

Practical No. 10

NAME-Shriharsh J. Deshmukh

ROLL NO-71

DIV-A

BATCH-A3

Aggregate Function

1) Sum

-> db.customer.aggregate([{ \$group : { _id: null, totalMarks: { \$sum : "\$marks" } } }])

O/P : [{ _id: null, totalMarks: 362 }]

2) AVG

-> db.customer.aggregate([{ \$group : { _id: null, avgMarks: { \$avg : "\$marks" } } }])

O/P : [{ _id: null, avgMarks: 90.5 }]

3) MIN

-> db.customer.aggregate([{ \$group : { _id: null, minMarks: { \$min : "\$marks" } } }])

O/P : [{ _id: null, minMarks: 85 }]

4) MAX

-> db.customer.aggregate([{ \$group : { _id: null, maxMarks: { \$max : "\$marks" } } }])

O/P : [{ _id: null, maxMarks: 98 }]

5) count

-> db.customer.aggregate([{ \$group : { _id: null, countMarks: { \$count : "\$marks" } } }])

O/P : [{ _id: null, countMarks: 4 }]

Indexing

1) To Create Index

-> db.customer.createIndex({"marks" : 1})

2) To See Indexes

-> db.customer.getIndexes()

Practical No. 8

NAME-Shriharsh J. Deshmukh

ROLL NO-71

DIV-A

BATCH-A3

// Connection.php

<?php

\$servername = "localhost";

\$username = "root";

\$password = "";

\$dbname = "DBMS";

\$conn = new mysqli(\$servername, \$username, \$password, \$dbname);

if (\$conn->connect_error) { die("Connection

failed: " . \$conn->connect_error);

}

?>

//index.php

<?php include

'connection.php';

?>

<!DOCTYPE html> <html

lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<link rel="stylesheet" href="styles.css">

<link rel="stylesheet"

```
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
```

```
<title>crud operation</title>
```

```
</head>
```

```
<body>
```

```
<table>
```

```
<tr>
```

```
<td>
```

```
<form action="backend.php" method="post">
```

```
<div>
```

```
<label>Full Name</label>
```

```
<input type="text" name="fullName" id="fullName">
```

```
</div>
```

```
<div>
```

```
<label>Roll No</label>
```

```
<input type="text" name="rollno" id="empCode">
```

```
</div>
```

```
<div>
```

```
<label>Department</label>
```

```
<input type="text" name="dept" id="salary">
```

```
</div>
```

```
<div>
```

```
<label>Division</label>
```

```
<input type="text" name="div" id="city">
```

```
</div>
```

```
<div class="form-action-buttons">
```

```
<input type="submit" value="Submit">
```

```
</div>
```

```
</form>
```

```
</td>
```

</tr>

</table>

<table>

<tr>

<td>

<table class="list" id="employeeList">

<thead>

<tr>

<th>Full Name</th>

<th>Roll No</th>

<th>Department</th>

<th>Division</th>

</tr>

</thead>

<tbody>

<?php

\$sql = "SELECT * FROM stu_details";

\$result = \$conn->query(\$sql);

if (\$result->num_rows > 0) { while (\$row = \$result->fetch_assoc()) { echo

"<tr><td>" . \$row["name"]. "</td><td>" . \$row["rollno"]. "</td><td>" . \$row["dept"].

"</td><td>" . \$row["division"]. "</td></tr>";

}

}

?>

</tbody>

</table>

</td>

```

</tr>
</table>
</body>
</html>
//backend.php
<?php include
'connection.php';
// Get data from form
$name = $_POST['fullName'];
$rollno = $_POST['rollno'];
$dept = $_POST['dept'];
$div = $_POST['div'];
$sql = "INSERT INTO stu_details VALUES ('$name', $rollno, '$dept', '$div')";
if ($conn->query($sql) === TRUE) { echo
'<script>alert("Record inserted successfully");</script>';
header("Location: index.php");
} else {
echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
//style.css
body > table{
width: 80%;
}
table{
border-collapse: collapse;
}
table.list{

```

```
width: 100%;
}
td, th{ border: 1px solid
#dddddd; text-align: left;
padding: 8px;
}
input[type=text], input[type=number] {
width: 100%; padding: 12px 20px;
margin: 8px 0px; display: inline-block; border: 1px solid #ccc;
border-radius: 4px; cursor: pointer;
}
form div.form-action-buttons{ text-align: right;
}
a{ cursor: pointer; text-decoration: underline;
color: #0000ee; margin-right: 4px;
}
.hide{
display: none;
}
```

Practical No.3

USE SHRIHARSH;

CREATE TABLE Departments

(DepartmentID int,

DepartmentName varchar(100),

PRIMARY KEY(DepartmentID));

CREATE TABLE Employees

(EmployeeID int,

FirstName varchar(100),

LastName varchar(100),

DepartmentID int,

PRIMARY KEY(EmployeeID),

FOREIGN KEY(DepartmentID) REFERENCES Departments(DepartmentID));

INSERT INTO Departments

VALUES

(101,'HR'),

(102,'Sales'),

(103,'IT'),

(104,'Marketing');

INSERT INTO Employees

VALUES

(1,'John','Smith',101),

(2,'Jane','Doe',102),

(3,'Michael','Johnson',101),

(4,'Emily','Williams',103);

-- ----- Queries ----- --

-- 1. Give a Cartesian Product of Employees and Departments. (Cartesian Product)

SELECT * FROM Employees CROSS JOIN Departments;

-- -->Inner Join<-- --

-- 2. Provide all details of employees whose department ID is greater than 101 along with their respective departments. (theta join)

SELECT * FROM Employees e INNER JOIN Departments d
ON e.DepartmentID = d.DepartmentID AND e.DepartmentID > 101;

-- 3. Give all details of employees. (Equi Join)

SELECT * FROM Employees e INNER JOIN Departments d
ON e.DepartmentID = d.DepartmentID;

-- 4. Give all details of employees using a Natural Join. (Natural Join)

SELECT * FROM Employees e NATURAL JOIN Departments d;

-- -->outer join<-- --

-- 5. Show all employees and their respective department details, if available. (Left Outer Join)

SELECT * FROM Employees e LEFT JOIN Departments d
ON e.DepartmentID = d.DepartmentID;

-- 6. Show all departments and their respective employees, if available. (Right Outer Join)

SELECT * FROM Employees e RIGHT JOIN Departments d
ON e.DepartmentID = d.DepartmentID;

-- 7. Give the full details of employees along with their respective departments. (Full Outer Join)

```
SELECT * FROM Employees e
```

```
LEFT JOIN Departments d ON e.DepartmentID = d.DepartmentID
```

```
UNION
```

```
SELECT * FROM Employees e
```

```
RIGHT JOIN Departments d ON e.DepartmentID = d.DepartmentID
```

```
WHERE e.EmployeeID IS NULL;
```

