+---------------------+---------------+---------+----------------------+----------------------+--------------------+
| demo | cursor12 | PROCEDURE | root@localhost | 2024-04-20 13:19:22 | 2024-04-20 13:19:22 | DEFINER       |         | cp850                | cp850_general_ci     | utf8mb4_0900_ai_ci |
| demo | proc2    | PROCEDURE | root@localhost | 2024-04-20 15:05:38 | 2024-04-20 15:05:38 | DEFINER       |         | cp850                | cp850_general_ci     | utf8mb4_0900_ai_ci |
+------+----------+-----------+----------------+---------------------+---------------------+---------------+---------+----------------------+----------------------+--------------------+
2 rows in set (0.00 sec)

mysql> call cursor12;
+------+--------+
| ID   | Name   |
+------+--------+
| 5011 | Shanti |
+------+--------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

Procedure 2:

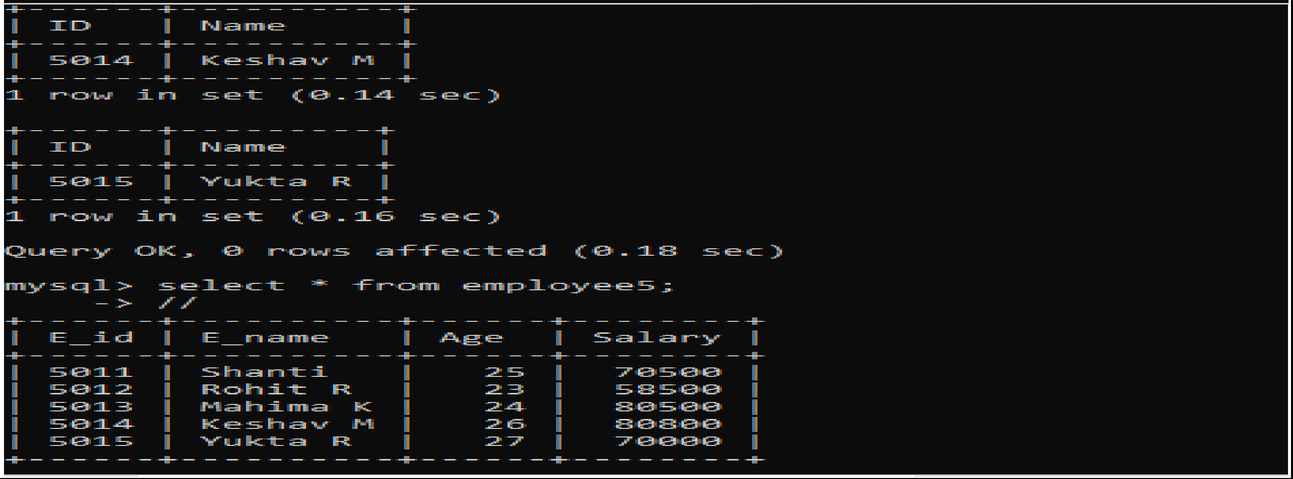
```
mysql> delimiter //
mysql> create procedure proc2()
    -> begin
    -> declare finished int default 0;
    -> declare ID int;
    -> declare Name varchar(20);
    -> declare cursor2 cursor for select E_id,E_name from employee5;
    -> declare continue handler for not found set finished=1;
    -> open cursor2;
    -> loop1:loop fetch cursor2 into ID,Name;
    -> if finished then leave loop1;
    -> end if;
    -> select ID,Name;
    -> end loop;
    -> close cursor2;
    -> end //
Query OK, 0 rows affected (0.12 sec)
```

mysql> call proc2();//



MySQL 8.0 Command Line Client

```
mysql> delimiter //
mysql> create procedure proc2()
    -> begin
    -> declare finished int default 0;
    -> declare ID int;
    -> declare Name varchar(20);
    -> declare cursor2 cursor for select E_id,E_name from employee5;
    -> declare continue handler for not found set finished=1;
    -> open cursor2;
    -> loop1:loop fetch cursor2 into ID,Name;
    -> if finished then leave loop1;
    -> end if;
    -> select ID,Name;
    -> end loop;
    -> close cursor2;
    -> end //
Query OK, 0 rows affected (0.12 sec)

mysql> call proc2();
    -> //
+------+---------+
| ID   | Name    |
+------+---------+
| 5011 | Shanti  |
+------+---------+
1 row in set (0.09 sec)

+------+----------+
| ID   | Name     |
+------+----------+
| 5012 | Rohit R  |
+------+----------+
1 row in set (0.10 sec)

+------+-----------+
| ID   | Name      |
+------+-----------+
| 5013 | Mahima K  |
+------+-----------+
1 row in set (0.12 sec)

+------+-----------+
| ID   | Name      |
+------+-----------+
| 5014 | Keshav M  |
+------+-----------+
```



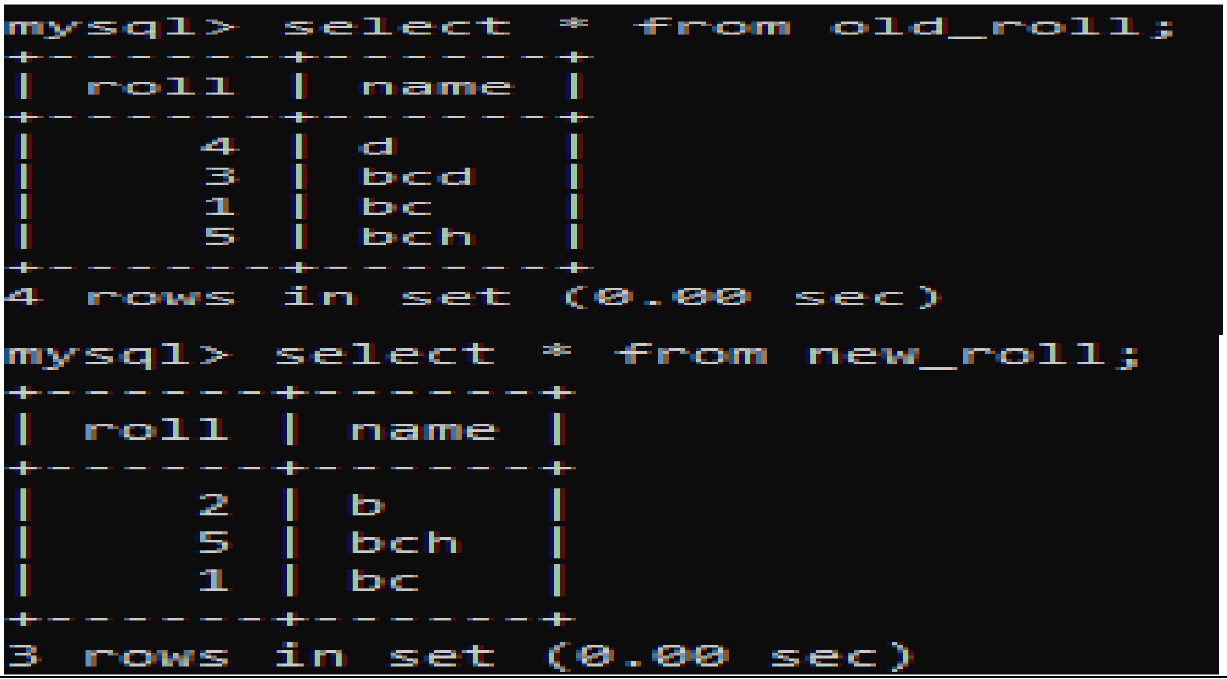
```
+------+-----------+
| ID   | Name      |
+------+-----------+
| 5014 | Keshav M  |
+------+-----------+
1 row in set (0.14 sec)

+------+----------+
| ID   | Name     |
+------+----------+
| 5015 | Yukta R  |
+------+----------+
1 row in set (0.16 sec)

Query OK, 0 rows affected (0.18 sec)

mysql> select * from employee5;
    -> //
+------+-----------+------+--------+
| E_id | E_name    | Age  | Salary |
+------+-----------+------+--------+
| 5011 | Shanti    |   25 |  70500 |
| 5012 | Rohit R   |   23 |  58500 |
| 5013 | Mahima K  |   24 |  80500 |
| 5014 | Keshav M  |   26 |  80800 |
| 5015 | Yukta R   |   27 |  70000 |
+------+-----------+------+--------+
```

