

20MCA134-ADVANCED DBMS LABORATORY RECORD

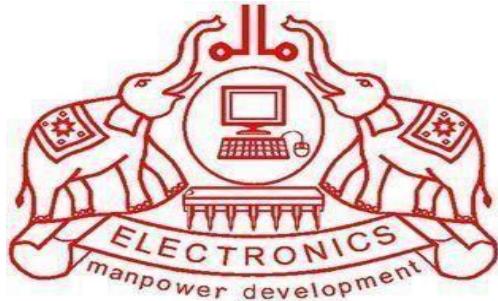
*Submitted in partial fulfillment of the requirement for the award of
Master of Computer Applications*

at

COLLEGE OF ENGINEERING POONJAR

Managed by I.H.R.D A Govt. of Kerala undertaking

(Affiliated to APJ Abdul Kalam Technological University)



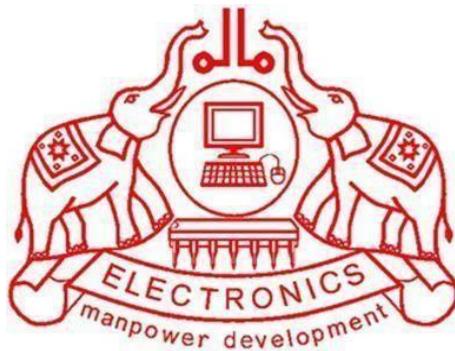
SUBMITTED BY
SHAMNA T S(PJR24MCA-2018)

DEPARTMENT OF COMPUTER SCIENCE
COLLEGE OF ENGINEERING POONJAR

COLLEGE OF ENGINEERING POONJAR

Managed by I.H.R.D., A Govt. Of Kerala undertaking

(Affiliated to APJ Abdul Kalam Technological University)



CERTIFICATE

Certified that this is a Bonafide record of practical work done in Advanced DBMS Lab (20MCA134) Laboratory by **SHAMNA T S** RegNo.**PJR24MCA-2018** of College of Engineering, Poonjar during the academic year 2024- 2026.

Dr. Annie Julie Joseph

Thasni Noushad

Head of the Department

Staff member in Charge

Submitted to the University Examination held on:

INTERNAL EXAMINAR

EXTERNAL EXAMINAR

CONTENTS

Sl. no	List of Experiments	Pg. no
1	SQL Query operations on Employee Table	1
2	SQL Query operations on tables Sailors, Reserves, Boats	13
3	SQL Query operations on table Customers, Orders, Salesman	19
4	DCL and TCL Operations	23
5	Create a View Virtual emp and its operations	27
6	Join Operations	30
7	PL/SQL Program to Hello World	32
8	PL/SQL program to find sum of two numbers using Function	33
9	PL/SQL program to find sum of two numbers using procedure	34
10	PL/SQL program to find factorial of a number using Procedure	35
11	PL/SQL program to find student info using Cursor	36
12	PL/SQL program to Check age before insert on people using Trigger	37
13	Mongo db CURD Operations	38
14	Aggregate Operations	47
15	Create users and Roles	54
16	Python Mongo db Connection	55
17	Sorting	57

SET-1

AIM

Create an employee table ‘EMP’ with following fields: empno NUMBER(4) ename VARCHAR2(25) jobVARCHAR2(12)salaryNUMBER(10,2)commissionNUMBER(7,2)dept no NUMBER(2)

```
mysql> create table emp(empno int primary key,ename varchar (20), job
varchar(50),salary int,commision int,deptno int);
Query OK, 0 rows affected (0.30 sec)
```

1. Display the structure of ‘EMP’

```
mysql> desc emp;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| empno | int(11) | NO | PRI | NULL |       |
| ename | varchar(20) | YES |     | NULL |       |
| job | varchar(50) | YES |     | NULL |       |
| salary | int(11) | YES |     | NULL |       |
| commision | int(11) | YES |     | NULL |       |
| deptno | int(11) | YES |     | NULL |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

2. Insert the following record into ‘EMP’

```
mysql> insert into emp values(7369,"smith","clerk",2000,800,20);
Query OK, 1 row affected (0.04 sec)

mysql> select * from emp;
+-----+-----+-----+-----+-----+
| empno | ename | job | salary | commision | deptno |
+-----+-----+-----+-----+-----+
| 7369 | smith | clerk | 2000 | 800 | 20 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

3. Insert the rest of the records using the substitution variable.

```
mysql> insert into emp values(7499,"allen","salesman",1600,300,30);
Query OK, 1 row affected (0.03 sec)

mysql> insert into emp values(7521,"ward","salesman",1250,500,30);
Query OK, 1 row affected (0.03 sec)

mysql> insert into emp values(7566,"jones","manager",2975,500,20);
Query OK, 1 row affected (0.02 sec)

mysql> insert into emp values(7654,"martin","salesman",1250,1400,30);
Query OK, 1 row affected (0.03 sec)

mysql> insert into emp values(7698,"blake","manager",2850,NULL,30);
Query OK, 1 row affected (0.03 sec)

mysql> insert into emp values(7782,"clark","manager",2450,NULL,10);
Query OK, 1 row affected (0.03 sec)

mysql> insert into emp values(7788,"scott","analyst",3000,NULL,20);
Query OK, 1 row affected (0.03 sec)

mysql> insert into emp values(7839,"king","president",5000,NULL,10);
Query OK, 1 row affected (0.05 sec)

mysql> insert into emp values(7844,"turner","salesman",1500,NULL,30);
Query OK, 1 row affected (0.02 sec)

mysql> insert into emp values(7876,"adams","clerk",1100,NULL,20);
Query OK, 1 row affected (0.03 sec)

mysql> insert into emp values(7900,"james","NULL",950,NULL,30);
Query OK, 1 row affected (0.02 sec)

mysql> insert into emp values(7902,"ford","analyst",3000,NULL,20);
Query OK, 1 row affected (0.02 sec)

mysql> insert into emp values(7934,"miller","clerk",1300,NULL,10);
Query OK, 1 row affected (0.04 sec)
```

```
mysql> select * from emp;
+-----+-----+-----+-----+-----+-----+
| empno | ename | job   | salary | commision | deptno |
+-----+-----+-----+-----+-----+-----+
| 7369  | smith  | clerk | 2000  |      800  |    20  |
| 7499  | allen  | salesman | 1600  |      300  |    30  |
| 7521  | ward   | salesman | 1250  |      500  |    30  |
| 7566  | jones  | manager | 2975  |      NULL |    20  |
| 7654  | martin | salesman | 1250  |     1400  |    30  |
| 7698  | blake  | manager | 2850  |      NULL |    30  |
| 7782  | clark  | manager | 2450  |      NULL |    10  |
| 7788  | scott  | analyst | 3000  |      NULL |    20  |
| 7839  | king    | president | 5000  |      NULL |    10  |
| 7844  | turner | salesman | 1500  |      NULL |    30  |
| 7876  | adams  | clerk  | 1100  |      NULL |    20  |
| 7900  | james   | NULL   | 950   |      NULL |    30  |
| 7902  | ford    | analyst | 3000  |      NULL |    20  |
| 7934  | miller | clerk  | 1300  |      NULL |    10  |
+-----+-----+-----+-----+-----+-----+
14 rows in set (0.00 sec)
```

4. Insert job as 'CLERK' for all 'NULL' job types.

```
mysql> update emp set job="clerk" where ename="james";
Query OK, 0 rows affected (0.03 sec)
Rows matched: 1  Changed: 0  Warnings: 0
```

```
mysql> select * from emp;
+-----+-----+-----+-----+-----+-----+
| empno | ename | job   | salary | commision | deptno |
+-----+-----+-----+-----+-----+-----+
| 7369  | smith  | clerk | 2000  |      800  |    20  |
| 7499  | allen  | salesman | 1600  |      300  |    30  |
| 7521  | ward   | salesman | 1250  |      500  |    30  |
| 7566  | jones  | manager | 2975  |      NULL |    20  |
| 7654  | martin | salesman | 1250  |     1400  |    30  |
| 7698  | blake  | manager | 2850  |      NULL |    30  |
| 7782  | clark  | manager | 2450  |      NULL |    10  |
| 7788  | scott  | analyst | 3000  |      NULL |    20  |
| 7839  | king    | president | 5000  |      NULL |    10  |
| 7844  | turner | salesman | 1500  |      NULL |    30  |
| 7876  | adams  | clerk  | 1100  |      NULL |    20  |
| 7900  | james   | clerk  | 950   |      NULL |    30  |
| 7902  | ford    | analyst | 3000  |      NULL |    20  |
| 7934  | miller | clerk  | 1300  |      NULL |    10  |
+-----+-----+-----+-----+-----+-----+
14 rows in set (0.00 sec)
```

5. Add a new field ‘date_join’ with following values date_join17-DEC-8020-FEB-8122-FEB-81 02-APR-81 28-SEP-81 01-MAY-81 09-JUN-81 19-APR-87 17-NOV-81 08-SEP-81

```
mysql> update emp set date_joining='1981-02-22' where empno=7521;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> update emp set date_joining='1981-04-02' where empno=7566;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> update emp set date_joining='1981-09-28' where empno=7654;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> update emp set date_joining='1981-05-01' where empno=7698;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> update emp set date_joining='1981-06-09' where empno=7782;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> update emp set date_joining='1987-04-19' where empno=7788;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> update emp set date_joining='1981-11-17' where empno=7839;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

6. Display details of all employees

