

**M.Sc. (Five Year Integrated) in Computer Science  
(Artificial Intelligence & Data Science)**

**Third Semester**

**Laboratory Record**

**21-805-0307: DATABASE SYSTEMS LAB**

*Submitted in partial fulfillment  
of the requirements for the award of degree in  
Master of Science (Five Year Integrated)  
in Computer Science (Artificial Intelligence & Data Science) of  
Cochin University of Science and Technology (CUSAT)  
Kochi*



*Submitted by*

**ABHIN P.T  
(80521002)**

**DEPARTMENT OF COMPUTER SCIENCE  
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY (CUSAT)  
KOCHI-682022**

**JANUARY 2023**

**DEPARTMENT OF COMPUTER SCIENCE**  
**COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY (CUSAT)**  
**KOCHI, KERALA-682022**



*This is to certify that the software laboratory record for **21-805-0307: Database Systems Lab** is a record of work carried out by **ABHIN P.T (80521002)**, in partial fulfillment of the requirements for the award of degree in **Master of Science (Five Year Integrated) in Computer Science (Artificial Intelligence & Data Science)** of Cochin University of Science and Technology (CUSAT), Kochi. The lab record has been approved as it satisfies the academic requirements in respect of the second semester laboratory prescribed for the Master of Science (Five Year Integrated) in Computer Science degree.*

**Faculty Member in-charge**

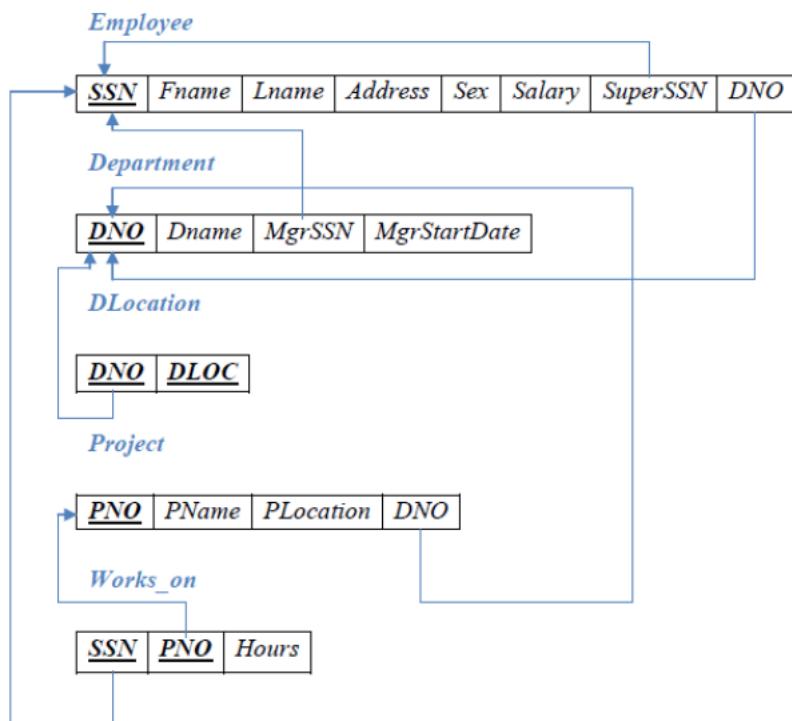
Sajmi Salaam  
Guest Faculty  
Department of Computer Science  
CUSAT

Dr. Philip Samuel  
Professor and Head  
Department of Computer Science  
CUSAT

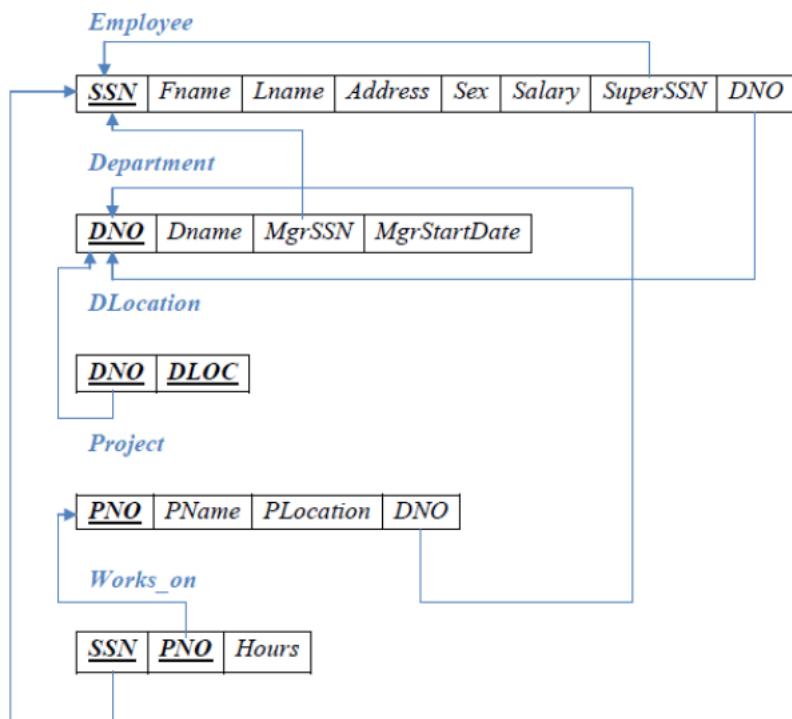
**Table of Contents**

| <b>Sl.No.</b> | <b>Program</b>                                    | <b>Page.No.</b> |
|---------------|---|-----------------|
| 1             | Schema Diagram and ER Diagram                     | 1               |
| 2             | Queries to implement DDL Commands                 | 2               |
| 3             | Queries to implement DML Commands                 | 6               |
| 4             | Queries to implement DCL Commands.                | 11              |
| 5             | Queries to implement Group Functions              | 12              |
| 6             | Program to implement Nested Queries               | 15              |
| 7             | Program to implement Views                        | 26              |
| 8             | Programs of Functions And Procedures              | 29              |
| 9             | Implementation of Cursor                          | 32              |
| 10            | Implementation of Trigger                         | 35              |
| 11            | Queries to implement TCL Commands                 | 41              |
| 12            | Operations on NOSQL Systems                       | 44              |
| 13            | Simple Structure of GraphQL program               | 47              |
| 14            | Programs demonstrating Java Database Connectivity | 52              |
| 15            | Project Report on Application Software            | 52              |

## SCHEMA DIAGRAM



## ER DIAGRAM



## DDL COMMANDS

### AIM

Develop SQL Queries to execute and verify the Data Definition Language commands and also implement Data Constraints.

### Questions : 1

Create five tables using constraints like primary key, not null, check, default, null, unique, foreign key as per the above schema

### QUERY

```
create database company;

use company;

create table employee(SSN varchar(20),Name char(30),Address varchar(60),sex char(10),
salary int(90),superSSN varchar(20),DNo varchar(20));

create table department(DNo varchar(20),Dname char(50),
MgrSSN varchar(30),MgrStartDate date);

create table dlocation(DNo varchar(30),DLoc varchar(90));

create table project(PNo varchar(30),Pname char(90),Plocation varchar(80),
DNo varchar(50));

create table works_on(SSN varchar(60),PNo varchar(50),Hours time);

alter table employee add constraint primary key(SSN);
alter table department add constraint primary key(DNo);
alter table project add constraint primary key(PNo);
alter table employee add foreign key(DNo) references department(DNo);
alter table department add foreign key(MgrSSN)
references employee(SSN);
alter table dlocation add foreign key(DNo) references employee(DNo);
alter table works_on add foreign key(SSN) references employee(SSN);

desc employee;
desc works_on;
desc project;
desc dlocation;
desc department;
```

**DATABASE TABLES**

```
mysql> desc employee;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| SSN   | varchar(20) | NO   | PRI   | NULL    |
| Name  | char(30)    | YES  |       | NULL    |
| Address | varchar(60) | YES  |       | NULL    |
| sex   | char(10)    | YES  |       | NULL    |
| salary | int      | YES  |       | NULL    |
| superSSN | varchar(20) | YES  |       | NULL    |
| DNo   | varchar(20) | YES  | MUL   | NULL    |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> desc works_on;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| SSN   | varchar(60) | YES  | MUL   | NULL    |
| PNo   | varchar(50)  | YES  |       | NULL    |
| Hours | time     | YES  |       | NULL    |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> desc project;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| PNo   | varchar(30) | NO   | PRI   | NULL    |
| Pname | char(90)   | YES  |       | NULL    |
| Plocation | varchar(80) | YES  |       | NULL    |
| DNo   | varchar(50)  | YES  |       | NULL    |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc department;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| DNo   | varchar(20) | NO   | PRI   | NULL    |
| Dname | char(50)   | YES  |       | NULL    |
| MgrSSN | varchar(30) | YES  | MUL   | NULL    |
| MgrStartDate | date | YES  |       | NULL    |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc dlocation;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| DNo   | varchar(30) | YES  | MUL   | NULL    |
| DLoc  | varchar(90)  | YES  |       | NULL    |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

**Questions : 2**

Add another column Age with datatype integer in employee table

**QUERY**

```
alter table employee add column Age int;
desc employee;
```

**DATABASE TABLES**

```
mysql> desc employee;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| SSN   | varchar(20) | NO   | PRI  | NULL    |       |
| Name  | char(30)    | YES  |       | NULL    |       |
| Address | varchar(60) | YES  |       | NULL    |       |
| sex   | char(10)    | YES  |       | NULL    |       |
| salary | int        | YES  |       | NULL    |       |
| superSSN | varchar(20) | YES  |       | NULL    |       |
| DNo   | varchar(20) | YES  | MUL  | NULL    |       |
| Age   | int        | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

**Questions : 3**

Drop a table named Project

**QUERY**

```
drop table project;
desc project;
```

Query OK, 0 rows affected (0.01 sec)

**DATABASE TABLES**

```
mysql> desc project;
ERROR 1146 (42S02): Table 'company.project' doesn't exist
```

**Questions : 4**

Truncate a table named WORKS\_ON

**QUERY**

```
TRUNCATE TABLE works_on;
```

**DATABASE TABLES**

```
mysql> desc works_on;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| SSN   | varchar(60) | YES  | MUL | NULL    |
| PNo   | varchar(50)  | YES  |      | NULL    |
| Hours | time       | YES  |      | NULL    |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

**Questions : 5**

View the structure of the table Department

**QUERY**

```
desc department;
```

**DATABASE TABLES**

```
mysql> desc department;
+-----+-----+-----+-----+-----+
| Field     | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| DNo       | varchar(20) | NO   | PRI | NULL    |
| Dname     | char(50)   | YES  |      | NULL    |
| MgrSSN    | varchar(30) | YES  | MUL | NULL    |
| MgrStartDate | date     | YES  |      | NULL    |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## DML COMMANDS

### AIM

Develop SQL Queries to execute and verify the Data Manipulation Language commands.

