

**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

**AKSHAY K S
(80521005)**

**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 **AKSHAY K S(80521005)**, 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

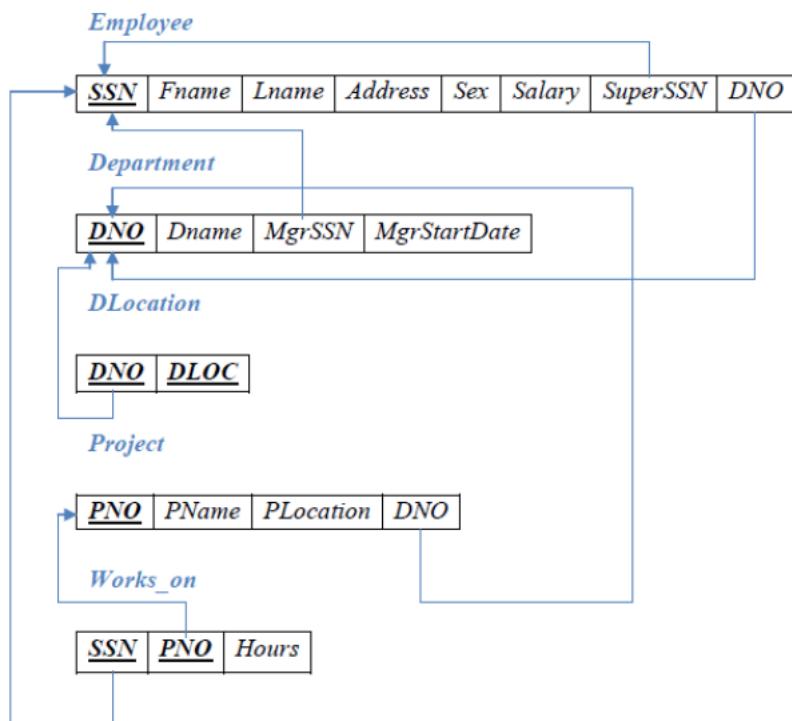
Sajmi Salaam
Guest Faculty
Department of Computer Science
CUSAT

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

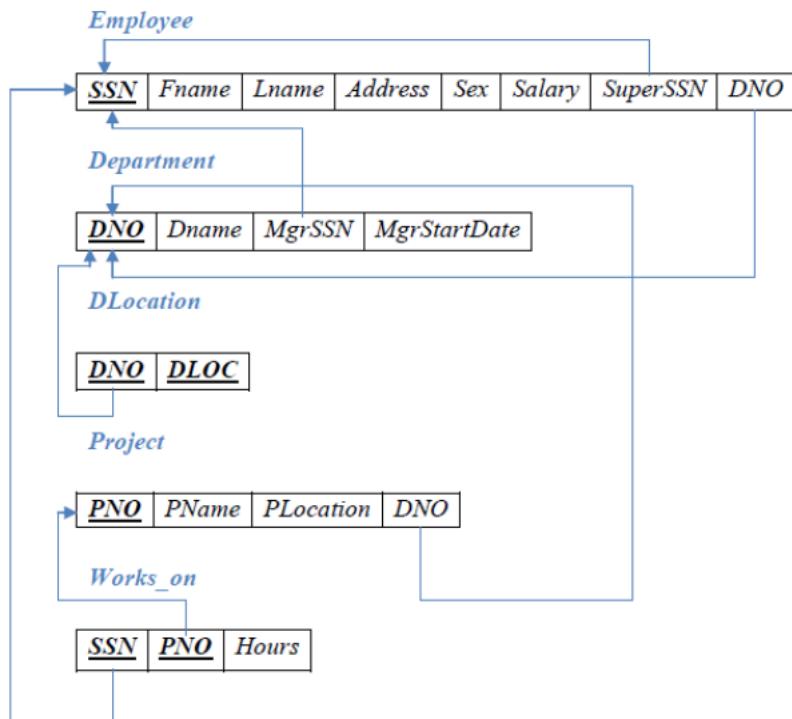
Table of Contents

Sl.No.	Program	Page.No.
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

```
mysql> use COMPANY;
```

```
Database changed
```

```
mysql> create table EMPLOYEE(SSN varchar(50) primary key not null,Fname  
varchar(30),Lname varchar(30) null,Address varchar(50),Sex varchar(20)  
default 'Male',Salary int(30) check(Salary>10000),SuperSSN varchar(20)  
unique,DNO varchar(20));
```

```
Query OK, 0 rows affected, 1 warning (0.02 sec)
```

```
mysql> create table DEPARTMENT(DNO varchar(20) primary key,Dname varchar(20),  
MgrSSN varchar(20), MgrStartData date not null);
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> create table PROJECT(PNO varchar(20) primary key,PName varchar(30),  
PLocation varchar(20),DNO varchar(20),constraint dno_project foreign key(DNO)  
references DEPARTMENT(DNO));
```

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

```
mysql> create table WORKS_ON(SSN varchar(20) primary key,PNO varchar(20),  
Hours integer);
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> create table DLOCATION(DNO varchar(20) primary key,DLOC varchar(20));
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> desc EMPLOYEE;  
  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> desc DEPARTMENT;  
  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> desc PROJECT;  
  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> desc WORKS_ON;  
  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> desc DLOCATION;  
  
Query OK, 0 rows affected (0.01 sec)
```