```
mysql> select * from emp;
+-----+-----+-----+-----+-----+-----+-----+
| empno | ename | job      | salary | commision | deptno | date_joining |
+-----+-----+-----+-----+-----+-----+-----+
| 7369 | smith | clerk    | 2000  |     800  |    20  | 1980-12-17
| 7499 | allen | salesman | 1600  |     300  |    30  | 1981-02-20
| 7521 | ward  | salesman | 1250  |     500  |    30  | 1981-02-22
| 7566 | jones | manager   | 2975  |    NULL  |    20  | 1981-04-02
| 7654 | martin | salesman | 1250  |    1400  |    30  | 1981-09-28
| 7698 | blake  | manager   | 2850  |    NULL  |    30  | 1981-05-01
| 7782 | clark  | manager   | 2450  |    NULL  |    10  | 1981-06-09
| 7788 | scott  | analyst   | 3000  |    NULL  |    20  | 1987-04-19
| 7839 | king    | president | 5000  |    NULL  |    10  | 1981-11-17
| 7844 | turner | salesman | 1500  |    NULL  |    30  | 1981-09-08
| 7876 | adams  | clerk    | 1100  |    NULL  |    20  | 1987-05-23
| 7900 | james   | clerk    | 950   |    NULL  |    30  | 1981-12-03
| 7902 | ford    | analyst   | 3000  |    NULL  |    20  | 1981-12-03
| 7934 | miller | clerk    | 1300  |    NULL  |    10  | 1982-01-23
+-----+-----+-----+-----+-----+-----+-----+
14 rows in set (0.00 sec)
```

7. Display all the distinct job types in ‘EMP’

```
mysql> select distinct job from emp;
+-----+
| job |
+-----+
| clerk |
| salesman |
| manager |
| analyst |
| president |
+-----+
5 rows in set (0.00 sec)
```

8. Display names of all employees in dept20 and 30

```
mysql> select ename from emp where deptno in(20,30);
+-----+
| ename |
+-----+
| smith |
| allen |
| ward |
| jones |
| martin |
| blake |
| scott |
| turner |
| adams |
| james |
| ford |
+-----+
11 rows in set (0.00 sec)
```

9. List name and Total of salary i.e sal+commission

```
mysql> select ename,sum(salary+commision) from emp group by ename;
+-----+-----+
| ename | sum(salary+commision) |
+-----+-----+
| adams | NULL |
| allen | 1900 |
| blake | NULL |
| clark | NULL |
| ford | NULL |
| james | NULL |
| jones | NULL |
| king | NULL |
| martin | 2650 |
| miller | NULL |
| scott | NULL |
| smith | 2800 |
| turner | NULL |
| ward | 1750 |
+-----+-----+
14 rows in set (0.00 sec)
```

10. List name and Annual Salary i.e sal*12

```
mysql> select ename,sum(salary*12)from emp group by ename;
+-----+-----+
| ename | sum(salary*12) |
+-----+-----+
| adams | 13200 |
| allen | 19200 |
| blake | 34200 |
| clark | 29400 |
| ford | 36000 |
| james | 11400 |
| jones | 35700 |
| king | 60000 |
| martin | 15000 |
| miller | 15600 |
| scott | 36000 |
| smith | 24000 |
| turner | 18000 |
| ward | 15000 |
+-----+-----+
14 rows in set (0.00 sec)
```

11. List the employees who joined on the date '03-DEC-81'

```
mysql> select ename from emp where date_joining="1981-12-03";
+-----+
| ename |
+-----+
| james |
| ford  |
+-----+
2 rows in set (0.00 sec)
```

12. Display the total salary of 'Miller'

```
mysql> select salary from emp where ename="miller";
+-----+
| salary |
+-----+
| 1300  |
+-----+
1 row in set (0.00 sec)
```

13. Delete the employee 'Miller' from 'EMP'

```
mysql> delete from emp where ename="miller";
Query OK, 1 row affected (0.03 sec)

mysql> select * from emp;
+-----+-----+-----+-----+-----+-----+-----+
| empno | ename | job      | salary | commision | deptno | date_joining |
+-----+-----+-----+-----+-----+-----+-----+
| 7369 | smith | clerk    | 2000  |     800 |    20 | 1980-12-17
| 7499 | allen | salesman | 1600  |     300 |    30 | 1981-02-20
| 7521 | ward  | salesman | 1250  |     500 |    30 | 1981-02-22
| 7566 | jones | manager  | 2975  |    NULL |    20 | 1981-04-02
| 7654 | martin | salesman | 1250  |    1400 |    30 | 1981-09-28
| 7698 | blake  | manager  | 2850  |    NULL |    30 | 1981-05-01
| 7782 | clark  | manager  | 2450  |    NULL |    10 | 1981-06-09
| 7788 | scott  | analyst  | 3000  |    NULL |    20 | 1987-04-19
| 7839 | king   | president| 5000  |    NULL |    10 | 1981-11-17
| 7844 | turner | salesman | 1500  |    NULL |    30 | 1981-09-08
| 7876 | adams  | clerk    | 1100  |    NULL |    20 | 1987-05-23
| 7900 | james  | clerk    | 950   |    NULL |    30 | 1981-12-03
| 7902 | ford   | analyst  | 3000  |    NULL |    20 | 1981-12-03
+-----+-----+-----+-----+-----+-----+-----+
13 rows in set (0.00 sec)
```

14. Display name and dept no of all employees.

```
mysql> select ename,deptno from emp;
+-----+-----+
| ename | deptno |
+-----+-----+
| smith |      20 |
| allen |      30 |
| ward  |      30 |
| jones |      20 |
| martin|      30 |
| blake |      30 |
| clark |      10 |
| scott |      20 |
| king  |      10 |
| turner|      30 |
| adams |      20 |
| james |      30 |
| ford  |      20 |
+-----+-----+
13 rows in set (0.00 sec)
```

15. Remove the field ‘commission’ from ‘EMP’ after updating salary with total salary, i.e sal+commission

```
mysql> update emp SET salary= CASE WHEN commision is NOT NULL THEN salary+commision ELSE salary END;
Query OK, 4 rows affected (0.03 sec)
Rows matched: 13  Changed: 4  Warnings: 0

mysql> select * from emp;
+-----+-----+-----+-----+-----+-----+-----+
| empno | ename | job   | salary | commision | deptno | date_joining |
+-----+-----+-----+-----+-----+-----+-----+
| 7369 | smith | clerk | 2800  |     800 |      20 | 1980-12-17 |
| 7499 | allen | salesman | 1900  |     300 |      30 | 1981-02-20 |
| 7521 | ward  | salesman | 1750  |     500 |      30 | 1981-02-22 |
| 7566 | jones | manager | 2975  |    NULL |      20 | 1981-04-02 |
| 7654 | martin| salesman | 2650  |    1400 |      30 | 1981-09-28 |
| 7698 | blake | manager | 2850  |    NULL |      30 | 1981-05-01 |
| 7782 | clark | manager | 2450  |    NULL |      10 | 1981-06-09 |
| 7788 | scott | analyst | 3000  |    NULL |      20 | 1987-04-19 |
| 7839 | King   | president | 5000  |    NULL |      10 | 1981-11-17 |
| 7844 | turner| salesman | 1500  |    NULL |      30 | 1981-09-08 |
| 7876 | adams | clerk   | 1100  |    NULL |      20 | 1987-05-23 |
| 7900 | james | clerk   | 950   |    NULL |      30 | 1981-12-03 |
| 7902 | ford  | analyst | 3000  |    NULL |      20 | 1981-12-03 |
+-----+-----+-----+-----+-----+-----+-----+
13 rows in set (0.01 sec)
```

```
mysql> alter table emp drop column commision;
Query OK, 0 rows affected (0.47 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> select * from emp;
```

empno	ename	job	salary	deptno	date_joining
7369	smith	clerk	2800	20	1980-12-17
7499	allen	salesman	1900	30	1981-02-20
7521	ward	salesman	1750	30	1981-02-22
7566	jones	manager	2975	20	1981-04-02
7654	martin	salesman	2650	30	1981-09-28
7698	blake	manager	2850	30	1981-05-01
7782	clark	manager	2450	10	1981-06-09
7788	scott	analyst	3000	20	1987-04-19
7839	king	president	5000	10	1981-11-17
7844	turner	salesman	1500	30	1981-09-08
7876	adams	clerk	1100	20	1987-05-23
7900	james	clerk	950	30	1981-12-03
7902	ford	analyst	3000	20	1981-12-03

13 rows in set (0.01 sec)

16. Display the name of employees having the same amount of salary(don't use subqueries)

```
mysql> select e1.ename,e1.salary from emp e1
    -> join emp e2 on
    -> e1.salary=e2.salary
    -> and e1.empno <> e2.empno;
+-----+
| ename | salary |
+-----+
| ford  | 3000  |
| scott | 3000  |
+-----+
2 rows in set (0.01 sec)
```

17. Display the name and employee no as ‘name’and‘emp_id’

```
mysql> select ename as name,empno as empid from emp;
+-----+-----+
| name | empid |
+-----+-----+
| smith | 7369 |
| allen | 7499 |
| ward | 7521 |
| jones | 7566 |
| martin | 7654 |
| blake | 7698 |
| clark | 7782 |
| scott | 7788 |
| king | 7839 |
| turner | 7844 |
| adams | 7876 |
| james | 7900 |
| ford | 7902 |
+-----+-----+
13 rows in set (0.00 sec)
```

18. Rename table ‘EMP’ to ‘EMPLOYEE’

