PRACTICAL NO:2

Name:

Roll no:

```
(base) rmdstic@rmdstic-OptiPlex-3020:~$ sudo mysql

[sudo] password for rmdstic:

Welcome to the MySQL monitor.  Commands end with ; or \g.

Your MySQL connection id is 11

Server version: 8.0.33-0ubuntu0.20.04.2 (Ubuntu)

Copyright © 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 database db_students;

Query OK, 1 row affected (0.09 sec)

Mysql> use db_students;

Database changed

Mysql> create table tbl_student_details( stud_prn integer(10) primary key, stud_name varchar(20) not
null, stud_class varchar(20) not null, stud_stream varchar(20) not null);

Query OK, 0 rows affected, 1 warning (0.32 sec)

Mysql> use db_students;

Database changed

Mysql> create view vw_students_details as( select stud_name, stud_class from tbl_student_details );

Query OK, 0 rows affected (0.09 sec)

Mysql> select * from vw_students_details;

Empty set (0.01 sec)

Mysql> desc vw_students_details;

+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| stud_name  | varchar(20) | NO   |     | NULL    |       |
| stud_class | varchar(20) | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
Mysql> desc tbl_student_details;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| stud_prn    | int         | NO   | PRI | NULL    |       |
| stud_name   | varchar(20) | NO   |     | NULL    |       |
| stud_class  | varchar(20) | NO   |     | NULL    |       |
| stud_stream | varchar(20) | NO   |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
Mysql> create INDEX indx_student_details on tbl_student_details (stud_prn);
Query OK, 0 rows affected (0.24 sec)
Records: 0  Duplicates: 0  Warnings: 0
Mysql> create TABLE dummy_table( col1 int(1), col2 int(5), col3 varchar(5), INDEX(col2,col3) );
Query OK, 0 rows affected, 2 warnings (0.51 sec)
Mysql> use db_students;

Database changed

Mysql> create table tbl_info( sr_no integer(2) AUTO_INCREMENT PRIMARY KEY, first_name varchar(10),
last_name varchar(10));

Query OK, 0 rows affected, 1 warning (0.33 sec)
```

```
Mysql> insert into tbl_info(first_name,last_name) values('omkar','ghodake');

Query OK, 1 row affected (0.06 sec)

Mysql> insert into tbl_info(first_name,last_name) values('deeapk','dukare');

Query OK, 1 row affected (0.09 sec)

Mysql> select* from tbl_info;

| sr_no | first_name | last_name |

|     1 | omkar      | ghodake   |

|     2 | deeapk     | dukare    |

Cm2  rows in set (0.00 sec)

Mysql>
```

PRACTICAL NO:3

Name:

Roll no:

```
(base) rmdstic@rmdstic-OptiPlex-3020:~$ sudo mysql
[sudo] password for rmdstic:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.33-0ubuntu0.20.04.2 (Ubuntu)

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> show databases;
+--------------------+
| Database           |
+--------------------+
| DatabaseName       |
| Database_Name      |
| Stud               |
| StudDB             |
| T3                 |
| abcd               |
| aka                |
| cl                 |
| class              |

| client             |

| d1                 |

| db_students        |

| dc                 |

| information_schema |

| myDB               |

| mysql              |

| performance_schema |

| stud               |

| sys                |

| t2                 |

| test               |

+--------------------+

21 rows in set (0.00 sec)


mysql> use test;

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A


Database changed

mysql> create table student(roll int(5) primary key, name varchar(30),marks int(3));

Query OK, 0 rows affected, 2 warnings (0.31 sec)


mysql> desc student;

+-------+-------------+------+-----+---------+-------+

| Field | Type        | Null | Key | Default | Extra |

+-------+-------------+------+-----+---------+-------+

| roll  | int         | NO   | PRI | NULL    |       |
| name  | varchar(30) | YES  |     | NULL    |       |
| marks | int         | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> INSERT INTO student(roll,name,marks) VAlUES('1','omkar','98');
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO student(roll,name,marks) VAlUES('2','deepak','93');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO student(roll,name,marks) VAlUES('3','abhinav','90');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO student(roll,name,marks) VAlUES('4','amit','78');
Query OK, 1 row affected (0.09 sec)

mysql> select * from student;
+------+---------+-------+
| roll | name    | marks |
+------+---------+-------+
|    1 | omkar   |    98 |
|    2 | deepak  |    93 |
|    3 | abhinav |    90 |
|    4 | amit    |    78 |
+------+---------+-------+
4 rows in set (0.00 sec)

mysql> INSERT INTO student(roll,name,marks) VAlUES('6','arnav','88');
```

```
Query OK, 1 row affected (0.06 sec)

mysql> select * from student;
+------+---------+-------+
| roll | name    | marks |
+------+---------+-------+
|    1 | omkar   |    98 |
|    2 | deepak  |    93 |
|    3 | abhinav |    90 |
|    4 | amit    |    78 |
|    6 | arnav   |    88 |
+------+---------+-------+
5 rows in set (0.00 sec)

mysql> UPDATE student SET roll=5 WHERE name='arnav';
Query OK, 1 row affected (0.09 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student;
+------+---------+-------+
| roll | name    | marks |
+------+---------+-------+
|    1 | omkar   |    98 |
|    2 | deepak  |    93 |
|    3 | abhinav |    90 |
|    4 | amit    |    78 |
|    5 | arnav   |    88 |
+------+---------+-------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT roll,name from student WHERE roll>=3;
+------+---------+
| roll | name    |
+------+---------+
|    3 | abhinav |
|    4 | amit    |
|    5 | arnav   |
+------+---------+
3 rows in set (0.00 sec)

mysql> SELECT name from student ORDER BY name;
+---------+
| name    |
+---------+
| abhinav |
| amit    |
| arnav   |
| deepak  |
| omkar   |
+---------+
5 rows in set (0.01 sec)

mysql> SELECT name from student ORDER BY name DESC;
+---------+
| name    |
+---------+
| omkar   |
| deepak  |
| arnav   |

| amit    |

| abhinav |

+---------+

5 rows in set (0.00 sec)


mysql> DELETE from student where roll='5';

Query OK, 1 row affected (0.07 sec)


mysql> select * from student;

+------+---------+-------+

| roll | name    | marks |

+------+---------+-------+

|    1 | omkar   |    98 |

|    2 | deepak  |    93 |

|    3 | abhinav |    90 |

|    4 | amit    |    78 |

+------+---------+-------+

4 rows in set (0.00 sec)


mysql> SELECT COUNT(marks) from student;

+--------------+

| COUNT(marks) |

+--------------+

|            4 |

+--------------+

1 row in set (0.00 sec)


mysql> SELECT COUNT(*) from student;
```

```
+----------+
| COUNT(*) |
+----------+
|        4 |
+----------+
1 row in set (0.00 sec)

mysql> SELECT AVG(marks) from student;
+------------+
| AVG(marks) |
+------------+
|    89.7500 |
+------------+
1 row in set (0.00 sec)

mysql> SELECT MIN(marks) from student;
+------------+
| MIN(marks) |
+------------+
|         78 |
+------------+
1 row in set (0.00 sec)

mysql> SELECT MAX(marks) from student;
+------------+
| MAX(marks) |
+------------+
|         98 |
+------------+
1 row in set (0.00 sec)


mysql> SELECT SUM(marks) from student;

+------------+

| SUM(marks) |

+------------+

|        359 |

+------------+

1 row in set (0.00 sec)


mysql>
```

PRACTICAL NO:4

Name:

Roll No:

```
(base) rmdstic@rmdstic-OptiPlex-3020:~$ sudo mysql;

[sudo] password for rmdstic:

Welcome to the MySQL monitor.  Commands end with ; or \g.

Your MySQL connection id is 13

Server version: 8.0.33-0ubuntu0.20.04.2 (Ubuntu)

Copyright © 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> show database

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MySQL server version for the right syntax to use near ‘database’ at line 1

Mysql> show databases;

| Database          |

| RMD               |

| RMDSCOE           |

| RMDSSOE           |

| Stud              |

| T2                |

| d1                |

| db_student        |

| deepak            |
| information_schema |

| myDB               |

| mydb               |

| mysql              |

| performance_schema |

| stud               |

| tt1                |

15 rows in set (0.00 sec)

Mysql> use stud;

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with -A

Database changed

Mysql> create table Borrower(roll_no int,name varchar(10),date_of_issue date, name_of_book
varchar(10),status char);

Query OK, 0 rows affected (0.52 sec)

Mysql> insert into Borrower values(1,'AA','2023-08-01','Book1','I'),(2,'BB','2023-08-
12','Book2','I'),(3,'CC','2023-07-25','Book3','I');

Query OK, 3 rows affected (0.07 sec)

Records: 3  Duplicates: 0  Warnings: 0

Mysql> create table Fine(roll_no int, date_of_issue date ,Amt int);

Query OK, 0 rows affected (0.28 sec)

Mysql> DELIMITER @

Mysql> CREATE PROCEDURE Calculate_Fine(in r int, in nb varchar(10))

  ➔   BEGIN
  ➔   DECLARE d date;
  ➔   DECLARE diff int;
  ➔   DECLARE fine_amt int;
  ➔   Select date_of_issue into d from Borrower where roll_no=r and name_of_book=nb;
  ➔   Select DATEDIFF(CURDATE(),d) into diff;
  ➔   If (diff>15) and (diff<=30) then
➔      Set fine_amt= 5 * diff;
➔      Insert into Fine values(r,d,fine_amt);
➔      Elseif (diff>30) then
➔      Set fine_amt= 50 * diff;
➔      Insert into Fine values(r,d,fine_amt);
➔       End if;
➔      Update Borrower set status='R' where roll_no=r;
➔       END @

Query OK, 0 rows affected (0.11 sec)

Mysql> DELIMITER ;

Mysql> Select * from Borrower;

+---------+------+---------------+--------- -----+--------+

| roll_no | name | date_of_issue | name_of_book | status |

+---------+------+---------------+--------------+--------+

|      1 | AA   | 2023-08-01    | Book1        | I      |

|      2 | BB   | 2023-08-12    | Book2        | I      |

|      3 | CC   | 2023-07-25    | Book3        | I      |

+---------+------+---------------+--------------+--------+

3 rows in set (0.00 sec)

Mysql> call Calculate_Fine(2,'Book2');

Query OK, 1 row affected (0.16 sec)

Mysql> select * from Fine;


| roll_no | date_of_issue | Amt  |


|      2 | 2023-08-12    | 1550 |


1 row in set (0.00 sec)

Mysql> call Calculate_Fine(1,'Book1');
```

```
Query OK, 1 row affected (0.12 sec)

Mysql> call Calculate_Fine(2,'Book2');

Query OK, 0 rows affected (0.07 sec)

Mysql> call Calculate_Fine(3,'Book3');

Query OK, 1 row affected (0.14 sec)

Mysql>  select * from Fine;

| roll_no | date_of_issue | Amt  |

|      2 | 2023-08-12    | 1550 |

|      1 | 2023-08-01    | 2100 |

|      2 | 2023-08-12    | 1550 |

|      3 | 2023-07-25    | 2450 |

4 rows in set (0.00 sec)

Mysql>
```

Name :
Roll No :
Div : B

PRACTICAL NO:-05

Que. Unnamed PL/SQLcode block: Use of Control structure and Exception handling is mandatory.

```
SQL> create table ar(radius number(5), area number(14,2));

Table created.

SQL> declare
  2     r number(5);
  3     ar number(14,2);
  4    pi constant number (4,2):=3.14;
  5    begin
  6    r:=5;
  7    while r<=9
  8    loop
  9   ar:=pi*power(r,2);
 10   insert into ar values(r,ar);
 11    r:=r+1;
 12    end loop;
 13    end;
 14    /

PL/SQL procedure successfully completed.

SQL> select * from ar;

    RADIUS       AREA
---------- ----------
         5       78.5
         6     113.04
         7     153.86
         8     200.96
         9     254.34
```

Name :
Roll No :
Div : B

PRACTICAL NO. 06

Que.Problem Statement:: Named PL/SQL Block: PL/SQL Stored Procedure and Stored Function.

```
create table stud_marks(name varchar2(20),total_marks number(5));
create table result(roll number(3),name varchar2(20),class varchar2(20));

Table created.

SQL> create or replace procedure proc_grade(rno number,name varchar2,marks number)is
  2  class varchar2(20);
  3
  4  begin
  5  if(marks<=1500 and marks>=990)then
  6  class:='distinction';
  7  elsif(marks<=989 and marks>=900)then
  8  class:='first';
  9  elsif(marks<=899 and marks>=825)then
 10  class:='higher second';
 11  end if;
 12  insert into stud_marks values(name,marks);
 13  insert into result values(rno,name,class);
 14  end;
 15  /

Procedure created.

SQL> exec proc_grade(1,'ram',1100);
SQL> exec proc_grade(2,'shyam',967);
SQL> exec proc_grade(3,'rohan',865);

PL/SQL procedure successfully completed.

SQL> select * from stud_marks;

NAME                 TOTAL_MARKS
-------------------- -----------
ram                  1100
shyam                  967
rohan                 865
```

```
SQL> select * from result;
ROLL   NAME                  CLASS
---------------------------------------------
1     ram                  distinction
2     shyam                  first
3     rohan                higher second
```