DATABASE TABLES

```
mysql> DESC EMPLOYEE;  
+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+  
| SSN | varchar(50) | NO | PRI | NULL |  
| Fname | varchar(30) | YES | | NULL |  
| Lname | varchar(30) | YES | | NULL |  
| Address | varchar(50) | YES | | NULL |  
| Sex | varchar(20) | YES | | Male |  
| Salary | int | YES | | NULL |  
| SuperSSN | varchar(20) | YES | UNI | NULL |  
| DNO | varchar(20) | YES | | NULL |  
+-----+-----+-----+-----+-----+  
  
mysql> desc DEPARTMENT;  
+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+  
| DNO | varchar(20) | NO | PRI | NULL |  
| Dname | varchar(20) | YES | | NULL |  
| MgrSSN | varchar(20) | YES | | NULL |  
| MgrStartData | date | NO | | NULL |  
+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)
```

Field	Type	Null	Key	Default	Extra
PNO	varchar(20)	NO	PRI	NULL	
PName	varchar(30)	YES		NULL	
PLocation	varchar(20)	YES		NULL	
DNO	varchar(20)	YES	MUL	NULL	

Field	Type	Null	Key	Default	Extra
SSN	varchar(20)	NO	PRI	NULL	
PNO	varchar(20)	YES		NULL	
Hours	int(11)	YES		NULL	

Field	Type	Null	Key	Default	Extra
DNO	varchar(20)	NO	PRI	NULL	
DLOC	varchar(20)	YES		NULL	

Questions : 2

Add another column Age with datatype integer in employee table

QUERY

```
mysql> alter table EMPLOYEE add(AGE integer);
```

```
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

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

```
mysql> DROP TABLE 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

```
mysql> TRUNCATE TABLE WORKS_ON;
```

```
Query OK, 0 rows affected (0.01 sec)
```

DATABASE TABLES

Field	Type	Null	Key	Default	Extra
SSN	varchar(20)	NO	PRI	NULL	
PNO	varchar(20)	YES		NULL	
Hours	int(11)	YES		NULL	

Questions : 5

View the structure of the table Department

QUERY

```
mysql> DESC DEPARTMENT;
```

DATABASE TABLES

```
mysql> desc DEPARTMENT;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| DNO        | varchar(20) | NO   | PRI | NULL    |       |
| Dname      | varchar(20) | YES  |     | NULL    |       |
| MgrSSN     | varchar(20) | YES  |     | NULL    |       |
| MgrStartData | date      | NO   |     | 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

```
mysql> insert into EMPLOYEE values ("e1001","Raj","Sharma","Mumbai","Male",
15000,"s1001","d01",22);
Query OK, 1 row affected (0.00 sec)

mysql> insert into EMPLOYEE values ("e1002","Ria","Chopra","Chennai","Female",
20000,"s1002","d02",22);
Query OK, 1 row affected (0.01 sec)

mysql> insert into EMPLOYEE values ("e1003","Hira","Mohammed","Kochi","Female",
17000,"s1003","d03",20);
Query OK, 1 row affected (0.00 sec)

mysql> insert into EMPLOYEE values ("e1004","Simran","S","Chennai","Female",
15000,"s1004","d04",21);
Query OK, 1 row affected (0.01 sec)

mysql> insert into DEPARTMENT values("d01","Marketing","m1001","2022-07-04");
Query OK, 1 row affected (0.00 sec)

mysql> insert into DEPARTMENT values("d02","Finance","m1002","2022-04-07");
Query OK, 1 row affected (0.00 sec)

mysql> insert into DEPARTMENT values("d03","Sales","m1003","2022-08-18");
Query OK, 1 row affected (0.00 sec)

mysql> insert into DEPARTMENT values("d04","Human Resources","m1004","2022-02-1
7");
Query OK, 1 row affected (0.00 sec)
```

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into DEPARTMENT values("d05","Management","m1005","2021-12-17");
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into PROJECT values("p1001","Project Synergy","Chennai","d01");
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into PROJECT values ("p1002","Project Point","Mumbai","d02");
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into PROJECT values ("p1003","Strategic Program","Delhi","d03");
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into PROJECT values ("p1004","Apollo","Mumbai","d04");
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into PROJECT values ("p1005","Astron","Chennai","d05");
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into WORKS_ON values("e1001","p1001",7);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into WORKS_ON values("e1002","p1002",10);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into WORKS_ON values("e1003","p1003",9);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into WORKS_ON values("e1004","p1004",10);
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into WORKS_ON values("e1005","p1005",4);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into DLOCATION values("d01","Chennai");
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into DLOCATION values("d02","Mumbai");
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into DLOCATION values("d03","Delhi");
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into DLOCATION values("d04","Mumbai");
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into DLOCATION values("d05","Chennai");
Query OK, 1 row affected (0.01 sec)
```

Questions : 2

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

QUERY

```
mysql> SELECT * FROM EMPLOYEE;
mysql> SELECT * FROM DEPARTMENT;
mysql> SELECT * FROM PROJECT;
mysql> SELECT * FROM WORKS_ON;
mysql> SELECT * FROM DLOCATION;
```

