Partical 2 :-

create database PL;

use PL;

create table Employee(Emp\_id int primary key, Emp\_name varchar(20),Emp\_Address varchar(20), Emp\_Salary int);

insert into Employee values(1,'Bhumika','Pune',60000);

insert into Employee values(2,'Akanksha','Delhi',90000);

insert into Employee values(3,'Priya','Mumbai',20000);

insert into Employee values(4,'Kunal','Pimpari',80000);

insert into Employee values(5,'Harsh','Kolkata',10000);

insert into Employee values(6,'Akku','Gujarat',5000);

insert into Employee values(7,'Bittu','Solapure',4000);

select\*from Employee;

delete from Employee where Emp\_id=6;

update Employee set Emp\_salary=8000 where Emp\_id=7;

select min(Emp\_salary),max(Emp\_salary),avg(Emp\_salary),sum(Emp\_salary) from Employee;

SELECT Emp\_Salary FROM Employee ORDER BY Emp\_Salary ASC;

SELECT Emp\_Salary FROM Employee ORDER BY Emp\_Salary Desc;

select\*from Employee where Emp\_Salary < 5000;

select\*from Employee where Emp\_Salary > 5000;

select\*from Employee where Emp\_Salary between 5000 and 6000;

select\*from Employee where Emp\_Salary not between 5000 and 6000;

select\*from Employee where Emp\_Salary > 5000 and Address='Pune';

SELECT \* FROM Employee WHERE Emp\_Salary > 5000 AND Emp\_Address = 'Pune';

SELECT \* FROM Employee WHERE Emp\_Salary < 5000 AND Emp\_Address = 'Pune';

SELECT \* FROM Employee WHERE Emp\_Salary > 5000 OR Emp\_Address = 'Pune';

SELECT \* FROM Employee WHERE Emp\_Salary < 5000 OR Emp\_Address = 'Pune';

select\*from Employee where not Emp\_Address='pune';

create table Bus(bus\_no int primary key, source varchar(40), Distanaction varchar(40), Coach\_Type varchar (40));

alter table Bus;

ALTER TABLE Bus DROP COLUMN Coach\_Type;

select\*From bus;

ALTER TABLE Bus RENAME TO Bus1;

insert into Bus1 values(1, "hadapsar" , "Jejuri");

select\*From Bus1;

Alter Table Bus1 modify column Source varchar(20);

desc bus1;

drop table Bus1;

Prac3

create database PL;

use PL;

create table emp (emp\_id int primary key , emp\_name varchar(20),dept\_id int,emp\_location varchar(20));

desc emp;

insert into emp value(1,'Bhumika',101,'Pune');

insert into emp value(2, 'Amit', 1200, 'Mumbai');

insert into emp value(3, 'Raj', 5000, 'Delhi');

insert into emp value(4, 'Priya', 4500, 'Bangalore');

insert into emp value(5, 'Kiran', 3500, 'Chennai');

insert into emp value(6, 'Simran', 5500, 'Kolkata');

insert into emp value(7, 'Ravi', 4000, 'Hyderabad');

select\*from emp;

create table Dept(dept\_id int primary key , dept\_name varchar(20), dept\_emp varchar (20));

desc Dept;

insert into Dept value(101,'comp','bhumika');

insert into Dept value(1200, 'IT', 'Amit');

insert into Dept value(5000, 'HR', 'Raj');

insert into Dept value(4500, 'Finance', 'Priya');

insert into Dept value(105, 'Marketing', 'Kiran');

insert into Dept value(106, 'Sales', 'Simran');

insert into Dept value(107, 'R&D', 'Ravi');

insert into Dept value(108, 'Operations', 'Neha');

select\*from Dept;

select emp.emp\_id,emp.emp\_name,Dept.dept\_id,Dept.dept\_name from emp inner join Dept on emp . dept\_id=Dept.dept\_id order by Dept.dept\_id;

select\*from emp left outer join Dept on emp .emp\_id=Dept.dept\_id union select\*from emp right outer join Dept on emp.dept\_id=Dept.dept\_id;

select\*from emp left outer join Dept on emp.dept\_id=Dept.dept\_id;

select\*from emp right outer join Dept on emp.dept\_id=Dept.dept\_id;

SELECT \* FROM emp LEFT OUTER JOIN Dept ON emp.emp\_id = Dept.dept\_id UNION SELECT \* FROM emp RIGHT OUTER JOIN Dept ON emp.dept\_id = Dept.dept\_id;