<u>Experiment No 6</u>

Write a PL/SQL block of code using parameterized Cursor, that will merge the data available in the newly created table N_RollCall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped.

```
create table old_roll(roll int,name varchar(10));
create table new_roll(roll int,name varchar(10));
```

```
insert into old_roll values(4,'d');
insert into old_roll values(3,'bcd');
insert into old_roll values(1,'bc');
insert into old_roll values(5,'bch');
insert into new_roll values(2,'b');
insert into new_roll values(5,'bch');
insert into new_roll values(1,'bc');
```



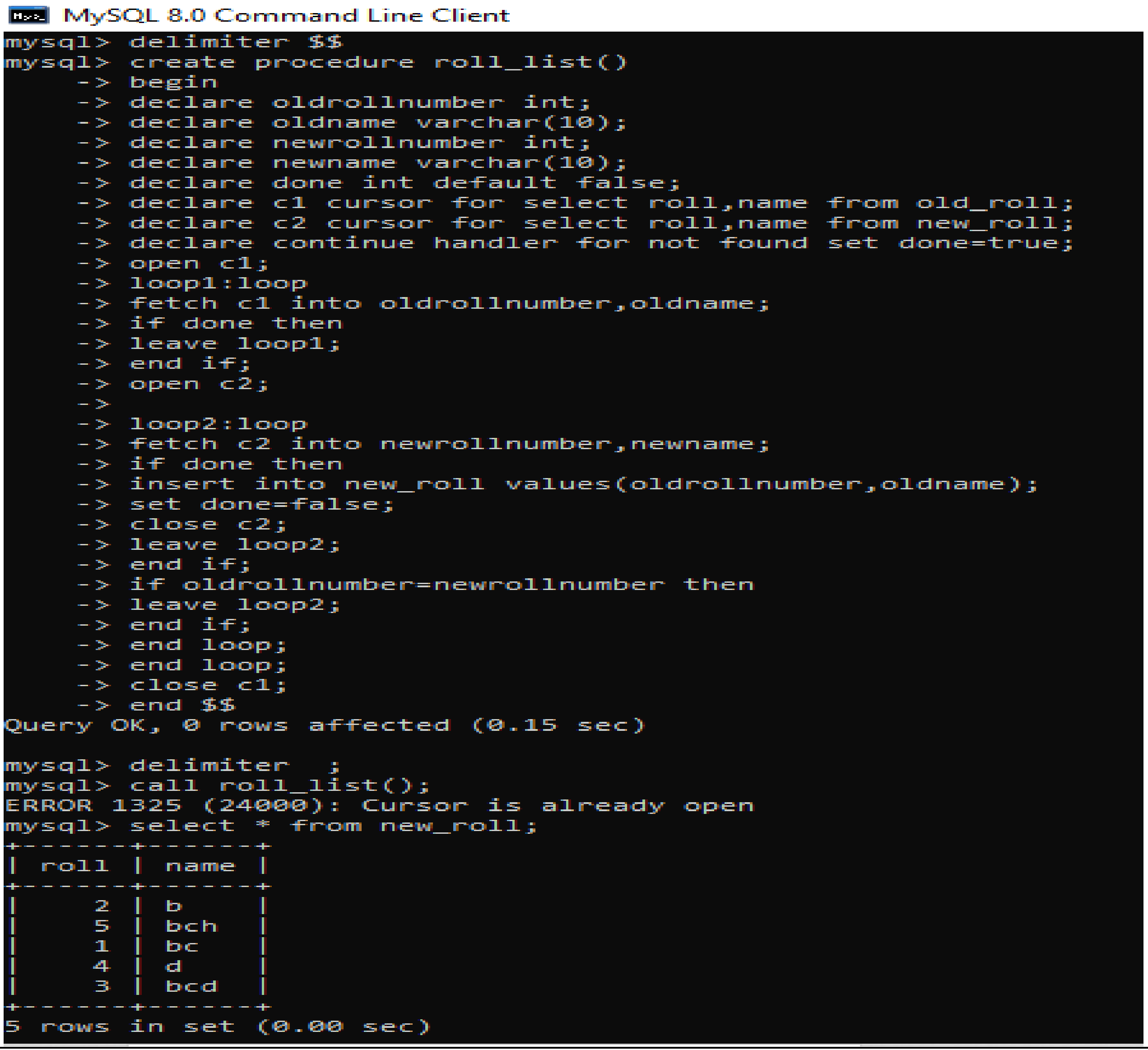
```
mysql> select * from old_roll;
+------+------+
| roll | name |
+------+------+
|    4 | d    |
|    3 | bcd  |
|    1 | bc   |
|    5 | bch  |
+------+------+
4 rows in set (0.00 sec)

mysql> select * from new_roll;
+------+------+
| roll | name |
+------+------+
|    2 | b    |
|    5 | bch  |
|    1 | bc   |
+------+------+
3 rows in set (0.00 sec)
```

Parameterized Cursor code:

```
delimiter $$
                    create procedure roll_list()
                    begin
                    declare oldrollnumber int;
                    declare oldname varchar(10);
                    declare newrollnumber int;
                    declare newname varchar(10);
                    declare done int default false;
                    declare c1 cursor for select roll,name from old_roll;
                    declare c2 cursor for select roll,name from new_roll;
                    declare continue handler for not found set done=true;
                    open c1;
                    loop1:loop
                    fetch c1 into oldrollnumber,oldname;
                    if done then
                    leave loop1;
                    end if;
                    open c2;

                    loop2:loop
                    fetch c2 into newrollnumber,newname;
                    if done then
                    insert into new_roll values(oldrollnumber,oldname);
```

```
set done=false;
close c2;
leave loop2;
end if;
if oldrollnumber=newrollnumber then
leave loop2;
end if;
end loop;
end loop;
close c1;
end $$
delimiter  ;
```

Parameterized Cursor code execution:



MySQL 8.0 Command Line Client

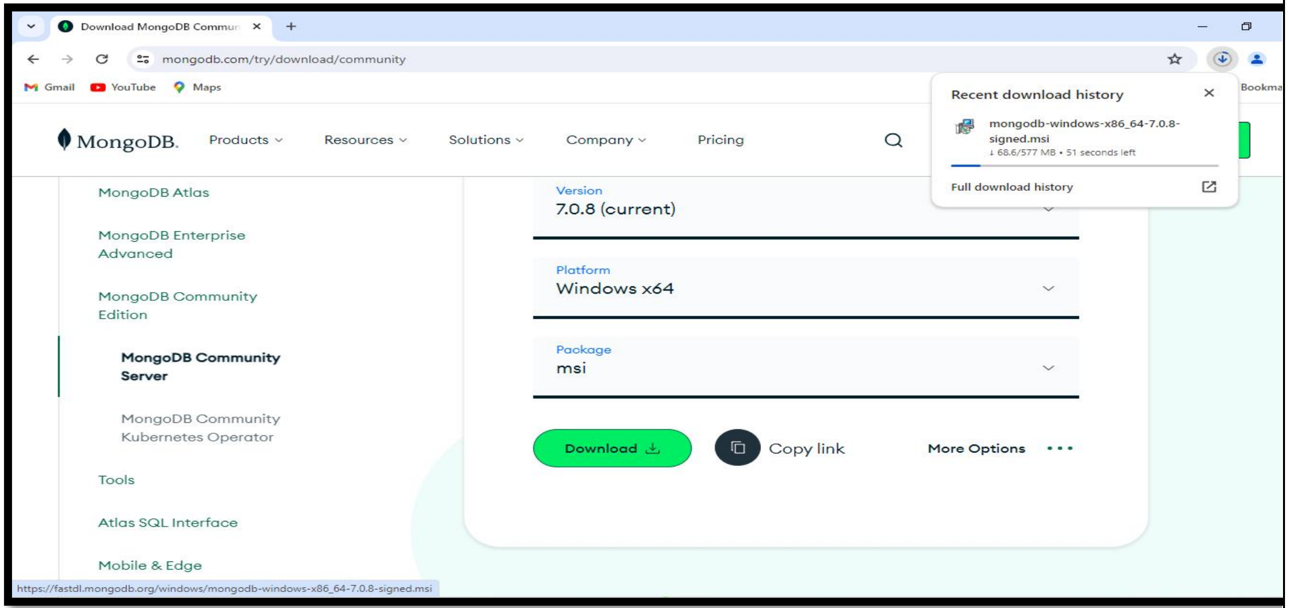
```
mysql> delimiter $$
mysql> create procedure roll_list()
    -> begin
    -> declare oldrollnumber int;
    -> declare oldname varchar(10);
    -> declare newrollnumber int;
    -> declare newname varchar(10);
    -> declare done int default false;
    -> declare c1 cursor for select roll,name from old_roll;
    -> declare c2 cursor for select roll,name from new_roll;
    -> declare continue handler for not found set done=true;
    -> open c1;
    -> loop1:loop
    -> fetch c1 into oldrollnumber,oldname;
    -> if done then
    -> leave loop1;
    -> end if;
    -> open c2;
    ->
    -> loop2:loop
    -> fetch c2 into newrollnumber,newname;
    -> if done then
    -> insert into new_roll values(oldrollnumber,oldname);
    -> set done=false;
    -> close c2;
    -> leave loop2;
    -> end if;
    -> if oldrollnumber=newrollnumber then
    -> leave loop2;
    -> end if;
    -> end loop;
    -> end loop;
    -> close c1;
    -> end $$
Query OK, 0 rows affected (0.15 sec)

mysql> delimiter  ;
mysql> call roll_list();
ERROR 1325 (24000): Cursor is already open
mysql> select * from new_roll;
+------+------+
| roll | name |
+------+------+
|    2 | b    |
|    5 | bch  |
|    1 | bc   |
|    4 | d    |
|    3 | bcd  |
+------+------+
5 rows in set (0.00 sec)
```

Experiment No 7

Install an Open Source NoSQL Data base MangoDB & perform basic CRUD(Create, Read, Update & Delete) operations. Execute MangoDB basic Queries using CRUD operations.

1)Install MongoDB:

Step 1:To Download& Install MongoDB : https://www.mongodb.com/try/download/community



mongodb.com/try/download/community

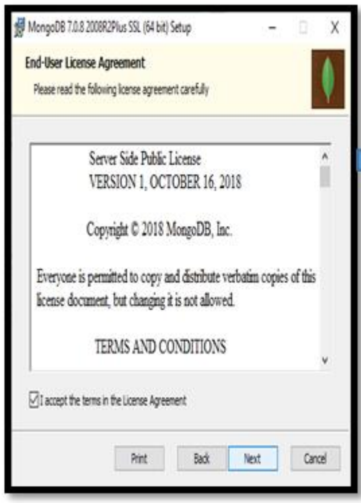
Recent download history
mongodb-windows-x86_64-7.0.8-signed.msi
↓ 68.6/577 MB • 51 seconds left
Full download history

Version
7.0.8 (current)

Platform
Windows x64

Package
msi

https://fastdl.mongodb.org/windows/mongodb-windows-x86_64-7.0.8-signed.msi



MongoDB 7.0.8 2008R2Plus SSL (64 bit) Setup

End-User License Agreement
Please read the following license agreement carefully

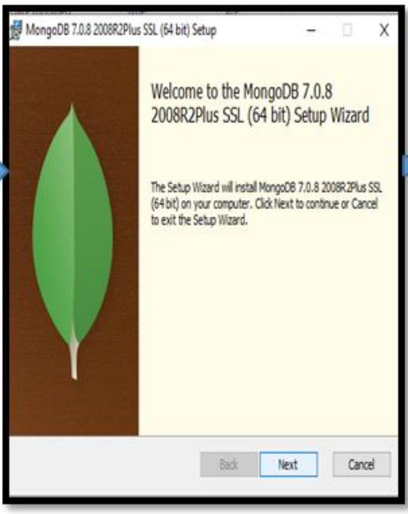
Server Side Public License
VERSION 1, OCTOBER 16, 2018

Copyright © 2018 MongoDB, Inc.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

TERMS AND CONDITIONS

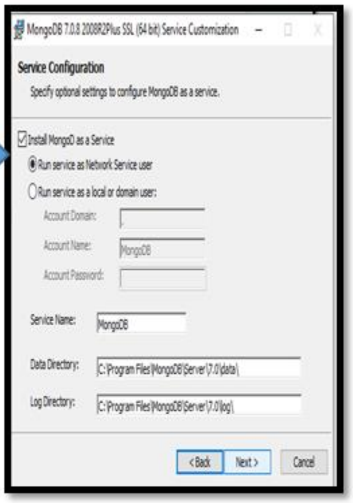
☑ I accept the terms in the License Agreement



MongoDB 7.0.8 2008R2Plus SSL (64 bit) Setup

Welcome to the MongoDB 7.0.8 2008R2Plus SSL (64 bit) Setup Wizard

The Setup Wizard will install MongoDB 7.0.8 2008R2Plus SSL (64 bit) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

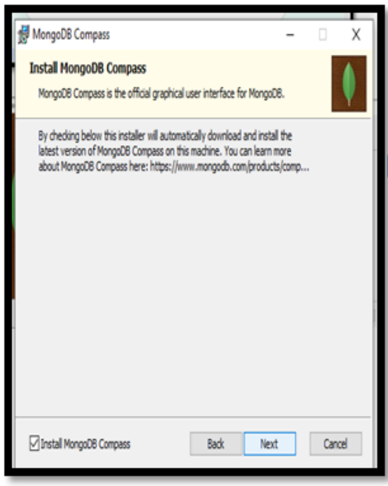


MongoDB 7.0.8 2008R2Plus SSL (64 bit) Service Customization

Service Configuration
Specify optional settings to configure MongoDB as a service.

☑ Install MongoD as a Service
● Run service as Network Service user
○ Run service as a local or domain user:
Account Domain:
Account Name: MongoDB
Account Password:

Service Name: MongoDB
Data Directory: C:\Program Files\MongoDB\Server\7.0\data\
Log Directory: C:\Program Files\MongoDB\Server\7.0\log\

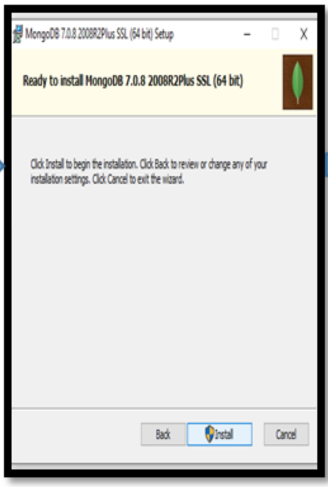


MongoDB Compass

Install MongoDB Compass
MongoDB Compass is the official graphical user interface for MongoDB.

By checking below this installer will automatically download and install the latest version of MongoDB Compass on this machine. You can learn more about MongoDB Compass here: https://www.mongodb.com/products/comp...