DATABASE TABLES

```
mysql> select * FROM EMPLOYEE;
+-----+-----+-----+-----+-----+-----+-----+-----+
| SSN | Fname | Lname | Address | Sex | Salary | SuperSSN | DNO | AGE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| e1001 | Raj | Sharma | Mumbai | Male | 15000 | s1001 | d01 | 22 |
| e1002 | Ria | Chopra | Chennai | Female | 20000 | s1002 | d02 | 22 |
| e1003 | Hira | Mohammed | Kochi | Female | 17000 | s1003 | d03 | 20 |
| e1004 | Simran | S | Chennai | Female | 15000 | s1004 | d04 | 21 |
| e1005 | Ashwin | K | Mumbai | Male | 25000 | s1005 | d05 | 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM DEPARTMENT;
+-----+-----+-----+-----+
| DNO | Dname | MgrSSN | MgrStartDate |
+-----+-----+-----+-----+
| d01 | Marketing | m1001 | 2022-07-04 |
| d02 | Finance | m1002 | 2022-04-07 |
| d03 | Sales | m1003 | 2022-08-18 |
| d04 | Human Resources | m1004 | 2022-02-01 |
| d05 | Management | m1005 | 2021-12-17 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
|mysql> SELECT * FROM PROJECT;
+-----+-----+-----+-----+
| PNO   | PName        | PLocation | DNO   |
+-----+-----+-----+-----+
| p1001 | Project Synergy | Chennai    | d01   |
| p1002 | Project Point   | Mumbai     | d02   |
| p1003 | Strategic Program | Delhi      | d03   |
| p1004 | Apollo         | Mumbai     | d04   |
| p1005 | Astron         | Chennai    | d05   |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
|mysql> select *from WORKS_ON;
+-----+-----+-----+
| SSN   | PNO    | Hours  |
+-----+-----+-----+
| e1001 | p1001 |      7 |
| e1002 | p1002 |     10 |
| e1003 | p1003 |      9 |
| e1004 | p1004 |     10 |
| e1005 | p1005 |      4 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
|mysql> SELECT * FROM DLOCATION;
+-----+-----+
| DNO  | DLOC    |
+-----+-----+
| d01  | Chennai |
| d02  | Mumbai  |
| d03  | Delhi   |
| d04  | Mumbai  |
| d05  | Chennai |
+-----+-----+
5 rows in set (0.00 sec)
```

Questions : 3

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

QUERY

```
mysql> Update EMPLOYEE set SALARY=25000 where SSN="e1002";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

DATABASE TABLES

```
mysql> SELECT * FROM EMPLOYEE;
+-----+-----+-----+-----+-----+-----+-----+-----+
| SSN | Fname | Lname | Address | Sex | Salary | SuperSSN | DNO | AGE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| e1001 | Raj | Sharma | Mumbai | Male | 15000 | s1001 | d01 | 22 |
| e1002 | Ria | Chopra | Chennai | Female | 25000 | s1002 | d02 | 22 |
| e1003 | Hira | Mohammed | Kochi | Female | 17000 | s1003 | d03 | 20 |
| e1004 | Simran | S | Chennai | Female | 15000 | s1004 | d04 | 21 |
| e1005 | Ashwin | K | Mumbai | Male | 25000 | s1005 | d05 | 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Questions : 4

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

QUERY

```
mysql> Update EMPLOYEE set SALARY=25000 where SSN="e1002";
Query OK, 1 row affected (0.00 sec)

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

DATABASE TABLES

```
mysql> SELECT * FROM EMPLOYEE;
+-----+-----+-----+-----+-----+-----+-----+-----+
| SSN | Fname | Lname | Address | Sex | Salary | SuperSSN | DNO | AGE |
+-----+-----+-----+-----+-----+-----+-----+-----+
| e1001 | Raj | Sharma | Mumbai | Male | 15000 | s1001 | d01 | 22 |
| e1003 | Hira | Mohammed | Kochi | Female | 17000 | s1003 | d03 | 20 |
| e1004 | Simran | S | Chennai | Female | 15000 | s1004 | d04 | 21 |
| e1005 | Ashwin | K | Mumbai | Male | 25000 | s1005 | d05 | 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+
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

```
mysql> create user 'user1'@'localhost' identified by 'password';
Query OK, 0 rows affected (0.01 sec)

mysql> grant select on COMPANY.EMPLOYEE to 'user1'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

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

```
mysql> grant all on COMPANY.EMPLOYEE to 'user1'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> Revoke all on EMPLOYEE from 'user1'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

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

```
mysql> select Fname from EMPLOYEE where Fname like '%a_';
```

DATABASE TABLES

Fname
Raj
Simran

Questions : 2

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

QUERY

```
mysql> select Sex,count(*) from EMPLOYEE group by Sex;
```

DATABASE TABLES

Sex	count(*)
Female	3
Male	2

Questions : 3

Calculate the average salary of the female employees.

QUERY

```
mysql> select avg(Salary) from EMPLOYEE where Sex="Female";
```

DATABASE TABLES

avg(Salary)
17333.3333

Questions : 4

Calculate the sum of salaries of male employees.

QUERY

```
mysql> select sum(Salary) from EMPLOYEE where Sex="Male";
```

DATABASE TABLES

sum(Salary)
40000

Questions : 5

Display the maximum and minimum salaries of male employees.

QUERY

```
mysql> select max(Salary),min(Salary) from EMPLOYEE where Sex="Male";
```

DATABASE TABLES

	max(Salary)	min(Salary)
	25000	15000

Questions : 6

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

QUERY

```
mysql> select * from EMPLOYEE where Salary between 25000 and 50000;
```

DATABASE TABLES

SSN	Fname	Lname	Address	Sex	Salary	SuperSSN	DNO	AGE
e1005	Ashwin	K	Mumbai	Male	25000	s1005	d05	21

Questions : 7

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

QUERY

```
mysql> select Lname from EMPLOYEE where Salary=30000 or Salary=40000  
or Salary = 50000;
```

DATABASE TABLES

Lname

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

```
mysql> Update EMPLOYEE,PROJECT set Salary=Salary+(Salary*0.25)/100 where  
EMPLOYEE.DNo=PROJECT.DNo and PLocation="Chennai";  
Query OK, 2 rows affected (0.01 sec)  
Rows matched: 2 Changed: 2 Warnings: 0
```