Prac 4&5

create database LB;

use LB;

create table borrower(Roll\_no int, BName varchar(20), DateofIssue date, NameofBook varchar(30), Status char(1));

create table fine (Roll\_no int,CurrDate date,Amt int);

alter table borrower add primary key (Roll\_no);

insert into borrower values(1,'bhumika','2023-08-01','DBMS','I');

insert into borrower values(2,'Sonali','2023-07-11','MYSQL','I'),(3,'Pragati','2023-08-15','PL/SQL','I');

select \* from borrower;

select \* from fine;

delimiter $

create procedure calfine(Rollno int(10), Bname1 varchar(30))

begin

declare Idate date;

declare fine int(20);

declare day int(20);

select DateofIssue into Idate from borrower where Roll\_no=Rollno and

Bname=Bname1;

set day=DATEDIFF(CURDATE(),Idate);

IF(day>=15 and day<=30) then

set fine=day\*5;

ELSEIF(day>30)then

set fine=day\*50;

end IF;

update borrower set status='R' where Roll\_no=Rollno and Bname=Bname1;

IF (fine is not null) then

insert into fine values (Rollno, CURDATE(),fine);

end IF;

end $

call calfine(3,'Pragati')$

select \* from fine$

select \* from borrower$

Prac6

create database PL;

use PL;

create table stud\_marks(roll\_no int primary key,sname varchar(30),total\_marks int);

create table result(roll\_no int,class varchar(30));

insert into stud\_marks values(1,'BHU',930),(2,'Sona',1130),(3,'harsh',950),(4,'Tan',840),(5,'Kartik',1000),(6,'Aryan',860);

select \* from stud\_marks;

delimiter $

create procedure credit (IN roll int)

begin

declare m int;

declare c varchar(20);

select total\_marks into m from stud\_marks where roll\_no=roll;

if m>=990 and m <=1500 then

set c='Distinction';

insert into result values (roll,c);

elseif m>=900 and m<=989 then

set c='First Class';

insert into result values(roll,c);

elseif m>=825 and m<=899 then

set c='Higher Second Class';

insert into result values(roll,c);

end if;

end $

create function disp\_grade2(roll1 int)

RETURNS varchar(20)

DETERMINISTIC

READS SQL DATA

begin

declare m1 int;

declare c1 varchar(20);

select total\_marks into m1 from stud\_marks where roll1=roll\_no;

if m1>=990 and m1<=1500 then

set c1='Distinction';

elseif m1>=900 and m1<=989 then

set c1='First Class';

elseif m1>=825 and m1<=899 then

set c1='Higher Second Class';

end if;

return c1;

end $

select disp\_grade2(1);

select disp\_grade2(2);

select disp\_grade2(3);

Prac 7

create database cur1;

use cur1;

create table o\_rollcall(roll int,name varchar(10));

create table n\_rollcall(roll int,name varchar(10));

insert into o\_rollcall values (63,'BHUMIKA'),(65,'SONALI'),(66,'Pragati'),(72,'Diksha');

insert into n\_rollcall values (64,'HARSH'),(68,'KArtik'),(65,'Aryan'),(60,'Yash') ,(63,'Abhi');

select \* from o\_rollcall;

select \* from n\_rollcall;

Delimiter $

create procedure roll\_list()

begin

declare oldrollno int;

declare oldname varchar(10);

declare newrollno int;

declare newname varchar(10);

declare done int default false;

declare c1 cursor for select roll,name from o\_rollcall;

declare c2 cursor for select roll,name from n\_rollcall;

declare continue handler for not found set done=true;

open c1;

loop1:loop

fetch c1 into oldrollno,oldname;

if done then

leave loop1;

end if;

open c2;

loop2:loop

fetch c2 into newrollno,newname;

if done then

insert into n\_rollcall values(oldrollno,oldname);

set done=false;

close c2;

leave loop2;

end if;

end loop;

end loop;

close c1;

end ;$

call roll\_list();

select \* from n\_rollcall;

Prac8

create database cur1;

use cur1;

CREATE TABLE Library (book\_id INT AUTO\_INCREMENT PRIMARY KEY,title VARCHAR(255),author VARCHAR(255),publication\_year INT);

CREATE TABLE Library\_Audit (

audit\_id INT AUTO\_INCREMENT PRIMARY KEY,

book\_id INT,

title VARCHAR(255),

author VARCHAR(255),

publication\_year INT,

action VARCHAR(50), -- "DELETE" or "UPDATE"

audit\_timestamp TIMESTAMP

);