Name :
Roll No :
Div : B

PRACTICAL NO:-07

Que. Cursors:(All types: Implicit, Explicit, Cursor FOR Loop, Parameterized Cursor)

```
SQL> conn  system
Enter password:
Connected.
SQL> create table O_ROLLCALL(ROLL_NO NUMBER(3), NAME VARCHAR2(20));

Table created.

SQL> INSERT INTO O_ROLLCALL values(1, 'Rohit');

1 row created.

SQL> INSERT INTO O_ROLLCALL values(1, 'Anisha');

1 row created.

SQL> INSERT INTO O_ROLLCALL values(3, 'Payal');

1 row created.

SQL> select * from O_ROLLCALL;

   ROLL_NO NAME
---------- --------------------
         1 Rohit
         1 Anisha
         3 Payal

SQL> update O_ROLLCALL set ROLL_NO=2 where NAME='Anisha';

1 row updated.

SQL> select * from O_ROLLCALL;

   ROLL_NO NAME
---------- --------------------
         1 Rohit
         2 Anisha
         3 Payal
```

```
SQL> create table N_ROLLCALL(roll_no number(3), name varchar2(20));

Table created.

SQL> insert into N_ROLLCALL values(1, 'Rohit');

1 row created.

SQL> select * from O_ROLLCALL;

   ROLL_NO NAME
---------- --------------------
         1 Rohit
         2 Anisha
         3 Payal

SQL> select * from N_ROLLCALL;

   ROLL_NO NAME
---------- --------------------
         1 Rohit

SQL> set serveroutput on;
SQL> declare
  2  cursor cur1 is
  3  select roll_no, name from O_ROLLCALL;
  4  cursor cur2 is
  5  select roll_no from N_ROLLCALL;
  6  r number(3);
  7  rno number(3);
  8  nm varchar2(20);
  9  begin
 10  open cur1;
 11  open cur2;
 12  loop
 13  fetch cur1 into rno, nm;
 14  fetch cur2 into r;
 15  exit when cur1%found=false;
 16  if r<> rno then
 17  insert into N_ROLLCALL values(rno, nm);
 18  end if;
 19  end loop;
 20  close cur1;
 21  end;
 22  /
```

```
PL/SQL procedure successfully completed.

SQL> select * from O_ROLLCALL;

   ROLL_NO NAME
---------- --------------------
         1 Rohit
         2 Anisha
         3 Payal

SQL> select * from N_ROLLCALL;

   ROLL_NO NAME
---------- --------------------
         1 Rohit
         2 Anisha
         3 Payal
```

Name :
Roll No :
Div : B

Practical No : 08

Que. Database Trigger (All Types: Row level and Statement level triggers, Before and After Triggers).

```
create table library(bno number(5),bname varchar2(20),author varchar2(20),allowed_days number(5));
Table created.

SQL> insert into library values (1,'java','Mr.Patil',10);
1 row created.

SQL> insert into library values (2,'UB','Mr.Sharma',15);
1 row created.

SQL> insert into library values (3,'CPP','Mr.Surve',15);
1 row created.

SQL> create table library_audit(bno number(5),o_days number(5),n_days number(5));
Table created.

SQL> create or replace trigger tr1
  2   before update or delete on library
  3   for each row
  4   begin
  5   insert into library_audit values(:new.bno,:old.allowed_days,:new.allowed_days);
  6   end;
  7   /
Trigger created.
SQL> select * from library;

       BNO BNAME                AUTHOR               ALLOWED_DAYS
---------- -------------------- -------------------- ------------
         1 java                 Mr.Patil                       10
         2 UB                   Mr.Sharma                      15
         3 CPP                  Mr.Surve                       15

SQL> update library set allowed_days=15 where bno=1;
1 row updated.

SQL> select * from library;

       BNO BNAME                AUTHOR               ALLOWED_DAYS
---------- -------------------- -------------------- ------------
         1 java                 Mr.Patil                       15
         2 UB                   Mr.Sharma                      15
         3 CPP                  Mr.Surve                       15

SQL> select * from library_audit;

       BNO     O_DAYS     N_DAYS
```

```
---------- ---------- ----------
         1         10         15

SQL> delete from library where bno=1;
1 row deleted.

SQL> select * from library_audit;

       BNO     O_DAYS     N_DAYS
---------- ---------- ----------
         1         10         15
                   15
```