-- 8. Provide the first names and last names of HR employees without using a join. (Subquery)

```
SELECT FirstName, LastName FROM Employees
```

```
WHERE DepartmentID = (SELECT DepartmentID FROM Departments WHERE DepartmentName = 'HR');
```

-- 9. Create and display a view of the Employees table. (Views)

```
CREATE VIEW EmployeeView AS
```

```
SELECT e.EmployeeID, e.FirstName, e.LastName, d.DepartmentName
```

```
FROM Employees e
```

```
INNER JOIN Departments d ON e.DepartmentID = d.DepartmentID;
```

```
SELECT * FROM EmployeeView;
```

PRACTICAL NO. 6

CODE:

```
shriharsh@shriharsh-mac:~$ sudo mysql
[sudo] password for shriharsh:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.34-0ubuntu0.22.04.1 (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> USE SHRIHARSH;
Database changed
mysql> CREATE TABLE Old_Roll
-> (
-> Roll_No INT PRIMARY KEY,
-> Name VARCHAR(20)
-> );
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> CREATE TABLE New_Roll
-> (
-> Roll_No INT PRIMARY KEY,
-> Name VARCHAR(20)
-> );
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> INSERT INTO Old_Roll VALUES (01,"Rajesh"),(02,"Shubham");
Query OK, 2 rows affected (0.42 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO New_Roll VALUES (03,"Saurabh"),(02,"Shubham");
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

```

mysql> DELIMITER //
mysql> CREATE PROCEDURE Create_List()
-> BEGIN
->   DECLARE tmp_Roll INT;
->   DECLARE tmp_Name VARCHAR(20);
->   DECLARE New_Roll INT;
->
->   DECLARE done INT DEFAULT 0;
->   DECLARE cur_old CURSOR FOR SELECT * FROM Old_Roll;
->   DECLARE cur_new CURSOR FOR SELECT Roll_No FROM New_Roll;
->
->   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
->
->   OPEN cur_old;
->   read_loop: LOOP
->     FETCH cur_old INTO tmp_Roll, tmp_Name;
->     IF done THEN
->       LEAVE read_loop;
->     END IF;
->
->     OPEN cur_new;
->     FETCH cur_new INTO New_Roll;
->
->     IF New_Roll IS NOT NULL AND New_Roll = tmp_Roll THEN
->       SELECT 'Roll No.: ', New_Roll, ' Already Present';
->     ELSE
->       SELECT 'Roll No.: ', tmp_Roll, ' Inserted Successfully!';
->       INSERT INTO New_Roll VALUES (tmp_Roll, tmp_Name);
->     END IF;
->
->     CLOSE cur_new;
->   END LOOP;
->
->   CLOSE cur_old;
->   COMMIT;
-> END //

```

Query OK, 0 rows affected (0.02 sec)

```
mysql> DELIMITER ;
```

```
mysql> CALL Create_List();
```

```
+-----+-----+-----+
| Roll No.: | tmp_Roll | Inserted Successfully! |
+-----+-----+-----+
| Roll No.: |          1 | Inserted Successfully! |
+-----+-----+-----+
1 row in set (0.00 sec)
```

```
+-----+-----+-----+
| Roll No.: | tmp_Roll | Inserted Successfully! |
+-----+-----+-----+
| Roll No.: |          2 | Inserted Successfully! |
+-----+-----+-----+
1 row in set (0.01 sec)
```

ERROR 1062 (23000): Duplicate entry '2' for key 'New_Roll.PRIMARY'

```
mysql> Select * From Old_Roll;
```

```
+-----+-----+
| Roll_No | Name   |
+-----+-----+
|         1 | Rajesh |
|         2 | Shubham |
+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> Select * From New_Roll;
```

```
+-----+-----+
| Roll_No | Name   |
+-----+-----+
|         1 | Rajesh |
|         2 | Shubham |
|         3 | Saurabh |
+-----+-----+
3 rows in set (0.00 sec)
```

```
shriharsh@shriharsh-mac: ~  
shriharsh@shriharsh-mac:~$ sudo mysql  
[sudo] password for shriharsh:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 13  
Server version: 8.0.34-0ubuntu0.22.04.1 (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> USE SHRIHARSH;  
Database changed  
mysql> CREATE TABLE Old_Roll  
-> (  
-> Roll_No INT PRIMARY KEY,  
-> Name VARCHAR(20)  
-> );  
Query OK, 0 rows affected (0.08 sec)  
  
mysql> CREATE TABLE New_Roll  
-> (  
-> Roll_No INT PRIMARY KEY,  
-> Name VARCHAR(20)  
-> );  
Query OK, 0 rows affected (0.09 sec)  
  
mysql> INSERT INTO Old_Roll VALUES (01,"Rajesh"),(02,"Shubham");  
Query OK, 2 rows affected (0.42 sec)  
Records: 2  Duplicates: 0  Warnings: 0  
  
mysql> INSERT INTO New_Roll VALUES (03,"Saurabh"),(02,"Shubham");  
Query OK, 2 rows affected (0.02 sec)  
Records: 2  Duplicates: 0  Warnings: 0  
  
mysql> DELIMITER //  
mysql> CREATE PROCEDURE Create_List()  
-> BEGIN  
-> DECLARE tmp_Roll INT;  
-> DECLARE tmp_Name VARCHAR(20);  
-> DECLARE New_Roll INT;  
->  
-> DECLARE done INT DEFAULT 0;  
-> DECLARE cur_old CURSOR FOR SELECT * FROM Old_Roll;  
-> DECLARE cur_new CURSOR FOR SELECT Roll_No FROM New_Roll;  
->  
-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;  
->  
-> OPEN cur_old;  
-> read_loop: LOOP  
-> FETCH cur_old INTO tmp_Roll, tmp_Name;  
-> IF done THEN  
-> LEAVE read_loop;  
-> END IF;  
->  
-> OPEN cur_new;  
-> FETCH cur_new INTO New_Roll;  
->  
-> IF New_Roll IS NOT NULL AND New_Roll = tmp_Roll THEN  
-> SELECT 'Roll No.: ', New_Roll, ' Already Present';  
-> ELSE  
-> SELECT 'Roll No.: ', tmp_Roll, ' Inserted Successfully!';  
-> INSERT INTO New_Roll VALUES (tmp_Roll, tmp_Name);  
-> END IF;  
->  
-> CLOSE cur_new;  
-> END LOOP;  
->  
-> CLOSE cur_old;  
-> COMMIT;  
-> END //
```

