## 20MCA134 – ADVANCED DBMS LABORATORY RECORD

*Submitted in partial fulfilment of the requirements for the award of Masters 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**

**SREELAKSHMI KRISHNAN(PJR24MCA-2020)**

**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 **SREELAKSHMI KRISHNAN** Reg No. **PJR24MCA-2020** of College of Engineering, Poonjar during the academic year 2023- 2025.

Dr. Annie Julie Joseph

**Head of the Department**

Thasni Noushad

**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 Virtualemp and its operations | 27 |
| 6 | Join Operations | 30 |
| 7 | PL/SQL Program | 32 |
| 8 | PL/SQL Function and Procedure | 33 |
| 9 | PL/SQL Cursor and Trigger | 35 |
| 10 | Mongodb Installation | 37 |
| 11 | Mongodb CURD Operations | 39 |
| 12 | Aggregate Operations | 48 |
| 13 | Create users and Roles | 55 |
| 14 | Pymongo Library | 56 |
| 15 | Python Mongodb Connection and Sort | 58 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Date: 12-02-2024

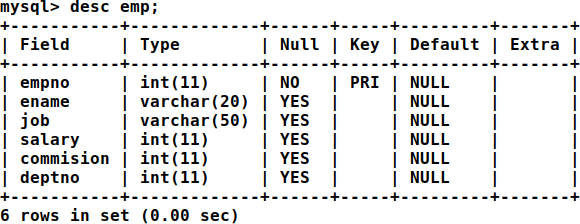
# SET – 1

**AIM**

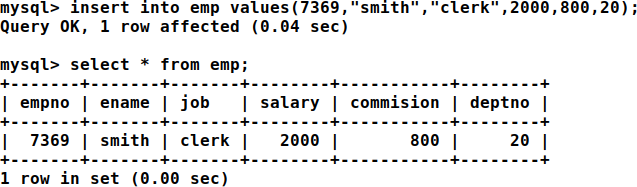
Create an employee table ‘EMP’ with following fields : empno NUMBER(4) ename VARCHAR2(25) job VARCHAR2(12) salary NUMBER(10,2) commission NUMBER(7,2) deptno NUMBER(2)



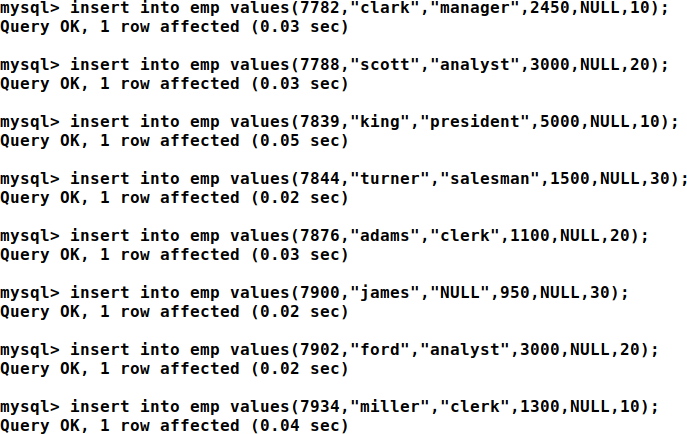
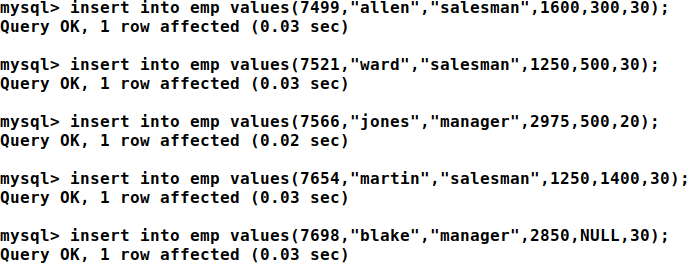
1. Display the structure of ‘EMP’

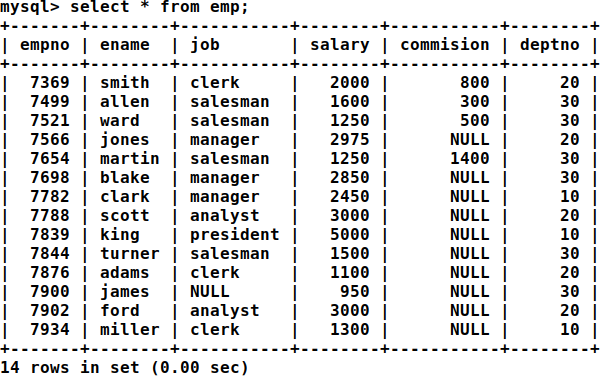


1. Insert the following record into ‘EMP’

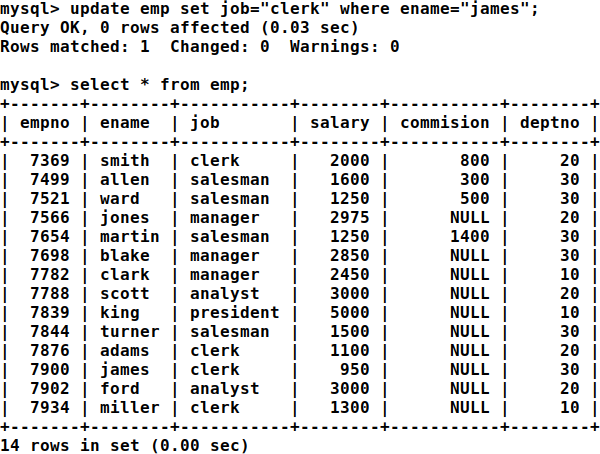


1. Insert the rest of the records using the substitution variable.

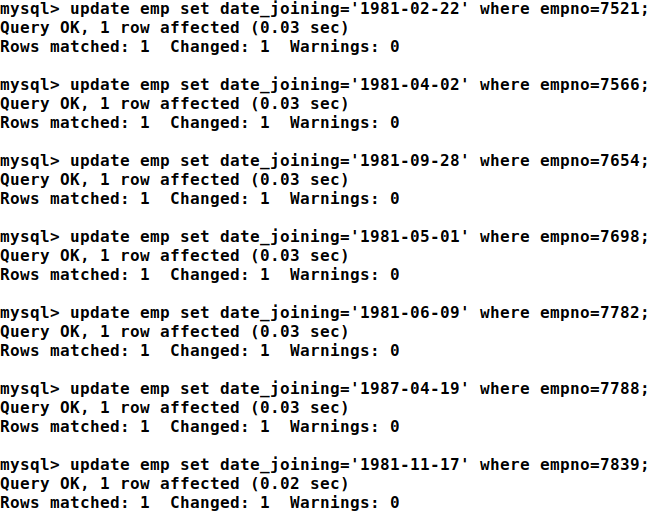




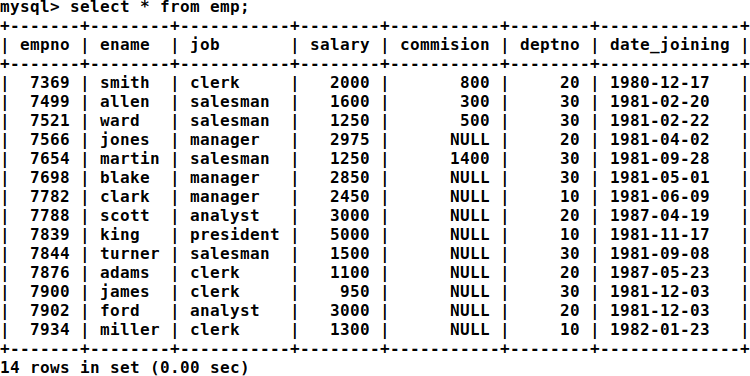
1. Insert job as ‘CLERK’ for all ‘NULL’ job types.



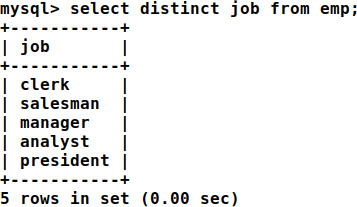
1. Add a new field ‘date\_join’ with following values date\_join 17-DEC-80 20-FEB-81 22- FEB-81 02-APR-81 28-SEP-81 01-MAY-81 09-JUN-81 19-APR-87 17-NOV-81 08- SEP-81



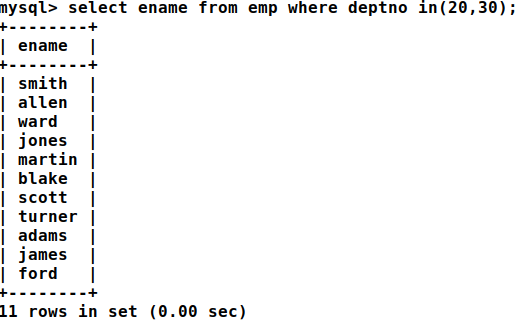
1. Display details of all employees



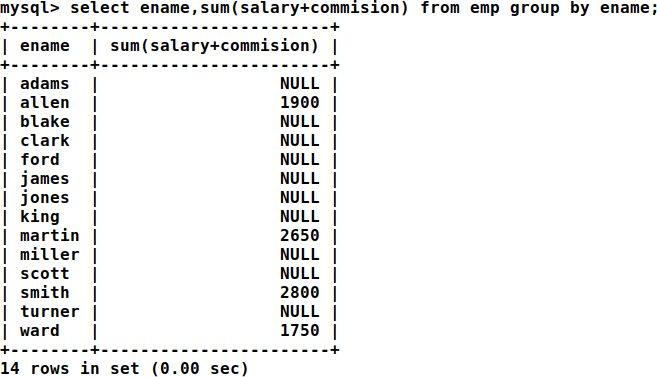
1. Display all the distinct job types in ‘EMP’



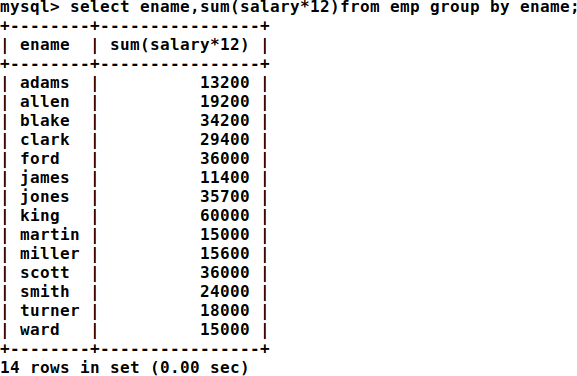
1. Display names of all employees in dept 20 and 30