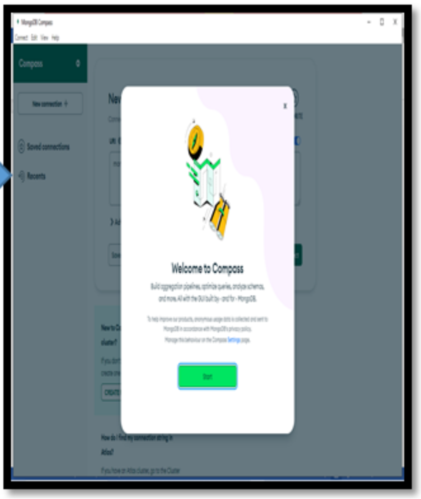
☑ Install MongoDB Compass



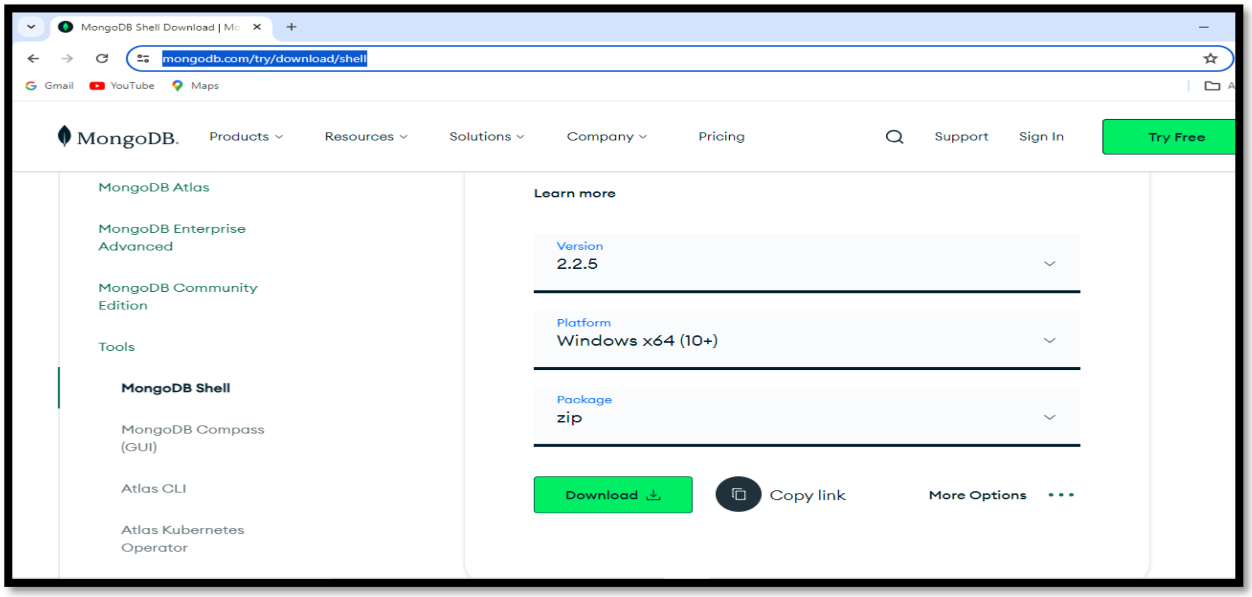
MongoDB 7.0.8 2008R2Plus SSL (64 bit) Setup

Ready to install MongoDB 7.0.8 2008R2Plus SSL (64 bit)

Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.



Step 2:To Download& Install MongoDB shell :
https://www.mongodb.com/try/download/shell



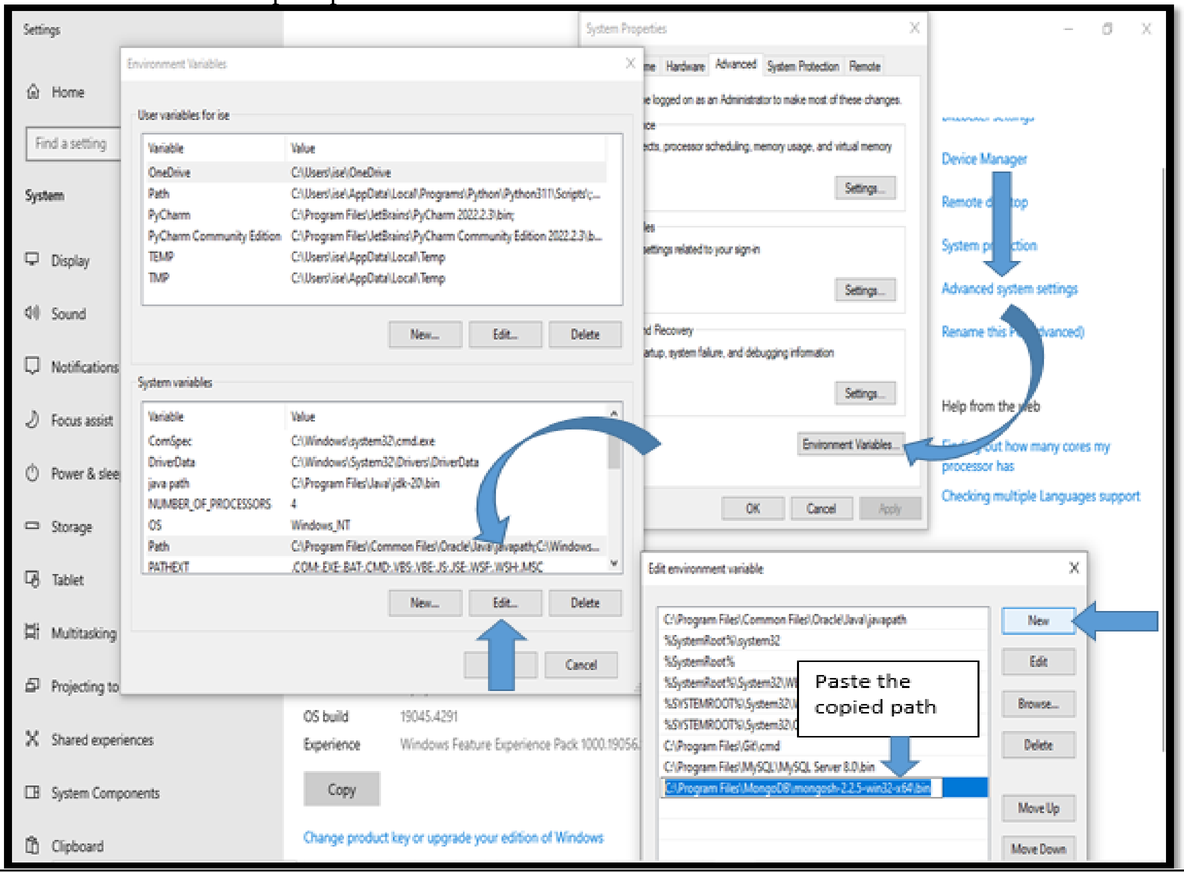
Version
2.2.5

Platform
Windows x64 (10+)

Package
zip

Step 3:Open and copy the location: C:\Program Files\MongoDB\mongosh-2.2.5-win32-x64\bin

Paste the copied path in **Edit Environment variable**



User variables for ise

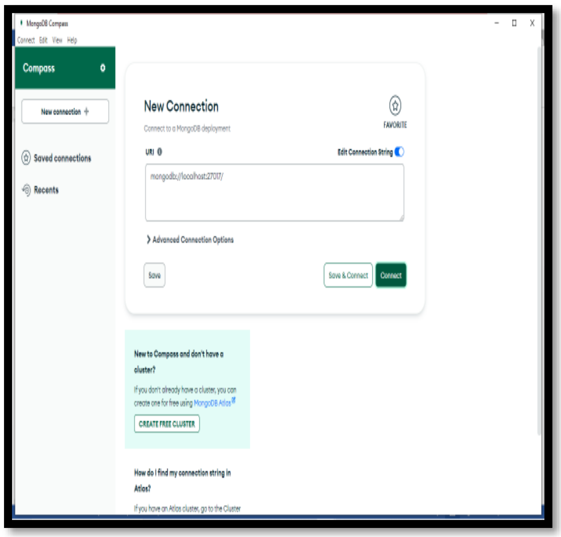
Variable	Value
OneDrive	C:\Users\ise\OneDrive
Path	C:\Users\ise\AppData\Local\Programs\Python\Python311\Scripts\;...
PyCharm	C:\Program Files\JetBrains\PyCharm 2022.2.3\bin;
PyCharm Community Edition	C:\Program Files\JetBrains\PyCharm Community Edition 2022.2.3\b...
TEMP	C:\Users\ise\AppData\Local\Temp
TMP	C:\Users\ise\AppData\Local\Temp