PRACTICAL NO : 09

Name :

Roll No :

Problem Statement : Design and develop mongodb queries using CRUD operations.

```
test> show dbs

admin   40.00 KiB

config  60.00 KiB

local   40.00 KiB

test> use mydb

switched to db mydb

mydb> show dbs;

admin   40.00 KiB

config  60.00 KiB

local   40.00 KiB

mydb> db.createCollection("employees")

{ ok: 1 }

mydb> show collections

employees

mydb> db.employees.insert({"name":"vaishali","Emp_Id":"e100"})

DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.

{

  acknowledged: true,

  insertedIds: { '0': ObjectId("651f919cd8accb638978f371") }

}

mydb> show dbs;

admin    40.00 KiB

config  108.00 KiB

local    40.00 KiB

mydb     40.00 KiB

mydb> db.employees.find()
```

```
[
  {
    _id: ObjectId("651f919cd8accb638978f371"),
    name: 'vaishali',
    Emp_Id: 'e100'
  }
]
mydb>
db.employees.insert([{"name":"neha","Emp_Id":"e101"},{"name":"sweeti","Emp_Id":"e102"},{"nam
e":"smith","Emp_Id":"e103"},{"name":"rohit","Emp_Id":"e104"},{"name":"virat","Emp_Id":"e105"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("651f9419d8accb638978f372"),
    '1': ObjectId("651f9419d8accb638978f373"),
    '2': ObjectId("651f9419d8accb638978f374"),
    '3': ObjectId("651f9419d8accb638978f375"),
    '4': ObjectId("651f9419d8accb638978f376")
  }
}
mydb> db.employees.find()
[
  {
    _id: ObjectId("651f919cd8accb638978f371"),
    name: 'vaishali',
    Emp_Id: 'e100'
  },
  {
    _id: ObjectId("651f9419d8accb638978f372"),
    name: 'neha',
    Emp_Id: 'e101'
  },
  {
    _id: ObjectId("651f9419d8accb638978f373"),
    name: 'sweeti',
    Emp_Id: 'e102'
  },
  {
    _id: ObjectId("651f9419d8accb638978f374"),
    name: 'smith',
    Emp_Id: 'e103'
  },
  {
    _id: ObjectId("651f9419d8accb638978f375"),
    name: 'rohit',
    Emp_Id: 'e104'
  },
  {
    _id: ObjectId("651f9419d8accb638978f376"),
    name: 'virat',
    Emp_Id: 'e105'
  }
]

mydb> db.employees.find({$and:[{"name":"virat"},{"Emp_Id":"e105"}]})
[
  {
    _id: ObjectId("651f9419d8accb638978f376"),
    name: 'virat',
    Emp_Id: 'e105'
  }
]
mydb> db.employees.find({$or:[{"name":"virat"},{"Emp_Id":"e104"}]})
```

```
[
  {
    _id: ObjectId("651f9419d8accb638978f375"),
    name: 'rohit',
    Emp_Id: 'e104'
  },
  {
    _id: ObjectId("651f9419d8accb638978f376"),
    name: 'virat',
    Emp_Id: 'e105'
  }
]
mydb>
db.employees.update({Emp_Id:"e102"},{$set:{name:"KL_Rahul"},$currentDate:{lastupdated:true}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
mydb> db.employees.find()
[
  {
    _id: ObjectId("651f919cd8accb638978f371"),
    name: 'vaishali',
    Emp_Id: 'e100'
  },
  {
    _id: ObjectId("651f9419d8accb638978f372"),
    name: 'neha',
    Emp_Id: 'e101'
  },
  {
    _id: ObjectId("651f9419d8accb638978f373"),
    name: 'KL_Rahul',
    Emp_Id: 'e102',
    lastupdated: ISODate("2023-10-06T05:19:49.751Z")
  },
  {
    _id: ObjectId("651f9419d8accb638978f374"),
    name: 'smith',
    Emp_Id: 'e103'
  },
  {
    _id: ObjectId("651f9419d8accb638978f375"),
    name: 'rohit',
    Emp_Id: 'e104'
  },
  {
    _id: ObjectId("651f9419d8accb638978f376"),
    name: 'virat',
    Emp_Id: 'e105'
  }
]
mydb> db.employees.remove({name:"neha"})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany,
findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
mydb>
```