```
Query OK, 0 rows affected (0.02 sec)  
  
mysql> DELIMITER ;  
mysql> CALL Create_List();  
+-----+  
| Roll No.: | tmp_Roll | Inserted Successfully! |  
+-----+  
| Roll No.: |      1 | Inserted Successfully! |  
+-----+  
1 row in set (0.00 sec)  
  
+-----+  
| Roll No.: | tmp_Roll | Inserted Successfully! |  
+-----+  
| Roll No.: |      2 | Inserted Successfully! |  
+-----+  
1 row in set (0.01 sec)  
  
ERROR 1062 (23000): Duplicate entry '2' for key 'New_Roll.PRIMARY'  
mysql> Select * From Old_Roll;  
+-----+  
| Roll_No | Name |  
+-----+  
|      1 | Rajesh |  
|      2 | Shubham |  
+-----+  
2 rows in set (0.00 sec)  
  
mysql> Select * From New_Roll;  
+-----+  
| Roll_No | Name |  
+-----+  
|      1 | Rajesh |  
|      2 | Shubham |  
|      3 | Saurabh |  
+-----+  
3 rows in set (0.00 sec)
```

PRACTICAL NO. 7

CODE:

```
shriharsh@shriharsh-mac:~$ sudo mysql
[sudo] password for shriharsh:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.34-0ubuntu0.22.04.1 (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> use SHRIHARSH;
Database changed
mysql> create table Lib_Master(Roll INT, Name varchar(30),Book_name varchar(30),status varchar(20));
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> create table Backup(Roll INT, Name varchar(30),Book_name varchar(30),status varchar(20));
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> desc Lib_Master;
```

Field	Type	Null	Key	Default	Extra
Roll	int	YES		NULL	
Name	varchar(30)	YES		NULL	
Book_name	varchar(30)	YES		NULL	
status	varchar(20)	YES		NULL	

4 rows in set (0.01 sec)

```
mysql> desc Backup;
```

Field	Type	Null	Key	Default	Extra
Roll	int	YES		NULL	
Name	varchar(30)	YES		NULL	
Book_name	varchar(30)	YES		NULL	
status	varchar(20)	YES		NULL	

4 rows in set (0.01 sec)

```
mysql> insert into Lib_Master values(01,"Shubham","Times","I"),(02,"Rajesh","DBMS","I");
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

Roll	Name	Book_name	status
1	Shubham	Times	I
2	Rajesh	DBMS	I

2 rows in set (0.00 sec)

```
mysql> select * from Backup;
Empty set (0.00 sec)
```

```
mysql> delimiter //
mysql> create trigger secure before update on Lib_Master for each row
-> begin
-> insert into Backup values(OLD.Roll, OLD.Name, OLD.Book_name, OLD.status);
-> end;
-> //
```

Query OK, 0 rows affected (0.03 sec)

```
mysql> delimiter ;
mysql> update Lib_Master set name="Saurabh" where Roll=2;
Query OK, 1 row affected (0.08 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> select * from Lib_Master;
```

Roll	Name	Book_name	status
1	Shubham	Times	I
2	Saurabh	DBMS	I

2 rows in set (0.00 sec)

```
mysql> select * from Backup;
```

Roll	Name	Book_name	status
2	Rajesh	DBMS	I

1 row in set (0.00 sec)

```
mysql> create table After_Update_Backup(Roll INT, Name varchar(30),Book_name varchar(30),status
-> varchar(20));
Query OK, 0 rows affected (0.15 sec)
```

```
mysql> delimiter //
```

```
mysql> create trigger newupdates after update on Lib_Master for each row
```

```
-> begin
```

```
-> insert into After_Update_Backup values(NEW.Roll, NEW.Name, NEW.Book_name, NEW.status);
```

```
-> end;
```

```
-> //
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> delimiter ;
```

```
mysql> insert into Lib_Master values(03,"Raj","Bright_Minds","I"),(04,"Ram","Updated_Life","I");
```

```
Query OK, 2 rows affected (0.02 sec)
```

```
Records: 2 Duplicates: 0 Warnings: 0
```

```
mysql> update Lib_Master set name="Rajesh" where Roll=3;
```

```
Query OK, 1 row affected (0.03 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> select * from After_Update_Backup;
```