```
mysql> alter table emp rename to employe;
Query OK, 0 rows affected (0.08 sec)

mysql> select * from emp;
ERROR 1146 (42S02): Table 'employee.emp' doesn't exist
mysql> select * from employe;
+-----+-----+-----+-----+-----+-----+
| empno | ename | job | salary | deptno | date_joining |
+-----+-----+-----+-----+-----+-----+
| 7369 | smith | clerk | 2800 | 20 | 1980-12-17 |
| 7499 | allen | salesman | 1900 | 30 | 1981-02-20 |
| 7521 | ward | salesman | 1750 | 30 | 1981-02-22 |
| 7566 | jones | manager | 2975 | 20 | 1981-04-02 |
| 7654 | martin | salesman | 2650 | 30 | 1981-09-28 |
| 7698 | blake | manager | 2850 | 30 | 1981-05-01 |
| 7782 | clark | manager | 2450 | 10 | 1981-06-09 |
| 7788 | scott | analyst | 3000 | 20 | 1987-04-19 |
| 7839 | king | president | 5000 | 10 | 1981-11-17 |
| 7844 | turner | salesman | 1500 | 30 | 1981-09-08 |
| 7876 | adams | clerk | 1100 | 20 | 1987-05-23 |
| 7900 | james | clerk | 950 | 30 | 1981-12-03 |
| 7902 | ford | analyst | 3000 | 20 | 1981-12-03 |
+-----+-----+-----+-----+-----+
13 rows in set (0.00 sec)
```

19. Create a new table ‘EMP_TAB’ from table ‘EMPLOYEE’

```
mysql> create table emp_tab as(select * from employee);
Query OK, 13 rows affected (0.24 sec)
Records: 13  Duplicates: 0  Warnings: 0

mysql> select * from emp_tab;
+-----+-----+-----+-----+-----+-----+
| empno | ename | job   | salary | deptno | date_joining |
+-----+-----+-----+-----+-----+-----+
| 7369  | smith  | clerk | 2800  | 20    | 1980-12-17
| 7499  | allen  | salesman | 1900  | 30    | 1981-02-20
| 7521  | ward   | salesman | 1750  | 30    | 1981-02-22
| 7566  | jones  | manager | 2975  | 20    | 1981-04-02
| 7654  | martin | salesman | 2650  | 30    | 1981-09-28
| 7698  | blake  | manager | 2850  | 30    | 1981-05-01
| 7782  | clark  | manager | 2450  | 10    | 1981-06-09
| 7788  | scott  | analyst | 3000  | 20    | 1987-04-19
| 7839  | king    | president | 5000  | 10    | 1981-11-17
| 7844  | turner | salesman | 1500  | 30    | 1981-09-08
| 7876  | adams  | clerk   | 1100  | 20    | 1987-05-23
| 7900  | james   | clerk   | 950   | 30    | 1981-12-03
| 7902  | ford    | analyst | 3000  | 20    | 1981-12-03
+-----+-----+-----+-----+-----+-----+
13 rows in set (0.00 sec)
```

20. List the details of ‘EMPLOYEE’ and ‘EMPTAB’

```
mysql> select * from emp_tab;
+-----+-----+-----+-----+-----+-----+
| empno | ename | job   | salary | deptno | date_joining |
+-----+-----+-----+-----+-----+-----+
| 7369  | smith  | clerk | 2800  | 20    | 1980-12-17
| 7499  | allen  | salesman | 1900  | 30    | 1981-02-20
| 7521  | ward   | salesman | 1750  | 30    | 1981-02-22
| 7566  | jones  | manager | 2975  | 20    | 1981-04-02
| 7654  | martin | salesman | 2650  | 30    | 1981-09-28
| 7698  | blake  | manager | 2850  | 30    | 1981-05-01
| 7782  | clark  | manager | 2450  | 10    | 1981-06-09
| 7788  | scott  | analyst | 3000  | 20    | 1987-04-19
| 7839  | king    | president | 5000  | 10    | 1981-11-17
| 7844  | turner | salesman | 1500  | 30    | 1981-09-08
| 7876  | adams  | clerk   | 1100  | 20    | 1987-05-23
| 7900  | james   | clerk   | 950   | 30    | 1981-12-03
| 7902  | ford    | analyst | 3000  | 20    | 1981-12-03
+-----+-----+-----+-----+-----+-----+
13 rows in set (0.00 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+-----+
| empno | ename | job   | salary | deptno | date_joining |
+-----+-----+-----+-----+-----+-----+
| 7369  | smith  | clerk | 2800  | 20    | 1980-12-17
| 7499  | allen  | salesman | 1900  | 30    | 1981-02-20
| 7521  | ward   | salesman | 1750  | 30    | 1981-02-22
| 7566  | jones  | manager | 2975  | 20    | 1981-04-02
| 7654  | martin | salesman | 2650  | 30    | 1981-09-28
| 7698  | blake  | manager | 2850  | 30    | 1981-05-01
| 7782  | clark  | manager | 2450  | 10    | 1981-06-09
| 7788  | scott  | analyst | 3000  | 20    | 1987-04-19
| 7839  | king    | president | 5000  | 10    | 1981-11-17
| 7844  | turner | salesman | 1500  | 30    | 1981-09-08
| 7876  | adams  | clerk   | 1100  | 20    | 1987-05-23
| 7900  | james   | clerk   | 950   | 30    | 1981-12-03
| 7902  | ford    | analyst | 3000  | 20    | 1981-12-03
+-----+-----+-----+-----+-----+-----+
13 rows in set (0.00 sec)
```

21. Delete all records from EMP

```
mysql> delete from emp_tab;
Query OK, 13 rows affected (0.04 sec)
```

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

22. Delete the table ‘EMP’

```
mysql> drop table emp_tab;
Query OK, 0 rows affected (0.13 sec)
```

```
mysql> select * from emp_tab;
ERROR 1146 (42S02): Table 'employee.emp_tab' doesn't exist
mysql> select * from employee;
```

SET-2

AIM

Create the following tables and execute the queries given below SAILORS

1. Find the names and ages of all sailors

```
mysql> select sname,age from sailors;
+-----+-----+
| sname | age |
+-----+-----+
| dustin | 45 |
| brutas | 33 |
| lubber | 55 |
| andy | 25 |
| rusty | 35 |
| horatio | 35 |
| zorba | 16 |
| horatio | 35 |
| art | 26 |
| bob | 64 |
+-----+
10 rows in set (0.00 sec)
```

2. Find all information of sailors who have reserved boat number101

```
mysql> select * from sailors s,reserves r where s.sid=r.sid and r.bid=101;
+-----+-----+-----+-----+-----+-----+
| sid | sname | rating | age | sid | bid | day |
+-----+-----+-----+-----+-----+-----+
| 22 | dustin | 7 | 45 | 22 | 101 | 1998-10-10 |
| 64 | horatio | 7 | 35 | 64 | 101 | 1998-05-09 |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

3. Find all sailors with rating above 7.

```
mysql> select * from sailors where rating>7;
+----+-----+-----+-----+
| sid | sname | rating | age  |
+----+-----+-----+-----+
| 31  | lubber |      8 | 55   |
| 32  | andy   |      8 | 25   |
| 58  | rusty   |     10 | 35   |
| 71  | zorba   |     10 | 16   |
| 74  | horatio |      9 | 35   |
+----+-----+-----+-----+
5 rows in set (0.01 sec)
```

4. Find the names of sailors who have reserved boat no103

```
mysql> select s.sname from sailors s,reserves r where s.sid=r.sid and r.bid=103;
+-----+
| sname |
+-----+
| dustin |
| lubber |
| horatio |
+-----+
3 rows in set (0.00 sec)
```

5. Find the names of sailors who have reserved a red boat, and list in the order of age

```
mysql> select distinct s.sname,s.age from sailors s, boats b,reserves r where
    s.sid=r.sid and r.bid=b.bid and b.color="red" order by s.age;
+-----+-----+
| sname | age  |
+-----+-----+
| horatio | 35  |
| dustin  | 45  |
| lubber  | 55  |
+-----+-----+
3 rows in set (0.00 sec)
```

6. Find the names of sailors who have reserved either a red or green boat.

```
mysql> select distinct s.sname from sailors s,reserves r,boats b where s.sid=r.sid and r.bid=b.bid and (b.color="red" or b.color="green");
+-----+
| sname |
+-----+
| dustin |
| lubber |
| horatio |
+-----+
3 rows in set (0.00 sec)
```

7. Find the colors of boats reserved by “Lubber”

```
mysql> select distinct b.color from sailors s,boats b,reserves r where s.sid=r.sid and r.bid=b.bid and s.sname="lubber";
+-----+
| color |
+-----+
| red   |
| green |
+-----+
2 rows in set (0.00 sec)
```

8. Find the names of sailors who have reserved at least one boat

```
mysql> select distinct s.sname from sailors s,reserves r where s.sid=r.sid;
+-----+
| sname |
+-----+
| dustin |
| lubber |
| horatio |
+-----+
3 rows in set (0.00 sec)
```

9. Find the ids and names of sailors who have reserved two different boats on the same day.

```
mysql> select distinct s.sname,s.sid from sailors s,reserves r1,reserves r2
where s.sid=r1.sid and s.sid=r2.sid and r1.day=r2.day and r1.bid<r2.bid;
+-----+----+
| sname | sid |
+-----+----+
| dustin | 22 |
+-----+----+
1 row in set (0.00 sec)
```

10. Find the name and the age of the youngest sailor.

```
mysql> select sname,age from sailors where age=(select min(age)from sailors);
+-----+-----+
| sname | age  |
+-----+-----+
| zorba | 16  |
+-----+-----+
1 row in set (0.00 sec)
```

11. Find the names and ratings of a sailor whose rating is better than some sailor called Horatio.

```
mysql> select sname,rating from sailors where rating >any(select rating from
sailors where sname="horatio");
+-----+-----+
| sname | rating |
+-----+-----+
| lubber |     8 |
| andy   |     8 |
| rusty  |    10 |
| zorba  |    10 |
| horatio |     9 |
+-----+-----+
5 rows in set (0.00 sec)
```