DATABASE TABLES

SSN	Fname	Lname	Address	Sex	Salary	SuperSSN	DNo	AGE
e1001	Raj	Sharma	Mumbai	Male	15000	s1001	d01	22
e1002	Ria	Chopra	Chennai	Female	20000	s1002	d02	22
e1003	nazeem	Mohammed	Kochi	Female	17000	s1003	d03	20
e1004	Simran	S	Chennai	Female	15000	s1004	d04	21
e1005	Ashwin	K	Mumbai	Male	25000	s1005	d05	21

Questions : 2

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

QUERY

```
mysql> select EMPLOYEE.Fname,PROJECT.PLocation FROM EMPLOYEE,WORKS_ON,PROJECT  
where WORKS_ON.Hours > 5 and EMPLOYEE.SSN = WORKS_ON.SSN  
and WORKS_ON.PNo=PROJECT.PNo;
```

DATABASE TABLES

	Fname	PLocation

Questions : 3

Left join employee table and works_on table

QUERY

```
mysql> select * from EMPLOYEE left join WORKS_ON on EMPLOYEE.SSN = WORKS_ON.SSN;
```

DATABASE TABLES

SSN	Fname	Lname	Address	Sex	Salary	SuperSSN	DNO	AGE	SSN	PNO	Hours
e1001	Raj	Sharma	Mumbai	Male	15000	s1001	d01	22	NULL	NULL	NULL
e1002	Ria	Chopra	Chennai	Female	20000	s1002	d02	22	NULL	NULL	NULL
e1003	nazeem	Mohammed	Kochi	Female	17000	s1003	d03	20	NULL	NULL	NULL
e1004	Simran	S	Chennai	Female	15000	s1004	d04	21	NULL	NULL	NULL
e1005	Ashwin	K	Mumbai	Male	25000	s1005	d05	21	NULL	NULL	NULL

Questions : 4

Right join works_on table and employee table

QUERY

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

DATABASE TABLES

SSN	PNO	Hours	SSN	Fname	Lname	Address	Sex	Salary	SuperSSN	DNO	AGE
NULL	NULL	NULL	e1001	Raj	Sharma	Mumbai	Male	15000	s1001	d01	22
NULL	NULL	NULL	e1002	Ria	Chopra	Chennai	Female	20000	s1002	d02	22
NULL	NULL	NULL	e1003	nazeem	Mohammed	Kochi	Female	17000	s1003	d03	20
NULL	NULL	NULL	e1004	Simran	S	Chennai	Female	15000	s1004	d04	21
NULL	NULL	NULL	e1005	Ashwin	K	Mumbai	Male	25000	s1005	d05	21

Questions : 5

Full join works_on table and employee table

QUERY

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

DATABASE TABLES

SSN	PNO	Hours	SSN	Fname	Lname	Address	Sex	Salary	SuperSSN	DNO	AGE
e1001	p1001	7	e1001	Raj	Sharma	Mumbai	Male	15000	s1001	d01	22
e1002	p1002	10	e1001	Raj	Sharma	Mumbai	Male	15000	s1001	d01	22
e1003	p1003	9	e1001	Raj	Sharma	Mumbai	Male	15000	s1001	d01	22
e1004	p1004	10	e1001	Raj	Sharma	Mumbai	Male	15000	s1001	d01	22
e1005	p1005	4	e1001	Raj	Sharma	Mumbai	Male	15000	s1001	d01	22
e1001	p1001	7	e1002	Ria	Chopra	Chennai	Female	20000	s1002	d02	22
e1002	p1002	10	e1002	Ria	Chopra	Chennai	Female	20000	s1002	d02	22
e1003	p1003	9	e1002	Ria	Chopra	Chennai	Female	20000	s1002	d02	22
e1004	p1004	10	e1002	Ria	Chopra	Chennai	Female	20000	s1002	d02	22
e1005	p1005	4	e1002	Ria	Chopra	Chennai	Female	20000	s1002	d02	22
e1001	p1001	7	e1003	nazeem	Moha...	Kochi	Female	17000	s1003	d03	20
e1002	p1002	10	e1003	nazeem	Moha...	Kochi	Female	17000	s1003	d03	20
e1003	p1003	9	e1003	nazeem	Moha...	Kochi	Female	17000	s1003	d03	20
e1004	p1004	10	e1003	nazeem	Moha...	Kochi	Female	17000	s1003	d03	20
e1005	p1005	4	e1003	nazeem	Moha...	Kochi	Female	17000	s1003	d03	20
e1001	p1001	7	e1004	Simran	S	Chennai	Female	15000	s1004	d04	21
e1002	p1002	10	e1004	Simran	S	Chennai	Female	15000	s1004	d04	21
e1003	p1003	9	e1004	Simran	S	Chennai	Female	15000	s1004	d04	21
e1004	p1004	10	e1004	Simran	S	Chennai	Female	15000	s1004	d04	21
e1005	p1005	4	e1004	Simran	S	Chennai	Female	15000	s1004	d04	21
e1001	p1001	7	e1005	Ashwin	K	Mumbai	Male	25000	s1005	d05	21
e1002	p1002	10	e1005	Ashwin	K	Mumbai	Male	25000	s1005	d05	21
e1003	p1003	9	e1005	Ashwin	K	Mumbai	Male	25000	s1005	d05	21
e1004	p1004	10	e1005	Ashwin	K	Mumbai	Male	25000	s1005	d05	21
e1005	p1005	4	e1005	Ashwin	K	Mumbai	Male	25000	s1005	d05	21

VIEWS

AIM

Develop SQL Queries for creating and dropping Views

Questions : 1

Create a view VW_emp on employee table

QUERY

```
mysql> create view VW_emp as select*from EMPLOYEE;
Query OK, 0 rows affected (0.00 sec)
```