### Questions : 1

Insert five records in the tables as per the above schema

### QUERY

```
SET FOREIGN_KEY_CHECKS=0;
insert into employee values('1001','Vasu','hn1,vadayam,kuttiadi,kozhikode',
'm',50000,'s001','d001',45);
insert into employee values('1002','Haridasan','hn2,vadayam,kuttiadi,kozhikode',
'm',55000,'s002','d002',51);
insert into employee values('1003','Sajeevan','hn3,vadayam,kuttiadi,kozhikode',
'm',59000,'s003','d003',48);
insert into employee values('1004','Dineshan','hn4,vadayam,kuttiadi,kozhikode',
'm',55000,'s004','d004',46);
insert into employee values('1005','Janaki','hn5,vadayam,kuttiadi,kozhikode',
'f',38000,'s005','d005',53);
insert into department values('d1','computer science','1005','2022-04-16');
insert into department values('d2','DOE','1002','2022-01-15');
insert into department values('d3','SMS','1001','2022-08-10');
insert into department values('d4','SLS','1003','2022-12-12');
insert into department values('d5','computer science','1004','2022-04-16');
insert into dlocation values('d1','near university library');
insert into dlocation values('d2','1st floor of dcs dept');
insert into dlocation values('d3','near sbi cusat');
insert into dlocation values('d4','opposite of cusat sports arena');
insert into dlocation values('d5','opposite of cusat sports arena');
insert into works_on values('1001','p1','10:00:00');
insert into works_on values('1002','p3','06:00:00');
insert into works_on values('1003','p2','12:00:00');
insert into works_on values('1004','p5','09:00:00');
insert into works_on values('1005','p4','11:00:00');
SET FOREIGN_KEY_CHECKS=1;
```

**Questions : 2**

Display the entire content of the tables as per the above schema

**QUERY**

```
select * from employee;
select * from department;
select * from works_on;
select * from dlocation;
```

**DATABASE TABLES**

```
mysql> select * from employee;
+----+-----+-----+-----+-----+-----+-----+-----+
| SSN | Name      | Address          | sex | salary | superSSN | DNo | Age |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1001 | Vasu       | hn1,vadayam,kuttiadi,kozhikode | m   | 1000   | s001    | d001 | NULL |
| 1002 | Haridasan  | hn2,vadayam,kuttiadi,kozhikode | m   | 55000  | s002    | d002 | 51   |
| 1003 | Sajeevan    | hn3,vadayam,kuttiadi,kozhikode | m   | 59000  | s003    | d003 | NULL |
| 1004 | Dineshan    | hn4,vadayam,kuttiadi,kozhikode | m   | 55000  | s004    | d004 | NULL |
| 1005 | Janaki     | hn5,vadayam,kuttiadi,kozhikode | f   | 38000  | s005    | d005 | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select * from department;
+----+-----+-----+-----+
| DNo | Dname           | MgrSSN | MgrStartDate |
+----+-----+-----+-----+
| d1  | computer science | 1005   | 2022-04-16   |
| d2  | DOE              | 1002   | 2022-01-15   |
| d3  | SMS               | 1001   | 2022-08-10   |
| d4  | SLS               | 1003   | 2022-12-12   |
| d5  | computer science | 1004   | 2022-04-16   |
+----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select * from works_on;
+----+----+-----+
| SSN | PNo | Hours        |
+----+----+-----+
| 1001 | p1  | 10:00:00 |
| 1002 | p3  | 06:00:00 |
| 1003 | p2  | 12:00:00 |
| 1004 | p5  | 09:00:00 |
| 1005 | p4  | 11:00:00 |
+----+----+-----+
5 rows in set (0.00 sec)
```

```
mysql> select * from dlocation;
+----+-----+
| DNo | DLoc          |
+----+-----+
| d1  | near university library |
| d2  | 1st floor of dcs dept   |
| d3  | near sbi cusat         |
| d4  | oposite of cusat sports arena |
| d5  | oposite of cusat sports arena |
+----+-----+
5 rows in set (0.00 sec)
```

**Questions : 3**

Modify the salary of the employee as 25000 whose SSN is e1001

**QUERY**

```
update employee set salary = 25000 where SSN = '1001';
```

**DATABASE TABLES**

```
mysql> select * from employee;
+----+-----+-----+-----+-----+-----+-----+-----+
| SSN | Name   | Address          | sex | salary | superSSN | DNo | Age |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1001 | Vasu    | hn1,vadayam,kuttiadi,kozhikode | m   | 25000 | s001    | d001 | NULL |
| 1002 | Haridasan | hn2,vadayam,kuttiadi,kozhikode | m   | 55000 | s002    | d002 | 51   |
| 1003 | Sajeevan | hn3,vadayam,kuttiadi,kozhikode | m   | 59000 | s003    | d003 | NULL |
| 1004 | Dineshan | hn4,vadayam,kuttiadi,kozhikode | m   | 55000 | s004    | d004 | NULL |
| 1005 | Janaki   | hn5,vadayam,kuttiadi,kozhikode | f   | 38000 | s005    | d005 | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

**Questions : 4**

Delete the details of the employee whose SSN is "e1002"

**QUERY**

```
delete from employee where SSN = '1002';
```

**DATABASE TABLES**

```
mysql> select * from employee;
+----+-----+-----+-----+-----+-----+-----+-----+
| SSN | Name   | Address          | sex | salary | superSSN | DNo | Age |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1001 | Vasu    | hn1,vadayam,kuttiadi,kozhikode | m   | 25000 | s001    | d001 | NULL |
| 1003 | Sajeevan | hn3,vadayam,kuttiadi,kozhikode | m   | 59000 | s003    | d003 | NULL |
| 1004 | Dineshan | hn4,vadayam,kuttiadi,kozhikode | m   | 55000 | s004    | d004 | NULL |
| 1005 | Janaki   | hn5,vadayam,kuttiadi,kozhikode | f   | 38000 | s005    | d005 | NULL |
+----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## DCL COMMANDS

### AIM

Develop SQL Queries to implement Data Control Language commands

### Questions : 1

To grant a SELECT permission on employee table to user1

### QUERY

```
create user 'user1'@'localhost' IDENTIFIED by 'password';  
  
grant select on company.employee to 'user1'@'localhost';
```

### DATABASE TABLES

```
mysql> show grants for 'user1'@'localhost';  
+-----+  
| Grants for user1@localhost |  
+-----+  
| GRANT USAGE ON *.* TO 'user1'@'localhost'  
| GRANT SELECT ON 'company'.'employee' TO 'user1'@'localhost'  
+-----+  
2 rows in set (0.00 sec)
```

### Questions : 2

Revoking a privilege to all users in a table

### QUERY

```
revoke select on employee from 'user1'@'localhost';
```

### DATABASE TABLES

```
mysql> show grants for 'user1'@'localhost';  
+-----+  
| Grants for user1@localhost |  
+-----+  
| GRANT USAGE ON *.* TO 'user1'@'localhost' |  
+-----+  
1 row in set (0.00 sec)
```

## GROUP FUNCTION OR AGGREGATE FUNCTION

### AIM

Develop SQL Queries to execute computation on table data with built-in functions

### Questions : 1

List the fname of all the employee having ‘a’ as the second last character in their name.

### QUERY

```
select name from employee where name like "%a_";
```

### DATABASE TABLES

|   | name     |
|---|----------|
| ▶ | Sajeevan |
|   | Dineshan |

### Questions : 2

Count the total number of male and female employees in the Employee table.

### QUERY

```
select Sex,count(Sex) from employee group by Sex;
```

### DATABASE TABLES

|   | Sex | count(Sex) |
|---|-----|------------|
| ▶ | m   | 3          |
|   | f   | 1          |

### Questions : 3

Calculate the average salary of the female employees.

### QUERY

```
select avg(Salary) from employee where Sex = "F" ;
```

**DATABASE TABLES**

|   |             |
|---|-------------|
|   | avg(Salary) |
| ▶ | 38000.0000  |

**Questions : 4**

Calculate the sum of salaries of male employees.

**QUERY**

```
select sum(Salary) from employee where Sex = "M";
```

**DATABASE TABLES**

|   |             |
|---|-------------|
|   | sum(Salary) |
| ▶ | 139000      |

**Questions : 5**

Display the maximum and minimum salaries of male employees.

**QUERY**

```
select max(Salary),min(Salary) from employee where Sex = "M";
```

**DATABASE TABLES**

|   |             |             |
|---|-------------|-------------|
|   | max(Salary) | min(Salary) |
| ▶ | 59000       | 25000       |

**Questions : 6**

Display the details of all employees whose salary between 25000 and 50000

**QUERY**

```
select * from employee where salary between 25000 and 50000;
```

**DATABASE TABLES**

|   | SSN  | Name   | Address                        | sex | salary | superSSN | DNo  | Age  |
|---|------|--------|--------------------------------|-----|--------|----------|------|------|
| ▶ | 1001 | Vasu   | hn1,vadayam,kuttiadi,kozhikode | m   | 25000  | s001     | d001 | NULL |
|   | 1005 | Janaki | hn5,vadayam,kuttiadi,kozhikode | f   | 38000  | s005     | d005 | NULL |

**Questions : 7**

Display the lname of the employees whose salaries are 30000 or 40000 or 50000.

**QUERY**

```
select * from employee where salary = 30000 or salary = 40000 or salary = 50000;
```

**DATABASE TABLES**

|   | SSN  | Name   | Address                        | sex | salary | superSSN | DNo  | Age  |
|---|------|--------|--------------------------------|-----|--------|----------|------|------|
| ▶ | 1005 | Janaki | hn5,vadayam,kuttiadi,kozhikode | f   | 40000  | s005     | d005 | NULL |

## NESTED QUERIES

### AIM

Develop SQL Queries to implement Nested Queries/ Sub Queries and Joins

### Questions : 1

Update the salary by 0.25 times for all the employees whose Plocation is ‘Chennai’.

### QUERY

```
update employee set salary = salary + 0.25* salary  
where Address = 'chennai';
```

### DATABASE TABLES

|   | SSN  | Name     | Address                        | sex | salary | superSSN | DNo  | Age  |
|---|------|----------|--------------------------------|-----|--------|----------|------|------|
| ▶ | 1001 | Vasu     | hn1,vadayam,kuttiadi,kozhikode | m   | 25000  | s001     | d001 | NULL |
|   | 1003 | Sajeewan | chennai                        | m   | 115235 | s003     | d003 | NULL |
|   | 1004 | Dineshan | hn4,vadayam,kuttiadi,kozhikode | m   | 55000  | s004     | d004 | NULL |
|   | 1005 | Janaki   | chennai                        | f   | 78125  | s005     | d005 | NULL |

### Questions : 2

To display the name and project location of employees whose working hour is greater than 5

### QUERY

```
select name,Dloc from employee,dlocation,works_on where dlocation.Dno = employee.Dno and w
```

### DATABASE TABLES

|   | name     | Dloc                           |
|---|----------|--------------------------------|
| ▶ | Vasu     | near university library        |
|   | Sajeewan | near sbi cusat                 |
|   | Dineshan | opposite of cusat sports arena |
|   | Janaki   | opposite of cusat sports arena |

**Questions : 3**

Left join employee table and works\_on table

**QUERY**

```
select * from EMPLOYEE left join works_on on employee.SSN = works_on.SSN;
```

**DATABASE TABLES**

|   | SSN  | Name     | Address  | sex | salary | superSSN | DNo | Age  | SSN  | PNo | Hours    |
|---|------|----------|----------|-----|--------|----------|-----|------|------|-----|----------|
| ▶ | 1001 | Vasu     | vadakara | m   | 25000  | s001     | d1  | NULL | 1001 | p1  | 10:00:00 |
|   | 1003 | Sajeevan | chennai  | m   | 115235 | s003     | d3  | NULL | 1003 | p2  | 12:00:00 |
|   | 1004 | Dineshan | d4       | m   | 55000  | s004     | d4  | NULL | 1004 | p5  | 09:00:00 |
|   | 1005 | Janaki   | d5       | f   | 78125  | s005     | d5  | NULL | 1005 | p4  | 11:00:00 |

**Questions : 4**

Right join works\_on table and employee table

**QUERY**

```
select * from WORKS_ON RIGHT join EMPLOYEE on EMPLOYEE.SSN =WORKS_ON.SSN;
```

**DATABASE TABLES**

|   | SSN  | PNo | Hours    | SSN  | Name     | Address  | sex | salary | superSSN | DNo | Age  |
|---|------|-----|----------|------|----------|----------|-----|--------|----------|-----|------|
| ▶ | 1001 | p1  | 10:00:00 | 1001 | Vasu     | vadakara | m   | 25000  | s001     | d1  | NULL |
|   | 1003 | p2  | 12:00:00 | 1003 | Sajeevan | chennai  | m   | 115235 | s003     | d3  | NULL |
|   | 1004 | p5  | 09:00:00 | 1004 | Dineshan | d4       | m   | 55000  | s004     | d4  | NULL |
|   | 1005 | p4  | 11:00:00 | 1005 | Janaki   | d5       | f   | 78125  | s005     | d5  | NULL |