12. Find the names of sailors who have reserved all boats.

```
mysql> select distinct s.sname from sailors s where not exists(select *
from boats b where not exists(select * from reserves r where r.sid=s.sid
and r.bid=b.bid));
+-----+
| sname |
+-----+
| dustin |
+-----+
1 row in set (0.00 sec)
```

13. Count the number of different sailor names.

```
mysql> select count(distinct sname)from sailors;
+-----+
| count(distinct sname) |
+-----+
|          9           |
+-----+
1 row in set (0.00 sec)
```

14. Calculate the average age of all sailors.

```
mysql> select avg(age) from sailors;
+-----+
| avg(age) |
+-----+
| 36.9000  |
+-----+
1 row in set (0.00 sec)
```

15. Find the average age of sailors for each rating level

```
mysql> select rating ,avg(age) as avg_age from sailors group by rating;
+-----+
| rating | avg_age |
+-----+
|      1 | 33.0000 |
|      3 | 45.0000 |
|      7 | 40.0000 |
|      8 | 40.0000 |
|      9 | 35.0000 |
|     10 | 25.5000 |
+-----+
6 rows in set (0.00 sec)
```

16. Find the average age of sailors for each rating level that has at least two sailors.

```
mysql> select rating,avg(age) as avg_age from sailors group by rating
having count(*)>1;
+-----+-----+
| rating | avg_age |
+-----+-----+
|      3 | 45.0000 |
|      7 | 40.0000 |
|      8 | 40.0000 |
|     10 | 25.5000 |
+-----+-----+
4 rows in set (0.00 sec)
```

SET-3

AIM

Consider the following schema for Order Database: SALESMAN (Salesman_id, Name, City, Commission) CUSTOMER (Customer_id, Cust_Name, City, Grade, Salesman_id) ORDERS (Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id) Write SQL queries to

CUTOMER

```
mysql> select * from customer;
+-----+-----+-----+-----+-----+
| customer_id | cust_name | city      | Grade | salesman_id |
+-----+-----+-----+-----+-----+
|      10     | preethi   | bangalore | 100   |      1000   |
|      11     | vivek     | mangalore | 300   |      1000   |
|      12     | bhaskar   | chennai   | 400   |      2000   |
|      13     | chethan   | bangalore | 200   |      2000   |
|      14     | mamatha   | bangalore | 400   |      3000   |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

ORDERS

```
mysql> select * from orders;
+-----+-----+-----+-----+-----+
| ord_no | purchase_amt | ord_date    | customer_id | salesman_id |
+-----+-----+-----+-----+-----+
|      50     | 5000       | 2017-05-04 |      10     |      1000   |
|      51     | 450        | 2017-01-20 |      10     |      2000   |
|      52     | 1000       | 2017-02-24 |      13     |      2000   |
|      53     | 3500       | 2017-04-13 |      14     |      3000   |
|      54     | 550        | 2017-03-09 |      12     |      2000   |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

SALESMAN

```
mysql> select * from salesman;
+-----+-----+-----+-----+
| salesman_id | name    | city      | commission |
+-----+-----+-----+-----+
|      1000   | john    | Bangalore |      25    |
|      2000   | ravi    | bangalore |      20    |
|      3000   | kumar   | mysore   |      15    |
|      4000   | smith   | delhi    |      30    |
|      5000   | harsha  | hyderabad |      15    |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

1. Count the customers with grades above Bangalore's Average.

```
mysql> select grade,count(distinct customer_id)
-> from customer
-> group by grade
-> having grade > (select avg(grade)
-> from customer
-> where city='bangalore');
+-----+
| grade | count(distinct customer_id) |
+-----+
| 300  | 1 |
| 400  | 2 |
+-----+
2 rows in set (0.00 sec)
```

2. Find the name and numbers of all salesmen who had more than one customer

```
mysql> select salesman_id,name
-> from salesman A
-> where 1 < (select count(*)
-> from customer
-> where salesman_id=A.salesman_id);
+-----+
| salesman_id | name |
+-----+
| 1000 | john |
| 2000 | ravi |
+-----+
2 rows in set (0.00 sec)
```

3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)

```

mysql> select salesman.salesman_id, name, cust_name, commission
-> from salesman, customer
-> where salesman.city=customer.city
-> union
-> select salesman_id, name, 'NO MATCH', commission
-> from salesman
-> where not city=any
-> (select city from customer)
-> order by 2 desc;
+-----+-----+-----+-----+
| salesman_id | name   | cust_name | commission |
+-----+-----+-----+-----+
|      4000  | smith  | NO MATCH |        30 |
|      2000  | ravi   | preethi   |        20 |
|      2000  | ravi   | chethan   |        20 |
|      2000  | ravi   | mamatha   |        20 |
|      3000  | kumar  | NO MATCH |        15 |
|      1000  | john   | chethan   |        25 |
|      1000  | john   | mamatha   |        25 |
|      1000  | john   | preethi   |        25 |
|      5000  | harsha | NO MATCH |        15 |
+-----+-----+-----+-----+
9 rows in set (0.00 sec)

```

4. Create a view that finds the salesman who has the customer with the highest order of the day

```

mysql> create view elitsalesman as
-> select b.ord_date, a.salesman_id, a.name
-> from salesman a, orders b
-> where a.salesman_id=b.salesman_id
-> and b.purchase_amt=(select max(purchase_amt))
-> and b.purchase_amt=(select max(purchase_amt))
-> from orders c
-> where c.ord_date=b.ord_date);
Query OK, 0 rows affected (0.03 sec)

```

```

mysql> select * from elitsalesman;
+-----+-----+-----+
| ord_date | salesman_id | name   |
+-----+-----+-----+
| 2017-05-04 |          1000 | john  |
| 2017-01-20 |          2000 | ravi  |
| 2017-02-24 |          2000 | ravi  |
| 2017-04-13 |          3000 | kumar |
| 2017-03-09 |          2000 | ravi  |
+-----+-----+-----+
5 rows in set (0.02 sec)

```

5. Demonstrate the DELETE operation by removing salesmen with id 1000.All his orders must also be deleted.

```
mysql> delete from orders where salesman_id=1000;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from orders;
+-----+-----+-----+-----+-----+
| ord_no | purchase_amt | ord_date | customer_id | salesman_id |
+-----+-----+-----+-----+-----+
| 51 | 450 | 2017-01-20 | 10 | 2000 |
| 52 | 1000 | 2017-02-24 | 13 | 2000 |
| 53 | 3500 | 2017-04-13 | 14 | 3000 |
| 54 | 550 | 2017-03-09 | 12 | 2000 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

SET-4

DCL

1. Creating a Guest User and set permissions INSERT,DELETE,SELECT,UPDATE

```
cep@cep-Vostro-3470:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 5.7.42-0ubuntu0.18.04.1 (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 set4;
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 user guest identified by 'guest';
Query OK, 0 rows affected (0.00 sec)

mysql> grant insert,select,update,delete on student to guest;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
```

2. Perform INSERT,SELECT,UPDATE,DELETE operation in Guestmode

```
cep@cep-Vostro-3470:~$ mysql -u guest -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.7.42-0ubuntu0.18.04.1 (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 set4;
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> select * from student;
+-----+-----+-----+-----+
| rollno | fname | lname | email
+-----+-----+-----+-----+
| 100   | Amal  | Thomas | amal@gmail.com
| 101   | Athul | mohan  | athul@gmail.com
| 103   | godwin| paulose| godwin@gmail.com
| 104   | mathew| Joseph | mathew@gmail.com
| 105   | tebin | joy    | tebin@gmail.com
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> insert into student values(106,'Thomas','shelby','thomasshelby@gmail.com');
Query OK, 1 row affected (0.04 sec)

mysql> select * from student;
+-----+-----+-----+-----+
| rollno | fname | lname | email
+-----+-----+-----+-----+
| 100   | Amal  | Thomas | amal@gmail.com
| 101   | Athul | mohan  | athul@gmail.com
| 103   | godwin| paulose| godwin@gmail.com
| 104   | mathew| Joseph | mathew@gmail.com
| 105   | tebin | joy    | tebin@gmail.com
| 106   | Thomas | shelby | thomasshelby@gmail.com
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

3. Revoke the Permissions

```
cep@cep-Vostro-3470:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.7.42-0ubuntu0.18.04.1 (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 set4;
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> revoke insert,select,update,delete on student from guest;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
```

```
cep@cep-Vostro-3470:~$ mysql -u guest -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.7.42-0ubuntu0.18.04.1 (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 set4;
ERROR 1044 (42000): Access denied for user 'guest'@'%' to database 'set4'
mysql> quit;
Bye
```

TCL

```
mysql> select * from employee;
+-----+-----+-----+
| empid | empname | salary | year_of_experience |
+-----+-----+-----+
| 102   | cyril   | 15000  | 1                 |
| 103   | dominic  | 25000  | 3                 |
| 104   | emil    | 115000 | 15                |
| 106   | mathew   | 20000  | 2                 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)

mysql> update employee set salary=25000 where empid=106;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from employee;
+-----+-----+-----+
| empid | empname | salary | year_of_experience |
+-----+-----+-----+
| 102   | cyril   | 15000  | 1                 |
| 103   | dominic  | 25000  | 3                 |
| 104   | emil    | 115000 | 15                |
| 106   | mathew   | 25000  | 2                 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> rollback;
Query OK, 0 rows affected (0.04 sec)

mysql> select * from employee;
+-----+-----+-----+
| empid | empname | salary | year_of_experience |
+-----+-----+-----+
| 102   | cyril   | 15000  | 1                 |
| 103   | dominic  | 25000  | 3                 |
| 104   | emil    | 115000 | 15                |
| 106   | mathew   | 20000  | 2                 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