PRACTICAL NO : 10

Name :
Roll No :
Problem Statement : Implement aggregation and indexing with suitable example using mongodb

```
mydb> show dbs
admin    40.00 KiB
config  108.00 KiB
local    72.00 KiB
mydb    400.00 KiB
mydb> use comp
switched to db comp
comp> db.createCollection("employee")
{ ok: 1 }
comp>
db.employee.insertMany([{id:101,name:"vaishali",city:"mumbai",age:20,salary:25000},{id:102,na
me:"bhagyashri",city:"mumbai",age:20,salary:30000},{id:103,name:"purva",city:"delhi",age:19,sal
ary:40000},{id:104,name:"virat",city:"london",age:30,salary:25000},{id:105,name:"rohit",city:"del
hi",age:30,salary:28000},{id:106,name:"rahul",city:"pune",age:28,salary:35000}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6530bb5cfb41071db021be3d"),
    '1': ObjectId("6530bb5cfb41071db021be3e"),
    '2': ObjectId("6530bb5cfb41071db021be3f"),
    '3': ObjectId("6530bb5cfb41071db021be40"),
    '4': ObjectId("6530bb5cfb41071db021be41"),
    '5': ObjectId("6530bb5cfb41071db021be42")
  }
}
comp> db.employee.find()
[
  {
    _id: ObjectId("6530bb5cfb41071db021be3d"),
    id: 101,
    name: 'vaishali',
    city: 'mumbai',
    age: 20,
    salary: 25000
  },
  {
    _id: ObjectId("6530bb5cfb41071db021be3e"),
    id: 102,
    name: 'bhagyashri',
    city: 'mumbai',
    age: 20,
    salary: 30000
  },
  {
    _id: ObjectId("6530bb5cfb41071db021be3f"),
    id: 103,
    name: 'purva',
    city: 'delhi',
    age: 19,
    salary: 40000
  },
  {
    _id: ObjectId("6530bb5cfb41071db021be40"),
    id: 104,
    name: 'virat',
    city: 'london',
    age: 30,
    salary: 25000
  },
  {
    _id: ObjectId("6530bb5cfb41071db021be41"),
    id: 105,
    name: 'rohit',
    city: 'delhi',
    age: 30,
    salary: 28000
  },
  {
    _id: ObjectId("6530bb5cfb41071db021be42"),
    id: 106,
    name: 'rahul',
    city: 'pune',
    age: 28,
    salary: 35000
  }
]
comp> db.employee.aggregate([{$group:{_id:"$city"}},{$sort:{_id:-1}}])
[
  { _id: 'pune' },
  { _id: 'mumbai' },
  { _id: 'london' },
  { _id: 'delhi' }
]
comp> db.employee.aggregate([{$match:{salary:{$gt:25000}}},{$project:{_id:0,name:1,city:1}}])
[
  { name: 'bhagyashri', city: 'mumbai' },
  { name: 'purva', city: 'delhi' },
  { name: 'rohit', city: 'delhi' },
  { name: 'rahul', city: 'pune' }
]
comp> db.employee.aggregate([{$sort:{salary:1}},{$project:{_id:0,name:1,salary:1}},{$limit:3}])
[
  { name: 'vaishali', salary: 25000 },
  { name: 'virat', salary: 25000 },
  { name: 'rohit', salary: 28000 }
]
comp> db.employee.aggregate([{$sort:{salary:1}},{$project:{_id:0,name:1,salary:1}}])
[
  { name: 'vaishali', salary: 25000 },
  { name: 'virat', salary: 25000 },
  { name: 'rohit', salary: 28000 },
  { name: 'bhagyashri', salary: 30000 },
  { name: 'rahul', salary: 35000 },
  { name: 'purva', salary: 40000 }
]
```