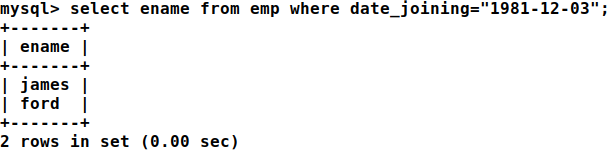
1. List name and Total of salary i.e sal+commission



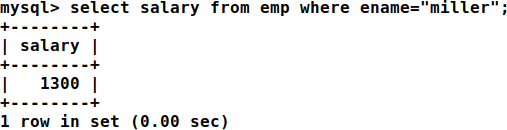
1. List name and Annual Salary i.e sal\*12



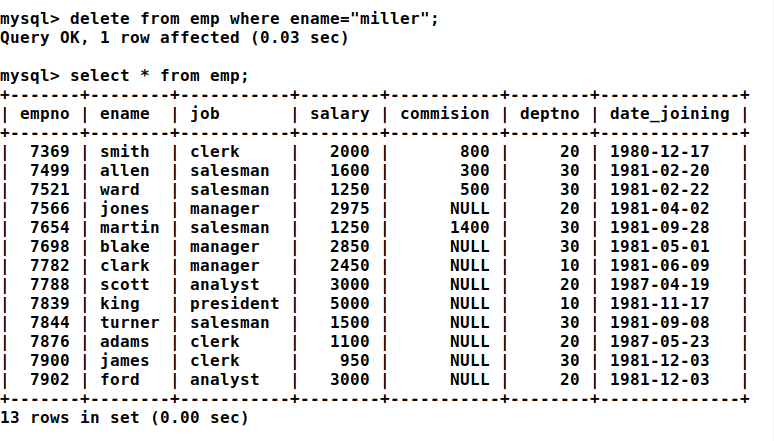
1. List the employees who joined on the date ‘03-DEC-81’



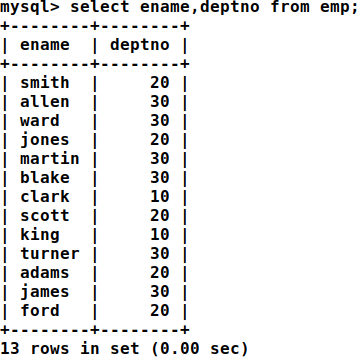
1. Display the total salary of ‘Miller’



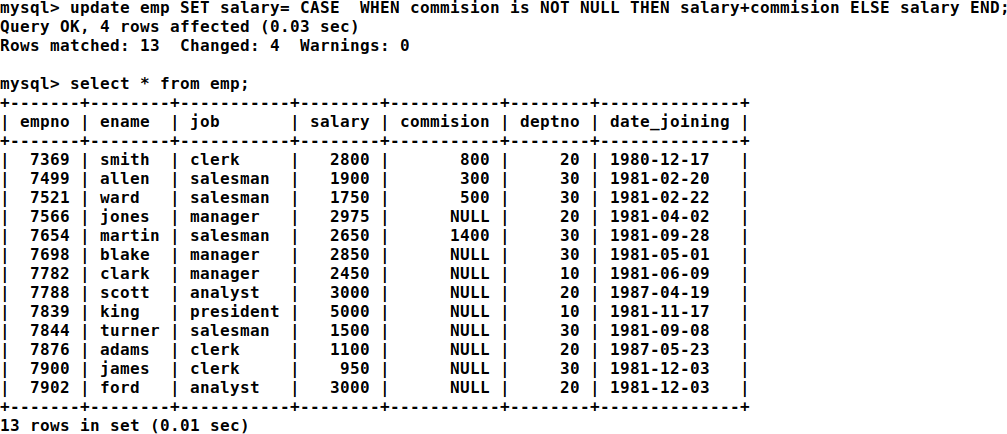
1. Delete the employee ‘Miller’ from ‘EMP’

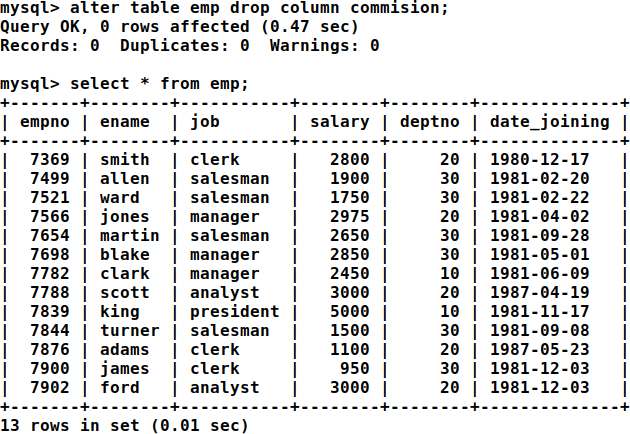


1. Display name and deptno of all employees.

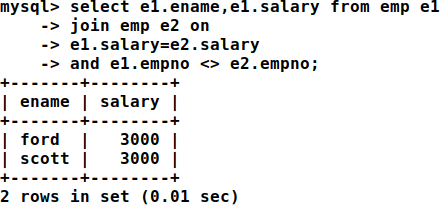


1. Remove the field ‘commission’ from ‘EMP’ after updating salary with total salary, i.e sal+commission

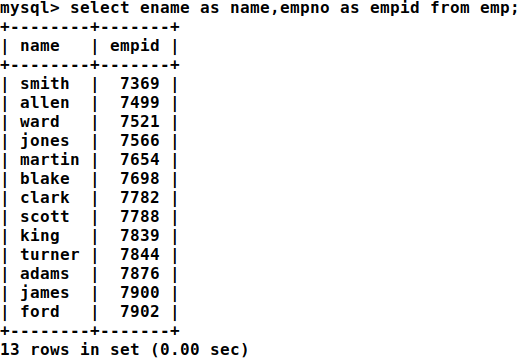




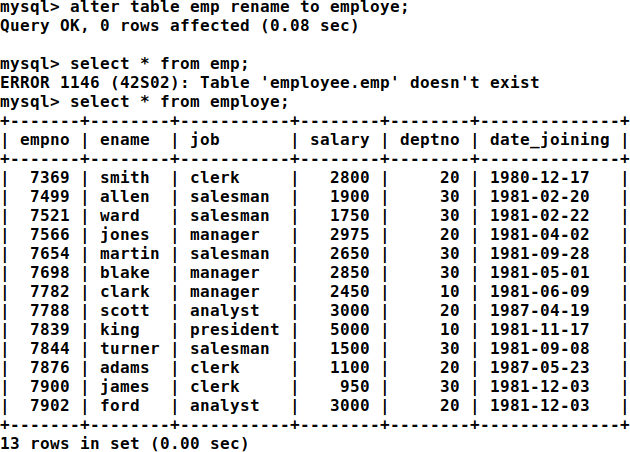
1. Display the name of employees having the same amount of salary ( don’t use subqueries)



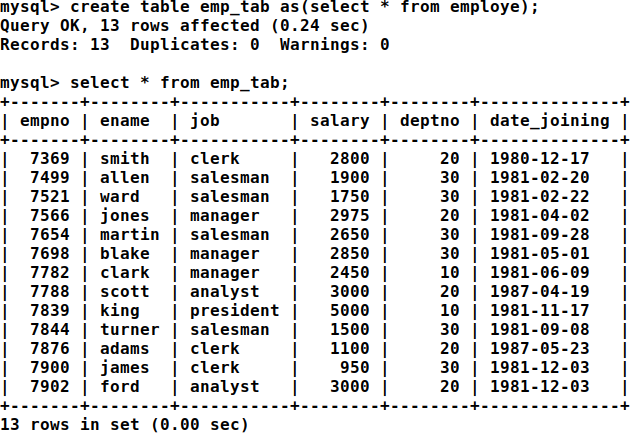
1. Display the name and employee no as ‘name’ and ‘emp\_id’



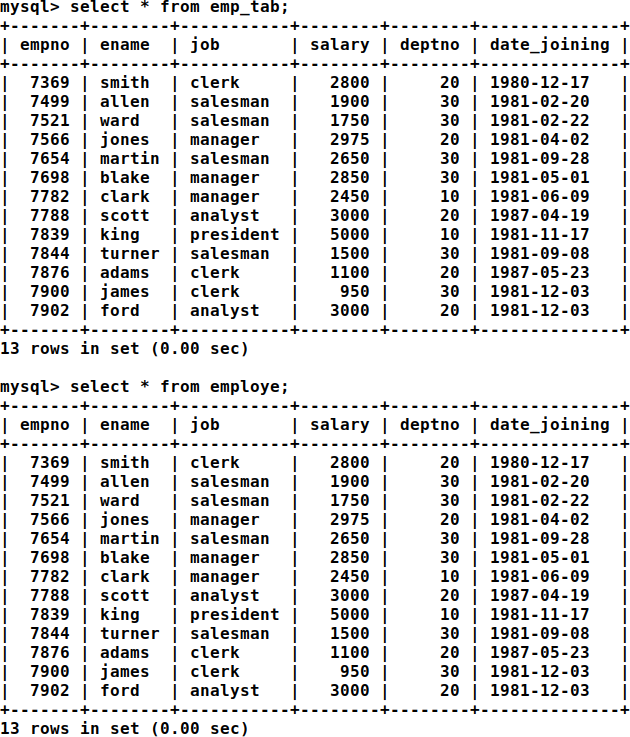
1. Rename table ‘EMP’ to ‘EMPLOYEE’



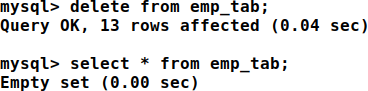
1. Create a new table ‘EMP\_TAB’ from table ‘EMPLOYEE’



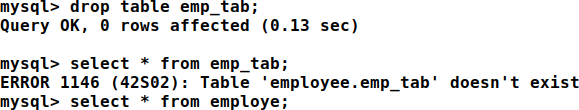
1. List the details of ‘EMPLOYEE’ and ‘EMPTAB’



1. Delete all records from EMP



1. Delete the table ‘EMP’



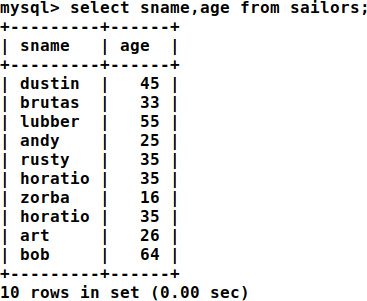
Date: 19-02-2024

# SET - 2

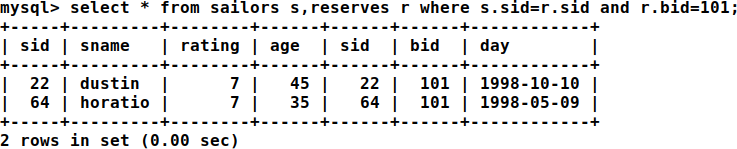
**AIM**

Create the following tables and execute the queries given below SAILORS

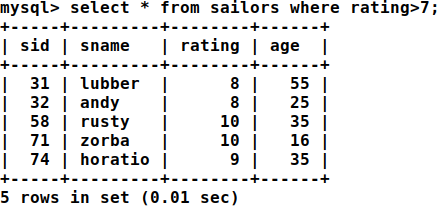
1. Find the names and ages of all sailors