DATABASE TABLES

SSN	Fname	Lname	Address	Sex	Salary	SuperSSN	DNO	AGE
e1001	Raj	Sharma	Mumbai	Male	15000	s1001	d01	22
e1002	Ria	Chopra	Chennai	Female	20000	s1002	d02	22
e1003	nazeem	Mohammed	Kochi	Female	17000	s1003	d03	20
e1004	Simran	S	Chennai	Female	15000	s1004	d04	21
e1005	Ashwin	K	Mumbai	Male	25000	s1005	d05	21

Questions : 2

Create another view VW_SSN contains SuperSSN and Dno of female employees

QUERY

```
mysql> create view VW_SSN as select SuperSSN, DNO from EMPLOYEE
where Sex = 'Female';
Query OK, 0 rows affected (0.01 sec)
```

DATABASE TABLES

SuperSSN	DNO
s1002	d02
s1003	d03
s1004	d04

Questions : 3

Update the address of employee to Chennai whose id is e100 in view VW_emp

QUERY

```
mysql> UPDATE VW_emp SET Address="Chennai" WHERE SSN='e1001';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1    Changed: 1    Warnings: 0
```

DATABASE TABLES

SSN	Fname	Lname	Address	Sex	Salary	SuperSSN	DNO	AGE
e1001	Raj	Sharma	Chennai	Male	15000	s1001	d01	22
e1002	Ria	Chopra	Chennai	Female	20000	s1002	d02	22
e1003	nazeem	Mohammed	Kochi	Female	17000	s1003	d03	20
e1004	Simran	S	Chennai	Female	15000	s1004	d04	21
e1005	Ashwin	K	Mumbai	Male	25000	s1005	d05	21

Questions : 4

Delete the view VW_emp

QUERY

```
mysql> drop view VW_emp;
Query OK, 0 rows affected (0.00 sec)
```

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*Plus: Release 11.2.0.2.0 Production on Fri Dec 16 14:29:44 2022
```

```
Copyright (c) 1982, 2014, Oracle. All rights reserved.
```

```
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*Plus: Release 11.2.0.2.0 Production on Fri Dec 16 14:29:44 2022
```

```
Copyright (c) 1982, 2014, Oracle. All rights reserved.
```

```
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*Plus: Release 11.2.0.2.0 Production on Fri Dec 16 14:29:44 2022
```

```
Copyright (c) 1982, 2014, Oracle. All rights reserved.
```

```
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;
    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*Plus: Release 11.2.0.2.0 Production on Fri Dec 16 14:29:44 2022

Copyright (c) 1982, 2014, Oracle. All rights reserved.

```
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

Questions : 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> 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		
LNAME			
ADDRESS		SEX	SALARY
SUPERSSN	DNO		
e1001	Archana		
Suresh			
13B, Highway Gardens, Kozhikode		Female	60000
SP1002	D001		
SSN	FNAME		
LNAME			
ADDRESS		SEX	SALARY
SUPERSSN	DNO		
e1002	Justin		
Varghese			
Rose Villa, Kochi		Male	50000
SP1001	D002		
SSN	FNAME		
LNAME			
ADDRESS		SEX	SALARY
SUPERSSN	DNO		
e1003	Meera		
Kumar			
11B, Arcadia Building, Mumbai		Female	70000
SP1004	D001		
SSN	FNAME		
LNAME			
ADDRESS		SEX	SALARY
SUPERSSN	DNO		
e1004	Kailas		
Nath			
V3, DD Homes, Bangalore		Male	30000
SP1003	D003		

```
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	D001				
SSN	FNAME	LNAME	ADDRESS	SEX	SALARY
SUPERSSN	DNO				
e1002	Justin	Varghese	Rose Villa,Kochi	Male	50000
SP1001	D002				