Roll	Name	Book_name	status
3	Rajesh	Bright_Minds	I

1 row in set (0.00 sec)

```
shriharsh@shriharsh-mac: ~  
shriharsh@shriharsh-mac:~$ sudo mysql  
[sudo] password for shriharsh:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 18  
Server version: 8.0.34-0ubuntu0.22.04.1 (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> use SHRIHARSH;  
Database changed  
mysql> create table Lib_Master(Roll INT, Name varchar(30),Book_name varchar(30),status varchar(20));  
Query OK, 0 rows affected (0.09 sec)  
  
mysql> create table Backup(Roll INT, Name varchar(30),Book_name varchar(30),status varchar(20));  
Query OK, 0 rows affected (0.08 sec)  
  
mysql> desc Lib_Master;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| Roll  | int  | YES  |     | NULL    |       |  
| Name  | varchar(30) | YES  |     | NULL    |       |  
| Book_name | varchar(30) | YES  |     | NULL    |       |  
| status | varchar(20) | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.01 sec)
```



```
mysql> desc Backup;
+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+
| Roll  | int           | YES  |     | NULL    |       |
| Name  | varchar(30)   | YES  |     | NULL    |       |
| Book_name | varchar(30) | YES  |     | NULL    |       |
| status | varchar(20)   | YES  |     | NULL    |       |
+-----+
4 rows in set (0.01 sec)

mysql> insert into Lib_Master values(01,"Shubham","Times","I"),(02,"Rajesh","DBMS","I");
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> select * from Lib_Master;
+-----+
| Roll | Name      | Book_name | status |
+-----+
| 1    | Shubham  | Times     | I      |
| 2    | Rajesh   | DBMS      | I      |
+-----+
2 rows in set (0.00 sec)

mysql> select * from Backup;
Empty set (0.00 sec)

mysql> delimiter //
mysql> create trigger secure before update on Lib_Master for each row
-> begin
-> insert into Backup values(OLD.Roll, OLD.Name, OLD.Book_name, OLD.status);
-> end;
-> //
Query OK, 0 rows affected (0.03 sec)

mysql> delimiter ;
mysql> update Lib_Master set name="Saurabh" where Roll=2;
Query OK, 1 row affected (0.08 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from Lib_Master;
+-----+
| Roll | Name      | Book_name | status |
+-----+
| 1    | Shubham  | Times     | I      |
| 2    | Saurabh  | DBMS      | I      |
+-----+
2 rows in set (0.00 sec)

mysql> select * from Backup;
+-----+
| Roll | Name      | Book_name | status |
+-----+
| 2    | Rajesh   | DBMS      | I      |
+-----+
1 row in set (0.00 sec)

mysql> create table After_Update_Backup(Roll INT, Name varchar(30),Book_name varchar(30),status
-> varchar(20));
Query OK, 0 rows affected (0.15 sec)

mysql> delimiter //
mysql> create trigger newupdates after update on Lib_Master for each row
-> begin
-> insert into After_Update_Backup values(NEW.Roll, NEW.Name, NEW.Book_name, NEW.status);
-> end;
-> //
Query OK, 0 rows affected (0.02 sec)

mysql> delimiter ;
mysql> insert into Lib_Master values(03,"Raj","Bright_Minds","I"),(04,"Ram","Updated_Life","I");
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> update Lib_Master set name="Rajesh" where Roll=3;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from After_Update_Backup;
+-----+
| Roll | Name      | Book_name | status |
+-----+
| 3    | Rajesh   | Bright_Minds | I      |
+-----+
1 row in set (0.00 sec)
```

```
CREATE DATABASE SHRIHARSH;  
USE SHRIHARSH;  
CREATE TABLE HOTEL  
(  
Hotel_No int,  
HName varchar(100),  
City varchar(100),  
PRIMARY KEY(Hotel_No)  
);
```

```
CREATE TABLE ROOM  
(  
Room_No int,  
Hotel_No int,  
RType varchar(100),  
Price int  
);
```

```
CREATE TABLE BOOKING  
(  
Hotel_No int,  
Guest_No int,  
Date_From int,  
Date_To int,  
Room_No int  
);
```

```
CREATE TABLE GUEST  
(  
Guest_No int,  
Guest_Name varchar(100),  
Guest_Address varchar(100),  
PRIMARY KEY(Guest_No)  
);
```

```
DESC HOTEL;
```

```
INSERT INTO HOTEL  
VALUES
```

```
(01,'Damodar','Pune'),  
(02,'SilverOak','Mumbai'),  
(03,'RKHotel','Pune');
```

```
INSERT INTO ROOM  
VALUES  
(07,01,'Luxary',1200),  
(08,02,'Double',800),  
(09,01,'Single',200),  
(10,03,'Triple',1000);
```

```
INSERT INTO BOOKING  
VALUES  
(01,11,'07-08-23','10-08-23',07),  
(01,12,'08-09-23','09-09-23',08),  
(02,13,'10-09-23','12-09-23',09),  
(03,14,'11-09-23','15-09-23',10);
```

```
INSERT INTO GUEST  
VALUES  
(11,'Arjun','Delhi'),  
(12,'Dhananjay','Pune'),  
(13,'Aakrit','Nagpur'),  
(14,'Rohit','Nashik');
```

```
SELECT * FROM ROOM;  
SELECT * FROM HOTEL;  
SELECT * FROM BOOKING;  
SELECT * FROM GUEST;
```

-- New Day--

use SHRIHARSH;

ALTER TABLE booking

MODIFY COLUMN DATE_FROM DATE, MODIFY COLUMN DATE_TO DATE;

-- Updating Values in Table --

UPDATE booking

SET DATE_FROM = '2023-08-07'

WHERE Guest_No=11;

UPDATE booking

SET DATE_FROM = '2023-09-08'

```
WHERE Guest_No=12;
UPDATE booking
SET DATE_FROM = '2023-09-10'
WHERE Guest_No=13;
UPDATE booking
SET DATE_FROM = '2023-09-11'
WHERE Guest_No=14;
```

```
UPDATE booking
SET DATE_TO = '2023-08-10'
WHERE Guest_No=11;
UPDATE booking
SET DATE_TO = '2023-09-09'
WHERE Guest_No=12;
UPDATE booking
SET DATE_TO = '2023-09-12'
WHERE Guest_No=13;
UPDATE booking
SET DATE_TO = '2023-09-15'
WHERE Guest_No=14;
```

```
ALTER TABLE booking
ADD foreign key (Guest_No) references guest(Guest_No);
```

-- -- -- -- Queries -- -- -- --

```
-- What is the total revenue per night from all double rooms?
SELECT SUM(Price) from room
WHERE RType ='Double';
```

```
-- List the details of all rooms at the Damodar Hotel, including the name of the guest staying in
the room if the room is occupied.
SELECT * from hotel h INNER JOIN booking b
ON h.Hotel_No = b.Hotel_No
INNER JOIN guest g
ON b.Guest_No = g.Guest_No
WHERE HName = 'Damodar';
```

```
-- What is the average number of bookings for each hotel in September?
SELECT Hotel_No,AVG(Room_No) as Room_Average FROM booking
WHERE MONTH(DATE_FROM)>= 9 AND MONTH(DATE_FROM) <= 9
```

```
GROUP BY Hotel_No;
```

```
-- Create index on one of the field and show it's preformance in query
```

```
CREATE INDEX index_Hotel
```

```
on hotel(HName);
```

```
Select * FROM hotel
```

```
WHERE HName = 'Damodar';
```

```
-- List full details of all hotels
```

```
SELECT * from hotel h INNER JOIN booking b
```

```
ON h.Hotel_No = b.Hotel_No
```

```
INNER JOIN guest g
```

```
ON b.Guest_No = g.Guest_No
```

```
INNER JOIN room r
```

```
on b.Room_No = r.Room_No;
```

```
-- List full details of all hotels in Pune
```

```
SELECT * from hotel h INNER JOIN booking b
```

```
ON h.Hotel_No = b.Hotel_No
```

```
INNER JOIN guest g
```

```
ON b.Guest_No = g.Guest_No
```

```
INNER JOIN room r
```

```
on b.Room_No = r.Room_No
```

```
WHERE City = 'Pune';
```

```
-- Upadte the price of all rooms by 5%
```

```
UPDATE room SET Price = Price + Price*0.05;
```

```
select * from room;
```

```
-- List the number of rooms in each hotel in Mumbai.
```

```
Select h.HName, r.Room_No AS 'Rooms in Mumbai' FROM room r INNER JOIN hotel h
```

```
ON h.Hotel_No = r.Hotel_No
```

```
where r.Hotel_No = (SELECT Hotel_No FROM hotel
```

```
WHERE City = 'Mumbai');
```

```
-- List all Double or Single Rooms with a price below Rs.1000 per night in ASC order of price.
```

```
SELECT h.HName,r.Room_No,r.Price FROM room r INNER JOIN hotel h
```

```
ON h.Hotel_No = r.Hotel_No
```

```
WHERE RType ='Double' OR RType ='Single' AND Price <1000
```

```
ORDER BY Price ASC;
```