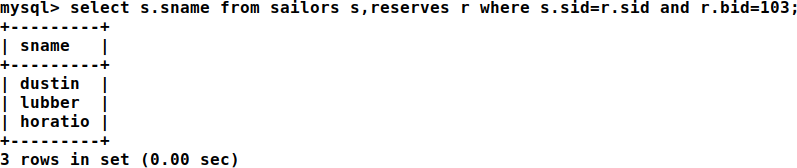
1. Find all information of sailors who have reserved boat number 101



1. Find all sailors with rating above 7.



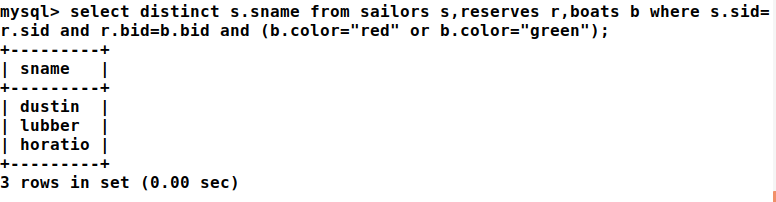
1. Find the names of sailors who have reserved boat no 103



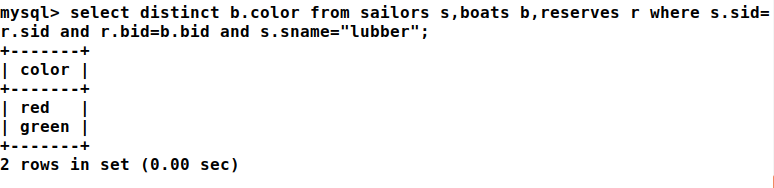
1. Find the names of sailors who have reserved a red boat, and list in the order of age



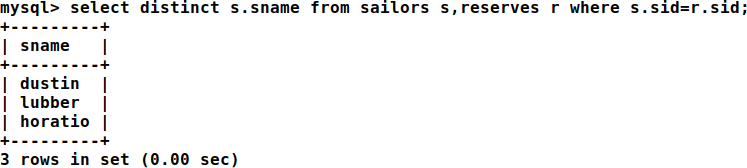
1. Find the names of sailors who have reserved either a red or green boat.



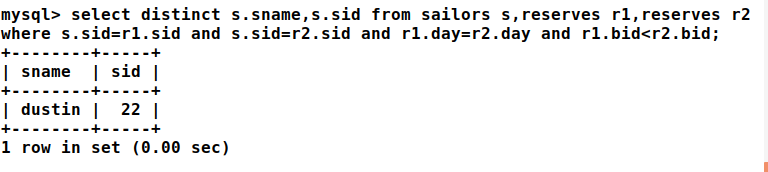
1. Find the colors of boats reserved by “Lubber”



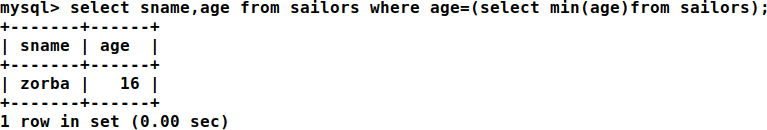
1. Find the names of sailors who have reserved at least one boat



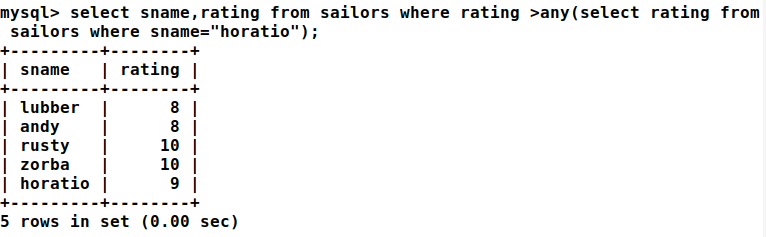
1. Find the ids and names of sailors who have reserved two different boats on the same day.



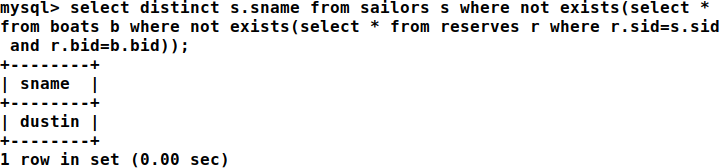
1. Find the name and the age of the youngest sailor.



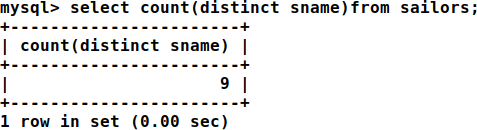
1. Find the names and ratings of a sailor whose rating is better than some sailor called Horatio.



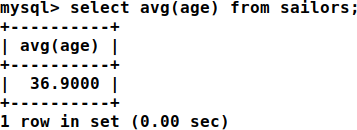
1. Find the names of sailors who have reserved all boats.



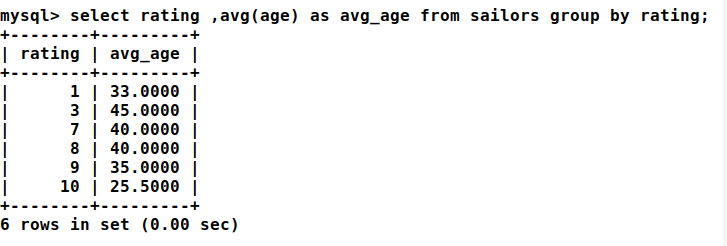
1. Count the number of different sailor names.



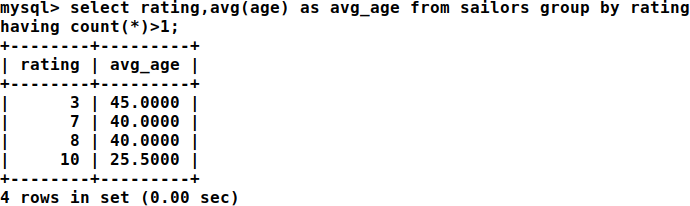
1. Calculate the average age of all sailors.



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



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



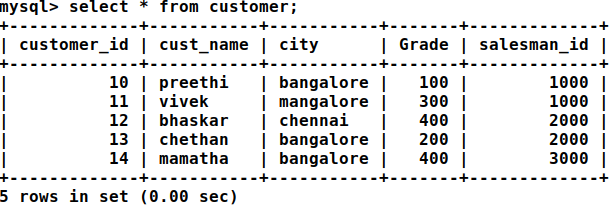
Date: 26-02-2024

# SET - 3

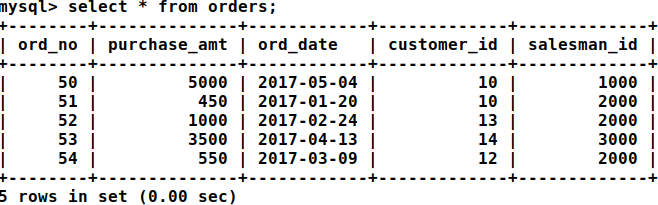
**AIM**

Consider the following schema for OrderDatabase: 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

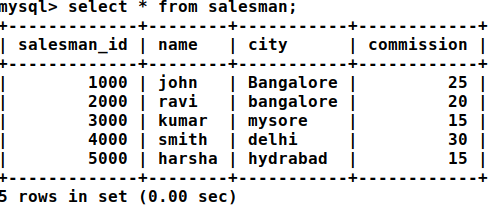
CUTOMER



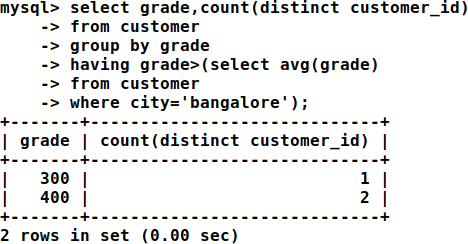
ORDERS



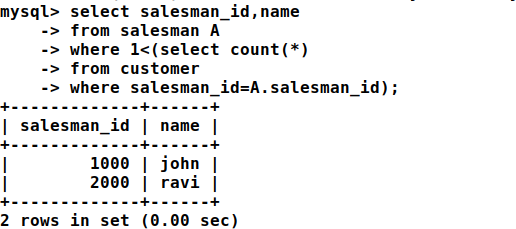
SALESMAN



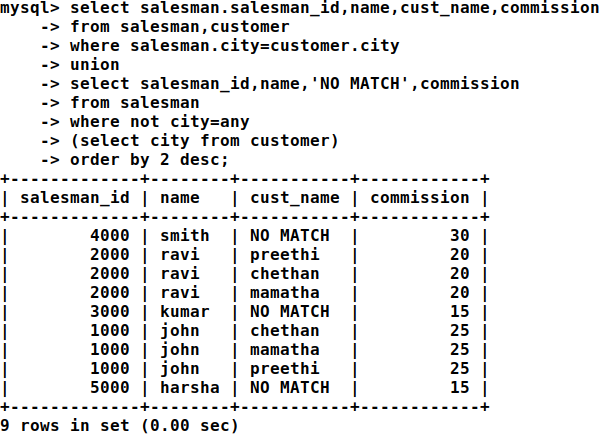
1. Count the customers with grades above Bangalore's Average.



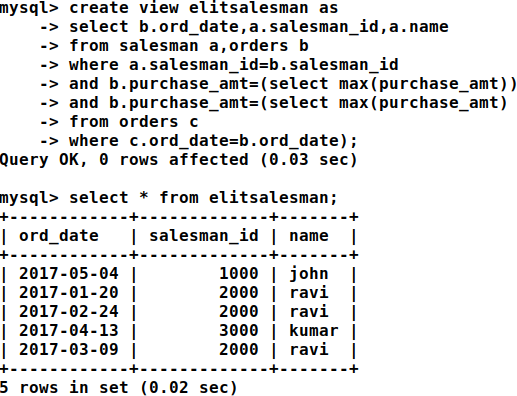
1. Find the name and numbers of all salesmen who had more than one customer



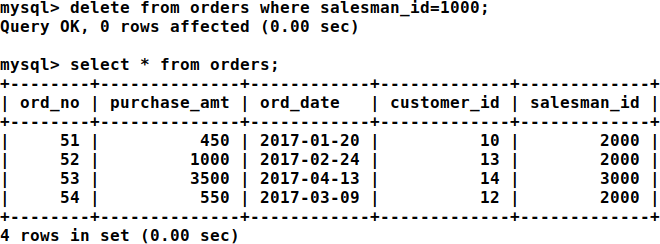
1. List all salesmen and indicate those who have and don’t have customers in their cities (Use UNION operation.)