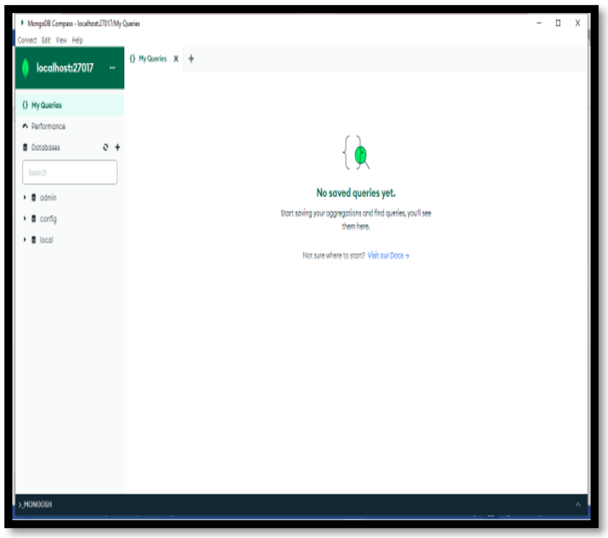
System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
java path	C:\Program Files\Java\jdk-20\bin
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Common Files\Oracle\Java\javapath;C:\Windows...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC

OS build 19045.4291
Experience Windows Feature Experience Pack 1000.19056.

```
C:\Program Files\Common Files\Oracle\Java\javapath
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\W
%SYSTEMROOT%\System32\
%SYSTEMROOT%\System32\O
C:\Program Files\Git\cmd
C:\Program Files\MySQL\MySQL Server 8.0\bin
C:\Program Files\MongoDB\mongosh-2.2.5-win32-x64\bin
```

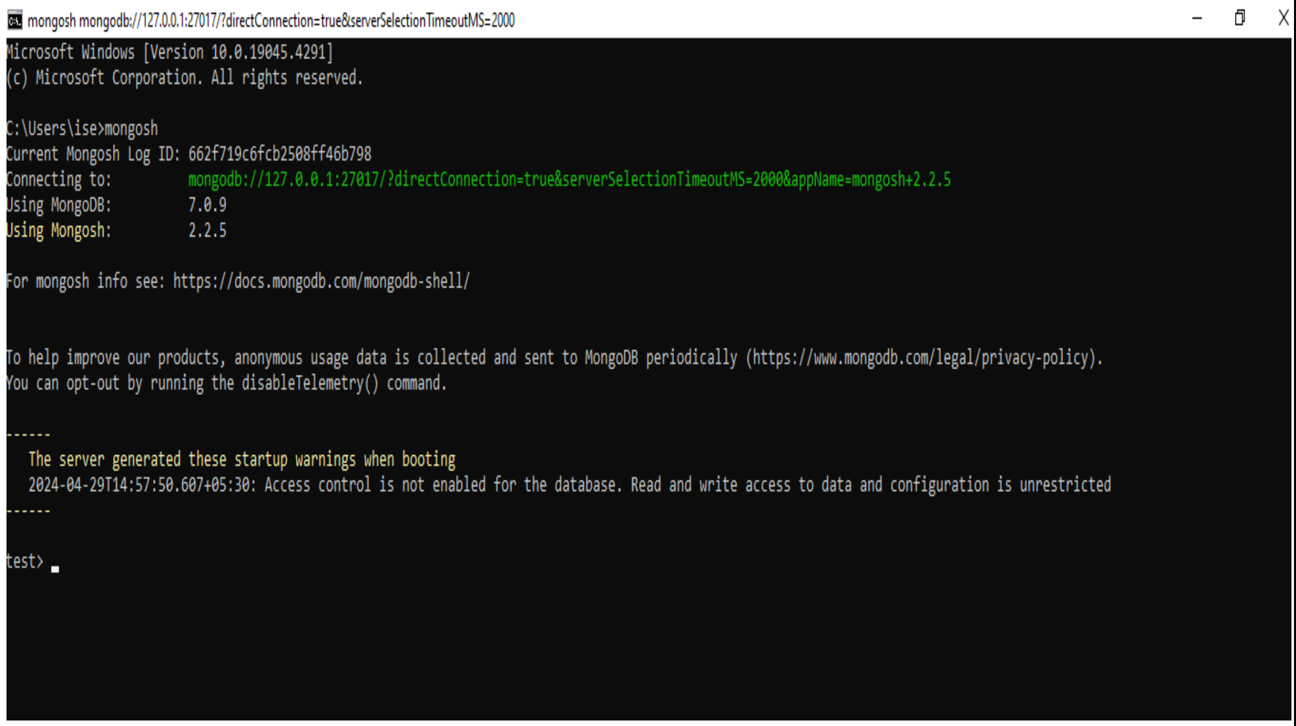

MongoDB Compass
Connect Edit View Help
Compass
New connection +
Saved connections
Recents
New Connection
Connect to a MongoDB deployment
FAVORITE
URI
Edit Connection String
mongodb://localhost:27017/
Advanced Connection Options
Save
Save & Connect
Connect
New to Compass and don't have a cluster?
If you don't already have a cluster, you can create one for free using MongoDB Atlas
CREATE FREE CLUSTER
How do I find my connection string in Atlas?
If you have an Atlas cluster, go to the Cluster


MongoDB Compass - localhost:27017/My Queries
Connect Edit View Help
localhost:27017
My Queries
Performance
Databases
Search
admin
config
local
No saved queries yet.
Start saving your aggregations and find queries, you'll see them here.
Not sure where to start? Visit our Docs →
>_MONGOSH

Step4:open MongoDB –click on CONNECT

2)MongoDB basic query execution with CRUD operation :

i)Open Command Prompt Connect MongoDB shell:
C:\Users\ise>mongosh


```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ise>mongosh
Current Mongosh Log ID: 662f719c6fcb2508ff46b798
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.5
Using MongoDB:          7.0.9
Using Mongosh:          2.2.5

For mongosh info see: https://docs.mongodb.com/mongodb-shell/


To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

------
   The server generated these startup warnings when booting
   2024-04-29T14:57:50.607+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test>
```

ii)To create collection& display collection:

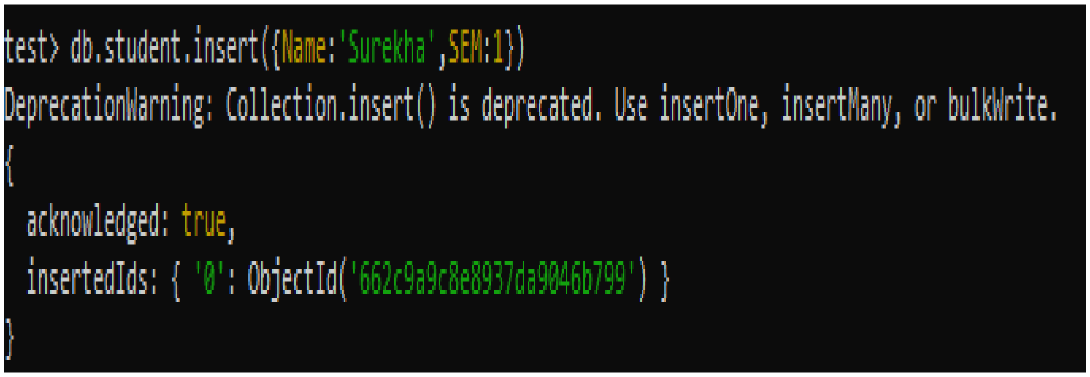
test> db.createCollection('Student')

test> show collections

iii)To Insert Records:

To Insert single record:

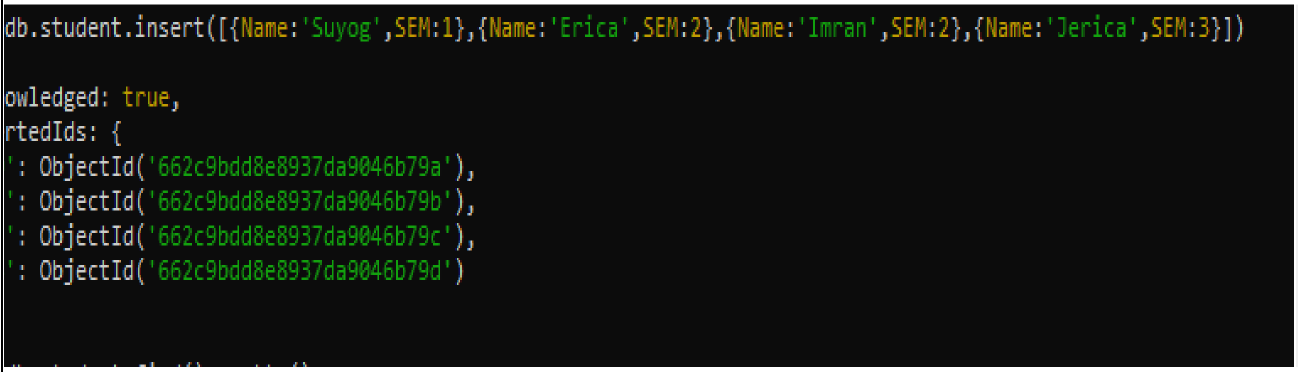
test> db.student.insert({Name:'Surekha',SEM:1})



```
test> db.student.insert({Name:'Surekha',SEM:1})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('662c9a9c8e8937da9046b799') }
}
```

To Insert Multi records:

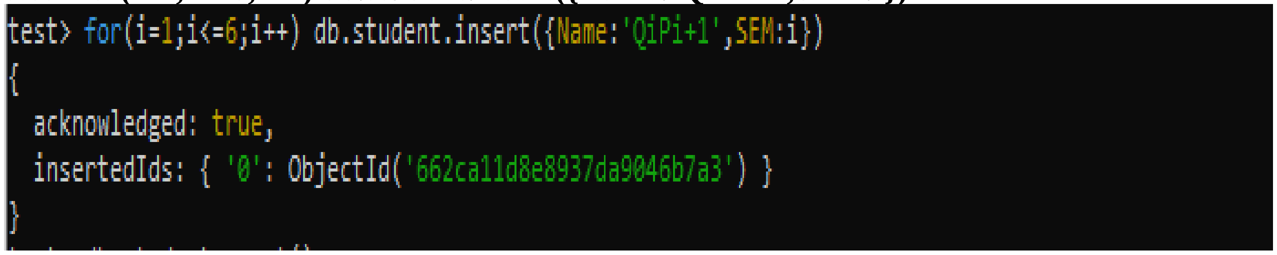
test>
db.student.insert([{Name:'Suyog',SEM:1},{Name:'Erica',SEM:2},{Name:'Imran',SEM:2},{Name:'Jerica',SEM:3}])



```
db.student.insert([{Name:'Suyog',SEM:1},{Name:'Erica',SEM:2},{Name:'Imran',SEM:2},{Name:'Jerica',SEM:3}])

owledged: true,
rtedIds: {
': ObjectId('662c9bdd8e8937da9046b79a'),
': ObjectId('662c9bdd8e8937da9046b79b'),
': ObjectId('662c9bdd8e8937da9046b79c'),
': ObjectId('662c9bdd8e8937da9046b79d')
```

To Insert Multi records using FOR loop:

test> for(i=1;i<=6;i++) db.student.insert({Name:'QiPi+1',SEM:i})

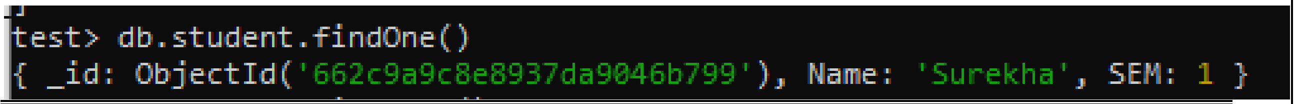


```
test> for(i=1;i<=6;i++) db.student.insert({Name:'QiPi+1',SEM:i})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('662ca11d8e8937da9046b7a3') }
}
```

iv)To Read Collection:

<u>To read single record:</u>

test> db.student.findOne()

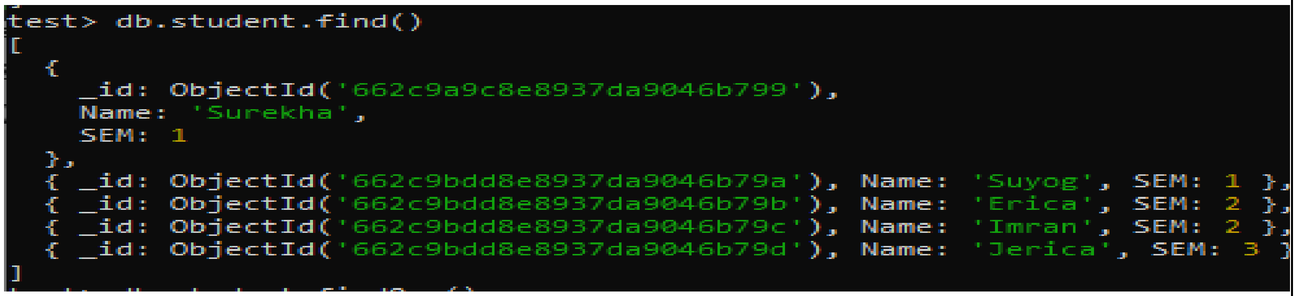


```
test> db.student.findOne()
{ _id: ObjectId('662c9a9c8e8937da9046b799'), Name: 'Surekha', SEM: 1 }
```

<u>To read ALL records:</u>

1.find():display the all documents in collection.

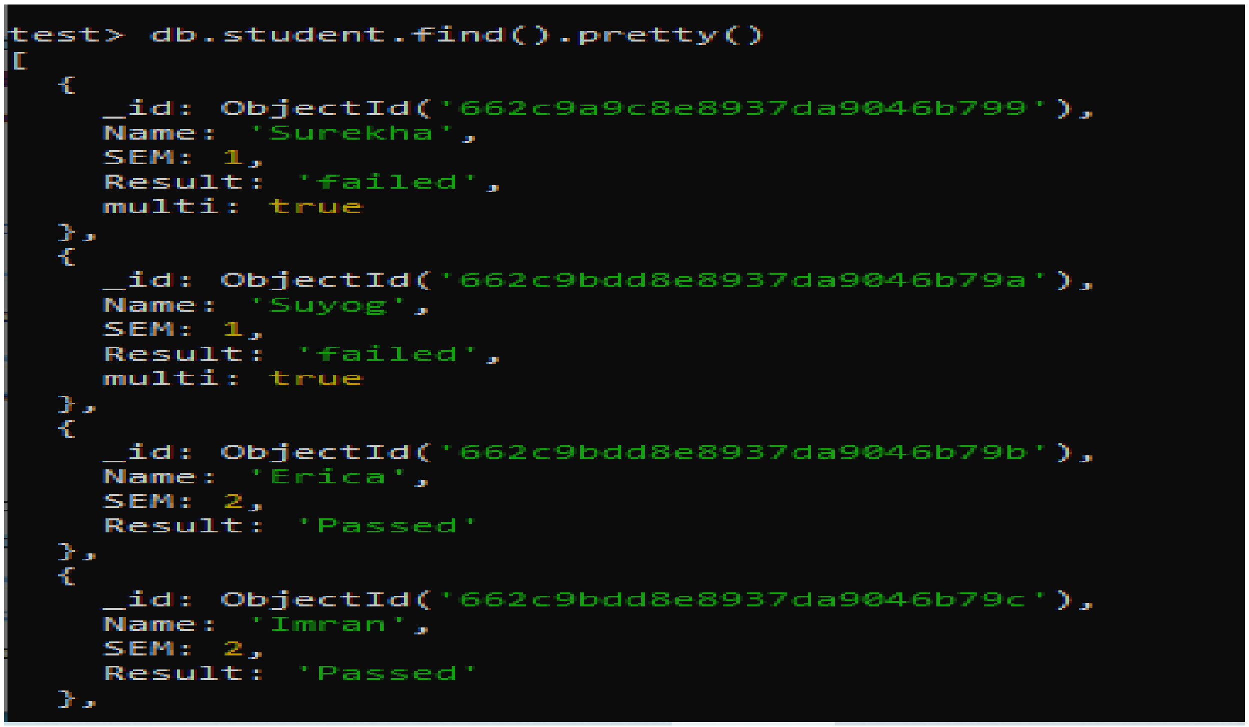
test> db.student.find()



```
test> db.student.find()
[
  {
    _id: ObjectId('662c9a9c8e8937da9046b799'),
    Name: 'Surekha',
    SEM: 1
  },
  { _id: ObjectId('662c9bdd8e8937da9046b79a'), Name: 'Suyog', SEM: 1 },
  { _id: ObjectId('662c9bdd8e8937da9046b79b'), Name: 'Erica', SEM: 2 },
  { _id: ObjectId('662c9bdd8e8937da9046b79c'), Name: 'Imran', SEM: 2 },
  { _id: ObjectId('662c9bdd8e8937da9046b79d'), Name: 'Jerica', SEM: 3 }
]
```