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

Questions : 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:**

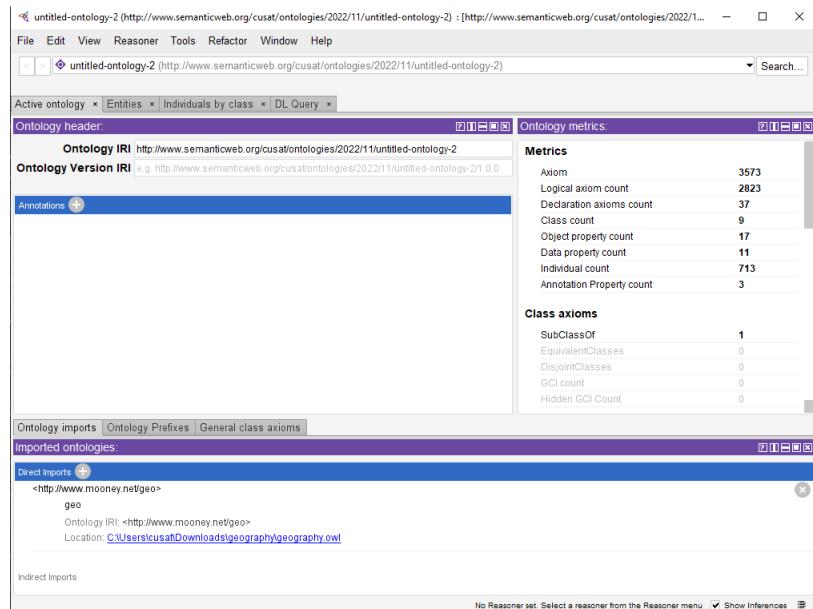
	City
1	< http://www.mooney.net/geo#alabama >
2	< http://www.mooney.net/geo#alabama >
3	< http://www.mooney.net/geo#alabama >
4	< http://www.mooney.net/geo#alabama >
5	< http://www.mooney.net/geo#alabama >
6	< http://www.mooney.net/geo#tennessee >
- Log Output:**

```

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
[023:55: INFO Server   :: Apache Jena Fuseki 4.6.1
[023:58: INFO Config  :: FUSEKI_HOME=C:\Users\cusat\Downloads\apache-jena-fuseki-4.6.1\apache-jena-fuseki-4.6.1\run
[023:58: INFO Config  :: FUSEKI_BASE=C:\Users\cusat\Downloads\apache-jena-fuseki-4.6.1\apache-jena-fuseki-4.6.1\run\shiro.ini
[024:00: INFO Config  :: Shiro file: file:///C:/Users/cusat/Downloads/apache-jena-fuseki-4.6.1/apache-jena-fuseki-4.6.1/run/shiro.ini
[024:00: INFO Config  :: Template file: templates/config-mem
[024:05: INFO Server   :: Database: in-memory
[024:05: INFO Server   :: Path = /ds
[024:05: INFO Server   :: System
[024:05: INFO Server   :: Memory: 1.2 GB
[024:05: INFO Server   :: Java: 19.0.1
[024:05: INFO Server   :: OS: Windows 10 10.0 amd64
[024:05: INFO Server   :: PID: 11092
[024:06: INFO Server   :: Started 2022/12/15 10:24:06 IST on port 3030
[024:44:40 INFO Admin   :: [3] Create database : name = /geography
[024:45:59 ERROR Fuseki  :: [line: 1, col: 1 ] Content is not allowed in prolog.
[024:50:00 INFO Fuseki  :: [7] Upload error: [line: 1, col: 1 ] Content is not allowed in prolog.
[024:50:00 INFO Fuseki  :: [7] 500 Server Error (632 ms)
[024:50:12 INFO Fuseki  :: [8] POST http://localhost:3030/geography/data
[024:50:12 INFO Fuseki  :: [line: 1, col: 1 ] Content is not allowed in prolog.
[024:50:12 INFO Fuseki  :: [8] Upload error: [line: 1, col: 1 ] Content is not allowed in prolog.
[024:50:12 INFO Fuseki  :: [8] 500 Server Error (417 ms)
[024:50:56 INFO Fuseki  :: [9] POST http://localhost:3030/geography/data
[024:50:56 INFO Fuseki  :: [9] Upload error: [line: 1, col: 1 ] Content is not allowed in prolog.
[024:50:56 INFO Fuseki  :: [9] 500 Server Error (433 ms)
[024:50:56 INFO Fuseki  :: [25] POST http://localhost:3030/geography/data
[024:50:56 INFO Fuseki  :: [line: 1, col: 1 ] Content is not allowed in prolog.
[024:50:56 INFO Fuseki  :: [25] Upload error: [line: 1, col: 1 ] Content is not allowed in prolog.
[024:50:56 INFO Fuseki  :: [25] 500 Server Error (477 ms)
[024:50:56 INFO Fuseki  :: [29] POST http://localhost:3030/geography/data
[024:50:56 INFO Fuseki  :: [line: 1, col: 1 ] Content is not allowed in prolog.
[024:50:56 INFO Fuseki  :: [29] Upload error: [line: 1, col: 1 ] Content is not allowed in prolog.
[024:50:56 INFO Fuseki  :: [29] 500 Server Error (438 ms)
[024:50:56 INFO Fuseki  :: [30] POST http://localhost:3030/geography/data
[024:50:56 INFO Fuseki  :: [line: 1, col: 1 ] Content is not allowed in prolog.
[024:50:56 INFO Fuseki  :: [30] Upload error: [line: 1, col: 1 ] Content is not allowed in prolog.
[024:50:56 INFO Fuseki  :: [30] 500 Server Error (488 ms)
[024:50:56 INFO Fuseki  :: [30] POST http://localhost:3030/geography/
[024:50:56:02 INFO Fuseki  :: [30] Filename: geography.owl, Content-Type=application/octet-stream, Charset=null => RDF/XML : Count=3589 Triples=3589 Quads=0
[024:50:56:03 INFO Fuseki  :: [30] 200 OK (1.272 s)
[024:50:56:36 INFO Fuseki  :: [31] POST http://localhost:3030/geography/
[024:50:56:36 INFO Fuseki  :: [31] Query - SELECT ?subject ?predicate ?object WHERE { ?subject ?predicate ?object } LIMIT 25
[024:50:56:36 INFO Fuseki  :: [31] 200 OK (65 ms)
[024:50:56:18 INFO Fuseki  :: [34] POST http://localhost:3030/geography/
[024:50:56:18 INFO Fuseki  :: [34] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }
[024:50:56:18 INFO Fuseki  :: [34] 200 OK (6 ms)
[024:50:56:59 INFO Fuseki  :: [35] POST http://localhost:3030/geography/
[024:50:56:59 INFO Fuseki  :: [35] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }
[024:50:56:59 INFO Fuseki  :: [35] 200 OK (4 ms)
[024:50:56:59 INFO Fuseki  :: [36] POST http://localhost:3030/geography/
[024:50:56:59 INFO Fuseki  :: [36] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }
[024:50:56:59 INFO Fuseki  :: [36] 200 OK (5 ms)
[024:50:56:59 INFO Fuseki  :: [36] POST http://localhost:3030/geography/
[024:50:56:59 INFO Fuseki  :: [36] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }
[024:50:56:59 INFO Fuseki  :: [36] 200 OK (5 ms)
[024:50:56:59:45 INFO Fuseki  :: [37] POST http://localhost:3030/geography/
[024:50:56:59:45 INFO Fuseki  :: [37] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }
[024:50:56:59:45 INFO Fuseki  :: [37] 200 OK (4 ms)
[024:50:56:59:48 INFO Fuseki  :: [38] POST http://localhost:3030/geography/
[024:50:56:59:48 INFO Fuseki  :: [38] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }
[024:50:56:59:48 INFO Fuseki  :: [38] 200 OK (5 ms)
[024:50:56:59:50 INFO Fuseki  :: [39] POST http://localhost:3030/geography/
[024:50:56:59:50 INFO Fuseki  :: [39] Query - prefix table:<https://www.mooney.net/geo> select ?name where { ?geo table:isCityOf ?city }
[024:50:56:59:50 INFO Fuseki  :: [39] 200 OK (5 ms)