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



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

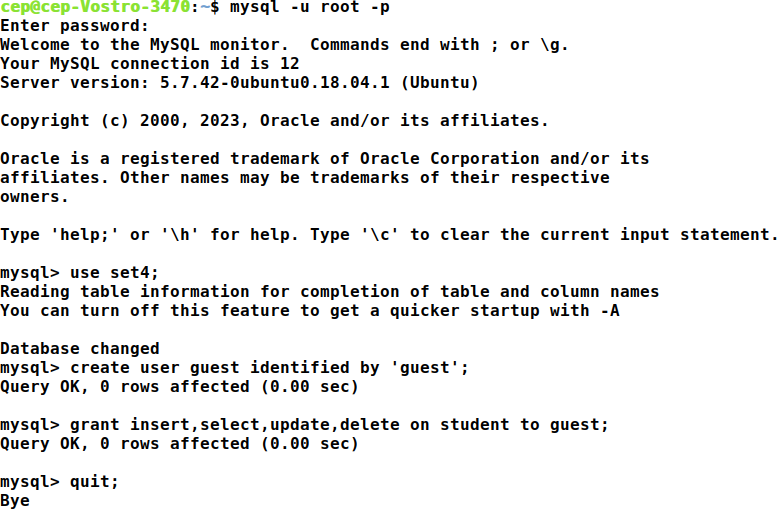


Date: 04-03-2024

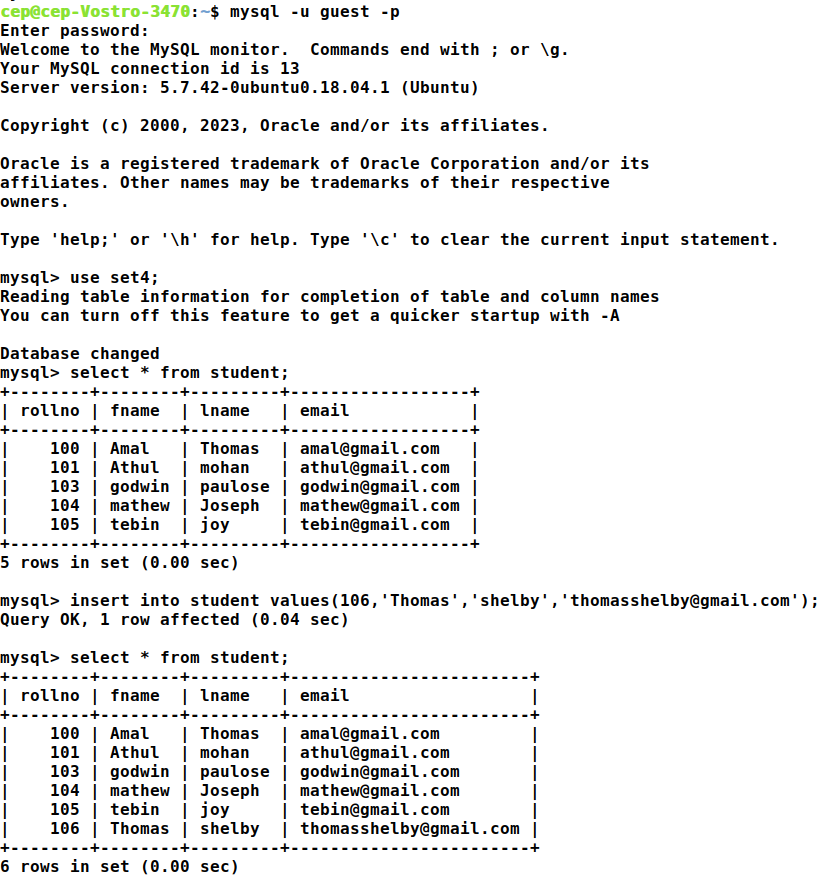
# SET - 4

## DCL

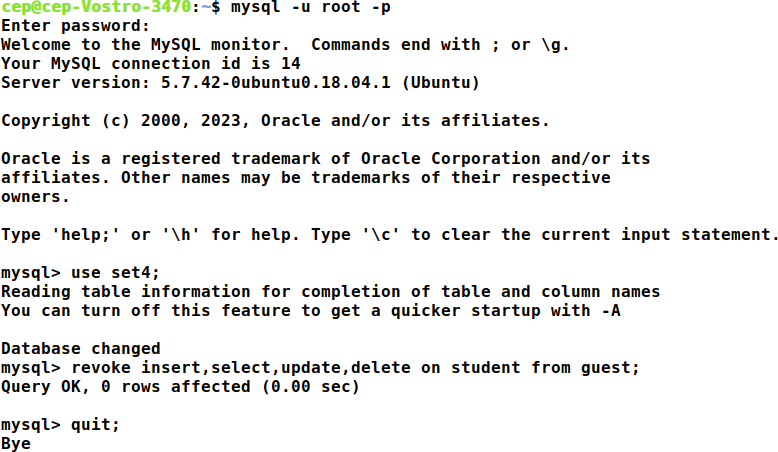
1. Creating a Guest User and set permissions INSERT, DELETE, SELECT, UPDATE

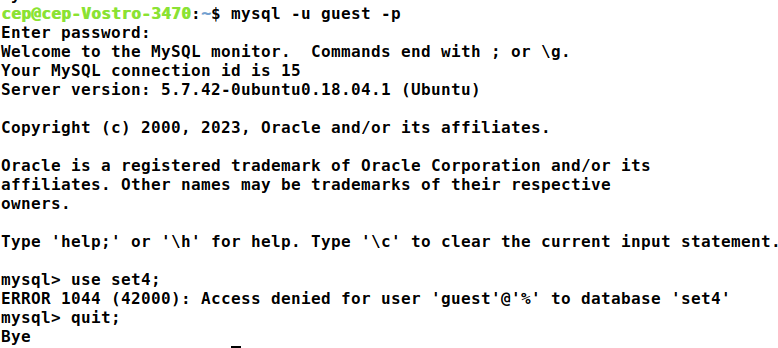


1. Perform INSERT, SELECT, UPDATE, DELETE operation in Guest mode

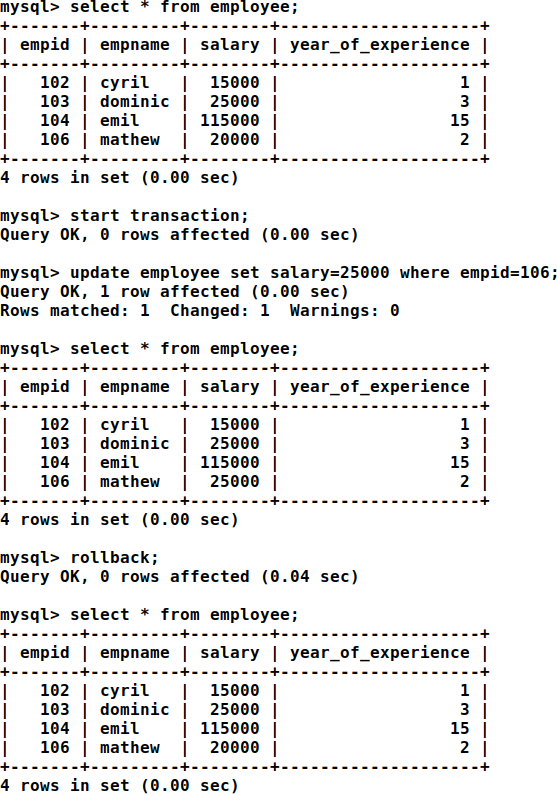


1. Revoke the Permissions





## TCL

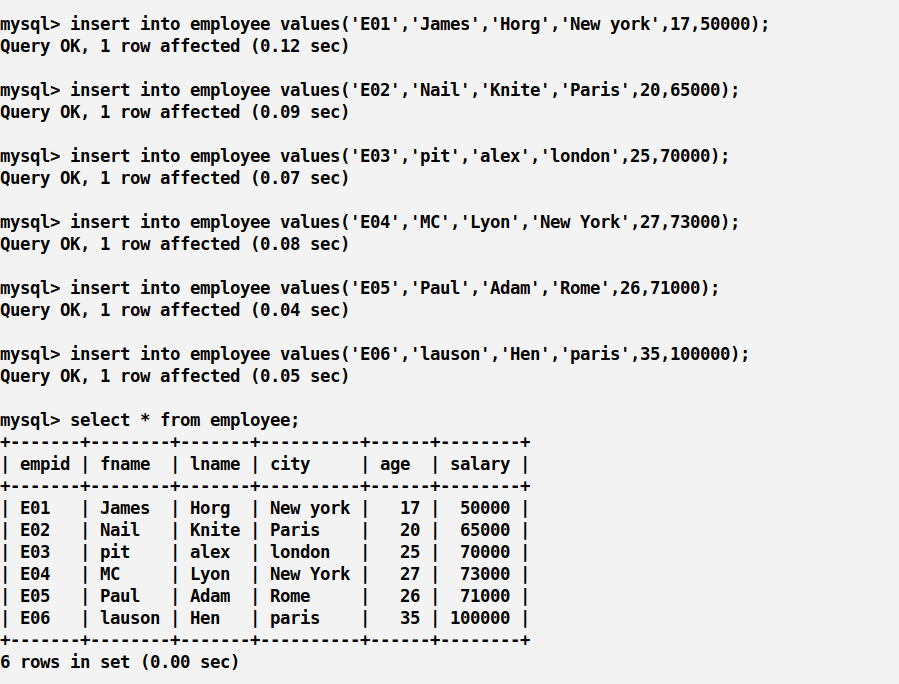
****

Date: 11-03-2024

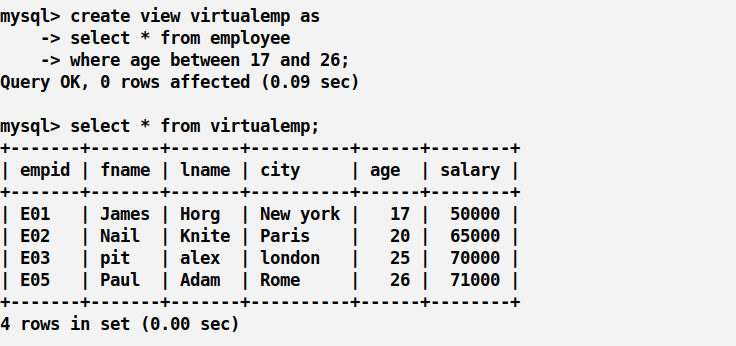
# SET – 5

**AIM**

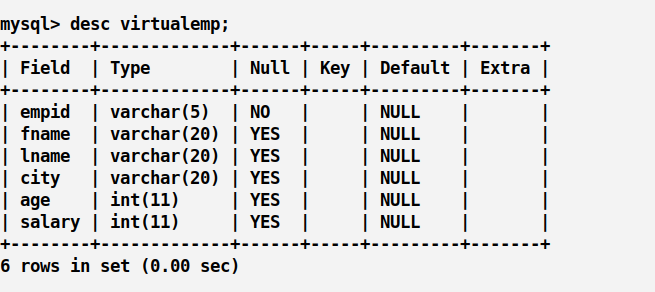
1. Create a table employee with following field. Empid, fname, lname, city, age, salary