Practical No. 12

Name : Shriharsh J. Deshmukh

Div : A

Roll No : 71

Batch : A3

Index.js

```
const express = require('express'); const
mongoose = require('mongoose'); const
bodyParser = require('body-parser');
const app = express();

// Connect to MongoDB

mongoose.connect('mongodb://localhost/student_details', { useNewUrlParser: true,
useUnifiedTopology:
true });

const db = mongoose.connection;

db.on('error', console.error.bind(console, 'MongoDB connection error:')); db.once('open',
() => console.log('Connected to MongoDB'));

// Body parser middleware

app.use(bodyParser.urlencoded({ extended: true }));

// Serve HTML file app.get('/',
(req, res) => {
  res.sendFile(__dirname + '/index.html');
});

// Student model

const Student = mongoose.model('Student', {
name: String, rollno : Number, class :
String, div : String
});

// Add a student app.post('/addStudent', (req, res) =>
```

```
{ const { name, rollno, className, div } =  
req.body;  
  
const student = new Student({ name, rollno, className, div });  
student.save();  
});
```

```
// Start the server const
```

```
PORT = 3000;
```

```
app.listen(PORT, () => {  
  console.log(`Server started on http://localhost:${PORT}`);  
});
```

```
Index.html
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<meta http-equiv="X-UA-Compatible" content="ie=edge">
```

```
<link rel="stylesheet"
```

```
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
```

```
<title>crud operation</title>
```

```
</head>
```

```
<style>
```

```
body {
```

```
  font-family: Arial, sans-serif; text-align: center;
```

```
}
```

```
.container { max-width: 400px; margin:
```

```
50px auto; padding:
```

```
20px; border: 1px
```

```
solid #ccc; border-radius: 10px;
```

```
}  
  
input { width: 100%;  
padding: 10px; margin-bottom: 10px; box-sizing: border-box;  
}  
  
button { padding: 10px  
20px; background-color:  
#007bff; color: #fff;  
border: none; border-radius: 5px; cursor:  
pointer;  
}  
  
</style>  
  
<body>  
  
<div class="container">  
  
<h2>Student Information</h2>  
  
<form action="/addStudent" method="post">  
  
<label for="name">Name:</label>  
  
<input type="text" id="name" name = "name" required><br>  
  
<label for="rollno">Roll No:</label>  
  
<input type="text" id="rollno" name = "rollno" required><br>  
  
<label for="class">Class:</label>  
  
<input type="text" id="class" name = "className" required><br>  
  
<label for="division">Division:</label>  
  
<input type="text" id="division" name = "div" required><br>  
  
<button type="submit" value="submit">Submit</button>  
  
</form>  
  
</div>  
  
</body>  
  
</html>
```


Practical No. 11

NAME-Shriharsh J. Deshmukh

ROLL NO-71

DIV-A

BATCH-A3

1) To see all databases >> show dbs

2) To use admin database >> use admin

3) To Create student Collections >> db.createCollection("student");

4) To insert many records in student collection

```
>> db.student.insertMany(  
  [  
    {id:1, "sec":'A', marks:80},  
    {id:2, "sec":'A', marks:90},  
    {id:1, "sec":'B', marks:99},  
    {id:1, "sec":'B', marks:95},  
    {id:1, "sec":'C', marks:90}  
  ]  
);
```

5) Find the maximum mark in each section and store in Collection

maxInEachSec

```
>> db.student.mapReduce(  
  function(){emit(this.sec, this.marks); },  
  function(sec,marks){return Math.max.apply(null, marks);},  
  {out : "maxInEachSec"});
```

6) To See records in Collection maxInEachSec >> db.maxInEachSec.find();

O/P :

```
{ "_id" : "A", "value" : 90 }  
{ "_id" : "B", "value" : 99 }
```

```
{ "_id" : "C", "value" : 90 }
```

7) Find the minimum mark in each section and store in Collection

minInEachSec

```
>> db.student.mapReduce(  
function(){emit(this.sec, this.marks); },  
function(sec,marks){return Math.min.apply(null, marks);},  
{out : "minInEachSec"});
```

8) To See records in Collection minInEachSec

```
>> db.minInEachSec.find();  
O/P : { "_id" : "A", "value" : 80 }  
{ "_id" : "B", "value" : 95 }  
{ "_id" : "C", "value" : 90 }
```

9) Find the average mark in each section and store in Collection avgInEachSec

```
>> db.student.mapReduce(  
function(){emit(this.sec, this.marks); },  
function(sec,marks){return Array.avg(marks);},  
  
{out : "avgInEachSec"});
```

10) To See records in Collection avgInEachSec

```
>> db.avgInEachSec.find();  
O/P : { "_id" : "A", "value" : 85 }  
{ "_id" : "B", "value" : 97 }  
{ "_id" : "C", "value" : 90 }
```

11) Find the sum of mark in each section and store in Collection sumInEachSec

```
>> db.student.mapReduce(  
function(){emit(this.sec, this.marks); },  
function(sec,marks){return Array.sum(marks);},  
  
{out : "sumInEachSec"});
```

12)To See records in Collection sumInEachSec

```
>> db.sumInEachSec.find();
```

O/P: { "_id" : "A", "value" : 170 }

{ "_id" : "B", "value" : 194 }

{ "_id" : "C", "value" : 90 }

Experiment No.9

Codes:

Please enter a MongoDB connection string (Default: mongodb://localhost/): mongodb://localhost:27017
mongodb://localhost:27017

Current Mongosh Log ID: 653c2dcb57be28126f2e70b4

Connecting to:

mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.2

Using MongoDB: 7.0.2

Using Mongosh: 2.0.2

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

----- The server generated these startup warnings when booting

2023-10-26T11:53:51.347+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted -----

test> show dbs

admin 40.00 KiB

config 108.00 KiB

local 72.00 KiB

test> use SHRIHARSH

switched to db SHRIHARSH

SHRIHARSH> db.students.insertMany([{"name":"Ram",age:12},{name:"Shyam",age:13},{name:"Raj",age:14},{name:"Rajesh",age:15}])

```
{
  acknowledged: true,
  insertedIds:
```

```
{ '0': ObjectId("653c2fa357be28126f2e70b5"),
  '1': ObjectId("653c2fa357be28126f2e70b6"),
  '2': ObjectId("653c2fa357be28126f2e70b7"),
  '3': ObjectId("653c2fa357be28126f2e70b8")
}
```

SHRIHARSH> db.students.find()

```
[{ _id: ObjectId("653c2fa357be28126f2e70b5"), name: 'Ram', age: 12 },
{ _id: ObjectId("653c2fa357be28126f2e70b6"), name: 'Shyam', age: 13 },
{ _id: ObjectId("653c2fa357be28126f2e70b7"), name: 'Raj', age: 14 },
{ _id: ObjectId("653c2fa357be28126f2e70b8"), name: 'Rajesh', age: 15 }]
```