PRACTICAL NO : 11

Name :
Roll No :
Problem Statement : Implementation of mapReduce operation with suitable example using mongodb

```
test> show dbs
admin    40.00 KiB
config   72.00 KiB
local    72.00 KiB
mydb    240.00 KiB
test> use mydb
switched to db mydb
mydb> show collections
employees
Employees
PDEA
student
Students
tushar
mydb> db.student.find()
[
  { _id: ObjectId("65277b64bd64de380e10da4e"), Name: 'Bhagyashri' },
  { _id: ObjectId("65277b64bd64de380e10da4f"), Name: 'Vaishali' }
]
mydb> db.createCollection("stud")
{ ok: 1 }
mydb>
db.stud.insertMany([{id:101,name:"vaishali",marks:88,age:20},{id:102,name:"bhagyashri",marks:8
9,age:20},{id:103,name:"purva",marks:78,age:19},{id:104,name:"virat",marks:68,age:30},{id:105,
name:"rohit",marks:91,age:30},{id:106,name:"rahul",marks:76,age:28}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6530ad4bfb41071db021be36"),
    '1': ObjectId("6530ad4bfb41071db021be37"),
    '2': ObjectId("6530ad4bfb41071db021be38"),
    '3': ObjectId("6530ad4bfb41071db021be39"),
    '4': ObjectId("6530ad4bfb41071db021be3a"),
    '5': ObjectId("6530ad4bfb41071db021be3b")
  }
}
mydb> db.stud.find()
[
  {
    _id: ObjectId("6530ad4bfb41071db021be36"),
    id: 101,
    name: 'vaishali',
    marks: 88,
    age: 20
  },
  {
    _id: ObjectId("6530ad4bfb41071db021be37"),
    id: 102,
    name: 'bhagyashri',
    marks: 89,
    age: 20
  },
  {
    _id: ObjectId("6530ad4bfb41071db021be38"),
    id: 103,
    name: 'purva',
    marks: 78,
    age: 19
  },
  {
    _id: ObjectId("6530ad4bfb41071db021be39"),
    id: 104,
    name: 'virat',
    marks: 68,
    age: 30
  },
  {
    _id: ObjectId("6530ad4bfb41071db021be3a"),
    id: 105,
    name: 'rohit',
    marks: 91,
    age: 30
  },
  {
    _id: ObjectId("6530ad4bfb41071db021be3b"),
    id: 106,
    name: 'rahul',
    marks: 76,
    age: 28
  }
]
mydb> var mapfunction=function(){emit(this.age,this.marks)}

mydb> var reducefunction=function(key,values){return Array.sum(values)}

mydb> db.stud.mapReduce(mapfunction,reducefunction,{'out':'Result1_mapReduce'})
DeprecationWarning: Collection.mapReduce() is deprecated. Use an aggregation instead.
See https://docs.mongodb.com/manual/core/map-reduce for details.
{ result: 'Result1_mapReduce', ok: 1 }

mydb> db.Result1_mapReduce.find()
[
  { _id: 20, value: 177 },
  { _id: 30, value: 159 },
  { _id: 19, value: 78 },
  { _id: 28, value: 76 }
]
```

```
mydb> db.stud.mapReduce(function () {emit(this.age, this.marks);}, function (key, values) { return
Array.avg(values);}, { query: { age: { $gt: 20 } }, out: 'Result2_mapReduce' })
{ result: 'Result2_mapReduce', ok: 1 }

mydb> db.Result2_mapReduce.find()
[ { _id: 30, value: 79.5 }, { _id: 28, value: 76 } ]
mydb>
```

PRACTICAL NO : 12

Name :
Roll No :
Problem Statement : Write a program to implement mysql/mongodb connectivity with java/php/python

```
import java.sql.*;
class MysqlConn{
public static void main(String args[]){
try{
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection("jdbc:mysql://lo-
calhost:3306/db","root1","surwase1");
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from emp");
while(rs.next())
System.out.println(rs.getInt(1)+" "+rs.getString(2)+"
"+rs.getString(3));
con.close();
}catch(Exception e){ System.out.println(e);}
}
}
```

OUTPUT :-

```
101  vaishali  survase
102  purva takale
103  bhagyashri shingade
```