SET-5

AIM

1. Create a table employee with following field. Empid, fname, lname, city, age, salary

```

mysql> insert into employee values('E01','James','Horg','New york',17,50000);
Query OK, 1 row affected (0.12 sec)

mysql> insert into employee values('E02','Nail','Knite','Paris',20,65000);
Query OK, 1 row affected (0.09 sec)

mysql> insert into employee values('E03','pit','alex','london',25,70000);
Query OK, 1 row affected (0.07 sec)

mysql> insert into employee values('E04','MC','Lyon','New York',27,73000);
Query OK, 1 row affected (0.08 sec)

mysql> insert into employee values('E05','Paul','Adam','Rome',26,71000);
Query OK, 1 row affected (0.04 sec)

mysql> insert into employee values('E06','lauson','Hen','paris',35,100000);
Query OK, 1 row affected (0.05 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| empid | fname | lname | city      | age   | salary |
+-----+-----+-----+-----+-----+
| E01  | James | Horg  | New york | 17   | 50000 |
| E02  | Nail  | Knite | Paris     | 20   | 65000 |
| E03  | pit   | alex   | london    | 25   | 70000 |
| E04  | MC    | Adam   | Lyon     | 27   | 73000 |
| E05  | Paul  | Adam   | Rome     | 26   | 71000 |
| E06  | lauson | Hen   | paris    | 35   | 100000 |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

2. From the above Table, create a View virtual emp for these where age falls between 17to 26

```

mysql> create view virtualemp as
-> select * from employee
-> where age between 17 and 26;
Query OK, 0 rows affected (0.09 sec)

mysql> select * from virtualemp;
+-----+-----+-----+-----+-----+
| empid | fname | lname | city      | age   | salary |
+-----+-----+-----+-----+-----+
| E01   | James | Horg   | New york | 17    | 50000  |
| E02   | Nail   | Knite  | Paris     | 20    | 65000  |
| E03   | pit    | alex   | london    | 25    | 70000  |
| E05   | Paul   | Adam   | Rome      | 26    | 71000  |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

3. Describe the structure of the view table

```

mysql> desc virtualemp;
+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| empid | varchar(5)  | NO   |   | NULL    |       |
| fname | varchar(20) | YES  |   | NULL    |       |
| lname | varchar(20) | YES  |   | NULL    |       |
| city  | varchar(20) | YES  |   | NULL    |       |
| age   | int(11)       | YES  |   | NULL    |       |
| salary | int(11)       | YES  |   | NULL    |       |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

4. Selecting from a view ,Add the ‘email’ column to the view table

```
mysql> alter view virtualemp as select * from employee;
Query OK, 0 rows affected (0.05 sec)

mysql> select * from virtualemp;
+-----+-----+-----+-----+-----+-----+-----+
| empid | fname | lname | city   | age  | salary | email
+-----+-----+-----+-----+-----+-----+-----+
| E01  | James | Horg  | New york | 17   | 50000  | james@gmail.com |
| E02  | Nail  | Knite | Paris    | 20   | 65000  | Nail@gmail.com |
| E03  | pit   | alex  | london   | 25   | 70000  | Pit@gmail.com |
| E04  | MC    | Lyon  | New York | 27   | 73000  | MC@gmail.com |
| E05  | Paul  | Adam  | Rome    | 26   | 71000  | Paul@gmail.com |
| E06  | lauson | Hen   | paris   | 35   | 100000 | lauson@gmail.com |
+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

5. Drop view Table

```
mysql> drop view virtualemp;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from virtualemp;
ERROR 1146 (42S02): Table 'set5.virtualemp' doesn't exist
mysql> █
```

SET-6

JOIN OPERATIONS

```

mysql> select * from employee;
+-----+-----+-----+-----+-----+-----+-----+
| employeeid | first_name | last_name | salary | joining_date | department | gender |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | vikas | ahlawat | 600000 | 2013-02-15 11:16:28 | IT | male |
| 2 | nikita | jain | 530000 | 2014-01-09 17:31:08 | HR | female |
| 3 | ashish | kumar | 1000000 | 2014-01-09 10:05:08 | IT | male |
| 4 | nikhil | sharma | 480000 | 2014-01-09 09:00:08 | HR | male |
| 5 | anish | kadian | 500000 | 2014-01-09 09:31:08 | Payroll | male |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from project;
+-----+-----+-----+
| projectid | employeeid | project_name |
+-----+-----+-----+
| 1 | 1 | Task track |
| 2 | 1 | CLP |
| 3 | 1 | Survey management |
| 4 | 2 | HR management |
| 5 | 3 | Task track |
| 6 | 3 | GRS |
| 7 | 3 | DDS |
| 8 | 4 | HR Management |
| 9 | 5 | GL Management |
+-----+-----+-----+
9 rows in set (0.01 sec)

```

- Get employee name, project name order by first name from "Employee Detail" and "Project Detail" for those employees which have assigned projects already.

```

mysql> select first_name,project_name from employee A inner join project B on A.employeeid=B.employeeid order by first_name;
+-----+-----+
| first_name | project_name |
+-----+-----+
| anish | GL Management |
| ashish | Task track |
| ashish | GRS |
| ashish | DDS |
| nikhil | HR Management |
| nikita | HR management |
| vikas | Task track |
| vikas | CLP |
| vikas | Survey management |
+-----+-----+
9 rows in set (0.00 sec)

```

- Get employee name, project name order by first name from "Employee Detail" and "Project Detail" for all employees even if they have not assigned a project.

```

mysql> select first_name,project_name from employee A LEFT OUTER JOIN project B on A.employeeid=B.employeeid order by first_name;
+-----+-----+
| first_name | project_name |
+-----+-----+
| anish | GL Management |
| ashish | Task track |
| ashish | GRS |
| ashish | DDS |
| nikhil | HR Management |
| nikita | HR management |
| vikas | Task track |
| vikas | CLP |
| vikas | Survey management |
+-----+-----+
9 rows in set (0.00 sec)

```

3. Get all project names even if they have not matching any employee id, in the left table, order by first name from "Employee Detail" and "Project Detail".

```
mysql> select first_name,project_name from employee A RIGHT OUTER JOIN project B ON A.employeeid=B.employeeid order by first_name;
+-----+-----+
| first_name | project_name |
+-----+-----+
| anish      | GL Management
| ashish     | GRS
| ashish     | DDS
| ashish     | Task track
| nikhil     | HR Management
| nikita     | HR management
| vikas      | Survey management
| vikas      | Task track
| vikas      | CLP
+-----+-----+
9 rows in set (0.00 sec)
```

4. Getcompletetrecord(employeeename,projectname)frombothtables([EmployeeDetail],[Project Detail]), if no match is found in any table then show NULL.

```
mysql> select A.first_name,B.project_name from employee A LEFT JOIN project B ON A.employeeid = B.employeeid UNION select A.first_name,B.project_name from project B LEFT JOIN employee A ON B.employeeid = B.employeeid WHERE A.employeeid IS NULL ORDER BY first_name;
+-----+-----+
| first_name | project_name |
+-----+-----+
| anish      | GL Management
| ashish     | GRS
| ashish     | DDS
| ashish     | Task track
| nikhil     | HR Management
| nikita     | HR management
| vikas      | CLP
| vikas      | Survey management
| vikas      | Task track
+-----+-----+
9 rows in set (0.00 sec)
```

Date:07-04-2025

SET-7

PL/SQL-PROGRAM

Write a Pl/SQL program to Q*Helloworld

```
mysql> DELIMITER //;
mysql> create procedure helloworld()
-> begin
-> declare message varchar(25);
-> set message='Hello,World!';
-> select message;
-> end//;
Query OK, 0 rows affected (0.04 sec)

mysql> call helloworld();
-> //;
+-----+
| message      |
+-----+
| Hello,World! |
+-----+
1 row in set (0.00 sec)

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

Date:11-04-2025

SET-8

PL/SQL-FUNCTION

Write PL/SQL program to find sum of two number using Function

```
mysql> DELIMITER //
mysql> create function addrl(a int,b int)
-> returns int deterministic
-> begin
-> declare c int;
-> set c = (a+b);
-> return c;
-> end //
Query OK, 0 rows affected (0.00 sec)

mysql> select addrl(5,6);
-> //;
+-----+
| addrl(5,6) |
+-----+
|      11 |
+-----+
1 row in set (0.00 sec)
```

SET-9

PL/SQL-PROCEDURE

Write PL/SQL program to find sum of two number using procedure

```
mysql> use setpl;
Database changed
mysql> delimiter //;
mysql> create procedure adds(in a int,in b int)
-> begin
-> declare r int;
-> set r=a+b;
-> select concat('the sum is',r);
-> end //;
Query OK, 0 rows affected (0.01 sec)

mysql> call adds(5,6)
-> //;
+-----+
| concat('the sum is',r) |
+-----+
| the sum is11           |
+-----+
1 row in set (0.00 sec)

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

Date:11-04-2025

SET-10

PL/SQL-PROCEDURE

Write PL/SQL program to find Factorial of a Number using Procedure

```
mysql> delimiter //;
mysql> create procedure fact(in x int)
-> begin
-> declare result int;
-> declare i int;
-> set result = 1;
-> set i = 1;
-> while i <= x do
-> set result = result * i;
-> set i = (i+1);
-> end while;
-> select concat('the factorial',x,'is',result);
-> end //;
Query OK, 0 rows affected (0.00 sec)

mysql> call fact(6);
-> //;
+-----+
| concat('the factorial',x,'is',result) |
+-----+
| the factorial6is720                      |
+-----+
1 row in set (0.00 sec)

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

SET-11

PL/SQL-CURSOR

Write PL/SQL program to find Student info using Cursor