**Questions : 5**

Full join works\_on table and employee table

**QUERY**

```
select * from WORKS_ON full join EMPLOYEE ;
```

**DATABASE TABLES**

|   | SSN  | PNo | Hours    | SSN  | Name     | Address  | sex | salary | superSSN | DNo | Age  |
|---|------|-----|----------|------|----------|----------|-----|--------|----------|-----|------|
| ▶ | 1005 | p4  | 11:00:00 | 1001 | Vasu     | vadakara | m   | 25000  | s001     | d1  | NULL |
|   | 1004 | p5  | 09:00:00 | 1001 | Vasu     | vadakara | m   | 25000  | s001     | d1  | NULL |
|   | 1003 | p2  | 12:00:00 | 1001 | Vasu     | vadakara | m   | 25000  | s001     | d1  | NULL |
|   | 1002 | p3  | 06:00:00 | 1001 | Vasu     | vadakara | m   | 25000  | s001     | d1  | NULL |
|   | 1001 | p1  | 10:00:00 | 1001 | Vasu     | vadakara | m   | 25000  | s001     | d1  | NULL |
|   | 1005 | p4  | 11:00:00 | 1003 | Sajeevan | chennai  | m   | 115235 | s003     | d3  | NULL |
|   | 1004 | p5  | 09:00:00 | 1003 | Sajeevan | chennai  | m   | 115235 | s003     | d3  | NULL |
|   | 1003 | p2  | 12:00:00 | 1003 | Sajeevan | chennai  | m   | 115235 | s003     | d3  | NULL |
|   | 1002 | p3  | 06:00:00 | 1003 | Sajeevan | chennai  | m   | 115235 | s003     | d3  | NULL |
|   | 1001 | p1  | 10:00:00 | 1003 | Sajeevan | chennai  | m   | 115235 | s003     | d3  | NULL |
|   | 1005 | p4  | 11:00:00 | 1004 | Dineshan | d4       | m   | 55000  | s004     | d4  | NULL |
|   | 1004 | p5  | 09:00:00 | 1004 | Dineshan | d4       | m   | 55000  | s004     | d4  | NULL |
|   | 1003 | p2  | 12:00:00 | 1004 | Dineshan | d4       | m   | 55000  | s004     | d4  | NULL |
|   | 1002 | p3  | 06:00:00 | 1004 | Dineshan | d4       | m   | 55000  | s004     | d4  | NULL |
|   | 1001 | p1  | 10:00:00 | 1004 | Dineshan | d4       | m   | 55000  | s004     | d4  | NULL |
|   | 1005 | p4  | 11:00:00 | 1005 | Janaki   | d5       | f   | 78125  | s005     | d5  | NULL |
|   | 1004 | p5  | 09:00:00 | 1005 | Janaki   | d5       | f   | 78125  | s005     | d5  | NULL |
|   | 1003 | p2  | 12:00:00 | 1005 | Janaki   | d5       | f   | 78125  | s005     | d5  | NULL |
|   | 1002 | p3  | 06:00:00 | 1005 | Janaki   | d5       | f   | 78125  | s005     | d5  | NULL |
|   | 1001 | p1  | 10:00:00 | 1005 | Janaki   | d5       | f   | 78125  | s005     | d5  | NULL |

## VIEWS

### AIM

Develop SQL Queries for creating and dropping Views

### Questions : 1

Create a view VW\_emp on employee table

### QUERY

```
create view VW_emp as select * from employee;
```

### DATABASE TABLES

```
mysql> select * from VW_emp;
+----+-----+-----+-----+-----+-----+-----+-----+
| SSN | Name   | Address | sex  | salary | superSSN | DNo  | Age   |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1001 | Vasu    | vadakara | m    | 25000  | s001     | d1    | NULL  |
| 1003 | Sajeevan | chennai  | m    | 115235 | s003     | d3    | NULL  |
| 1004 | Dineshan |           | d4   | 55000  | s004     | d4    | NULL  |
| 1005 | Janaki   |           | d5   | 78125  | s005     | d5    | NULL  |
+----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

### Questions : 2

Create another view VW\_SSN contains SuperSSN and Dno of female employees

### QUERY

```
create view VW_SSN as select SuperSSN,Dno from VW_emp where Sex = "F";
```

### DATABASE TABLES

```
mysql> select * from VW_SSN;
+-----+-----+
| SuperSSN | Dno  |
+-----+-----+
| s005     | d5   |
+-----+-----+
1 row in set (0.00 sec)
```

**Questions : 3**

Update the address of employee to Chennai whose id is e100 in view VW\_emp

**QUERY**

```
UPDATE VW_emp SET Address="Chennai" WHERE SSN = "1001";
```

**DATABASE TABLES**

```
mysql> select * from VW_emp;
+-----+-----+-----+-----+-----+-----+-----+-----+
| SSN | Name      | Address | sex   | salary | superSSN | DNo    | Age   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1001 | Vasu       | Chennai | m     | 25000  | s001    | d1     | NULL  |
| 1003 | Sajeevan   | chennai | m     | 115235 | s003    | d3     | NULL  |
| 1004 | Dineshan   | d4       | m     | 55000  | s004    | d4     | NULL  |
| 1005 | Janaki     | d5       | f     | 78125  | s005    | d5     | NULL  |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

**Questions : 4**

Delete the view VW\_emp

**QUERY**

```
drop view VW_EMP;
```

**DATABASE TABLES**

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

## FUNCTIONS AND PROCEDURES

### AIM

Develop PL/SQL program to familiarize with Function and Procedure

### Questions : 1

Write a PL/SQL function to find factorial of a number

### QUERY

```
SQL> connect
Enter user-name: system
Enter password:
Connected.
```

```
SQL> set serveroutput on
SQL> edit@factorial.sql

create or replace function get_factorial(N int)
return varchar
is
fact int := 1;
begin
for i in 1..N loop
fact := fact*i;
end loop;
return 'Factorial is ' || fact ;
end;
/
select get_factorial(5) from dual;
```

```
SQL> @XEfactorial.sql
```

Function created.

### DATABASE TABLES

```
SQL> @XEfactorial.sql
Function created.
GET_FACTORIAL(5)
Factorial is 120
```

### Questions : 2

Write a PL/SQL function to find maximum of two numbers

#### QUERY

```
SQL> connect
Enter user-name: system
Enter password:
Connected.
```

```
SQL> set serveroutput on
SQL> edit@max.sql

create or replace function maximum(n1 int, n2 int)
return varchar
is
m int := 0;
begin
if n1>n2 then
m := n1;
else
m := n2;
end if;
return 'Maximum is ' ||m;
end;
/
select maximum(4,9) from dual;

SQL> @XEmax.sql

Function created.
```

**DATABASE TABLES**

```
SQL> @XEmax.sql
Function created.
MAXIMUM(4,9)
Maximum is 9
```

**Questions : 3**

Write a PL/SQL procedure to print the prime

**QUERY**

```
SQL> connect
Enter user-name: system
Enter password:
Connected.
```

```
SQL> set serveroutput on
```

```
SQL> edit@prime.sql
```

```
DECLARE
    i NUMBER(3);
    j NUMBER(3);
BEGIN
    dbms_output.Put_line('The prime numbers are:');
    dbms_output.new_line;
    i := 2;
    LOOP
        j := 2;
        LOOP
            EXIT WHEN( ( MOD(i, j) = 0 )
                      OR ( j = i ) );
            j := j + 1;
        END LOOP;
        IF( j = i )THEN
            dbms_output.Put(i||' ');
        END IF;
    END LOOP;
```

```
    END IF;
    i := i + 1;
    exit WHEN i = 50;
END LOOP;
dbms_output.new_line;
END;
/
```

SQL> @XEprime.sql

Function created.

## DATABASE TABLES

```
SQL> @XEprime.sql
2
3
5
7
PL/SQL procedure successfully completed.■
```

## Questions : 4

Write a PL/SQL procedure to display numbers from 1 to 10 using while loop

## QUERY

```
SQL> connect
Enter user-name: system
Enter password:
Connected.
```

```
SQL> set serveroutput on
SQL> edit@numbers.sql
```

```
DECLARE
    i INTEGER := 1;
BEGIN
    WHILE i <= 10 LOOP
```

```
DBMS_OUTPUT.PUT_LINE(i);
i := i+1;
END LOOP;
END;
/
```

SQL> @XEnumbers.sql

Function created.

## DATABASE TABLES

```
SQL> @XEnumbers.sql
1
2
3
4
5
6
7
8
9
10
PL/SQL procedure successfully completed.
```

## CURSOR

### AIM

Develop PL/SQL program to implement Cursor

### Question : 1

Write a PL/SQL cursor program to update the salary of each employee of department number D001 in the Employee table as per the schema

### QUERY

```
SQL> create table Employee(SSN varchar(30),Fname varchar(30),Lname varchar(30),Address  
varchar(50),Sex varchar(15),Salary number(30),SuperSSN varchar(30),DNO varchar(20));
```

Table created.

```
SQL> create table Department(DNO varchar(20),Dname varchar(30),MgrSSN varchar(30),  
MgrStartDate varchar(20));
```

Table created.

```
SQL> insert into Employee values('e1001','Archana','Suresh','13B,Highway Gardens,  
Kozhikode','Female',60000,'SP1002','D001');
```

1 row created.

```
SQL> insert into Employee values('e1002','Justin','Varghese','Rose Villa,Kochi','Male',  
50000,'SP1001','D002');
```

1 row created.

```
SQL> insert into Employee values('e1003','Meera','Kumar','11B,Arcadia Building,Mumbai',  
'Female',70000,'SP1004','D001');
```

1 row created.

```
SQL> insert into Employee values('e1004','Kailas','Nath','V3,DD Homes,Bangalore','Male',  
30000,'SP1003','D003');
```

1 row created.

```
SQL> insert into Employee values('e1005 ','Sara','Khaild','Ashok Nagar,West Delhi',
'Female',45000,'SP1005','D004');
```

```
1 row created.
```

```
SQL> insert into Employee values('e1006 ','Rahul','Ashok','LV Road,Bengaluru','Male',
55000,'SP1005','D005');
```

```
1 row created.
```

```
SQL> create table Department(DNO varchar(20),Dname varchar(30),MgrSSN varchar(30),
MgrStartDate varchar(20));
```

```
Table created.
```

```
SQL> insert into Department values('D001','Accounts','M1003','2015-09-01');
1 row created.
```

```
SQL> insert into Department values('D002','HR','M1002','2016-12-05');
```

```
1 row created.
```

```
SQL> insert into Department values('D003','Marketing','M1005','2012-04-04');
```

```
1 row created.
```

```
SQL> insert into Department values('D004','Sales','M1004','2019-08-20');
```

```
1 row created.
```

```
SQL> insert into Department values('D005','Management','M1001','2017-03-09');
```

```
1 row created.
```

```
SQL> declare cursor employee_cur is
2 select SSN,Salary from Employee where DNO = 'D001'
3 for update;
4 incr_sal number;
5 begin
6 for employee_rec in employee_cur loop
```

```
7 if employee_rec.Salary < 50000 then
8 incr_sal := .15;
9 else
10 incr_sal := .10;
11 end if;
12 update Employee set Salary = Salary + Salary * incr_sal where current of
employee_cur;
13 end loop;
14 end;
15 /
```

PL/SQL procedure successfully completed.