```



JAVA DATABASE CONNECTIVITY

AIM

Develop program to implement Java Database Connectivity

Question : 2

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

DATABASE TABLES

Simple Library Management System

Enter ISBN :	ABC124
Enter Title :	Harry Potter and the half blood prince
Enter Author :	J.K. Rowling
Enter Edition :	2000
Enter Publication:	Bloomsbury
	<input type="button" value="submit"/> <input type="button" value="Reset"/>

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.

DATABASE TABLES

The screenshot shows a Java Swing application window titled "RETRIEVE DETAILS OF EMPLOYEE". At the top, there is a label "ENTER SSN" followed by a text input field and three buttons: "SEARCH" and "CLEAR ALL". Below this, the title "RETRIEVE DETAILS OF EMPLOYEE" is displayed in red. The main area contains six text input fields labeled "NAME", "ADDRESS", "AGE", "GENDER", "DEPARTMENT", and "DESIGNATION". To the right of these fields is a photograph of a modern building with the text "lenskart.com" visible on its facade. The entire application is set against a background of a green lawn and blue sky.

Question : 3

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

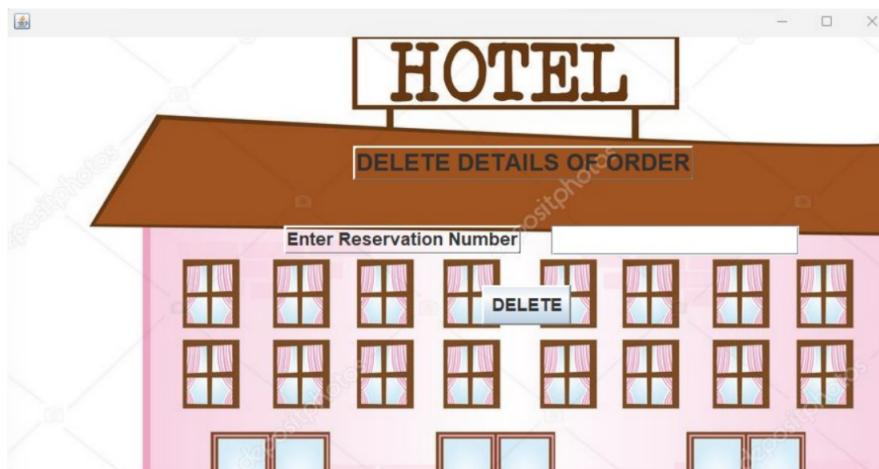
DATABASE TABLES



Question : 4

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

DATABASE TABLES



PROJECT

AIM

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

PROJECT DESCRIPTION

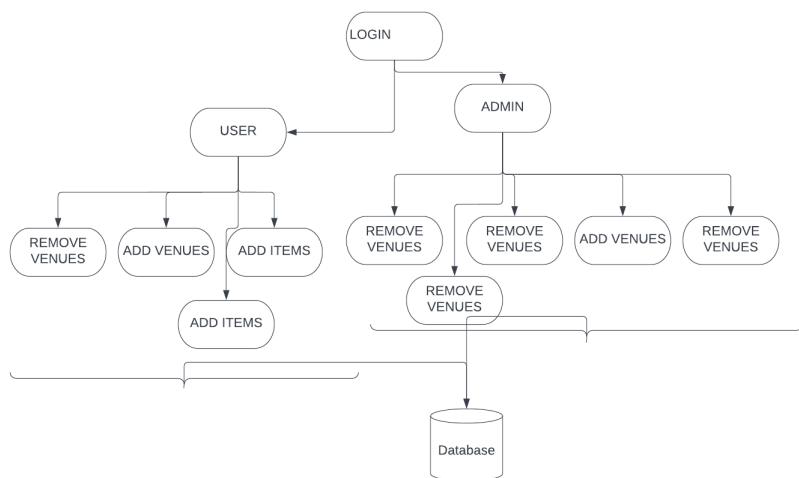
Software to allocate sports venue and sports items to costumers based on the availability. The costumers can easily view and select the required options. The software mainly aims for providing support to sports complexes which need to allocate it's venues properly.

USERS AND FUNCTIONALITIES

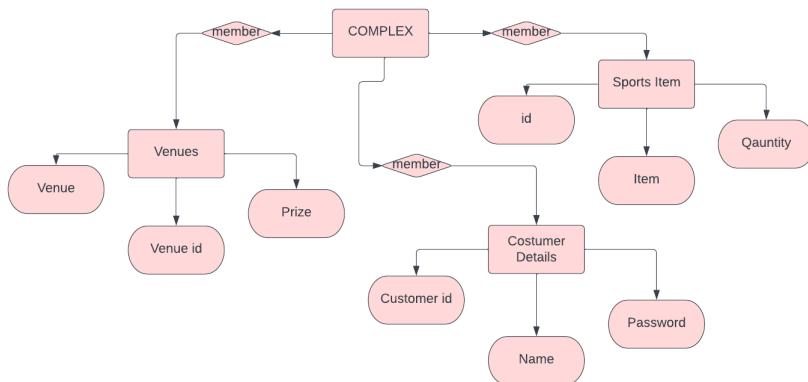
- ADMIN LOGIN
 - Add more Venues and items.
 - Remove Venues and items.
 - Add more Venues.
 - Remove users.
- USER LOGIN
 - Add more Venues and items to the list.
 - Remove Venues and items from the list.