```
mysql> delimiter //
mysql> create procedure pass_stude()
-> begin
-> declare mrk int;
-> declare nm varchar(10);
-> declare v_finish int default 0;
-> declare cur_1 cursor for select name, marks from studentinfo where marks>39;
-> declare continue handler for not found set v_finish=1;
-> open cur_1;
-> get_stud:loop
-> fetch cur_1 into nm,mrk;
-> select nm,mrk;
-> if v_finish=1 then leave get_stud;
-> end if;
-> end loop get_stud;
-> end//
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> delimiter ;
mysql> select * from studentinfo;
```

nm	mrk
John	45
Alice	72

Date:11-04-2025

SET-12

PL/SQL-TRIGGER

Write PL/SQL program to Check age before insert on people using Trigger

```
mysql> delimiter //
mysql> create trigger checkage before insert on people for each row if new.age<0
      then set new.age=0;end if;//
Query OK, 0 rows affected (0.05 sec)

mysql> delimiter ;
mysql> insert into people values(-20,'sidharth');
Query OK, 1 row affected (0.02 sec)

mysql> insert into people values(-10,'mikku');
Query OK, 1 row affected (0.02 sec)

mysql> select * from people;
+-----+-----+
| age | name  |
+-----+-----+
| 30  | Amal  |
|  0  | sidharth |
|  0  | mikku |
+-----+-----+
3 rows in set (0.00 sec)
```

SET-13

MONGODBCURDOPERATIONS

1. Student Database

Create a Student database with fields SRN,sname,degree,sem,cgpa

Use student

>db

student

```
> db.student_table.insert({"SRN":1,"sname":"appu","degree":"bca","sem":2,"cgpa":6.5})
WriteResult({ "nInserted" : 1 })
> db.student_table.insert({"SRN":2,"sname":"arun","degree":"bca","sem":2,"cgpa":6.5})
WriteResult({ "nInserted" : 1 })
> db.student_table.insert({"SRN":3,"sname":"sabu mon","degree":"bca","sem":2,"cgpa":6.3})
WriteResult({ "nInserted" : 1 })
> db.student_table.insert({"SRN":4,"sname":"rahul","degree":"bba","sem":2,"cgpa":7})
WriteResult({ "nInserted" : 1 })
> db.student_table.insert({"SRN":5,"sname":"Amal","degree":"bba","sem":5,"cgpa":6.7})
WriteResult({ "nInserted" : 1 })
> |
```

1. Display all the documents

```
> db.student_table.find()
{ "_id" : ObjectId("663dba8073197dc89cd4bead"), "SRN" : 1, "sname" : "appu", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dba9573197dc89cd4beae"), "SRN" : 2, "sname" : "arun", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dbad773197dc89cd4beaf"), "SRN" : 3, "sname" : "sabu mon", "degree" : "bca", "sem" : 2, "cgpa" : 6.3 }
{ "_id" : ObjectId("663dbaef73197dc89cd4beb0"), "SRN" : 4, "sname" : "rahul", "degree" : "bba", "sem" : 2, "cgpa" : 7 }
{ "_id" : ObjectId("663dbb0273197dc89cd4beb1"), "SRN" : 5, "sname" : "Amal", "degree" : "bba", "sem" : 5, "cgpa" : 6.7 }
> |
```

2. Display all the students in BCA

```
> db.student_table.find({degree:"bca"})
{ "_id" : ObjectId("663dba8073197dc89cd4bead"), "SRN" : 1, "sname" : "appu", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dba9573197dc89cd4beae"), "SRN" : 2, "sname" : "arun", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dbad773197dc89cd4beaf"), "SRN" : 3, "sname" : "sabu mon", "degree" : "bca", "sem" : 2, "cgpa" : 6.3 }
> |
```

3. Display all students in ascending order

```
> db.student_table.find().sort({sname:1})
{ "_id" : ObjectId("663dbb0273197dc89cd4beb1"), "SRN" : 5, "sname" : "Amal", "degree" : "bba", "sem" : 5, "cgpa" : 6.7 }
{ "_id" : ObjectId("663dba8073197dc89cd4bead"), "SRN" : 1, "sname" : "appu", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dba9573197dc89cd4beae"), "SRN" : 2, "sname" : "arun", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dbaef73197dc89cd4beb0"), "SRN" : 4, "sname" : "rahul", "degree" : "bba", "sem" : 2, "cgpa" : 7 }
{ "_id" : ObjectId("663dbad773197dc89cd4beaf"), "SRN" : 3, "sname" : "sabu mon", "degree" : "bca", "sem" : 2, "cgpa" : 6.3 }
> |
```

4. Display first 5 students

```
> db.student_table.find().limit(5)
{ "_id" : ObjectId("663dba8073197dc89cd4bead"), "SRN" : 1, "sname" : "appu", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dba9573197dc89cd4beae"), "SRN" : 2, "sname" : "arun", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dbad773197dc89cd4beaf"), "SRN" : 3, "sname" : "sabu mon", "degree" : "bca", "sem" : 2, "cgpa" : 6.3 }
{ "_id" : ObjectId("663dbaef73197dc89cd4beb0"), "SRN" : 4, "sname" : "rahul", "degree" : "bba", "sem" : 2, "cgpa" : 7 }
{ "_id" : ObjectId("663dbb0273197dc89cd4beb1"), "SRN" : 5, "sname" : "Amal", "degree" : "bba", "sem" : 5, "cgpa" : 6.7 }
> |
```

5. Display student 3,4,5

```
> db.student_table.find().skip(2).limit(3)
{ "_id" : ObjectId("663dbad773197dc89cd4beaf"), "SRN" : 3, "sname" : "sabu mon", "degree" : "bca", "sem" : 2, "cgpa" : 6.3 }
{ "_id" : ObjectId("663dbaef73197dc89cd4beb0"), "SRN" : 4, "sname" : "rahul", "degree" : "bba", "sem" : 2, "cgpa" : 7 }
{ "_id" : ObjectId("663dbb0273197dc89cd4beb1"), "SRN" : 5, "sname" : "Amal", "degree" : "bba", "sem" : 5, "cgpa" : 6.7 }
> |
```

6. List the degree of student Rahul

```
> db.student_table.find({sname:"rahul"})
{ "_id" : ObjectId("663dbaef73197dc89cd4beb0"), "SRN" : 4, "sname" : "rahul", "degree" : "bba", "sem" : 2, "cgpa" : 7 }
> |
```

7. Display student details of 3,4,5 in descending order of cgpa

```
> db.student_table.find().sort({cgpa:-1}).skip(2).limit(3)
{ "_id" : ObjectId("663dba8073197dc89cd4bead"), "SRN" : 1, "sname" : "appu", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dba9573197dc89cd4beae"), "SRN" : 2, "sname" : "arun", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dbad773197dc89cd4beaf"), "SRN" : 3, "sname" : "sabu mon", "degree" : "bca", "sem" : 2, "cgpa" : 6.3 }
> |
>
```

8. Display the no of students in BCA

```
> db.student_table.find({degree:"bca"}).count()
3
>
```

9. Display all degree without id

```
> db.student_table.find({}, {_id:0,degree:1})
{ "degree" : "bca" }
{ "degree" : "bca" }
{ "degree" : "bca" }
{ "degree" : "bba" }
{ "degree" : "bba" }
>
```

10. Display all distinct Degree

```
>
> db.student_table.distinct("degree")
[ "bca", "bba" ]
>
```

11. Display all the BCA students with cgpa <6 but greater than 7

```
>
> db.student_table.find({degree:"bca",cgpa:{$gt:6,$lt:7}})
{ "_id" : ObjectId("663dba8073197dc89cd4bead"), "SRN" : 1, "sname" : "appu", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dba9573197dc89cd4beae"), "SRN" : 2, "sname" : "arun", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dbad773197dc89cd4beaf"), "SRN" : 3, "sname" : "sabu mon", "degree" : "bca", "sem" : 2, "cgpa" : 6.3 }
> |
```

12. Display all the students in BCA and in 2nd Sem

```
> db.student_table.find({$and:[{degree:"bca"}, {sem:2}]})
{ "_id" : ObjectId("663dba8073197dc89cd4bead"), "SRN" : 1, "sname" : "appu", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dba9573197dc89cd4beae"), "SRN" : 2, "sname" : "arun", "degree" : "bca", "sem" : 2, "cgpa" : 6.5 }
{ "_id" : ObjectId("663dbad773197dc89cd4beaf"), "SRN" : 3, "sname" : "sabu mon", "degree" : "bca", "sem" : 2, "cgpa" : 6.3 }
>
>
```

2.employee database

Updatemodifiers(\$set,\$unset,\$inc.\$push,\$PushAll,\$pull,\$pullAll,\$addToSet)Create an employee database with fields:{ eid, ename,dept, desig, salary, yoj, address{dno, street, Locality, city}}

```
test> use employee
switched to db employee
employee> db
employee
employee> db.emp.insert({eid: 001, ename:"rahul", dept:"production", desig:"developer", salary:30000, yoj:2015, "address":{dno:397, street:2, locality:"rmnagar", city:"banglore"}})
```

1. Display all employees with salary in range(3000,50000)