SHRIHARSH> db.students.updateOne({name:"Ram"},{\$set:{ID_Cards:{has_Voter_Id_Card:false,has_Aadhar_Card:true}}})

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0 }
```

SHRIHARSH> db.students.findOne({name:"Ram"})

```
{ _id: ObjectId("653c2fa357be28126f2e70b5"),
  name: 'Ram', age: 12,
  ID_Cards: { has_Voter_Id_Card: false, has_Aadhar_Card: true }
}
```

SHRIHARSH> db.students.deleteMany({age:13})

```
{ acknowledged: true, deletedCount: 1 }
```

SHRIHARSH> db.students.find()

```
[{ _id: ObjectId("653c2fa357be28126f2e70b5"),
  name: 'Ram', age: 12,
  ID_Cards: { has_Voter_Id_Card: false, has_Aadhar_Card: true } },
{ _id: ObjectId("653c2fa357be28126f2e70b7"), name: 'Raj', age: 14 },
{ _id: ObjectId("653c2fa357be28126f2e70b8"), name: 'Rajesh', age: 15}]
```

SHRIHARSH> db.students.find({\$and:[{age:{\$gte:12}},{age:{\$lte:13}}]})

```
[
  {
    _id: ObjectId("653c3e6c3448605a27aa665f"),
    name: 'Ram',
    age: 12,
    ID_Cards: { has_Voter_Id_Card: false, has_Aadhar_Card: true }
  }
]
```

```

mongosh mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Please enter a MongoDB connection string (Default: mongodb://localhost/): mongodb://localhost:27017
mongodb://localhost:27017
Current Mongosh Log ID: 653c2dc57be28126f2e70b4
Connecting to:      mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.2
Using MongoDB:      7.0.2
Using Mongosh:      2.0.2

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

-----
The server generated these startup warnings when booting
2023-10-26T11:53:51.347+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> show dbs
admin    40.00 KiB
config  108.00 KiB
local    72.00 KiB
test> use SHRIHARSH
switched to db SHRIHARSH
SHRIHARSH> db.students.insertMany([({name:"Ram",age:12},{name:"Shyam",age:13},{name:"Raj",age:14},{name:"Rajesh",age:15})])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("653c2fa357be28126f2e70b5"),
    '1': ObjectId("653c2fa357be28126f2e70b6"),
    '2': ObjectId("653c2fa357be28126f2e70b7"),
    '3': ObjectId("653c2fa357be28126f2e70b8")
  }
}
SHRIHARSH> db.students.find()
[
  { _id: ObjectId("653c2fa357be28126f2e70b5"), name: 'Ram', age: 12 },
  { _id: ObjectId("653c2fa357be28126f2e70b6"), name: 'Shyam', age: 13 },
  { _id: ObjectId("653c2fa357be28126f2e70b7"), name: 'Raj', age: 14 },
  {
    _id: ObjectId("653c2fa357be28126f2e70b8"),
    name: 'Rajesh',
    age: 15
  }
]
SHRIHARSH> db.students.updateOne({name:"Ram"},{$set:{ID_Cards:{has_Voter_Id_Card:false, has_Aadhar_Card:true}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
SHRIHARSH> db.students.findOne({name:"Ram"})
{
  _id: ObjectId("653c2fa357be28126f2e70b5"),
  name: 'Ram',
  age: 12,
  ID_Cards: { has_Voter_Id_Card: false, has_Aadhar_Card: true }
}
SHRIHARSH> db.students.deleteMany({age:13})
{ acknowledged: true, deletedCount: 1 }
SHRIHARSH> db.students.find()
[
  {
    _id: ObjectId("653c2fa357be28126f2e70b5"),
    name: 'Ram',
    age: 12,
    ID_Cards: { has_Voter_Id_Card: false, has_Aadhar_Card: true }
  },
  { _id: ObjectId("653c2fa357be28126f2e70b7"), name: 'Raj', age: 14 },
  {
    _id: ObjectId("653c2fa357be28126f2e70b8"),
    name: 'Rajesh',
    age: 15
  }
]
SHRIHARSH>

SHRIHARSH> db.students.find({$and:[{age:{$gte:12}},{age:{$lte:13}}]})
[
  {
    _id: ObjectId("653c3e6c3448605a27aa665f"),
    name: 'Ram',
    age: 12,
    ID_Cards: { has_Voter_Id_Card: false, has_Aadhar_Card: true }
  }
]
SHRIHARSH>

```

EXPERIMENT NO. 4 CODE:

```
shriharsh@shriharsh-mac:~$ sudo mysql
```

```
[sudo] password for shriharsh:
```

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

```
Your MySQL connection id is 11
```

```
Server version: 8.0.34-0ubuntu0.22.04.1 (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> USE SHRIHARSH;
```

```
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 Library_Master(Roll_no INT, Name VARCHAR(30), Date_of_Issue DATE,  
Name_of_Book VARCHAR(30), Status VARCHAR(2));
```

```
Query OK, 0 rows affected (6.33 sec)
```

```
mysql> CREATE TABLE FINE(Roll_no INT, Name VARCHAR(30), dt_Date DATE, Amt INT);
```

```
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> INSERT INTO Library_Master VALUES(1,"Rajesh","2023-10-01","Wide  
Sea","I"),(2,"Shubham","2023-08-02","The Metamorphosis","I"),(5,"Saurabh","2023-09-  
23","Grand Union","I"),(8,"Revati","2023-09-25","The House On Mango  
Street","I"),(4,"Sharmila","2023-09-10","The Lottery","I");
```

```
Query OK, 5 rows affected (0.18 sec)
```

```
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> desc Library_Master;
```

Field	Type	Null	Key	Default	Extra
Roll_no	int	YES		NULL	
Name	varchar(30)	YES		NULL	
Date_of_Issue	date	YES		NULL	
Name_of_Book	varchar(30)	YES		NULL	
Status	varchar(2)	YES		NULL	

5 rows in set (0.00 sec)

```
mysql> desc FINE;
```

Field	Type	Null	Key	Default	Extra
Roll_no	int	YES		NULL	
Name	varchar(30)	YES		NULL	
dt_Date	date	YES		NULL	
Amt	int	YES		NULL	

4 rows in set (0.01 sec)

```
mysql> SELECT * FROM Library_Master;
```

Roll_no	Name	Date_of_Issue	Name_of_Book	Status
1	Rajesh	2023-09-01	Wide Sea	I
2	Shubham	2023-08-02	The Metamorphosis	I
5	Saurabh	2023-08-03	Grand Union	I
8	Revati	2023-08-25	The House On Mango Street	I
4	Sharmila	2023-09-10	The Lottery	I

5 rows in set (0.00 sec)