<u>2. find (). pretty ():</u> display the all-documents if collection is having more than 2 fields/columns.

test> db.student.find().pretty()



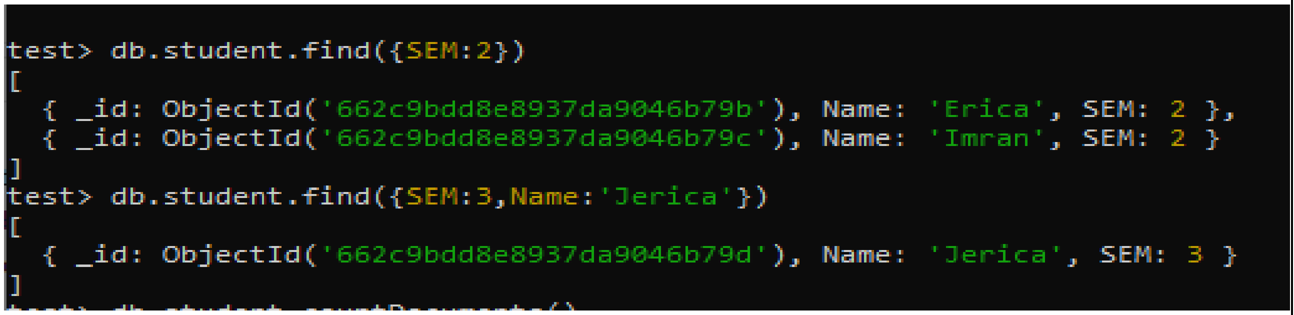
```
test> db.student.find().pretty()
[
  {
    _id: ObjectId('662c9a9c8e8937da9046b799'),
    Name: 'Surekha',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79a'),
    Name: 'Suyog',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79b'),
    Name: 'Erica',
    SEM: 2,
    Result: 'Passed'
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79c'),
    Name: 'Imran',
    SEM: 2,
    Result: 'Passed'
  },
```

3)read document with condition:

Example:

test> db.student.find({SEM:2})

test> db.student.find({SEM:3,Name:'Jerica'})



```
test> db.student.find({SEM:2})
[
  { _id: ObjectId('662c9bdd8e8937da9046b79b'), Name: 'Erica', SEM: 2 },
  { _id: ObjectId('662c9bdd8e8937da9046b79c'), Name: 'Imran', SEM: 2 }
]
test> db.student.find({SEM:3,Name:'Jerica'})
[
  { _id: ObjectId('662c9bdd8e8937da9046b79d'), Name: 'Jerica', SEM: 3 }
]
```

v)To Count the records:

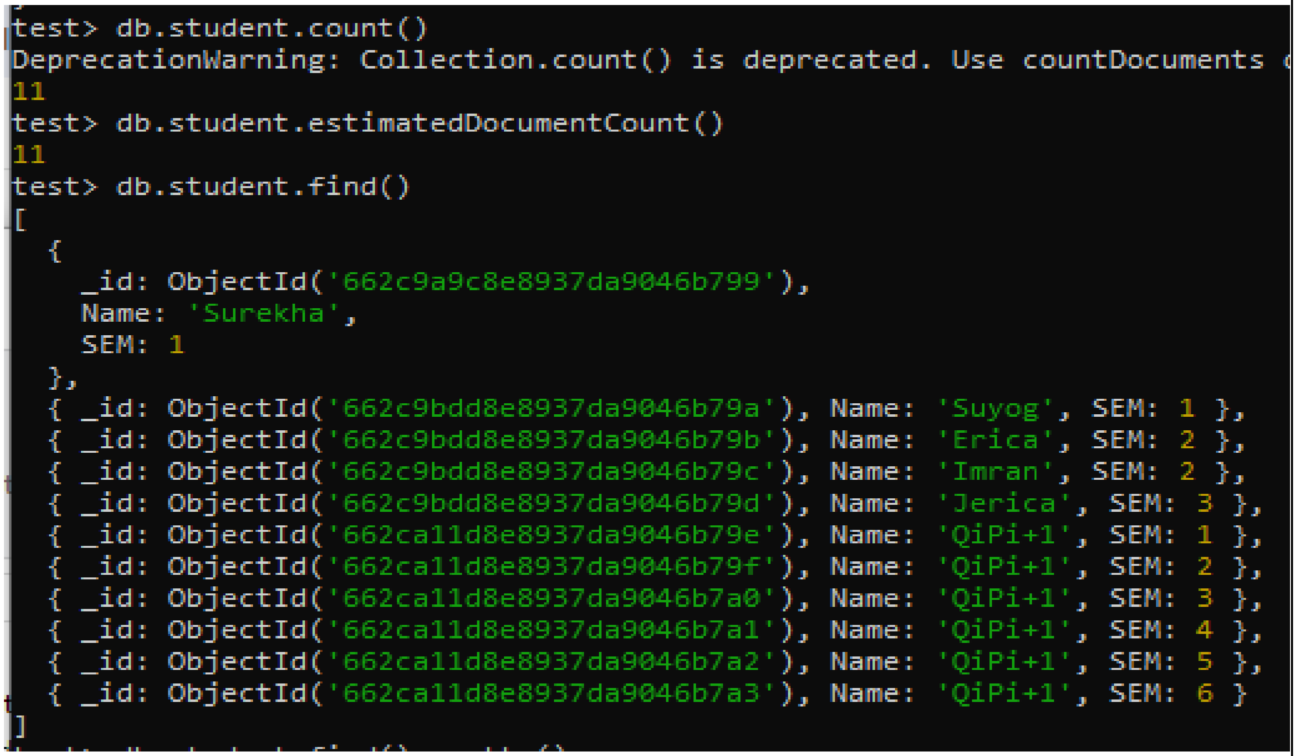
Use count Documents or estimatedDocumentCount or count.