```
employee> db.emp.find({salary:{$gt:30000,$lt:50000}})
[
  {
    _id: ObjectId('6682b5453bf052181ecc8989'),
    eid: 2,
    ename: 'amal',
    dept: 'production',
    desig: 'analyst',
    salary: 40000,
    yoj: 2013,
    address: { dno: 357, street: 4, locality: 'abnagar', city: 'banglore' }
  },
  {
    salary: 28000,
    _id: ObjectId('6682b5553bf052181ecc898a'),
    eid: 3, : { dno: 345, street: 2, locality: 'rmnagar', city: 'banglore' },
    ename: 'mathew',
    dept: 'sales',
    desig: 'executive',
    salary: 45000,
    yoj: 2012,
    address: { dno: 557, street: 4, locality: 'mmnagar', city: 'banglore' }
  }
]
```

2. Display all the employees with design developer

```
employee> db.emp.find({desig:"developer"})
[ {
  _id: ObjectId('6682b53f3bf052181ecc8988'),
  eid: 1,
  ename: 'rahul',
  dept: 'production',
  desig: 'developer',
  salary: 30000,
  yoj: 2015,
  address: { dno: 397, street: 2, locality: 'rmnagar', city: 'banglore' }
},
{
  _id: ObjectId('6682b55c3bf052181ecc898c'),
  eid: 5,
  ename: 'nixon',
  dept: 'production',
  desig: 'developer',
  salary: 28000,
  yoj: 2016,
  address: { dno: 345, street: 2, locality: 'rmnagar', city: 'banglore' }
}
]
```

3. Display the salary of “Rahul”

```
employee> db.emp.find({ename:"rahul"},{salary:1})
[ { _id: ObjectId('6682b53f3bf052181ecc8988'), salary: 30000 } ]
employee>
```

4. Display city of employee Rahul

```
employee> db.emp.find({ename:"rahul"}, {"address.city":1})
[ {
  _id: ObjectId('6682b53f3bf052181ecc8988'),
  address: { city: 'banglore' }
}
]
```

5. Update the salary of developers by 5000 increment

```
employee> db.emp.update({desig:"developer"},{$inc:{salary:5000}})
employee> db.emp.find({desig:"developer"})
[
  {
    _id: ObjectId('6682b53f3bf052181ecc8988'),
    eid: 1,
    ename: 'rahul',
    dept: 'production',
    desig: 'developer',
    salary: 35000,
    yoj: 2015,
    address: { dno: 397, street: 2, locality: 'rmnagar', city: 'banglore' }
  },
  {
    _id: ObjectId('6682b55c3bf052181ecc898c'),
    eid: 5,
    ename: 'nixon',
    dept: 'production',
    desig: 'developer',
    salary: 33000,
    yoj: 2016,
    address: { dno: 345, street: 2, locality: 'rmnagar', city: 'banglore' }
  }
]
```

6. Add field age to employee Rahul

```
employee> db.emp.update({ename:"rahul"},{$set:{age:"22"}})
employee> db.emp.find({ename:"rahul"})
[
  {
    _id: ObjectId('6682b53f3bf052181ecc8988'),
    eid: 1,
    ename: 'rahul',
    dept: 'production',
    desig: 'developer',
    salary: 35000,
    yoj: 2015,
    address: { dno: 397, street: 2, locality: 'rmnagar', city: 'banglore' },
    age: '22'
  }
]
```

7. Remove yoj from Rahul

```
employee> db.emp.update({ename:"rahul"},{$unset:{yoj:1}})
employee> db.emp.find({ename:"rahul"})
[
  {
    _id: ObjectId('6682b53f3bf052181ecc8988'),
    eid: 1,
    ename: 'rahul',
    dept: 'production',
    desig: 'developer',
    salary: 35000,
    address: { dno: 397, street: 2, locality: 'rmnagar', city: 'banglore' },
    age: '22'
  }
]
```

8. Add an array field project to Rahul

```
employee> db.emp.update({ename:"rahul"},{$push:{project:"p1"}})
employee> db.emp.find({ename:"rahul"})
[
  {
    _id: ObjectId('6682b53f3bf052181ecc8988'),
    eid: 1,
    ename: 'rahul',
    dept: 'production',
    desig: 'developer',
    salary: 35000,
    address: { dno: 397, street: 2, locality: 'rmnagar', city: 'banglore' },
    age: '22',
    project: [ 'p1' ]
  }
]
```

9. Add p2 and p3 project to Rahul

```
employee> db.emp.update({ename:"rahul"},{$pushAll:{project:["p2","p3"]}})
```

```

employee> db.emp.find({ename:"rahul"})
[
  {
    _id: ObjectId('6682b53f3bf052181ecc8988'),
    eid: 1,
    ename: 'rahul',
    dept: 'production',
    desig: 'developer',
    salary: 35000,
    address: { dno: 397, street: 2, locality: 'rmnagar', city: 'banglore' },
    age: '22',
    project: [ 'p1', 'p2', 'p3' ]
  }
]
1

```

10. Remove p3 from Rahul

```

employee> db.emp.update({ename:"rahul"},{$pull:{project:"p3"}})
employee> db.emp.find({ename:"rahul"})
[
  {
    _id: ObjectId('6682b53f3bf052181ecc8988'),
    eid: 1,
    ename: 'rahul',
    dept: 'production',
    desig: 'developer',
    salary: 35000,
    address: { dno: 397, street: 2, locality: 'rmnagar', city: 'banglore' },
    age: '22',
    project: [ 'p1', 'p2' ]
  }
]
1

```

11. Add a new embedded object “contacts” with “email” and “phone as array objects to Rahul

```

employee> db.emp.update({ename:"rahul"},{$push:{contacts:{phone:"998876655", email:"rahul@gmail.com"}}})

```

```
employee> db.emp.find({ename:"rahul"})
[ {
  _id: ObjectId('6682b53f3bf052181ecc8988'),
  eid: 1,
  ename: 'rahul',
  dept: 'production',
  desig: 'developer',
  salary: 35000,
  address: { dno: 397, street: 2, locality: 'rmnagar', city: 'banglore' },
  age: '22',
  project: [ 'p1', 'p2' ],
  contacts: [ { phone: '998876655', email: 'rahul@gmail.com' } ]
}
```

SET-14

AGGREGATE OPERATIONS

Assume that you have two collection, student and college

1. Documents in student collection contains 'name, address, mob, dateofbrith, Qualification (including course (starting from sslc, plus two, bsc, MCA, mark), location district)
2. Document in college collection include 'name, location, establishedyear, district'

```
>
> db.student.find({}, {_id:0})
{ "name" : "alice", "address" : "123 main street", "mob" : 1323424, "dob" : "1998-05-15", "qualification" : [ { "course" : "SSLC", "mark" : 85 }, { "course" : "plus two", "mark" : 82 }, { "course" : "B.sc", "mark" : 80 }, { "course" : "MCA", "mark" : 85 } ], "location" : "poonjar", "district" : "kerala" }
{ "name" : "alice", "address" : "123 main street", "mob" : 1323424, "dob" : "1998-05-15", "qualification" : [ { "course" : "SSLC", "mark" : 90 }, { "course" : "plus two", "mark" : 78 }, { "course" : "B.sc", "mark" : 75 }, { "course" : "MCA", "mark" : 80 } ], "location" : "poonjar", "district" : "kerala" }
{ "name" : "bob", "address" : "456 elm street", "mob" : 7623424, "dob" : "1996-08-20", "qualification" : [ { "course" : "SSLC", "mark" : 90 }, { "course" : "plus two", "mark" : 78 }, { "course" : "B.sc", "mark" : 75 }, { "course" : "MCA", "mark" : 80 } ], "location" : "poonjar", "district" : "kerala" }
> |
```



```
>
> db.college.find({}, {_id:0})
{ "name" : "poonjar college", "location" : "123 avenue", "established_year" : 2000, "district" : "kottayam" }
{ "name" : "sjct college", "location" : "456 boulevard", "established_year" : 1995, "district" : "kottayam" }
{ "name" : "sd college", "location" : "kanjirappally", "established_year" : 2014, "district" : "kottayam" }
{ "name" : "pmr college", "location" : "thiruvalam", "established_year" : 2005, "district" : "thiravndram" }
{ "name" : "kkm college", "location" : "thiruvalam", "established_year" : 2008, "district" : "thiravndram" }
>
>
>
```

1. Find out the total number of students in collection

```
>
> db.student.aggregate([{$count:"numberOfStudent"}])
{ "numberOfStudent" : 3 }
>
>
```

```
>
> db.student.aggregate([{$group:{_id:0,numberOfStudent:{$sum:1}}},{$project:{_id:0}}])
{ "numberOfStudent" : 3 }
```

2. How many colleges are in district TVM

```
>
> db.college.aggregate([{$match:{district:"thiravndram"}},{$count:"college of thrivandram:"}])
{ "college of thrivandram:" : 2 }
```

3. Display the details of student and college residing in same district

```
>
> db.student.aggregate([{$lookup:{from:"college",localField:"district",foreignField:"district",as:"college details"}}]).pretty()
{
    "_id" : ObjectId("661373ea963a71e775bc3d8c"),
    "name" : "alice",
    "address" : "123 main street",
    "mob" : 1323424,
    "dob" : "1998-05-15",
    "qualification" : [
        {
            "course" : "SSLC",
            "mark" : 85
        },
        {
            "course" : "plus two",
            "mark" : 82
        },
        {
            "course" : "B.sc",
            "mark" : 80
        },
        {
            "course" : "MCA",
            "mark" : 85
        }
    ],
    "location" : "poonjar",
    "district" : "kerala",
    "college details" : []
}
```