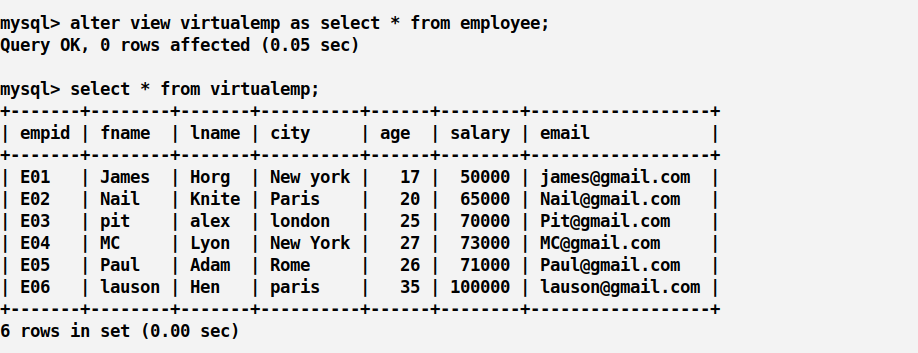
1. From the above Table, create a View virtualemp for these where age falls between 17 to 26



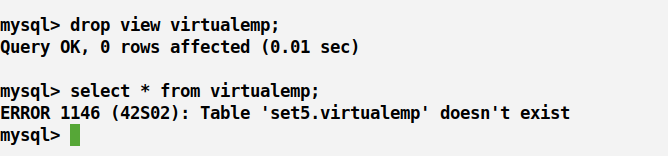
1. Describe the structure of the view table



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



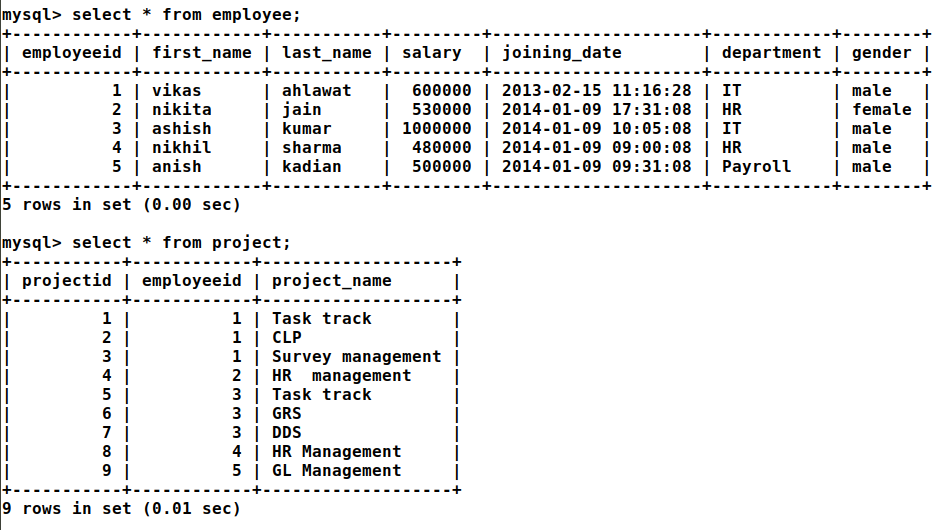
1. Drop view Table



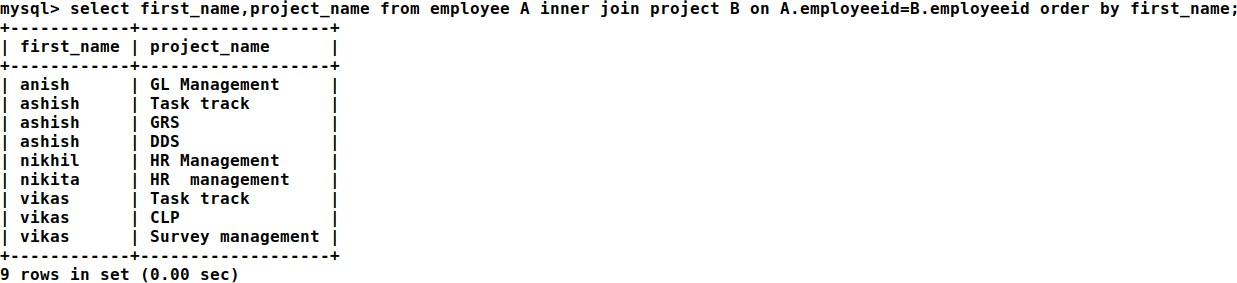
Date: 11-03-2024

# SET – 6

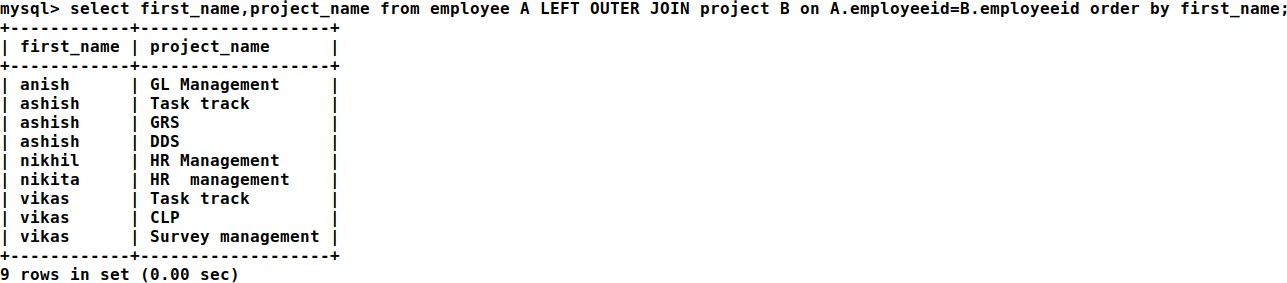
## JOIN OPERATIONS

****

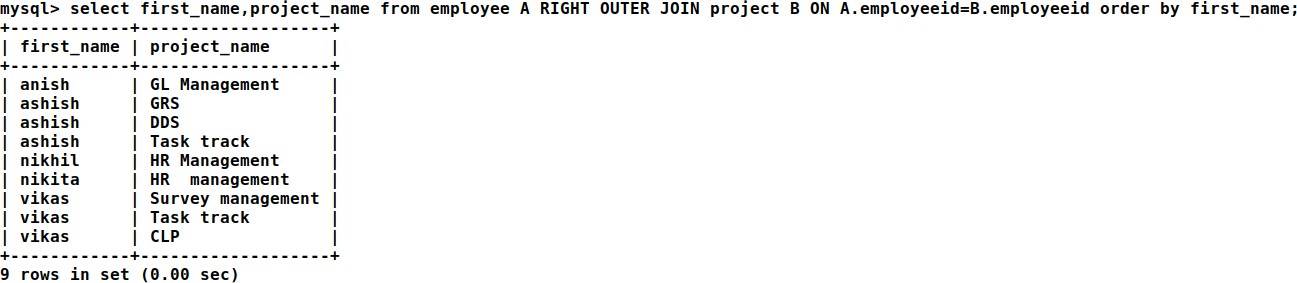
1. Get employee name, project name order by firstname from "EmployeeDetail" and "ProjectDetail" for those employees which have assigned projects already.



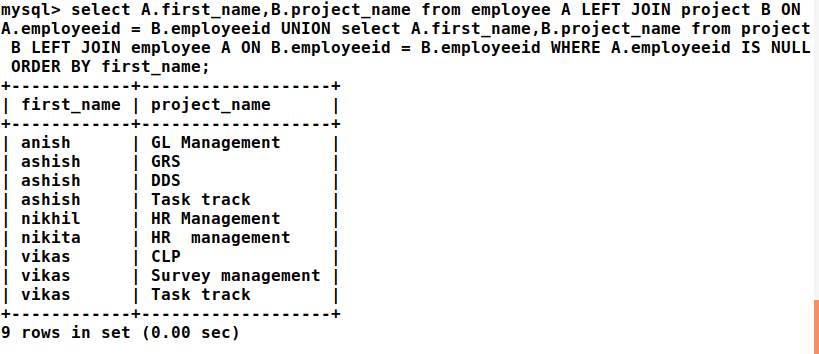
1. Get employee name, project name order by firstname from "EmployeeDetail" and "ProjectDetail" for all employees even if they have not assigned a project.



1. Get all project names even if they have not matching any employeeid, in the left table, order by firstname from "EmployeeDetail" and "ProjectDetail".



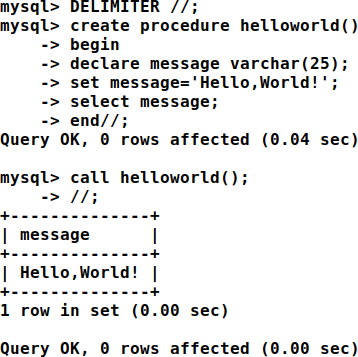
1. Get complete record(employeename, project name) from both tables([EmployeeDetail],[ProjectDetail]), if no match is found in any table then show NULL.