```

mysql> Delimiter //
mysql> CREATE PROCEDURE U(R_Roll INT,R_Name VARCHAR(30))
  -> BEGIN
  -> Declare X integer;
  -> Declare continue handler for not found
  -> BEGIN
  -> SELECT 'NOT FOUND';
  -> END;
  -> SELECT datediff(curdate(),Date_of_Issue) INTO X FROM Library_Master WHERE
Roll_no=R_Roll;
  -> IF (X>15 && X<30) THEN INSERT INTO FINE VALUES(R_Roll,R_Name,curdate() ,(X*5));
  -> ELSEIF(X>30) THEN INSERT INTO FINE VALUES(R_Roll,R_Name,curdate() ,(X*50));
  -> END IF;
  -> UPDATE Library_Master SET Status='R' WHERE Roll_no=R_Roll;
  -> END;
  -> //
Query OK, 0 rows affected, 1 warning (0.02 sec)

```

```

mysql> SELECT * FROM FINE;
  -> //
Empty set (0.00 sec)
mysql> CALL U(1,'Rajesh');
  -> //
Query OK, 1 row affected (0.02 sec)
mysql> CALL U(2,'Shubham');
  -> //
Query OK, 1 row affected (0.02 sec)
mysql> CALL U(5,'Saurabh');
  -> //
Query OK, 1 row affected (0.02 sec)
mysql> CALL U(8,'Revati');
  -> //
Query OK, 1 row affected (0.02 sec)
mysql> CALL U(4,'Sharmila');
  -> //
Query OK, 1 row affected (0.02 sec)

```

mysql> Select * from FINE;

Roll_no	Name	dt_Date	Amt
2	Shubham	2023-10-06	3250
4	Sharmila	2023-10-06	130

2 rows in set (0.01 sec)

```
shriharsh@shriharsh-mac: ~  
[sudo] password for shriharsh:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 15  
Server version: 8.0.34-0ubuntu0.22.04.1 (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> USE SHRIHARSH;  
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 Library_Master(Roll_no INT, Name VARCHAR(30), Date_of_Issue DATE, Name_of_Book VARCHAR(30), Status VARCHAR(2));  
Query OK, 0 rows affected (0.12 sec)  
  
mysql> CREATE TABLE FINE(Roll_no INT, Name VARCHAR(30), dt_Date DATE, Amt INT);  
Query OK, 0 rows affected (0.12 sec)  
  
mysql> INSERT INTO Library_Master VALUES(1,"Rajesh","2023-10-01","Wide Sea","I"),(2,"Shubham","2023-08-02","The Metamorphosis","I"),(5,"Saurabh","2023-09-23","Grand Union","I"),(8,"Revati","2023-09-25","The House On Mango Street","I"),(4,"Sharmila","2023-09-10","The Lottery","I");  
Query OK, 5 rows affected (0.01 sec)  
Records: 5  Duplicates: 0  Warnings: 0  
  
mysql> desc Library_Master;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| Roll_no | int | YES | | NULL | |  
| Name | varchar(30) | YES | | NULL | |  
| Date_of_Issue | date | YES | | NULL | |  
| Name_of_Book | varchar(30) | YES | | NULL | |  
| Status | varchar(2) | YES | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.01 sec)  
  
mysql> desc FINE;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| Roll_no | int | YES | | NULL | |  
| Name | varchar(30) | YES | | NULL | |  
| dt_Date | date | YES | | NULL | |  
| Amt | int | YES | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)  
  
mysql> SELECT * FROM Library_Master;  
+-----+-----+-----+-----+-----+  
| Roll_no | Name | Date_of_Issue | Name_of_Book | Status |  
+-----+-----+-----+-----+-----+  
| 1 | Rajesh | 2023-10-01 | Wide Sea | I |  
| 2 | Shubham | 2023-08-02 | The Metamorphosis | I |  
| 5 | Saurabh | 2023-09-23 | Grand Union | I |  
| 8 | Revati | 2023-09-25 | The House On Mango Street | I |  
| 4 | Sharmila | 2023-09-10 | The Lottery | I |  
+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

```
mysql> Delimiter //
mysql> CREATE PROCEDURE U(R_Roll INT,R_Name VARCHAR(30))
-> BEGIN
-> Declare X integer;
-> Declare continue handler for not found
-> BEGIN
-> SELECT 'NOT FOUND';
-> END;
-> SELECT datediff(curdate(),Date_of_Issue) INTO X FROM Library_Master WHERE Roll_no=R_Roll;
-> IF (X>15 && X<30) THEN INSERT INTO FINE VALUES(R_Roll,R_Name,curdate() ,(X*5));
-> ELSEIF(X>30) THEN INSERT INTO FINE VALUES(R_Roll,R_Name,curdate() ,(X*50));
-> END IF;
-> UPDATE Library_Master SET Status='R' WHERE Roll_no=R_Roll;
-> END;
-> //
Query OK, 0 rows affected, 1 warning (0.02 sec)

mysql> SELECT * FROM FINE;
-> //
Empty set (0.00 sec)

mysql> CALL U(1,'Rajesh');
-> //
Query OK, 1 row affected (0.01 sec)

mysql> Call U(2,'Shubham');
-> //
Query OK, 1 row affected (1.79 sec)

mysql> Call U(5,'Saurabh');
-> //
Query OK, 1 row affected (0.07 sec)

mysql> Call U(8,'Revati');
-> //
Query OK, 1 row affected (0.01 sec)

mysql> Call U(4,'Sharmila');
-> //
Query OK, 1 row affected (0.04 sec)

mysql> Select * from FINE;
-> //
+-----+-----+-----+-----+
| Roll_no | Name   | dt_Date | Amt |
+-----+-----+-----+-----+
|      2 | Shubham | 2023-10-06 | 3250 |
|      4 | Sharmila | 2023-10-06 | 130 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Practical No.5 Code

```
shriharsh@shriharsh-mac:~$ sudo mysql
```

```
[sudo] password for shriharsh:
```

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

```
Your MySQL connection id is 8
```

```
Server version: 8.0.34-0ubuntu0.22.04.1 (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> USE SHRIHARSH;
```

```
Database changed
```

```
mysql> CREATE TABLE Stud_Marks(Name VARCHAR(10),Total_marks INT, Roll INT);
```

```
Query OK, 0 rows affected (15.88 sec)
```

```
mysql> CREATE TABLE Result(Roll INT,Name VARCHAR(10), Class VARCHAR(30));
```