INSERT INTO Library (title, author, publication\_year) VALUES('Book 1', 'Author A',2000),('Book 2', 'Author B', 2010),('Book 3', 'Author C', 2020);

select\*From Library;

select\*from Library\_Audit;

DELIMITER $

CREATE TRIGGER BeforeLibraryDelete

BEFORE DELETE ON Library FOR EACH ROW

BEGIN

INSERT INTO Library\_Audit (book\_id, title, author, publication\_year, action,

audit\_timestamp)VALUES (OLD.book\_id, OLD.title, OLD.author, OLD.publication\_year,

'DELETE', NOW());

END; $

DELIMITER $

CREATE TRIGGER BeforeLibraryUpdate

BEFORE UPDATE ON Library FOR EACH ROW

BEGIN

INSERT INTO Library\_Audit (book\_id, title, author, publication\_year, action,

audit\_timestamp)VALUES (OLD.book\_id, OLD.title, OLD.author, OLD.publication\_year,

'UPDATE', NOW());

END$

-- Create an AFTER DELETE trigger

DELIMITER $

CREATE TRIGGER AfterLibraryDelete

AFTER DELETE ON Library

FOR EACH ROW

BEGIN

INSERT INTO Library\_Audit (book\_id, title, author, publication\_year, action, audit\_timestamp)

VALUES (OLD.book\_id, OLD.title, OLD.author, OLD.publication\_year, 'DELETE', NOW());

END $

-- Create an AFTER UPDATE trigger

DELIMITER $

CREATE TRIGGER AfterLibraryUpdate

AFTER UPDATE ON Library

FOR EACH ROW

BEGIN

INSERT INTO Library\_Audit (book\_id, title, author, publication\_year, action, audit\_timestamp)

VALUES (NEW.book\_id, NEW.title, NEW.author, NEW.publication\_year, 'UPDATE', NOW());

END $

DELIMITER $

UPDATE Library

SET title = 'Updated Book', author = 'New Author'

WHERE book\_id = 1;

DELETE FROM Library WHERE book\_id = 2;

SELECT \* FROM Library\_Audit;

Prac10

use stud

switched to db stud

db.createCollection("stud")

db.stud.insert({rno:1,name:'Bhumika'})

db.stud.find();

db.stud.insert({rno:2,name:'Akanksha'})

db.stud.find();

db.stud.insert({rno:3,name:'Ram'})

db.stud.find();

db.stud.find({rno:{$in:[1,3]}});

db.stud.update({rno:1},{$set:{name:'king'}})

db.stud.find();

db.stud.remove({rno:1});

db.stud.find();

db.stud.drop();

Prac11

db.createCollection("stud")

db.stud.insert({Rno:1,Name:"G",Address:"Pune",Marks:78})

db.stud.find()

db.stud.insert({Rno:2,Name:"S",Address:"Mumbai",Marks:85})

db.stud.insert({Rno:3,Name:"H",Address:"Nagpur",Marks:90})

db.stud.insert({Rno:4,Name:"F",Address:"Pune",Marks:56})

db.stud.insert({Rno:5,Name:"K",Address:"Sangli",Marks:85})

db.stud.find()

db.stud.aggregate([{ $project: {Rno: 1}}]);

db.stud.aggregate({$project:{Name:1}})

db.stud.aggregate({$project:{Name:0}})

db.stud.aggregate({$sort:{Name:- 1}})

db.stud.aggregate({$sort:{Rno:- 1}})

db.stud.aggregate({$limit:4},{$sort:{Rno:1}})

db.stud.aggregate({$limit:2},{$sort:{Rno:- 1}})

db.stud.aggregate([{$match:{}}])

db.stud.aggregate([{$match:{"Address":"Pune"}}])

db.stud.aggregate([{$group:{"\_id":"$Name"}}])

Prac12

db.createCollection("College")

db.College.insert({Rno:1,Name:"H",Subject:"DBMS",Marks:78})

db.College.insert({Rno:2,Name:"G",Subject:"CNS",Marks:90})

db.College.insert({Rno:1,Name:"L",Subject:"IOT",Marks:85})

db.College.insert({Rno:2,Name:"T",Subject:"TOC",Marks:56})

db.College.find()

var mapfunction=function(){emit(this.Rno,this.Marks);}

var reduce=function(Rno,Marks){return Array.sum(Marks);}

db.College.mapReduce(mapfunction,reduce,{out:"Result1"})

show tables