# SET - 7

## PL/SQL- PROGRAM

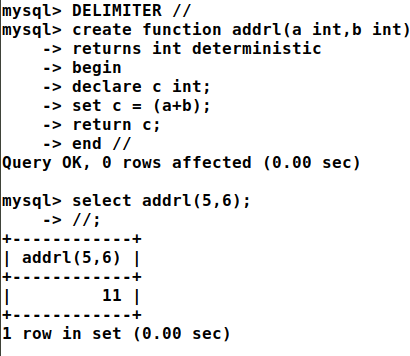
1. Write a Pl/SQL program to Q \*Hello world



# SET - 8

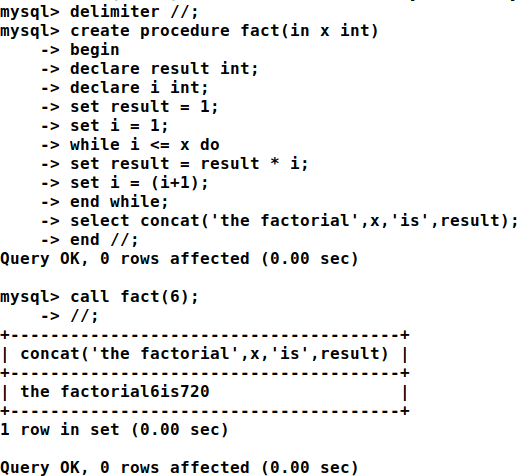
## PL/SQL-FUNCTION

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



## PL/SQL-PROCEDURE

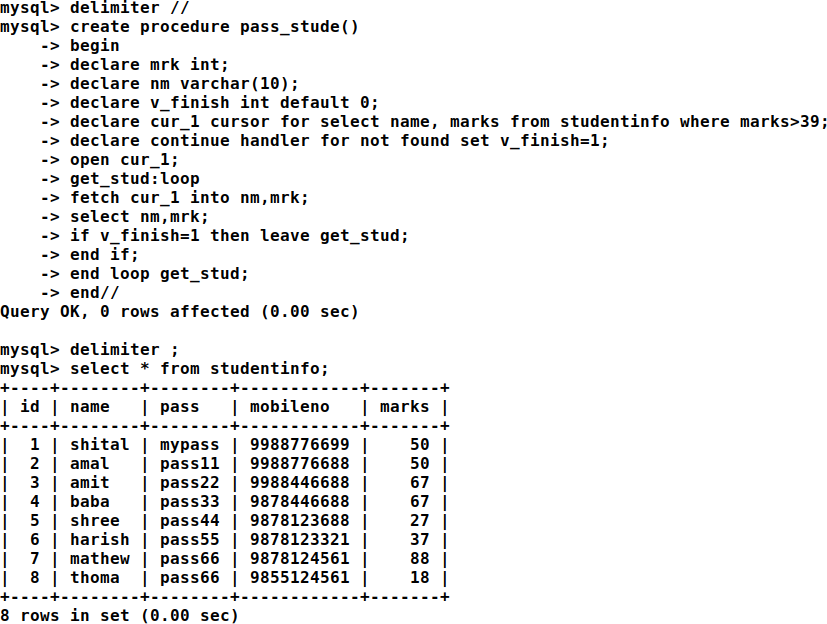
1. Write PL/SQL program to find Factorial of a Number using Procedure



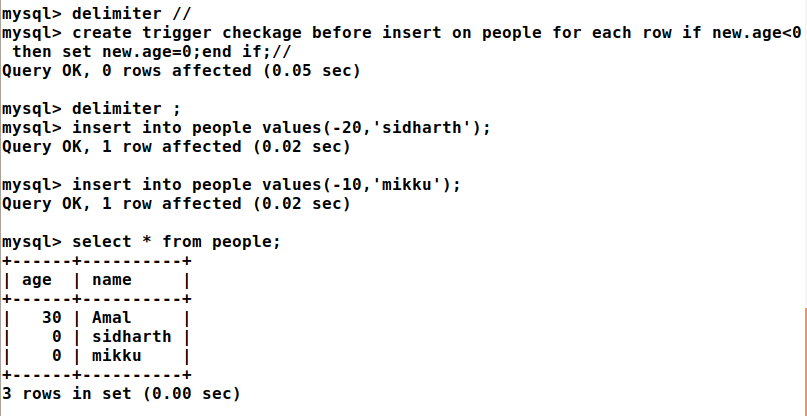
Date: 08-04-2024

# SET - 9

## PL/SQL-CURSOR

****

**PL/SQL -TRIGGER**

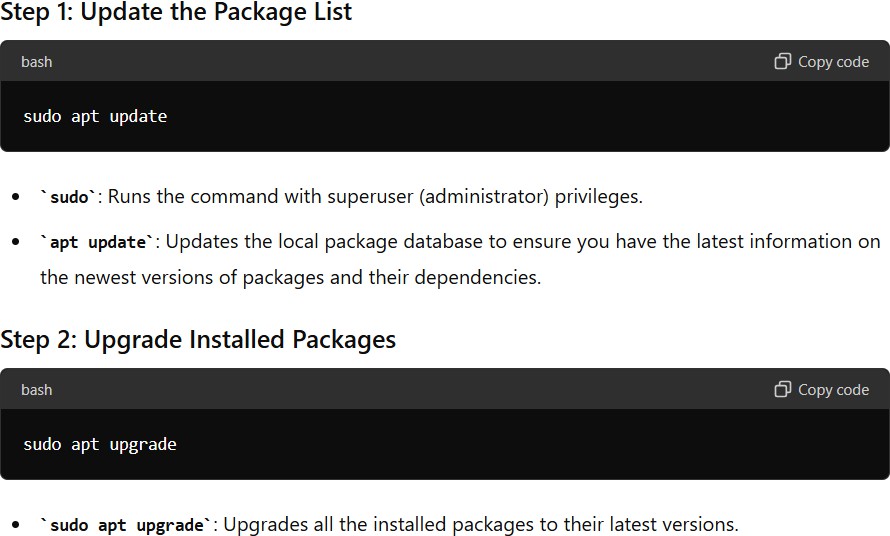
****

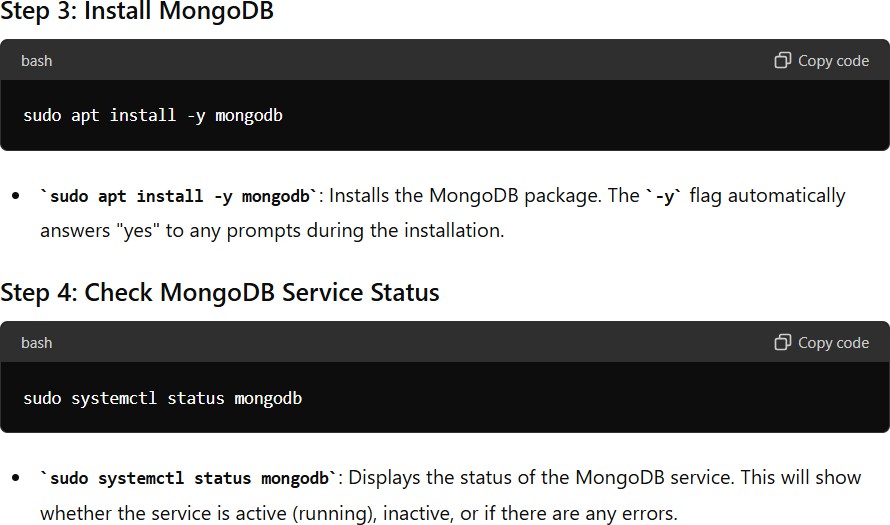
Date: 15-04-2024

# SET – 10

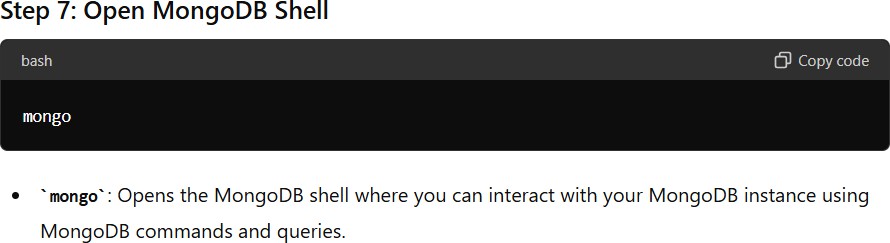
## MONGODB

***Installation***

******

******



******

Date: 15-04-2024

# SET – 11

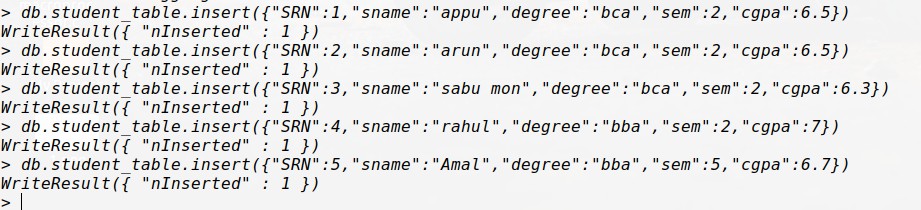
## MONGODB CURD OPERATIONS

### Student Database

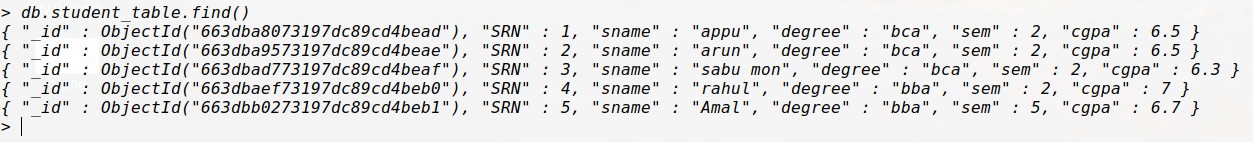
**Create a Student database with fields SRN, sname, degree, sem, cgpa**