## DATABASE TABLES

| SQL> select * from Employee;  |         |        |        |
|-------------------------------|---------|--------|--------|
| SSN                           | FNAME   | SEX    | SALARY |
| <hr/>                         |         |        |        |
| LNAME                         |         |        |        |
| ADDRESS                       |         |        |        |
| SUPERSSN                      | DNO     |        |        |
| e1001                         | Archana |        |        |
| Suresh                        |         |        |        |
| 13B,Highway Gardens,Kozhikode |         | Female | 60000  |
| SP1002                        | D001    |        |        |
| <hr/>                         |         |        |        |
| SSN                           | FNAME   | SEX    | SALARY |
| <hr/>                         |         |        |        |
| LNAME                         |         |        |        |
| ADDRESS                       |         |        |        |
| SUPERSSN                      | DNO     |        |        |
| e1002                         | Justin  |        |        |
| Varghese                      |         |        |        |
| Rose Villa,Kochi              |         | Male   | 50000  |
| SP1001                        | D002    |        |        |

| SSN   | FNAME  |                 |
|---|--------|-----------------|
| LNAME   |        |                 |
| ADDRESS   | SEX    | SALARY          |
| SUPERSSN  | DNO    |                 |
| e1003<br>Kumar<br>11B,Arcadia Building,Mumbai<br>SP1004 | Meera  | Female<br>70000 |
| D001  |        |                 |
| SSN   | FNAME  |                 |
| LNAME   |        |                 |
| ADDRESS   | SEX    | SALARY          |
| SUPERSSN  | DNO    |                 |
| e1004<br>Nath<br>V3,DD Homes,Bangalore<br>SP1003        | Kailas | Male<br>30000   |
| D003  |        |                 |
| SSN   | FNAME  |                 |
| LNAME   |        |                 |
| ADDRESS   | SEX    | SALARY          |
| SUPERSSN  | DNO    |                 |
| e1005<br>Khaild<br>Ashok Nagar,West Delhi<br>SP1005     | Sara   | Female<br>45000 |
| D004  |        |                 |
| SSN   | FNAME  |                 |
| LNAME   |        |                 |
| ADDRESS   | SEX    | SALARY          |
| SUPERSSN  | DNO    |                 |
| e1006<br>Ashok<br>LV Road,Bengaluru<br>SP1005           | Rahul  | Male<br>55000   |
| D005  |        |                 |

```
SQL> select * from Department;
```

| DNO  | DNAME      | MGRSSN | MGRSTARTDATE |
|------|------------|--------|--------------|
| D001 | Accounts   | M1003  | 2015-09-01   |
| D002 | HR         | M1002  | 2016-12-05   |
| D003 | Marketing  | M1005  | 2012-04-04   |
| DNO  | DNAME      | MGRSSN | MGRSTARTDATE |
| D004 | Sales      | M1004  | 2019-08-20   |
| D005 | Management | M1001  | 2017-03-09   |

```
SQL> select * from Employee;
```

| SSN      | FNAME   | LNAME    | ADDRESS                         | SEX    | SALARY |
|----------|---------|----------|---------------------------------|--------|--------|
| SUPERSSN | DNO     |          |                                 |        |        |
| e1001    | Archana | Suresh   | 13B, Highway Gardens, Kozhikode | Female | 66000  |
| SP1002   |         |          | SP1002                          | D001   |        |
| SSN      | FNAME   | LNAME    | ADDRESS                         | SEX    | SALARY |
| SUPERSSN | DNO     |          |                                 |        |        |
| e1002    | Justin  | Varghese | Rose Villa, Kochi               | Male   | 50000  |
| SP1001   |         |          | SP1001                          | D002   |        |

| SSN   | FNAME  |        |        |
|---|--------|--------|--------|
| LNAME   |        |        |        |
| ADDRESS   |        | SEX    | SALARY |
| SUPERSSN  | DNO    |        |        |
| e1003<br>Kumar<br>11B,Arcadia Building,Mumbai<br>SP1004 | Meera  | Female | 77000  |
|   | D001   |        |        |
| SSN   | FNAME  |        |        |
| LNAME   |        |        |        |
| ADDRESS   |        | SEX    | SALARY |
| SUPERSSN  | DNO    |        |        |
| e1004<br>Nath<br>V3,DD Homes,Bangalore<br>SP1003        | Kailas | Male   | 30000  |
|   | D003   |        |        |
| SSN   | FNAME  |        |        |
| LNAME   |        |        |        |
| ADDRESS   |        | SEX    | SALARY |
| SUPERSSN  | DNO    |        |        |
| e1005<br>Khaild<br>Ashok Nagar,West Delhi<br>SP1005     | Sara   | Female | 45000  |
|   | D004   |        |        |
| SSN   | FNAME  |        |        |
| LNAME   |        |        |        |
| ADDRESS   |        | SEX    | SALARY |
| SUPERSSN  | DNO    |        |        |
| e1006<br>Ashok<br>LV Road,Bengaluru<br>SP1005           | Rahul  | Male   | 55000  |
|   | D005   |        |        |

**Question : 2**

Write a PL/SQL cursor program to retrieve Dno and DName from Department table as per the schema

**QUERY**

```
SQL> declare cursor department_cur is
```

```
2 select DNO,Dname from Department;
3 data1 Department.DNO%type;
4 data2 Department.Dname%type;
5 begin
6 open department_cur;
7 loop
8 fetch department_cur into data1,data2;
9 exit when department_cur%notfound;
10 dbms_output.put_line('DNO : '||data1||'::Dname : '||data2);
11 end loop;
12 close department_cur;
13 end;
14 /
```

## DATABASE TABLES

```
DNO : D001::Dname : Accounts
DNO : D002::Dname : HR
DNO : D003::Dname : Marketing
DNO : D004::Dname : Sales
DNO : D005::Dname : Management

PL/SQL procedure successfully completed.
```

## TRIGGER

### AIM

Develop PL/SQL program to implement Trigger

### Question : 1

Write PL/SQL trigger program to display the salary differences between the old values and new values in the table employee as per the schema

### QUERY

```
CREATE OR REPLACE TRIGGER display_salary_changes
BEFORE DELETE OR INSERT OR UPDATE ON employeetable
FOR EACH ROW
WHEN (NEW.ID > 0)
DECLARE
sal_diff number;
BEGIN
sal_diff := :NEW.Salary - :OLD.Salary;
dbms_output.put_line('Old salary: ' || :OLD.salary);
dbms_output.put_line('New salary: ' || :NEW.salary);
dbms_output.put_line('Salary difference: ' || sal_diff);
END;
/
Trigger created.

DECLARE
BEGIN
UPDATE employeetable
SET Salary = Salary + 4000;
END;
/
```

### DATABASE TABLES

```
Trigger created.

SQL> set serveroutput on
SQL> @C:\Users\Sona\Desktop\h.sql
Old time: 7
New time: 5
Time difference: -2
Old time: 10
New time: 8
Time difference: -2
Old time: 9
New time: 7
Time difference: -2
Old time: 10
New time: 8
Time difference: -2
Old time: 4
New time: 2
Time difference: -2

PL/SQL procedure successfully completed.
```

**Question : 2**

Write PL/SQL trigger program to display the hour differences between the old values and new values in the table Works\_on as per the schema

**QUERY**

```
CREATE OR REPLACE TRIGGER display_hour_changes
BEFORE DELETE OR INSERT OR update on Work_on
for each row
when (NEW.HOURS > 0)
DECLARE
hour_diff number;
BEGIN
hour_diff := :NEW.HOURS - :OLD.HOURS;
dbms_output.put_line('Old time: ' || :OLD.HOURS);
dbms_output.put_line('New time: ' || :NEW.HOURS);
dbms_output.put_line('Salary difference: ' || hour_diff);
END;
/
Trigger created.

DECLARE
BEGIN
UPDATE Works_on
SET HOURS = HOURS - 4;
```

END;

/

## DATABASE TABLES

```
Old salary: 15000
New salary: 21000
Salary difference: 6000
Old salary: 20000
New salary: 26000
Salary difference: 6000
Old salary: 17000
New salary: 23000
Salary difference: 6000
Old salary: 15000
New salary: 21000
Salary difference: 6000
Old salary: 25000
New salary: 31000
Salary difference: 6000

PL/SQL procedure successfully completed.
```

## TCL

### AIM

Develop SQL Queries to understand the concept of Transaction Control Language

### Question : 1

Creating Check points in the program

### QUERY

```
mysql> start transaction;  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> savepoint save1;  
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> insert into Employee values("e1006","Anju","Rajesh",  
"Sobha Marina,Kochi","Female",  
80000,"SP1004","D005",29);  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> savepoint save2;  
Query OK, 0 rows affected (0.00 sec)
```

### DATABASE TABLES

```
mysql> select * from Employee;  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| SSN | Fname | Lname | Address          | Sex   | Salary | SuperSSN | DNO  | Age  |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| e1001 | Archana | Suresh | Chennai           | Female | 25000 | SP1002  | D001 | 28  |  
| e1002 | Akash   | Raj    | 4B,Renegade Villas,Pune | Male  | 40000 | SP1001  | D003 | 24  |  
| e1003 | Meera   | Kumar  | 11B,Arcadia Building,Mumbai | Female | 70000 | SP1004  | D005 | 31  |  
| e1004 | Kailas  | Nath   | V3,DD Homes,Bangalore | Male  | 30000 | SP1003  | D002 | 25  |  
| e1005 | Sara    | Khalid | Ashok Nagar,West Delhi | Female | 56250 | SP1005  | D004 | 27  |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)  
  
mysql> start transaction;  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> savepoint save1;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> insert into Employee values("e1006","Anju","Rajesh","Sobha Marina,Kochi","Female",80000,"SP1004","D005",29);  
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from Employee;
+-----+-----+-----+-----+-----+-----+-----+-----+
| SSN | Fname | Lname | Address          | Sex | Salary | SuperSSN | DNO | Age |
+-----+-----+-----+-----+-----+-----+-----+-----+
| e1001 | Archana | Suresh | Chennai           | Female | 25000 | SP1002 | D001 | 28 |
| e1002 | Akash   | Raj    | 4B,Renegade Villas,Pune | Male  | 40000 | SP1001 | D003 | 24 |
| e1003 | Meera   | Kumar  | 11B,Arcadia Building,Mumbai | Female | 70000 | SP1004 | D005 | 31 |
| e1004 | Kailas  | Nath   | V3,DD Homes,Bangalore | Male  | 30000 | SP1003 | D002 | 25 |
| e1005 | Sara    | Khalid | Ashok Nagar,West Delhi | Female | 56250 | SP1005 | D004 | 27 |
| e1006 | Anju    | Rajesh | Sobha Marina,Kochi      | Female | 80000 | SP1004 | D005 | 29 |
+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> savepoint save2;
Query OK, 0 rows affected (0.00 sec)
```

## Question : 2

Rollback to a previously created Checkpoint in the program

### QUERY

```
mysql> rollback to save1;
Query OK, 0 rows affected (0.01 sec)
```

## DATABASE TABLES

```
mysql> rollback to save1;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from Employee;
+-----+-----+-----+-----+-----+-----+-----+-----+
| SSN | Fname | Lname | Address          | Sex | Salary | SuperSSN | DNO | Age |
+-----+-----+-----+-----+-----+-----+-----+-----+
| e1001 | Archana | Suresh | Chennai           | Female | 25000 | SP1002 | D001 | 28 |
| e1002 | Akash   | Raj    | 4B,Renegade Villas,Pune | Male  | 40000 | SP1001 | D003 | 24 |
| e1003 | Meera   | Kumar  | 11B,Arcadia Building,Mumbai | Female | 70000 | SP1004 | D005 | 31 |
| e1004 | Kailas  | Nath   | V3,DD Homes,Bangalore | Male  | 30000 | SP1003 | D002 | 25 |
| e1005 | Sara    | Khalid | Ashok Nagar,West Delhi | Female | 56250 | SP1005 | D004 | 27 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

## Question : 3

Commit the program

### QUERY

```
mysql> commit;
Query OK, 0 rows affected (0.00 sec)
```

## DATABASE TABLES

```
mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

## MongoDB

### AIM

Develop program to perform operations in MongoDB

### Question : 1

Create a database emp

### QUERY

```
test> use emp
```

### DATABASE TABLES

```
test> use emp
switched to db emp
emp> db
emp
```

### Question : 2

Create new Collection

### QUERY

```
emp> db.createCollection("Department")
{ ok: 1 }
```

### DATABASE TABLES

```
emp> db.createCollection("Department")
{ ok: 1 }
emp> db.getCollectionNames()
[ 'Department' ]
```