```
Query OK, 0 rows affected (0.22 sec)
```

```
mysql> desc Stud_Marks;
```

Field	Type	Null	Key	Default	Extra
Name	varchar(10)	YES		NULL	
Total_marks	int	YES		NULL	
Roll	int	YES		NULL	

```
2 rows in set (0.07 sec)
```

```
mysql> desc Result;
```

Field	Type	Null	Key	Default	Extra
Roll	int	YES		NULL	
Name	varchar(10)	YES		NULL	
Class	varchar(30)	YES		NULL	

```
3 rows in set (0.00 sec)
```

```
mysql> INSERT INTO Stud_Marks VALUES('Rajesh',991,01),('Shubham',901,02),('Saurabh',826,03);
```

```
-> //
```

```
Query OK, 3 rows affected (0.01 sec)
```

```
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT * FROM Stud_Marks;
```

```
-> //
```

Name	Total_marks	Roll
Rajesh	991	1
Shubham	901	2
Saurabh	826	3

3 rows in set (0.00 sec)

```
mysql> delimiter //
```

```
mysql> CREATE PROCEDURE proc_Grade(IN Total_marks INT,OUT Class VARCHAR(30))
```

```
-> BEGIN IF(Total_marks<=1500 && Total_marks>=990) THEN SET Class="Distinction";
```

```
-> ELSEIF(Total_marks<=989 && Total_marks>=900) THEN SET Class="First Class";
```

```
-> ELSEIF(Total_marks<=899 && Total_marks>=825) THEN SET Class="Higher Second Class";
```

```
-> ELSE SET Class="Fail";
```

```
-> END IF;
```

```
-> END;
```

```
-> //
```

Query OK, 0 rows affected, 3 warnings (0.72 sec)

```
mysql> SET GLOBAL log_bin_trust_function_creators=1;
```

```
-> //
```

Query OK, 0 rows affected, 1 warning (0.00 sec)

```
mysql> CREATE FUNCTION F4R(X INT) RETURNS INT
```

```
-> BEGIN DECLARE tmp_Marks INT; DECLARE tmp_Name VARCHAR(10); DECLARE tmp_Grade  
VARCHAR(10);
```

```
-> SELECT Stud_Marks.Total_marks, Stud_Marks.Name INTO tmp_Marks,tmp_Name FROM  
Stud_Marks WHERE Stud_Marks.Roll=X;
```

```
-> CALL proc_Grade(tmp_Marks,@tmp_Grade);
```

```
-> INSERT INTO Result VALUES(X,tmp_Name,@tmp_Grade);
```

```
-> RETURN X;
```

```
-> END;
```

```
-> //
```

Query OK, 0 rows affected (0.02 sec)

```
mysql> SELECT F4R(1);
```

```
-> //
```

F4R(1)
1

1 row in set (0.06 sec)

```
mysql> SELECT F4R(2);
```

```
-> //
```

F4R(2)
2

```
1 row in set (0.01 sec)
```

```
mysql> SELECT F4R(3); //
```

F4R(3)
3

```
1 row in set (0.01 sec)
```

```
mysql> SELECT * FROM Result;
```

```
-> //
```

Roll	Name	Class
1	Rajesh	Distinction
2	Shubham	First Class
3	Saurabh	Higher Second Class

```
3 rows in set (0.00 sec)
```

```
*****
```

```
shriharsh@shriharsh-mac: ~  
shriharsh@shriharsh-mac:~$ sudo mysql  
[sudo] password for shriharsh:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 10  
Server version: 8.0.34-0ubuntu0.22.04.1 (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> USE SHRIHARSH;  
Database changed  
mysql> CREATE TABLE Stud_Marks(Name VARCHAR(10),Total_marks INT, Roll INT);  
Query OK, 0 rows affected (13.67 sec)  
  
mysql> CREATE TABLE Result(Roll INT,Name VARCHAR(10), Class VARCHAR(30));  
Query OK, 0 rows affected (16.40 sec)  
  
mysql> desc Stud_Marks;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| Name  | varchar(10) | YES | | NULL | |  
| Total_marks | int | YES | | NULL | |  
| Roll  | int | YES | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

```

mysql> desc Result;
+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Roll  | int           | YES  |     | NULL    |       |
| Name  | varchar(10)   | YES  |     | NULL    |       |
| Class | varchar(30)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> INSERT INTO Stud_Marks VALUES('Rajesh',991,01),('Shubham',901,02),('Saurabh',826,03);
Query OK, 3 rows affected (2.28 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM Stud_Marks;
+-----+-----+-----+
| Name   | Total_marks | Roll |
+-----+-----+-----+
| Rajesh |          991 |    1 |
| Shubham |          901 |    2 |
| Saurabh |          826 |    3 |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> delimiter //
mysql> CREATE PROCEDURE proc_Grade(IN Total_marks INT,OUT Class VARCHAR(30))
-> BEGIN IF(Total_marks<=1500 && Total_marks>=990) THEN SET Class="Distinction";
-> ELSEIF(Total_marks<=989 && Total_marks>=900) THEN SET Class="First Class";
-> ELSEIF(Total_marks<=899 && Total_marks>=825) THEN SET Class="Higher Second Class";
-> ELSE SET Class="Fail";
-> END IF;
-> END;
-> //
Query OK, 0 rows affected, 3 warnings (2.64 sec)

mysql> SET GLOBAL log_bin_trust_function_creators=1;
-> //
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> CREATE FUNCTION F4R(X INT) RETURNS INT
-> BEGIN DECLARE tmp_Marks INT; DECLARE tmp_Name VARCHAR(10); DECLARE tmp_Grade VARCHAR(10);
-> SELECT Stud_Marks.Total_marks, Stud_Marks.Name INTO tmp_Marks,tmp_Name FROM Stud_Marks WHERE Stud_Marks.Roll=X;
-> CALL proc_Grade(tmp_Marks,@tmp_Grade);
-> INSERT INTO Result VALUES(X,tmp_Name,@tmp_Grade);
-> RETURN X;
-> END;
-> //
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT F4R(1);
-> //
+-----+
| F4R(1) |
+-----+
|      1 |
+-----+
1 row in set (0.02 sec)

mysql> SELECT F4R(2);
-> //
+-----+
| F4R(2) |
+-----+
|      2 |
+-----+
1 row in set (0.01 sec)

mysql> SELECT F4R(3);
-> //
+-----+
| F4R(3) |
+-----+
|      3 |
+-----+
1 row in set (1.24 sec)

mysql> SELECT * FROM Result;
-> //
+-----+-----+-----+
| Roll | Name   | Class
+-----+-----+-----+
|    1 | Rajesh | Distinction
|    2 | Shubham | First Class
|    3 | Saurabh | Higher Second Class
+-----+-----+-----+
3 rows in set (0.00 sec)

```