Use student

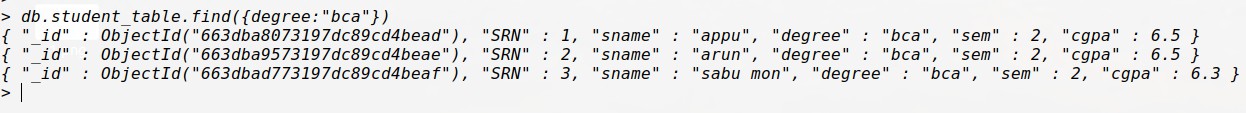
>db

student

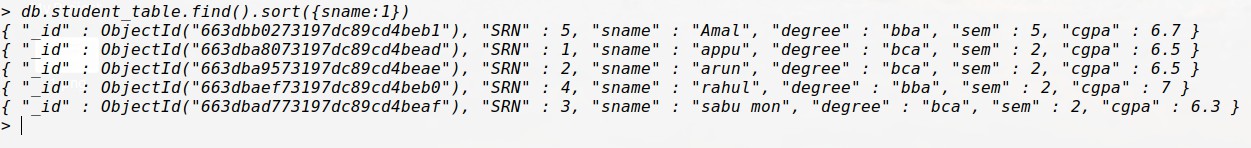
1. Display all the documents



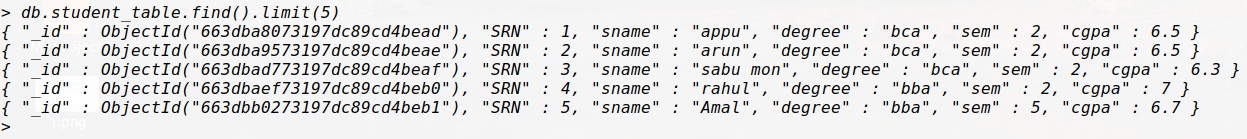
1. Display all the students in BCA



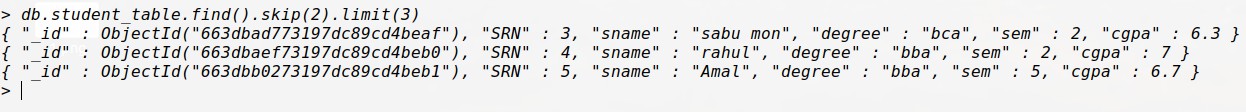
1. Display all students in ascending order



1. Display first 5 students



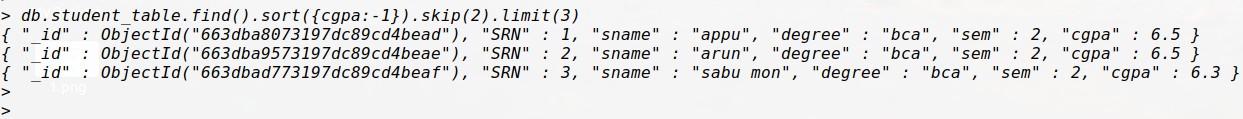
1. Display student 3,4,5



1. List the degree of student Rahul



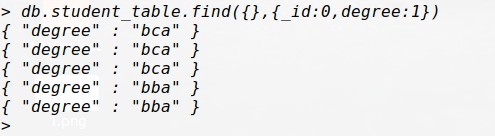
1. Display student details of 3,4,5 in descending order of cgpa



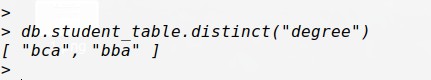
1. Display the no of students in BCA



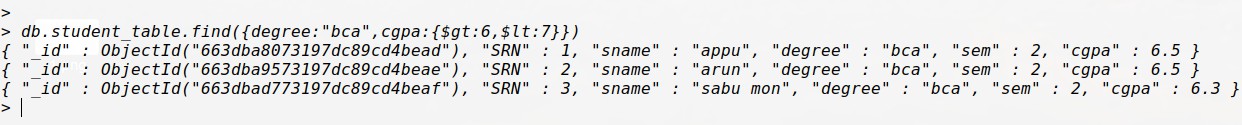
1. Display all degree without id



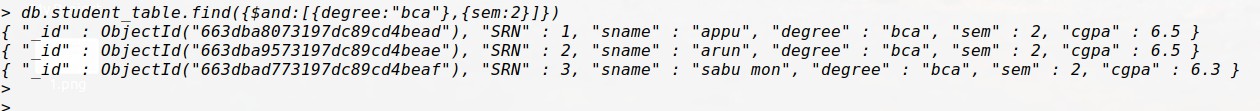
1. Display all distinct Degree



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



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



### Employee Database

**Update modifiers ($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}}**

****

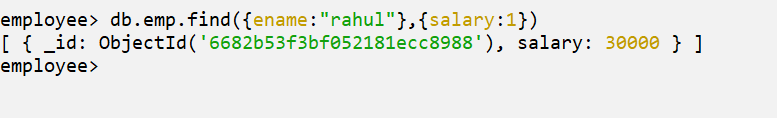
* 1. Display all employees with salary in range (3000,50000)



* 1. Display all the employees with desig developer



* 1. Display the salary of “Rahul”



* 1. Display city of employee Rahul



* 1. Update the salary of developers by 5000 increment



* 1. Add field age to employee Rahul



* 1. Remove yoj from Rahul



* 1. Add an array field project to Rahul



* 1. Add p2 and p3 project to Rahul





* 1. Remove p3 from Rahul



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





* 1. Add two phone numbers to Rahul



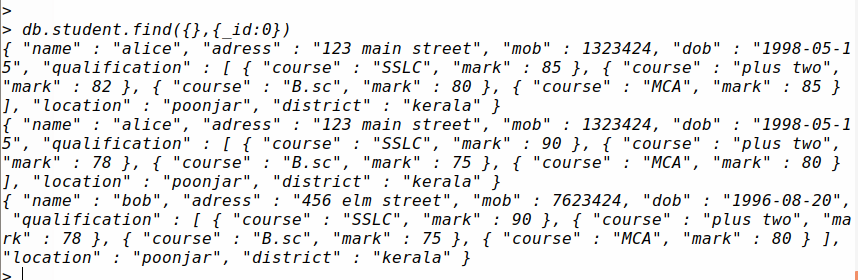
Date: 22-04-2024

# SET – 12

**AGGREGATE OPERATIONS**

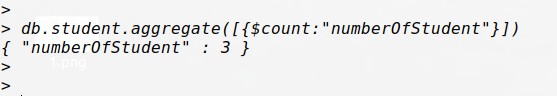
Assume that you have two collection, student and college

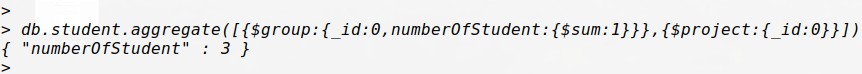
1. Documents in student collection contains 'name, address, mob, date of brith, Qualification (incluing course (starting from sslc,plus to, bsc, MCA,mark), location district
2. Document in college collection include 'name,location,established year,district'



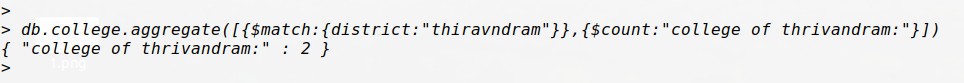


1. find out the total number of students in collection

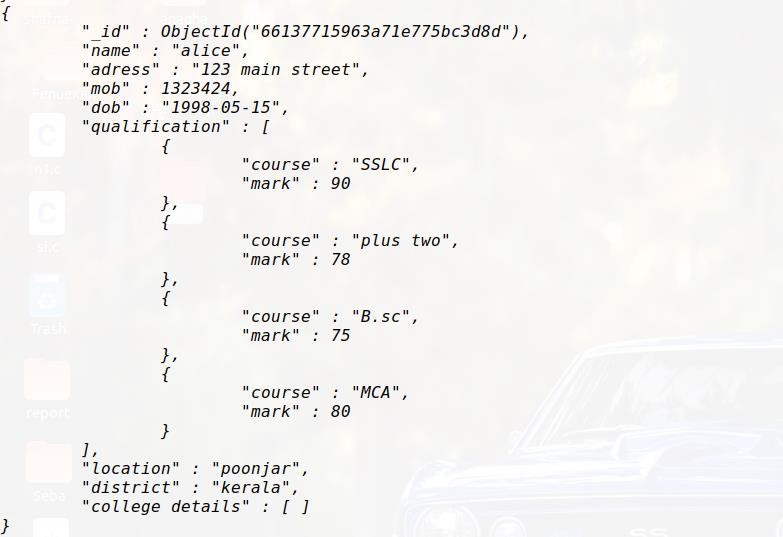
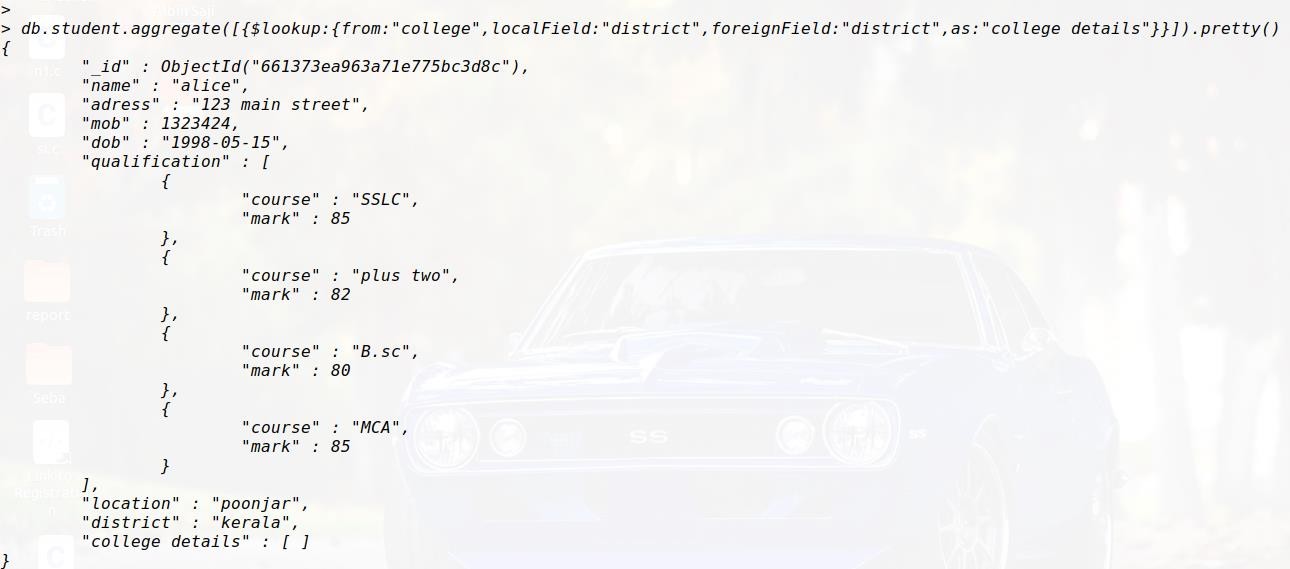