### Question : 3

Check the collection list created and drop collection

### QUERY

```
emp> db.getCollectionNames()
emp> db.Department.drop()
```

## DATABASE TABLES

```
emp> db.getCollectionNames()
[ 'Department' ]
emp> db.Department.drop()
true
```

### Question : 4

Insert document in selected Collection

## QUERY

```
emp> db.Employee.insertOne({"Empno" : "E1001" , "Empname" : "Archana" ,
"Salary" : 140000})
{
  acknowledged: true,
  insertedId: ObjectId("63c51ae5fd5856e66b201526")
}

emp> try{ db.Employee.insertMany([{"Empno" : "E1002" , "Empname" : "Rahul" ,
"Salary" : 120000},{"Empno" : "E1003" , "Empname" : "Sara" , "Salary" : 170000}]);
...
...
... catch(e){
...
print(e);
...
}
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63c51bb7fd5856e66b201527"),
    '1': ObjectId("63c51bb7fd5856e66b201528")
}
}
```

**DATABASE TABLES**

```
emp> db.Employee.find()
[  
  {  
    _id: ObjectId("63c51ae5fd5856e66b201526"),  
    Empno: 'E1001',  
    Empname: 'Archana',  
    Salary: 140000  
  },  
  {  
    _id: ObjectId("63c51bb7fd5856e66b201527"),  
    Empno: 'E1002',  
    Empname: 'Rahul',  
    Salary: 120000  
  },  
  {  
    _id: ObjectId("63c51bb7fd5856e66b201528"),  
    Empno: 'E1003',  
    Empname: 'Sara',  
    Salary: 170000  
  }  
]
```

**Question : 5**

To get the list documents in Collection

**QUERY**

```
emp> db.Employee.find()
```

**DATABASE TABLES**

```
emp> db.Employee.find()
[  
  {  
    _id: ObjectId("63c51ae5fd5856e66b201526"),  
    Empno: 'E1001',  
    Empname: 'Archana',  
    Salary: 140000  
  },  
  {  
    _id: ObjectId("63c51bb7fd5856e66b201527"),  
    Empno: 'E1002',  
    Empname: 'Rahul',  
    Salary: 120000  
  },  
  {  
    _id: ObjectId("63c51bb7fd5856e66b201528"),  
    Empno: 'E1003',  
    Empname: 'Sara',  
    Salary: 170000  
  }  
]
```

### Question : 6

Update the document in Collection

#### QUERY

```
emp> db.Employee.updateOne({"Empno" : "E1001"},  
... {  
... $set : {"Salary" : 160000},  
... $currentDate : {lastModified : true}  
... }  
... )  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

#### DATABASE TABLES

```
emp> db.Employee.find()  
[  
  {  
    _id: ObjectId("63c51ae5fd5856e66b201526"),  
    Empno: 'E1001',  
    Empname: 'Archana',  
    Salary: 160000,  
    lastModified: ISODate("2023-01-16T09:42:01.053Z")  
  },  
  {  
    _id: ObjectId("63c51bb7fd5856e66b201527"),  
    Empno: 'E1002',  
    Empname: 'Rahul',  
    Salary: 120000  
  },  
  {  
    _id: ObjectId("63c51bb7fd5856e66b201528"),  
    Empno: 'E1003',  
    Empname: 'Sara',  
    Salary: 170000  
  }  
]
```

### Question : 7

Delete the document in selected Collection

#### QUERY

```
emp> db.Employee.deleteOne({"Empname" : "Sara"});  
{ acknowledged: true, deletedCount: 1 }
```

## DATABASE TABLES

```
emp> db.Employee.find()
[  
  {  
    _id: ObjectId("63c51ae5fd5856e66b201526"),  
    Empno: 'E1001',  
    Empname: 'Archana',  
    Salary: 160000,  
    lastModified: ISODate("2023-01-16T09:42:01.053Z")  
  },  
  {  
    _id: ObjectId("63c51bb7fd5856e66b201527"),  
    Empno: 'E1002',  
    Empname: 'Rahul',  
    Salary: 120000  
  }  
]
```

### Question : 8

Projection using find() method

## QUERY

```
emp> db.Employee.find({}, {"Empname" : 1}).pretty()
```

## DATABASE TABLES

```
emp> db.Employee.find({}, {"Empname" : 1}).pretty()  
[  
  { _id: ObjectId("63c51ae5fd5856e66b201526"), Empname: 'Archana' },  
  { _id: ObjectId("63c51bb7fd5856e66b201527"), Empname: 'Rahul' }  
]
```

### Question : 9

Drop database emp

## QUERY

```
emp> db.dropDatabase()
```

## DATABASE TABLES

```
emp> db.dropDatabase()  
{ ok: 1, dropped: 'emp' }  
emp> |
```

# GRAPH SQL

## AIM

Develop a GraphQL program to print "Hello world"

## OUTPUT

### /geography

The screenshot shows a SPARQL query interface with the following details:

- SPARQL Endpoint:** /geography/
- Content Type (SELECT):** JSON
- Content Type (GRAPH):** Turtle
- Example Queries:**
  - Selection of triples
  - Selection of classes
- Prefixes:** rdf, rdfs, owl, xsd
- SPARQL Query:**

```

1+ prefix table:<http://www.mooney.net/geo#>
2 select ?name ?city
3 where
4 { ?geo table:isCityOf ?city
5 }
6

```
- Results:**

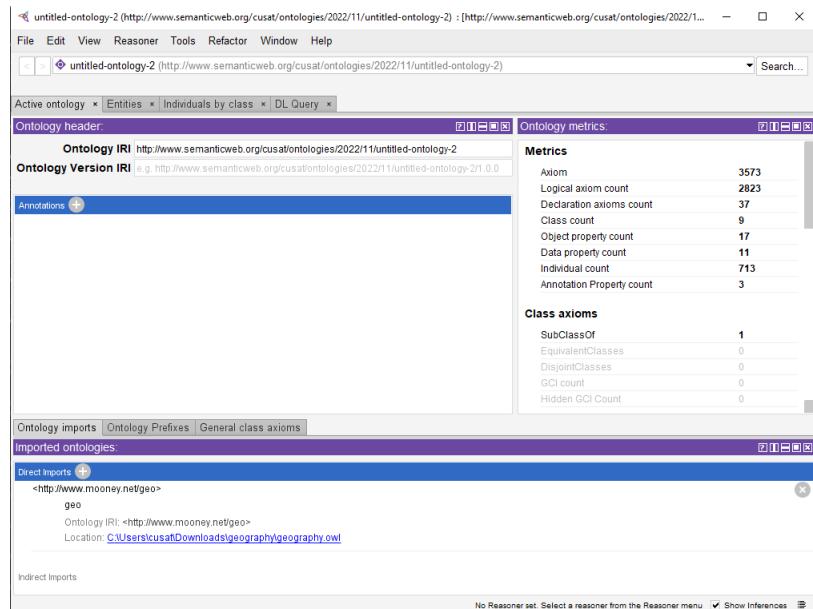
|   | City  |
|---|---|
| 1 | < <a href="http://www.mooney.net/geo#alabama">http://www.mooney.net/geo#alabama</a> >     |
| 2 | < <a href="http://www.mooney.net/geo#alabama">http://www.mooney.net/geo#alabama</a> >     |
| 3 | < <a href="http://www.mooney.net/geo#alabama">http://www.mooney.net/geo#alabama</a> >     |
| 4 | < <a href="http://www.mooney.net/geo#alabama">http://www.mooney.net/geo#alabama</a> >     |
| 5 | < <a href="http://www.mooney.net/geo#alabama">http://www.mooney.net/geo#alabama</a> >     |
| 6 | < <a href="http://www.mooney.net/geo#tennessee">http://www.mooney.net/geo#tennessee</a> > |
- Log Output:**

```

Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\cusat>cd C:\Users\cusat\Downloads\apache-jena-fuseki-4.6.1\apache-jena-fuseki-4.6.1
C:\Users\cusat>fuseki-server --update --mem /ds
[...]
10:23:55 INFO Server :: Apache Jena Fuseki 4.6.1
10:23:58 INFO Config :: FUSEKI_HOME=C:\Users\cusat\Downloads\apache-jena-fuseki-4.6.1\apache-jena-fuseki-4.6.1\run
10:23:58 INFO Config :: Shiro file: file:///C:/Users/cusat/Downloads/apache-jena-fuseki-4.6.1\apache-jena-fuseki-4.6.1\run/shiro.ini
10:24:00 INFO Config :: Template file: templates/config-mem
10:24:05 INFO Server :: Database: in-memory
10:24:05 INFO Server :: Path = /ds
10:24:05 INFO Server :: System
10:24:05 INFO Server :: Memory: 1.2 GiB
10:24:05 INFO Server :: Java: 19.0.1
10:24:05 INFO Server :: OS: Windows 10 10.0 amd64
10:24:05 INFO Server :: PID: 11092
10:24:06 INFO Server :: Started 2022/12/15 10:24:06 IST on port 3030
10:44:48 INFO Admin :: [3] Create database : name = /geography
10:44:59 ERROR Fuseki :: [line: 1, col: 1 ] Content is not allowed in prolog.
10:45:00 INFO Fuseki :: [7] Upload error: [line: 1, col: 1 ] Content is not allowed in prolog.
10:45:00 INFO Fuseki :: [7] 500 Server Error (632 ms)
10:46:12 INFO Fuseki :: [8] POST http://localhost:3030/geography/data
10:46:12 INFO Fuseki :: [line: 1, col: 1 ] Content is not allowed in prolog.
10:46:12 INFO Fuseki :: [8] Upload error: [line: 1, col: 1 ] Content is not allowed in prolog.
10:46:12 INFO Fuseki :: [8] 500 Server Error (417 ms)
10:46:56 INFO Fuseki :: [9] POST http://localhost:3030/geography/data
10:46:56 ERROR Fuseki :: [9] Upload error: [line: 1, col: 1 ] Content is not allowed in prolog.
10:46:56 INFO Fuseki :: [9] 500 Server Error (433 ms)
10:48:13 INFO Fuseki :: [25] POST http://localhost:3030/geography/data
10:48:13 ERROR Fuseki :: [line: 1, col: 1 ] Content is not allowed in prolog.
10:48:14 INFO Fuseki :: [25] Upload error: [line: 1, col: 1 ] Content is not allowed in prolog.
10:48:14 INFO Fuseki :: [25] 500 Server Error (477 ms)
10:48:48 INFO Fuseki :: [29] POST http://localhost:3030/geography/data
10:48:48 INFO Fuseki :: [30] Upload error: [line: 1, col: 1 ] Content is not allowed in prolog.
10:48:48 INFO Fuseki :: [30] 500 Server Error (408 ms)
10:48:49 INFO Fuseki :: [30] POST http://localhost:3030/geography/
10:48:49 INFO Fuseki :: [31] Upload error: [line: 1, col: 1 ] Content is not allowed in prolog.
10:48:49 INFO Fuseki :: [31] 500 Server Error (488 ms)
10:50:02 INFO Fuseki :: [36] POST http://localhost:3030/geography/
10:50:02 INFO Fuseki :: [36] Filename: geography.owl, Content-Type=application/octet-stream, Charset=null => RDF/XML : Count=3589 Triples=3589 Quads=0
10:50:03 INFO Fuseki :: [36] 200 OK (1.272 s)
10:50:36 INFO Fuseki :: [33] POST http://localhost:3030/geography/
10:50:36 INFO Fuseki :: [33] Query - SELECT ?subject ?predicate ?object WHERE { ?subject ?predicate ?object } LIMIT 25
10:50:36 INFO Fuseki :: [33] 200 OK (65 ms)
10:56:18 INFO Fuseki :: [34] POST http://localhost:3030/geography/
10:56:18 INFO Fuseki :: [34] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }
10:56:18 INFO Fuseki :: [34] 200 OK (6 ms)
10:56:59 INFO Fuseki :: [35] POST http://localhost:3030/geography/
10:56:59 INFO Fuseki :: [35] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }
10:56:59 INFO Fuseki :: [35] 200 OK (4 ms)
10:58:29 INFO Fuseki :: [36] POST http://localhost:3030/geography/
10:58:29 INFO Fuseki :: [36] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }
10:58:29 INFO Fuseki :: [36] 200 OK (5 ms)
10:59:45 INFO Fuseki :: [37] POST http://localhost:3030/geography/
10:59:45 INFO Fuseki :: [37] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }
10:59:45 INFO Fuseki :: [37] 200 OK (4 ms)
10:59:48 INFO Fuseki :: [38] POST http://localhost:3030/geography/
10:59:48 INFO Fuseki :: [38] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }
10:59:48 INFO Fuseki :: [38] 200 OK (5 ms)
10:59:50 INFO Fuseki :: [39] POST http://localhost:3030/geography/
10:59:50 INFO Fuseki :: [39] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }

```



## JAVA BASE CONNECTIVITY

### AIM

Develop program to implement Java Database Connectivity

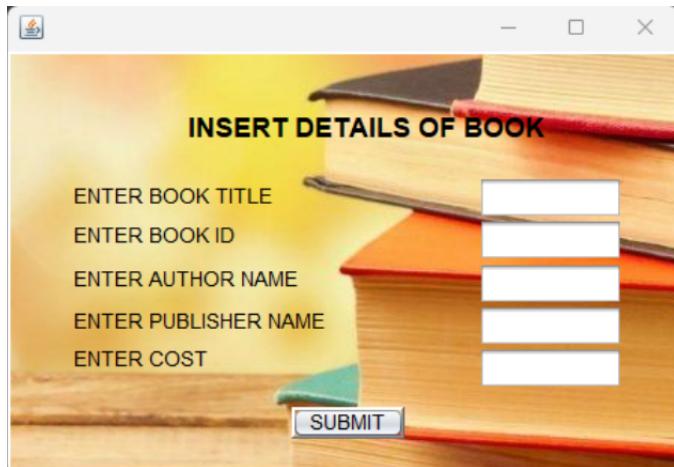
### Question : 1

Write a program which connects to an online book database and insert the details of the books in to the database.

### QUERY

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        Class.forName("com.mysql.cj.jdbc.Driver");  
    }  
    catch (ClassNotFoundException ex) {  
        Logger.getLogger(insert.class.getName()).log(Level.SEVERE, null, ex);  
    }  
    try (Connection con = DriverManager.getConnection  
        ("jdbc:mysql://localhost:3306/book","root","root")) {  
        String sql = "insert into books values(?,?,?,?,?)";  
        PreparedStatement ps = con.prepareStatement(sql);  
        ps.setString(1,jTextField1.getText());  
        ps.setString(2,jTextField2.getText());  
        ps.setString(3, jTextField3.getText());  
        ps.setString(4, jTextField4.getText());  
        ps.setInt(5,Integer.parseInt(jTextField5.getText()));  
        ps.execute();  
        JOptionPane.showMessageDialog(this,"data saved successfully");  
    }  
    catch(HeadlessException | NumberFormatException | SQLException e){  
        JOptionPane.showMessageDialog(this,e);  
    }  
}
```