```
{
    "_id" : ObjectId("66137715963a71e775bc3d8d"),
    "name" : "alice",
    "address" : "123 main street",
    "mob" : 1323424,
    "dob" : "1998-05-15",
    "qualification" : [
        {
            "course" : "SSLC",
            "mark" : 90
        },
        {
            "course" : "plus two",
            "mark" : 78
        },
        {
            "course" : "B.sc",
            "mark" : 75
        },
        {
            "course" : "MCA",
            "mark" : 80
        }
    ],
    "location" : "poonjar",
    "district" : "kerala",
    "college details" : []
}
```

```
{
    "_id" : ObjectId("66137792963a71e775bc3d8e"),
    "name" : "bob",
    "address" : "456 elm street",
    "mob" : 7623424,
    "dob" : "1996-08-20",
    "qualification" : [
        {
            "course" : "SSLC",
            "mark" : 90
        },
        {
            "course" : "plus two",
            "mark" : 78
        },
        {
            "course" : "B.sc",
            "mark" : 75
        },
        {
            "course" : "MCA",
            "mark" : 80
        }
    ],
    "location" : "poonjar",
    "district" : "kerala",
    "college details" : [ ]
}
```

4. Display the name and establishment year of college in district ways

```
> db.college.aggregate([{$sort:{"district":1}},{$project:{_id:0,name:1,established_year:1}}]).pretty()
{
    "name" : "poonjar college",
    "established_year" : 2000
}
{
    "name" : "sjcet college",
    "established_year" : 1995
}
{
    "name" : "sd college",
    "established_year" : 2014
}
{
    "name" : "pmr college",
    "established_year" : 2005
}
{
    "name" : "kkm college",
    "established_year" : 2008
}
> 1 ms
```

5. Display the course and marks

```
> db.student.aggregate([{$sort:{"mark":1}},{$project:{_id:0,qualification:1}}]).pretty()
{
    "qualification" : [
        {
            "course" : "SSLC",
            "mark" : 85
        },
        {
            "course" : "plus two",
            "mark" : 82
        },
        {
            "course" : "B.sc",
            "mark" : 80
        },
        {
            "course" : "MCA",
            "mark" : 85
        }
    ]
}
{
    "qualification" : [
        {
            "course" : "SSLC",
            "mark" : 90
        },
        {
            "course" : "plus two",
            "mark" : 78
        },
        {
            "course" : "B.sc",
            "mark" : 75
        },
        {
            "course" : "MCA",
            "mark" : 80
        }
    ]
}
{
    "qualification" : [
        {
            "course" : "SSLC",
            "mark" : 90
        },
        {
            "course" : "plus two",
            "mark" : 78
        },
        {
            "course" : "B.sc",
            "mark" : 75
        },
        {
            "course" : "MCA",
            "mark" : 80
        }
    ]
}
> |
```

6. Display the details of college in sorted order(name)

```
>
> db.college.aggregate([{$sort:[{"name":1}]}]).pretty()
{
    "_id" : ObjectId("6613805b963a71e775bc3d93"),
    "name" : "kkm college",
    "location" : "thiruvalam",
    "established_year" : 2008,
    "district" : "thiravndram"
}
{
    "_id" : ObjectId("66137e8b963a71e775bc3d92"),
    "name" : "pmr college",
    "location" : "thiruvalam",
    "established_year" : 2005,
    "district" : "thiravndram"
}
{
    "_id" : ObjectId("66137a45963a71e775bc3d8f"),
    "name" : "poonjar college",
    "location" : "123 avenue",
    "established_year" : 2000,
    "district" : "kottayam"
}
{
    "_id" : ObjectId("66137ad6963a71e775bc3d91"),
    "name" : "sd college",
    "location" : "kanjirappally",
    "established_year" : 2014,
    "district" : "kottayam"
}
{
    "_id" : ObjectId("66137aa8963a71e775bc3d90"),
    "name" : "sjcet college",
    "location" : "456 boulevard",
    "established_year" : 1995,
    "district" : "kottayam"
}
> |
```

7. List the details of college in descending order based on the establishment year.

```
>
> db.college.aggregate([{$sort:{"established_year": -1}}]).pretty()
{
    "_id" : ObjectId("66137ad6963a71e775bc3d91"),
    "name" : "sd college",
    "location" : "kanjirappally",
    "established_year" : 2014,
    "district" : "kottayam"
}
{
    "_id" : ObjectId("6613805b963a71e775bc3d93"),
    "name" : "kkm college",
    "location" : "thiruvalam",
    "established_year" : 2008,
    "district" : "thiravndram"
}
{
    "_id" : ObjectId("66137e8b963a71e775bc3d92"),
    "name" : "pmr college",
    "location" : "thiruvalam",
    "established_year" : 2005,
    "district" : "thiravndram"
}
{
    "_id" : ObjectId("66137a45963a71e775bc3d8f"),
    "name" : "poonjar college",
    "location" : "123 avenue",
    "established_year" : 2000,
    "district" : "kottayam"
}
{
    "_id" : ObjectId("66137aa8963a71e775bc3d90"),
    "name" : "sjcet college",
    "location" : "456 boulevard",
    "established_year" : 1995,
    "district" : "kottayam"
}
>
```

8. Display the colleges

```
>
>
> db.college.aggregate([{$project:{name:1}}]).pretty()
{
    "_id" : ObjectId("66137a45963a71e775bc3d8f"),
    "name" : "poonjar college"
}
{
    "_id" : ObjectId("66137aa8963a71e775bc3d90"),
    "name" : "sjcet college"
}
{
    "_id" : ObjectId("66137ad6963a71e775bc3d91"),
    "name" : "sd college"
}
{
    "_id" : ObjectId("66137e8b963a71e775bc3d92"),
    "name" : "pmr college"
}
{
    "_id" : ObjectId("6613805b963a71e775bc3d93"),
    "name" : "kkm college"
}
>
>
>
```

9. Display the marks of MCA

```
>
>
> db.student.aggregate([{$match:{'qualification.course':'MCA'}}, {$group:{'_id':0,min_mark:{$min:'$qualification.mark'}}}]).pretty()
{ "_id" : 0, "min_mark" : [ 85, 82, 80, 85 ] }
>
>
> |
```

Date:20-05-2025

SET-15

CREATE USERS AND ROLES

```
testdb1> db.createUser( {user: "test1", pwd: "test1", roles: [ {role: "readWrite", db: "testdb1"} ]} )
{ ok: 1 }
testdb1> db.getUsers()
{
  users: [
    {
      _id: 'testdb1.test1',
      userId: new UUID("17d51bd0-6e1a-4b08-b9ad-68caf656dd9d"),
      user: 'test1',
      db: 'testdb1',
      roles: [ { role: 'readWrite', db: 'testdb1' } ],
      mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
    }
  ],
  ok: 1
}
testdb1>
```

Date:20-05-2025

SET-16

MONGODB DATABASE CONNECTION



The screenshot shows a code editor window with a tab labeled "mongoconnection.py". The code is written in Python and performs the following steps:

- Imports the pymongo module.
- Creates a MongoClient object connected to "mongodb://localhost:27017/".
- Selects the "student" database from the client.
- Selects the "student" collection from the database.
- Creates a document with fields: "reg_no": 1007, "sname": "mathew", and "course": "MCA".
- Inserts the document into the collection.
- Finds the inserted document using the find_one() method.
- Prints the found document.

```
1 import pymongo
2 mongocon=pymongo.MongoClient("mongodb://localhost:27017/")
3 mydb=mongocon["student"]
4 mycol=mydb["student"]
5 mydoc={"reg_no":1007,"sname":"mathew","course":"MCA"}
6 insertmy=mycol.insert_one(mydoc)
7 x=mycol.find_one()
8 print(x)
9 |
```

OUTPUT

```
mongoconnection x
C:\Users\home\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\home\PycharmProject:
{'_id': ObjectId('6686bb65ebda567206cc8988'), 'reg_no': 1001, 'sname': 'Akash', 'course': 'MCA'}

Process finished with exit code 0
```

```
student> db.student.find()
[
  {
    _id: ObjectId('6686bb65ebda567206cc8988'),
    reg_no: 1001,
    sname: 'Akash',
    course: 'MCA'
  },
  {
    _id: ObjectId('6686bb90ebda567206cc8989'),
    reg_no: 1002,
    sname: 'Amal',
    course: 'MCA'
  },
  {
    _id: ObjectId('6686bbb1ebda567206cc898b'),
    reg_no: 1004,
    sname: 'dominic',
    course: 'MCA'
  },
  {
    _id: ObjectId('6686c3ce0b3fd467dfc905c5'),
    reg_no: 1007,
    sname: 'mathew',
    course: 'MCA'
  }
]
```

Date:20-05-2025

SET-17

SORTING



```
sort.py ×
1 import pymongo
2 myclient = pymongo.MongoClient("mongodb://localhost:27017/")
3 mydb = myclient["student"]
4 mycol = mydb["student"]
5 mydoc=mycol.find().sort("sname",1)
6 for x in mydoc:
7     print(x)
8 |
```

OUTPUT



```
sort ×
C:\Users\home\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\home\PycharmProjects\py
{'_id': ObjectId('6686bb65ebda567206cc8988'), 'reg_no': 1001, 'sname': 'Akash', 'course': 'MCA'}
{'_id': ObjectId('6686bb90ebda567206cc8989'), 'reg_no': 1002, 'sname': 'Amal', 'course': 'MCA'}
{'_id': ObjectId('6686bbb1ebda567206cc898b'), 'reg_no': 1004, 'sname': 'dominic', 'course': 'MCA'}
{'_id': ObjectId('6686c3ce0b3fd467dfc905c5'), 'reg_no': 1007, 'sname': 'mathew', 'course': 'MCA'}

Process finished with exit code 0
```