Example:

test> db.student.countDocuments( )

test> db.student.count()

test> db.student.estimatedDocumentCount()



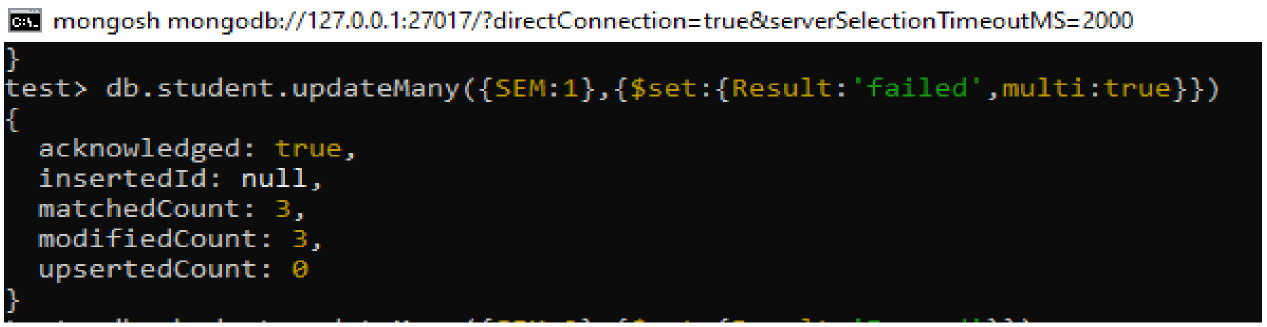
```
test> db.student.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments
11
test> db.student.estimatedDocumentCount()
11
test> db.student.find()
[
  {
    _id: ObjectId('662c9a9c8e8937da9046b799'),
    Name: 'Surekha',
    SEM: 1
  },
  { _id: ObjectId('662c9bdd8e8937da9046b79a'), Name: 'Suyog', SEM: 1 },
  { _id: ObjectId('662c9bdd8e8937da9046b79b'), Name: 'Erica', SEM: 2 },
  { _id: ObjectId('662c9bdd8e8937da9046b79c'), Name: 'Imran', SEM: 2 },
  { _id: ObjectId('662c9bdd8e8937da9046b79d'), Name: 'Jerica', SEM: 3 },
  { _id: ObjectId('662ca11d8e8937da9046b79e'), Name: 'QiPi+1', SEM: 1 },
  { _id: ObjectId('662ca11d8e8937da9046b79f'), Name: 'QiPi+1', SEM: 2 },
  { _id: ObjectId('662ca11d8e8937da9046b7a0'), Name: 'QiPi+1', SEM: 3 },
  { _id: ObjectId('662ca11d8e8937da9046b7a1'), Name: 'QiPi+1', SEM: 4 },
  { _id: ObjectId('662ca11d8e8937da9046b7a2'), Name: 'QiPi+1', SEM: 5 },
  { _id: ObjectId('662ca11d8e8937da9046b7a3'), Name: 'QiPi+1', SEM: 6 }
]
```

vi) UPDATE collection:

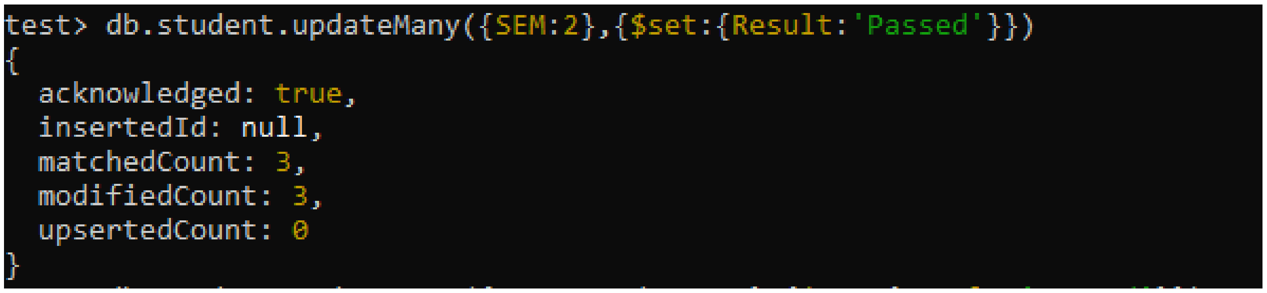
Example:

1.updateMany():

test> db.student.updateMany({SEM:1},{$set:{Result:'failed',multi:true}})



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
}
test> db.student.updateMany({SEM:1},{$set:{Result:'failed',multi:true}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
```



```
test> db.student.updateMany({SEM:2},{$set:{Result:'Passed'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
```

```
test> db.student.find().pretty()
[
  {
    _id: ObjectId('662c9a9c8e8937da9046b799'),
    Name: 'Surekha',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79a'),
    Name: 'Suyog',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79b'),
    Name: 'Erica',
    SEM: 2,
    Result: 'Passed'
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79c'),
    Name: 'Imran',
    SEM: 2,
    Result: 'Passed'
  },
  { _id: ObjectId('662c9bdd8e8937da9046b79d'), Name: 'Jerica', SEM: 3 },
  {
    _id: ObjectId('662ca11d8e8937da9046b79e'),
    Name: 'QiPi+1',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662ca11d8e8937da9046b79f'),
    Name: 'QiPi+1',
    SEM: 2,
    Result: 'Passed'
  },
  { _id: ObjectId('662ca11d8e8937da9046b7a0'), Name: 'QiPi+1', SEM: 3 },
  { _id: ObjectId('662ca11d8e8937da9046b7a1'), Name: 'QiPi+1', SEM: 4 },
  { _id: ObjectId('662ca11d8e8937da9046b7a2'), Name: 'QiPi+1', SEM: 5 },
  { _id: ObjectId('662ca11d8e8937da9046b7a3'), Name: 'QiPi+1', SEM: 6 }
```

2.***remove():***

```
test> db.student.remove({SEM:5})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany,
findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
test> db.student.remove({SEM:6})
{ acknowledged: true, deletedCount: 1 }
test> db.student.find().pretty()
[
  {
    _id: ObjectId('662c9a9c8e8937da9046b799'),
    Name: 'Surekha',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79a'),
    Name: 'Suyog',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79b'),
    Name: 'Erica',
    SEM: 2,
    Result: 'Passed'
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79c'),
    Name: 'Imran',
    SEM: 2,
    Result: 'Passed'
  },
  { _id: ObjectId('662c9bdd8e8937da9046b79d'), Name: 'Jerica', SEM: 3 },
  {
    _id: ObjectId('662ca11d8e8937da9046b79e'),
    Name: 'QiPi+1',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662ca11d8e8937da9046b79f'),
    Name: 'QiPi+1',
    SEM: 2,
    Result: 'Passed'
  },
  { _id: ObjectId('662ca11d8e8937da9046b7a0'), Name: 'QiPi+1', SEM: 3 },
  { _id: ObjectId('662ca11d8e8937da9046b7a1'), Name: 'QiPi+1', SEM: 4 }
]
test> db.student.remove({SEM:3},1)
{ acknowledged: true, deletedCount: 2 }
test> db.student.find().pretty()
[
  {
    _id: ObjectId('662c9a9c8e8937da9046b799'),
    Name: 'Surekha',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79a'),
    Name: 'Suyog',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79b'),
    Name: 'Erica',
    SEM: 2,
    Result: 'Passed'
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79c'),
    Name: 'Imran',
    SEM: 2,
    Result: 'Passed'
  },
  {
    _id: ObjectId('662ca11d8e8937da9046b79e'),
    Name: 'QiPi+1',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662ca11d8e8937da9046b79f'),
    Name: 'QiPi+1',
    SEM: 2,
    Result: 'Passed'
  },
  { _id: ObjectId('662ca11d8e8937da9046b7a1'), Name: 'QiPi+1', SEM: 4 }
]
```

3.delete():

```
test> db.student.deleteOne({SEM:4})
{ acknowledged: true, deletedCount: 1 }
test> db.student.find().pretty()
[
  {
    _id: ObjectId('662c9a9c8e8937da9046b799'),
    Name: 'Surekha',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79a'),
    Name: 'Suyog',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79b'),
    Name: 'Erica',
    SEM: 2,
    Result: 'Passed'
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79c'),
    Name: 'Imran',
    SEM: 2,
    Result: 'Passed'
  },
  {
    _id: ObjectId('662ca11d8e8937da9046b79e'),
    Name: 'QiPi+1',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662ca11d8e8937da9046b79f'),
    Name: 'QiPi+1',
    SEM: 2,
    Result: 'Passed'
  }
]
test> db.student.deleteMany({SEM:2})
{ acknowledged: true, deletedCount: 3 }
test> db.student.find().pretty()
[
  {
    _id: ObjectId('662c9a9c8e8937da9046b799'),
    Name: 'Surekha',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662c9bdd8e8937da9046b79a'),
    Name: 'Suyog',
    SEM: 1,
    Result: 'failed',
    multi: true
  },
  {
    _id: ObjectId('662ca11d8e8937da9046b79e'),
    Name: 'QiPi+1',
    SEM: 1,
    Result: 'failed',
    multi: true
  }
]
```