### DATABASE TABLES

**Question : 2**

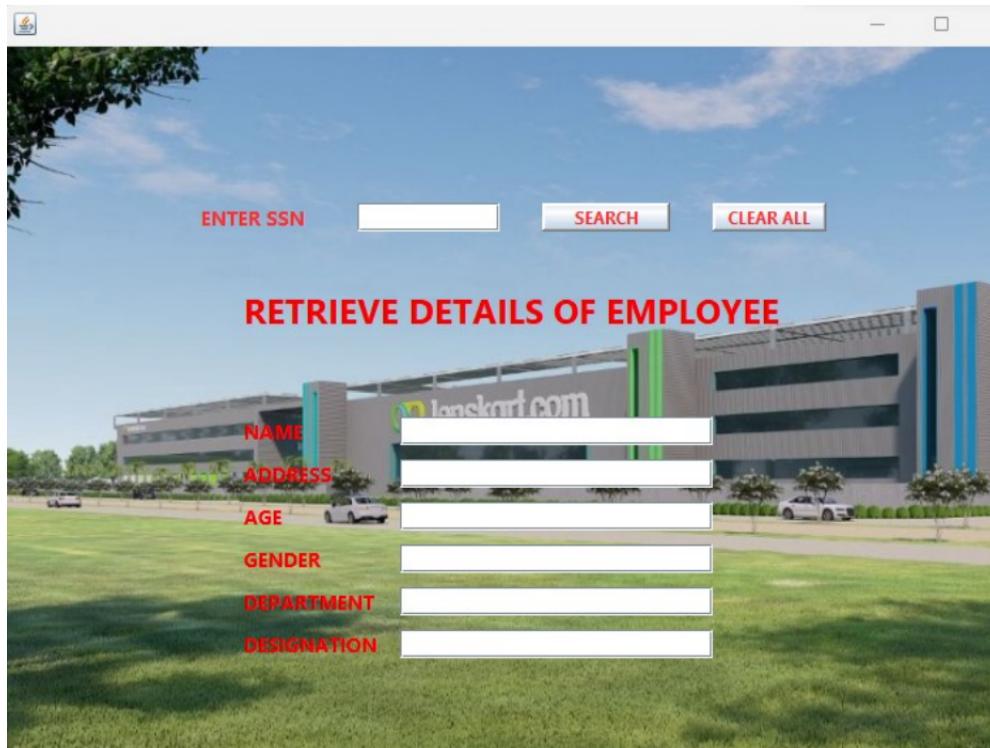
Write a program which connects to an online Employee database and retrieve the details of the employees in the database as per the schema.

**QUERY**

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        Class.forName("com.mysql.cj.jdbc.Driver");  
    }  
    catch (ClassNotFoundException ex) {  
        Logger.getLogger(Retrieve.class.getName()).log(Level.SEVERE, null, ex);  
    }  
    try (Connection con = DriverManager.getConnection  
        ("jdbc:mysql://localhost:3306/company", "root", "root")) {  
        String sql = "select * from employee where SSN = ?";  
        PreparedStatement ps = con.prepareStatement(sql);  
        ps.setString(1, ssn.getText());  
        ResultSet rs = ps.executeQuery();  
        if(rs.next()){  
            Name.setText(rs.getString("Name"));  
            address.setText(rs.getString("Address"));  
            Age.setText(rs.getString("Age"));  
            gender.setText(rs.getString("Sex"));  
        }  
        else{  
            JOptionPane.showMessageDialog(this, "data Not Found");  
        }  
    }  
    catch(HeadlessException | NumberFormatException | SQLException e){
```

```
JOptionPane.showMessageDialog(this,e);
}
}
```

## DATABASE TABLES



### Question : 3

Write a program which connects to an online hospital database and update the details of the patients in the database.

### QUERY

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
try {
Class.forName("com.mysql.cj.jdbc.Driver");
}
catch (ClassNotFoundException ex) {
Logger.getLogger(Update.class.getName()).log(Level.SEVERE, null, ex);
}
try (Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3306/hospital","root","root")) {
```

```
String sql = "select * from patients where phone = ?";  
PreparedStatement ps = con.prepareStatement(sql);  
ps.setString(1, phone.getText());  
ResultSet rs = ps.executeQuery();  
if(rs.next()) {  
    String sql2 = "update patients set Name=? ,Gender = ?,  
    bld_grp = ?,Age = ?,disease = ? where Phone = ?;";  
    PreparedStatement ps2 = con.prepareStatement(sql2);  
    ps2.setString(1, Name.getText());  
    ps2.setString(2, gender.getText());  
    ps2.setString(3, bld_grp.getText());  
    ps2.setInt(4, Integer.parseInt(Age.getText()));  
    ps2.setString(5, disease.getText());  
    ps2.setInt(6, Integer.parseInt(phone.getText()));  
    ps2.execute();  
    JOptionPane.showMessageDialog(this, "Data Updated Successfully");  
}  
else {  
    JOptionPane.showMessageDialog(this, "data Not Found");  
}  
}  
}  
}  
}  
catch (HeadlessException | NumberFormatException | SQLException e) {  
    JOptionPane.showMessageDialog(this, e);  
}  
}
```

## DATABASE TABLES



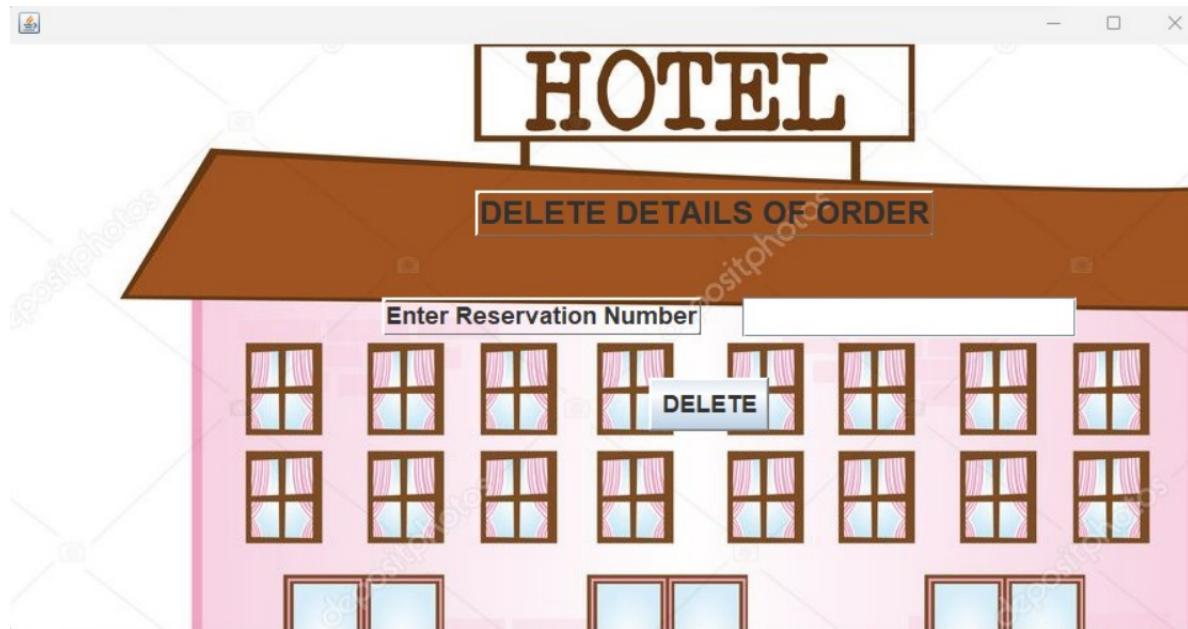
### Question : 4

Write a program which connects to an online Hotel database and delete the details of the orders from the database