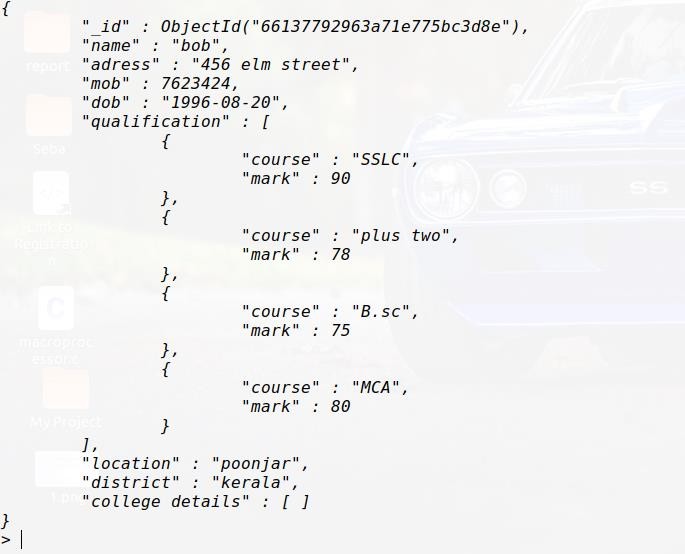


1. How many colleges are in district TVM

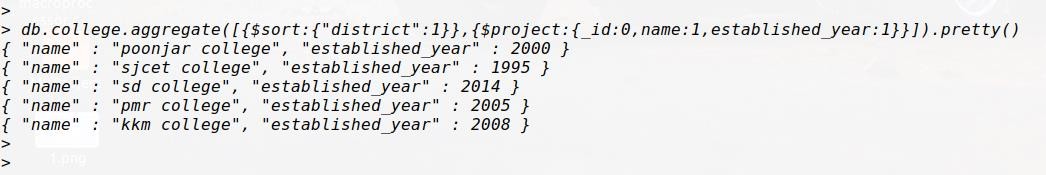


1. Dispaly the details of student and college residing in same district

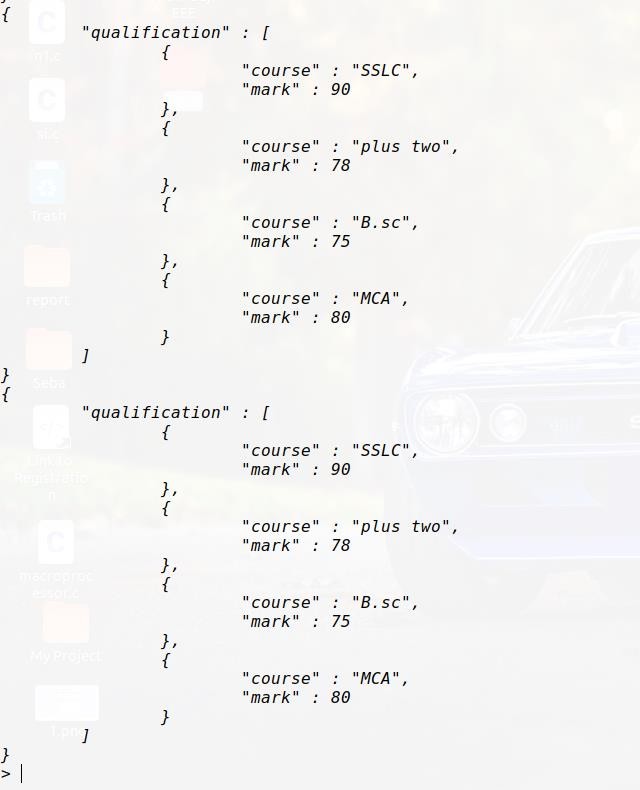
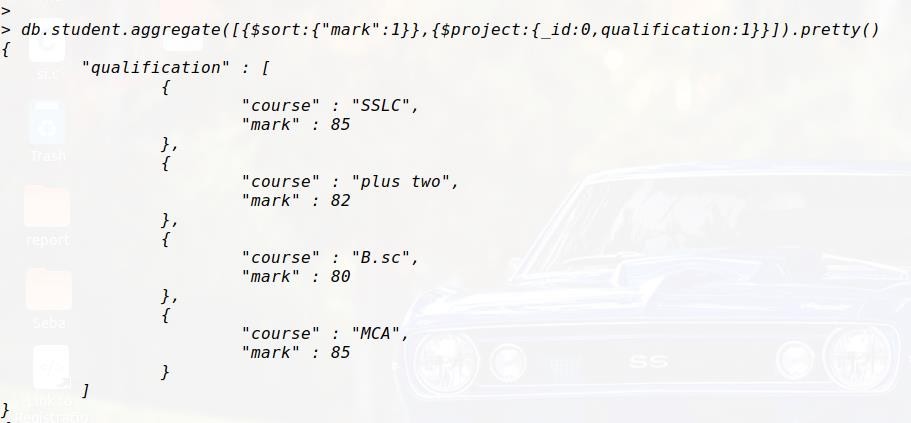




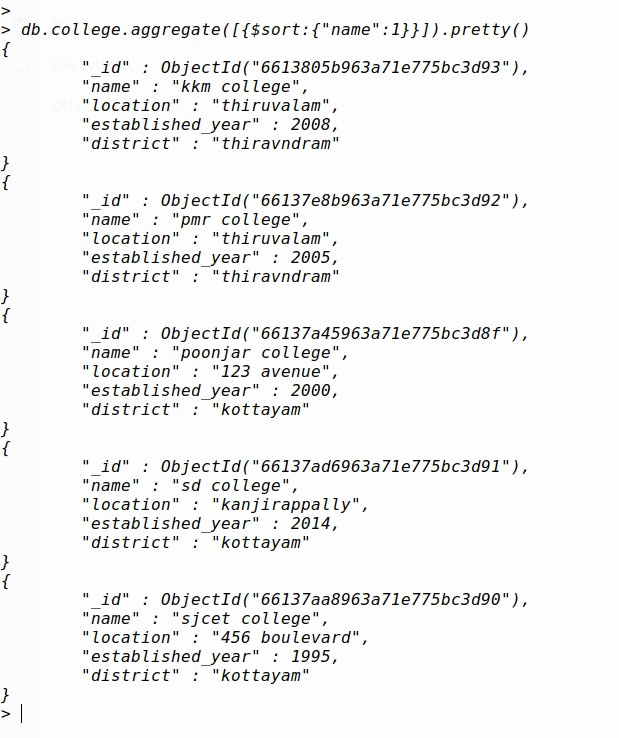
1. Display the name and establishment year of college in district ways



1. Display the course and marks



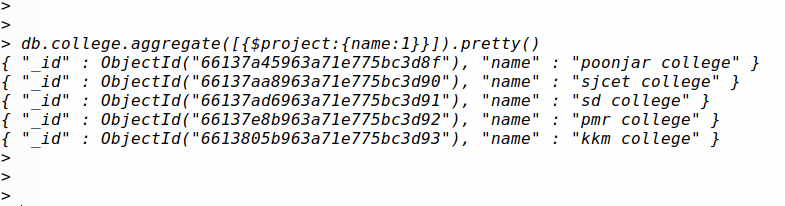
1. Display the details of college in sorted order( name)



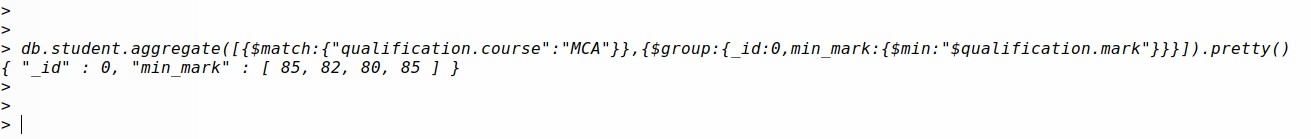
1. List the details of college in descenting order based on the establishment year.



1. Display the colleges

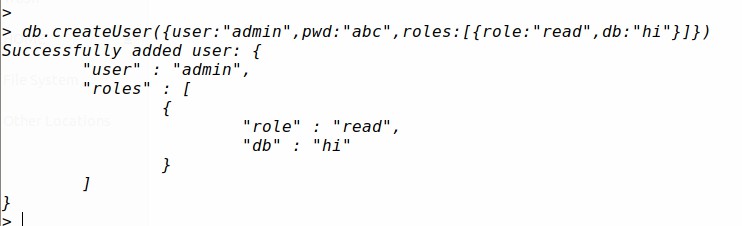


1. Display the marks of MCA



# SET – 13

1. Create Users and Roles



# SET – 14

## PYMONGO

PyMongo library, which is a Python driver for MongoDB, a popular NoSQL database

1. Importing the necessary modules

### Import pymongo

1. Creating a MongoDit client and connecting to a database:

### Myclient = pymongo.MongoClient("mongodb://localhost:27017/")

1. Accessing a specifie database:

### mydb = myclient["mydatabase"]

1. Accessing a specific collection within the database:

### mycol = mydb["customers"]

1. Inserting a single document into the collection:

### Myd ={"name": "Divya","address":"highway37"} q = mycol.insert\_one(myd) print(q.inserted\_id)

1. Inserting multiple documents into the collection:

**mydict = [**

**{"name": "John", "address": "highway 37"},**

**{"name": "Aby", "address": "Cross 30"},**

**{"name": "Jerry", "address": "River Road 45"}**

**]**

### x = mycol.insert\_many(mydict) print(x.inserted\_ids)

1. Retrieving a single document from the collection:

### y=mycol.find\_one() print(y)

1. Retrieving all documents from the collection:

### for z in mycol.find() print(z)

1. Retrieving documents while excluding the "name" field:

### for a in mycol.find({}, {"name":0}); print(a)

1. Deleting a document from the collection:

### myquery = {"name": "John"} mycol.delete\_one(myquery)

1. Sorting the retrieved documents by the "name" field:

### mydoc = mycol.find().sort("name") for l in mydoc: print(l)

1. Dropping the collection (deleting all documents within it):

### mycol.drop()

Date: 06-05-2024

# SET – 15

1. **MONGODB DATABASE CONNECTION**

****

**OUTPUT**

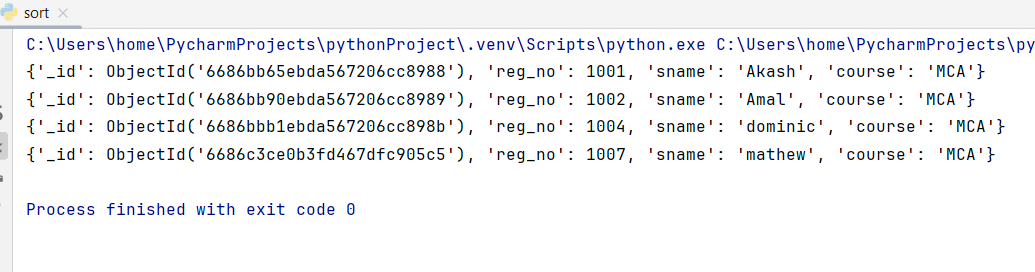


****

1. **SORTING**

****

**OUTPUT**

****