**QUERY**

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String res = jTextField1.getText();
    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
    }
    catch (ClassNotFoundException ex) {
        Logger.getLogger(delete_order.class.getName()).log(Level.SEVERE, null, ex);
    }
    try (Connection con = DriverManager.getConnection
        ("jdbc:mysql://localhost:3306/hotel","root","root")) {
        if (res.equals("")){
            JOptionPane.showMessageDialog(this,"Empty Field Not Allowed");
        }
        PreparedStatement statement = (PreparedStatement)con.prepareStatement
        ("select * from orders where reservation_no = ?;");
        statement.setString(1, res);
        ResultSet result = statement.executeQuery();
        if (!(result.next())){
            JOptionPane.showMessageDialog(this,"Data not in the DataBase");
        }
        else{
            PreparedStatement ps = (PreparedStatement)con.prepareStatement
            ("delete from orders where reservation_no = ?;");
            ps.setString(1,res);
            ps.execute();
            JOptionPane.showMessageDialog(this,"data deleted successfully");
            con.close();
            statement.close();
            result.close();
            ps.close();
        }
    }
    catch(HeadlessException | NumberFormatException | SQLException e){
        JOptionPane.showMessageDialog(this,e);
    }
}
```

**DATABASE TABLES**



## PROJECT

### AIM

Develop an Application software using java and mySQL for an Information Management Purpose.

### PROJECT DESCRIPTION

Software to allocate projects to students based on their field of expertise and interest and provide the domain expert mentors. The projects concentrates on matching key words by providing a set of unique features to both students and teachers which then is matched with each other and allocation is carried out accordingly. The project also work as a database collecting tool which contain all the key information regarding each member of the department. The application provides the Head of Department with a complete control over project allocation and professor assignment. The allocation system is purely based on field of interest of student and is not intervened by any external media. It is an user friendly platform which is comfortable to both beginners and experts.

### USERS AND FUNCTIONALITIES

\*User HOD :

- Access to all data
- Add/remove Professors
- Add/remove projects
- Customize student project details
- Sent messages using notice board

\*User Professor:

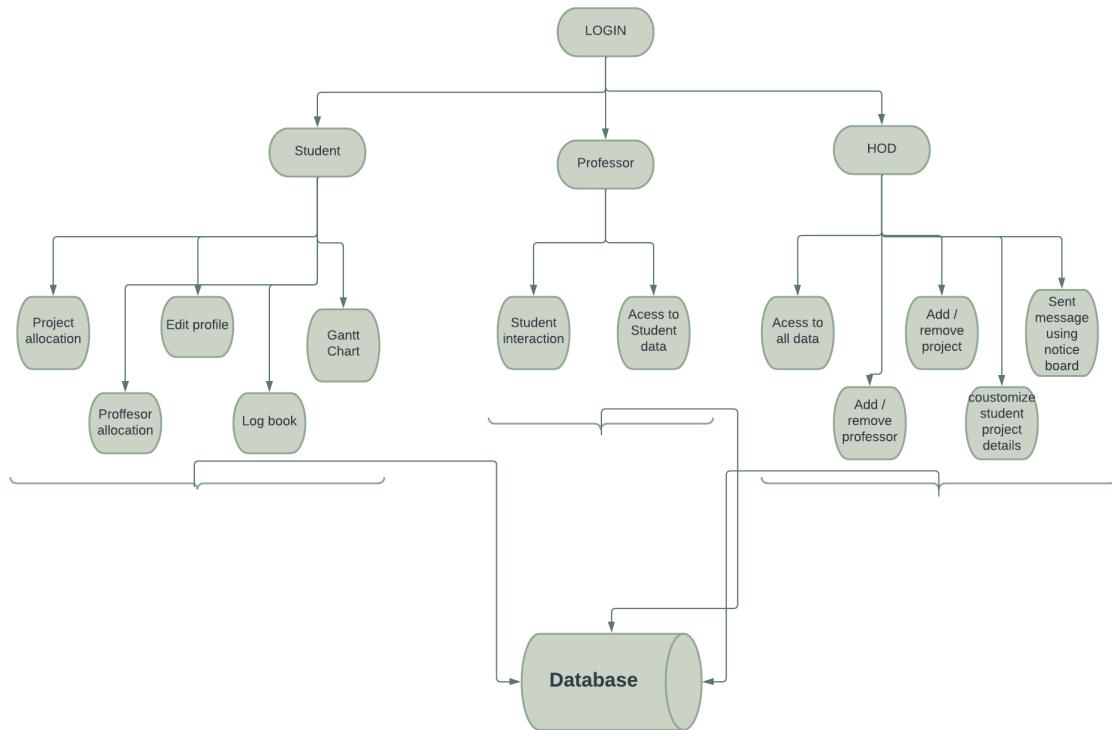
- Access to all students data
- Access to all project data
- Professor student interaction log

\*User Student:

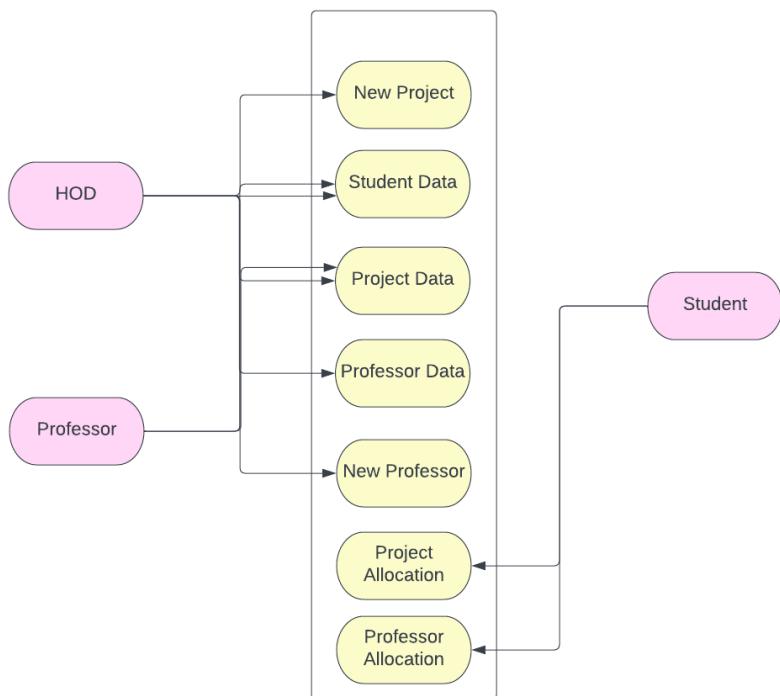
- Edit profile
- Log book
- Update Gantt chart and display

## REFERENCE DESIGN

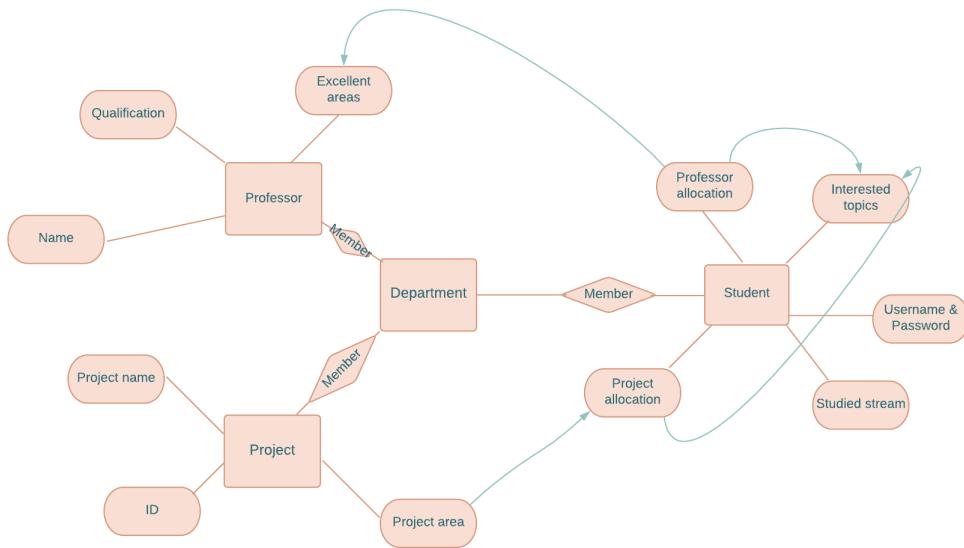
Activity diagram



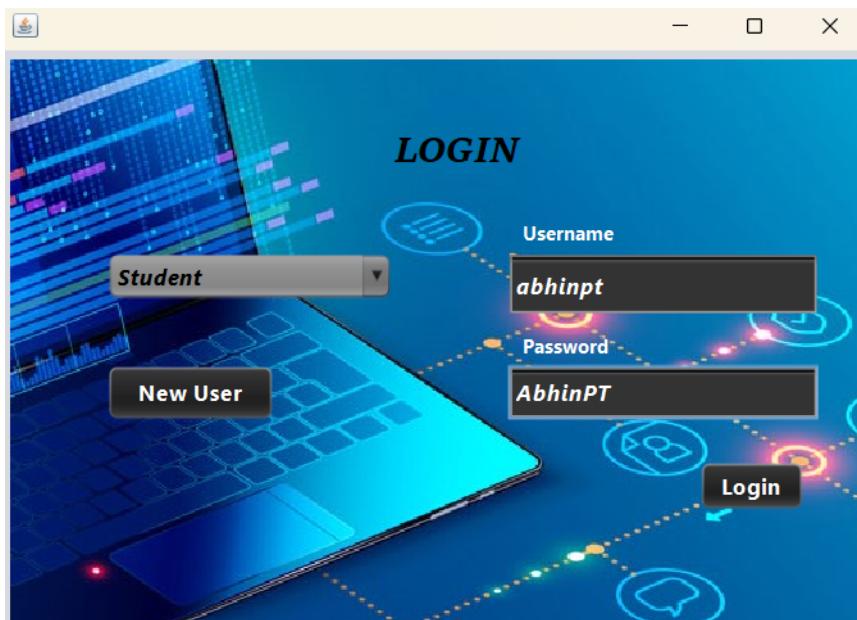
UML diagram

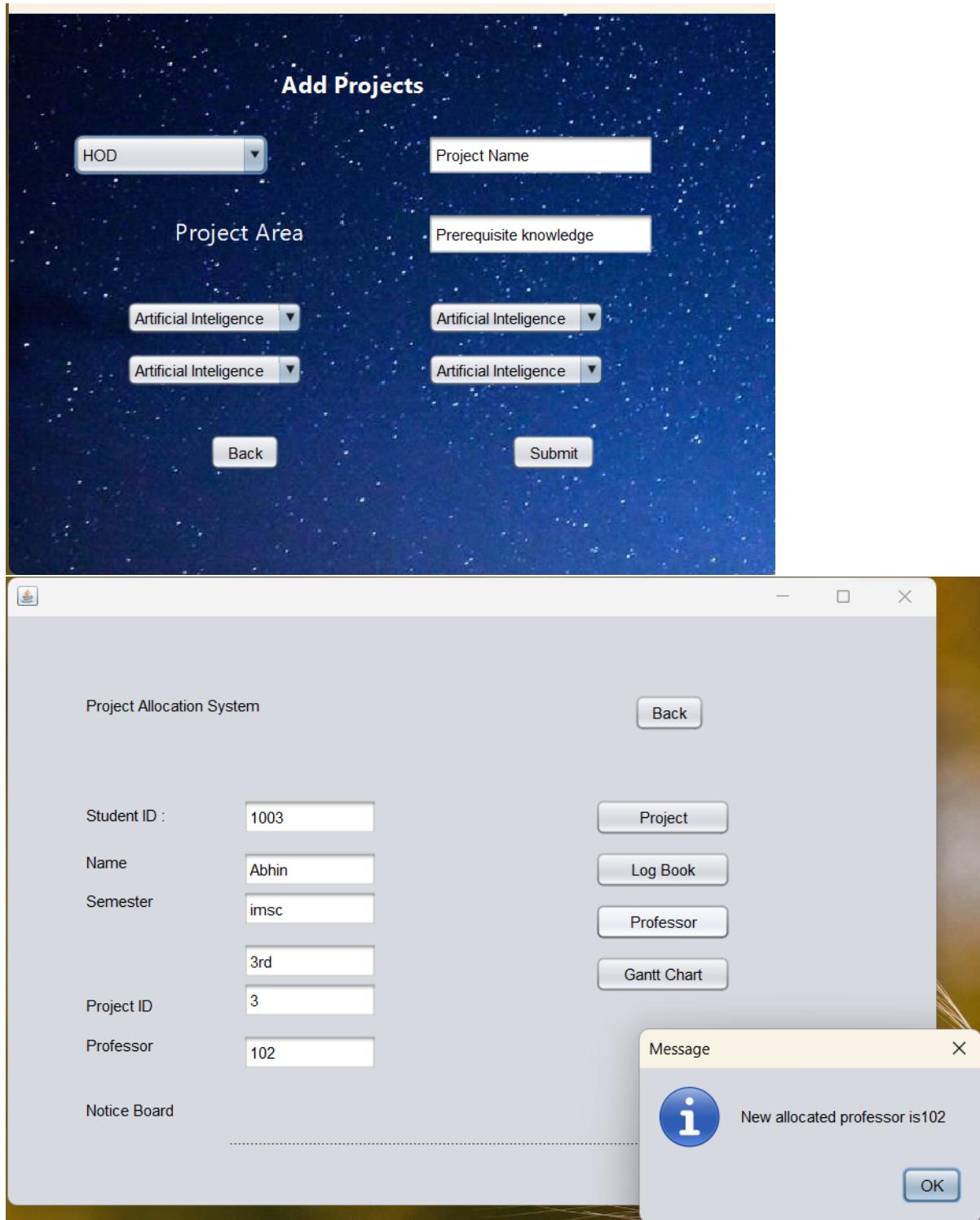


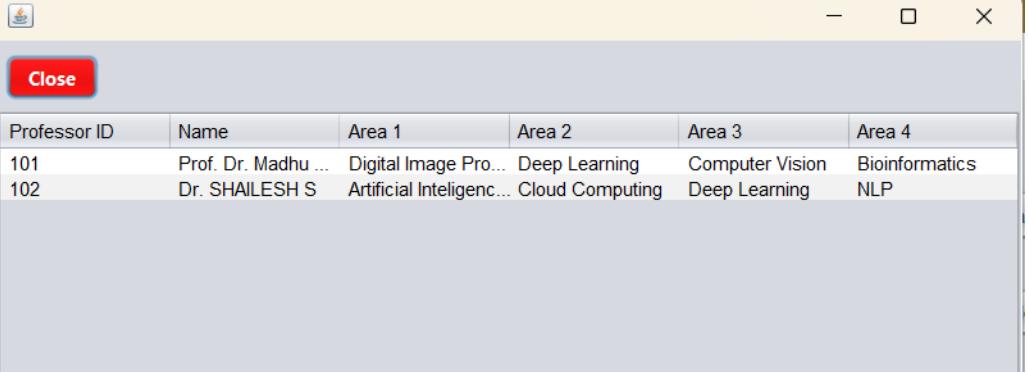
### ER diagram



### GUI







A screenshot of a Java Swing application window titled "Professor Data". The window contains a table with six columns: Professor ID, Name, Area 1, Area 2, Area 3, and Area 4. There are two rows of data:

| Professor ID | Name                | Area 1                   | Area 2          | Area 3          | Area 4         |
|--------------|---------------------|--------------------------|-----------------|-----------------|----------------|
| 101          | Prof. Dr. Madhu ... | Digital Image Pro...     | Deep Learning   | Computer Vision | Bioinformatics |
| 102          | Dr. SHAILESH S      | Artificial Inteligenc... | Cloud Computing | Deep Learning   | NLP            |

## SOFTWARE TOOLS

java.sql.Connection  
java.sql.DriverManager  
java.sql.SQLException  
javax.swing.JOptionPane  
javax.swing.table.DefaultTableModel  
java.sql.ResultSet

## IMPLEMENTATION

User login :

```
private void loginActionPerformed(java.awt.event.ActionEvent evt)
{
    String userName = username.getText();
    String pass = password.getText();
    String usertype = (String)login_combobox.getSelectedItem();

    if(usertype == "HOD"){
        if(userName.equals("hod") && pass.equals("hodpass")){
            System.out.println(userName+" "+pass);
            HOD h = new HOD();
            h.setVisible(true);
            this.setVisible(false);
        }
    }
}
```

```
JOptionPane.showMessageDialog(null  
,"Invalid Credentials");  
}  
}  
  
else if(usertype == "Student"){  
try{  
    Class.forName("com.mysql.jdbc.Driver");  
    Connection con=DriverManager.getConnection("jdbc:  
mysql://localhost:3306/  
projectallocationsystem","root","root");  
    Statement stml=con.createStatement();  
    ResultSet rs=stml.executeQuery("select * from student  
where username = '" +userName+ "' and  
passwords = '" +pass+ "'");  
  
    if(rs.next())  
    {  
        dispose();  
        Student S = new Student(rs);  
        S.setVisible(true);  
    }  
    else{  
        JOptionPane.showMessageDialog(null,  
        "Invalid Credentials");  
    }  
}  
catch(Exception e){  
    System.out.println(e);  
}  
}
```

```
else if(userType == "Professor"){

    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/projectallocationsystem","root","root");
        Statement stml=con.createStatement();
        ResultSet rs=stml.executeQuery("select * from professor where username = '" +userName+ "' and password = '" +pass+"'");
        if(rs.next())
        {
            dispose();
            Professor P = new Professor(rs);
            P.setVisible(true);
        }
        else{
            JOptionPane.showMessageDialog(null,
                "Invalid Credentials");
        }
    }

    catch(Exception e){
        System.out.println(e);
    }
}

}
```

## New User Sign In

```
private void SubmitBottonActionPerformed(java.awt.event.ActionEvent evt) {
```

```
event.ActionEvent evt) {  
    String name = Tname.getText();  
    String course = Tcourse.getText();  
    String semString = Tsem.getText();  
    //      int sem = Integer.parseInt(semString);  
    String stream = Tstream.getText();  
    String area1 = (String)Combo1.getSelectedItem();  
    String area2 = (String)Combo2.getSelectedItem();  
    String area3 = (String)Combo3.getSelectedItem();  
    String area4 = (String)Combo4.getSelectedItem();  
    String password = pass.getText();  
    String usernamee = username.getText();  
    String passComfirm = passcomf.getText();  
    if(password.equals(passComfirm)){  
        try{  
            Class.forName("com.mysql.jdbc.Driver");  
            Connection con=DriverManager.getConnection("jdbc:mysql:  
//localhost:3306/projectallocationsystem","root","root");  
            Statement stml=con.createStatement();  
            String sql1 = "insert into student(sname,course,  
semester,BtechOrPlustwo,  
topic1,topic2,topic3,topic4,username,passwords)  
values('"+name+"','"+course+"','"+semString+"',  
'"+stream+"','"+area1+"','"+area2+"','"+area3+"',  
'"+area4+"','"+usernamee+"','"+password+"')";  
            stml.executeUpdate(sql1);  
            JOptionPane.showMessageDialog(null,"Sign in successfull");  
        }  
        catch(Exception e){  
            System.out.println(e);  
        }  
    }  
}
```

```
    }
    else{
        JOptionPane.showMessageDialog(null,"Password
        and Confirmation password are different");
    }
}
```

## Project Allocation

```
private void getProjectActionPerformed(java.awt.
event.ActionEvent evt) {
    // TODO add your handling code here:
    int [] [] projectMatching = new int[3][2];
    for(int i=0;i<3;i++){
        projectMatching[i][1] = 0;
    }
    if(this.allocatedProject == 0){

try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con=DriverManager.getConnection("jdbc:
mysql://localhost:3306/projectallocationsystem"
    ,"root","root");
    Statement stml=con.createStatement();
    ResultSet rs=stml.executeQuery("select
    * from project");

    if(!rs.next())
    {
        JOptionPane.showMessageDialog(null,
        "No project available");
    }
}
}
```

```
    }

    else{

        rs_project = rs;

        int projectid=0;
        System.out.println("Projects : ");

        do{

            projectMatching[projectid][0] = rs.getInt("prid");

            for(int i =0;i<4;i++){
                for(int j=0;j<4;j++){
                    if(interestedTopics[i] ==
                        null ? rs.getString(j+4) == null :
                        interestedTopics[i].equals(rs.
                        getString(j+4))){
                        projectMatching[projectid][1]++;
                    }
                }
            }

            System.out.println(rs_project.getString(2));
            projectid++;
        }while(rs.next());

    }

    catch(HeadlessException | ClassNotFoundException
```

```
| SQLException e){  
    System.out.println(e);  
}  
  
  
for(int i =0;i<3;i++){  
    for(int j=0;j<2;j++){  
        System.out.print(projectMatching[i][j]+" ");  
    }  
    System.out.println(" ");  
}  
  
  
Arrays.sort(projectMatching, (final int []  
entry1, final int[] entry2) -> {  
    if(entry1[1] > entry2[1])  
        return 1;  
    else  
        return -1;  
});  
  
  
System.out.println(" ");  
System.out.println("After sort : ");  
System.out.println(" ");  
for(int i =0;i<3;i++){  
    for(int j=0;j<2;j++){  
        System.out.print(projectMatching[i][j]+" ");  
    }  
    System.out.println(" ");  
}  
  
  
try{
```

```
//Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection("jdbc:
mysql://localhost:3306/
projectallocationsystem","root","root");
Statement stml=con.createStatement();
String sql1 = "update student set
allocated_project = '" + projectMatching[2][0] +
where sid = '" + studentID + "' ;
stml.executeUpdate(sql1);
JOptionPane.showMessageDialog(null,"New Project
"+projectMatching[2][0]+" added");
jtextProject.setText(String.valueOf
(projectMatching[2][0]));
}
catch(Exception e){
    System.out.println(e);
}
this.allocatedProject = projectMatching[2][0];
}

else{
    JOptionPane.showMessageDialog(null,
    "Project is "+allocatedProject);
}
}
```

## RESULT AND OUTPUT

**Project Allocation System**

Student ID :  Project

Name  Log Book

Semester  Professor

Gantt Chart

Project ID  Professor

Professor

Notice Board

**Message**

 New allocated professor is 102

| Student        | Project ID | Professor ID | Interested Topic1       | Interested Topic2 | Interested Topic3 | Interested Topic4        |
|----------------|------------|--------------|-------------------------|-------------------|-------------------|--------------------------|
| Abhin          | 3          | 102          | Artificial Intelligence | ML                | Deep Learning     | Cloud Computing          |
| Noble          | 2          | 102          | Deep Learning           | Quantum Computing | NLP               | IOT                      |
| Abhishek       | 1          | 101          | Deep Learning           | Computer Vision   | Bioinformatics    | Digital Image Processing |
| Abhishek Mohan | 2          | 102          | Deep Learning           | ML                | IOT               | Artificial Intelligence  |

**Enter Project ID**

| Project ID | Project Name        | Area1                   | Area2                   | Area3                    | Area4           |
|------------|---------------------|-------------------------|-------------------------|--------------------------|-----------------|
| 1          | Face Detection      | ML                      | Deep Learning           | Digital Image Processing | Computer Vision |
| 2          | Chatbox             | Artificial Intelligence | NLP                     | IOT                      | Deep Learning   |
| 3          | Language Identifier | NLP                     | Artificial Intelligence | Deep Learning            | ML              |

## CRITICAL EVALUATION

Login

| No | Input                              | Expected o/p | Actual o/p |
|----|------------------------------------|--------------|------------|
| 1  | username = wrong, pswd = correct   | fail         | fail       |
| 2  | username = correct, pswd = wrong   | fail         | fail       |
| 3  | username = correct, pswd = correct | login        | login      |

### Project Searching and deleting using project id

| No | Input          | Expected o/p        | Actual o/p          |
|----|----------------|---------------------|---------------------|
| 1  | project id = 1 | Face detection      | Face detection      |
| 2  | project id = 2 | chatbox             | chatbox             |
| 3  | project id = 3 | Language identifier | Language identifier |

### Project allocation

| No | Input                       | Output             |
|----|-----------------------------|--------------------|
| 1  | interested topic of student | matching project 2 |
| 2  | interested topic of student | matching project 2 |
| 3  | interested topic of student | matching project 1 |

### Professor allocation

| No | Input                       | Output                 |
|----|-----------------------------|------------------------|
| 1  | interested topic of student | matching professor 102 |
| 2  | interested topic of student | matching professor 102 |
| 3  | interested topic of student | matching professor 101 |

## CONCLUSION

Project Allocation System is software that store basic details of a department also it allocate projects to students based on their field of expertise and interest and provide the domain expert mentors

## REFERENCES

MySQL

JAVA Database Connectivity

JAVA Swing