REFERENCE DESIGN

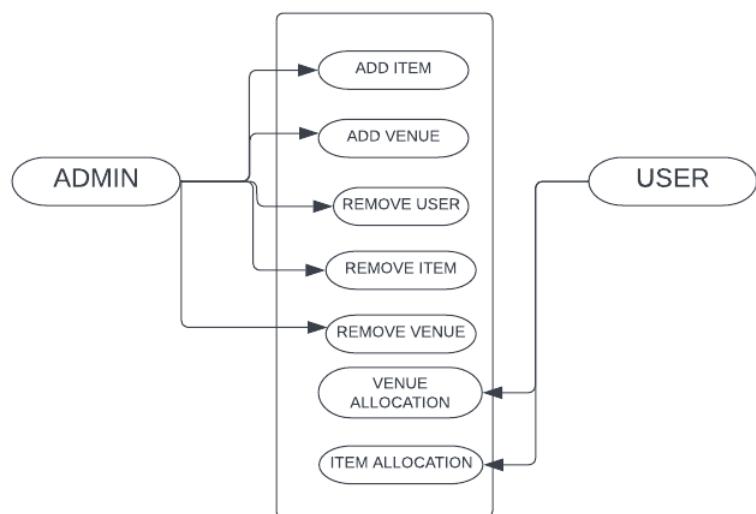
Activity diagram



ER DIAGRAM



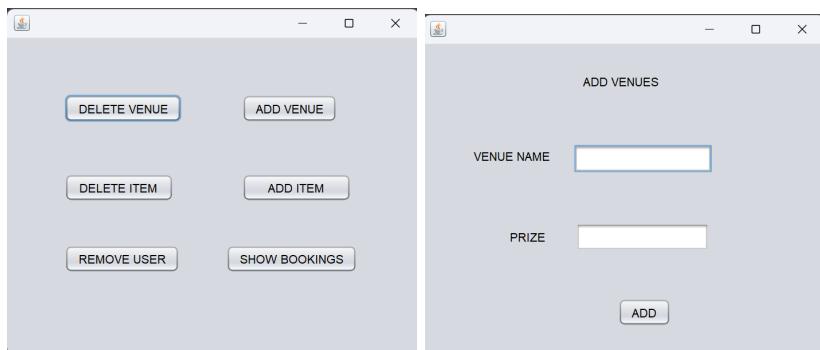
UML DIAGRAM



UI

The UI interface consists of two windows:

- SIGN UP** window:
 - Fields: **User Name**, **Password**, **Confirm Password**.
 - Buttons: **SIGN UP**.
- LOGIN PAGE** window:
 - Fields: **USERNAME**, **password**.
 - Buttons: **COSTUMER LOGIN**, **ADMIN LOGIN**, **SIGN UP**.



SOFTWARE TOOLS

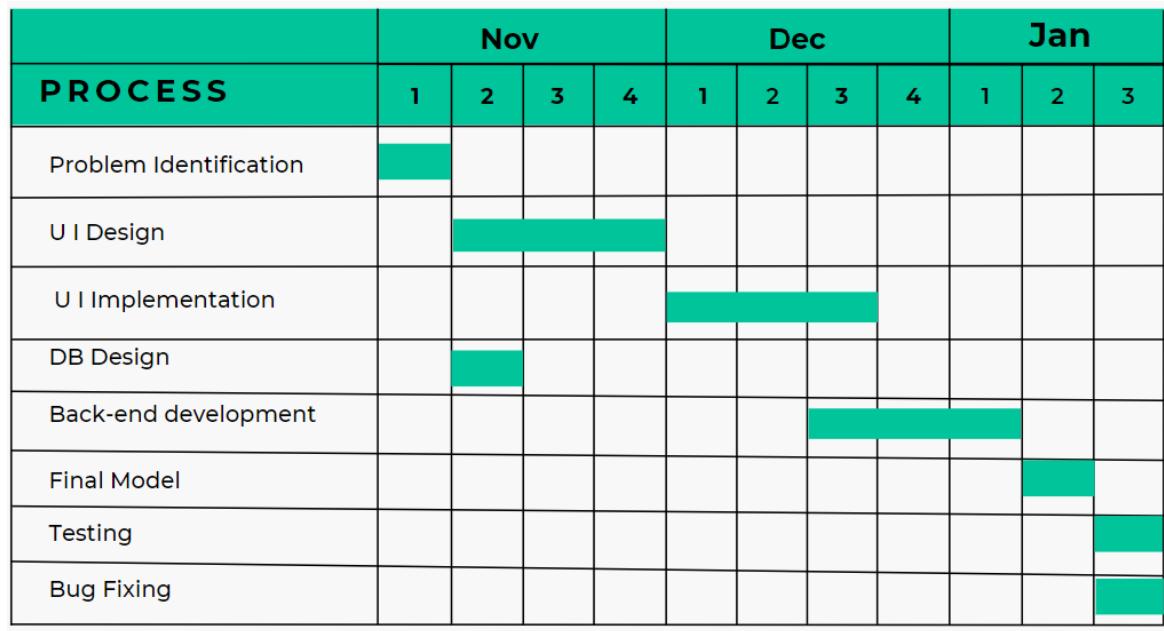
Apache Netbeans

MySQL

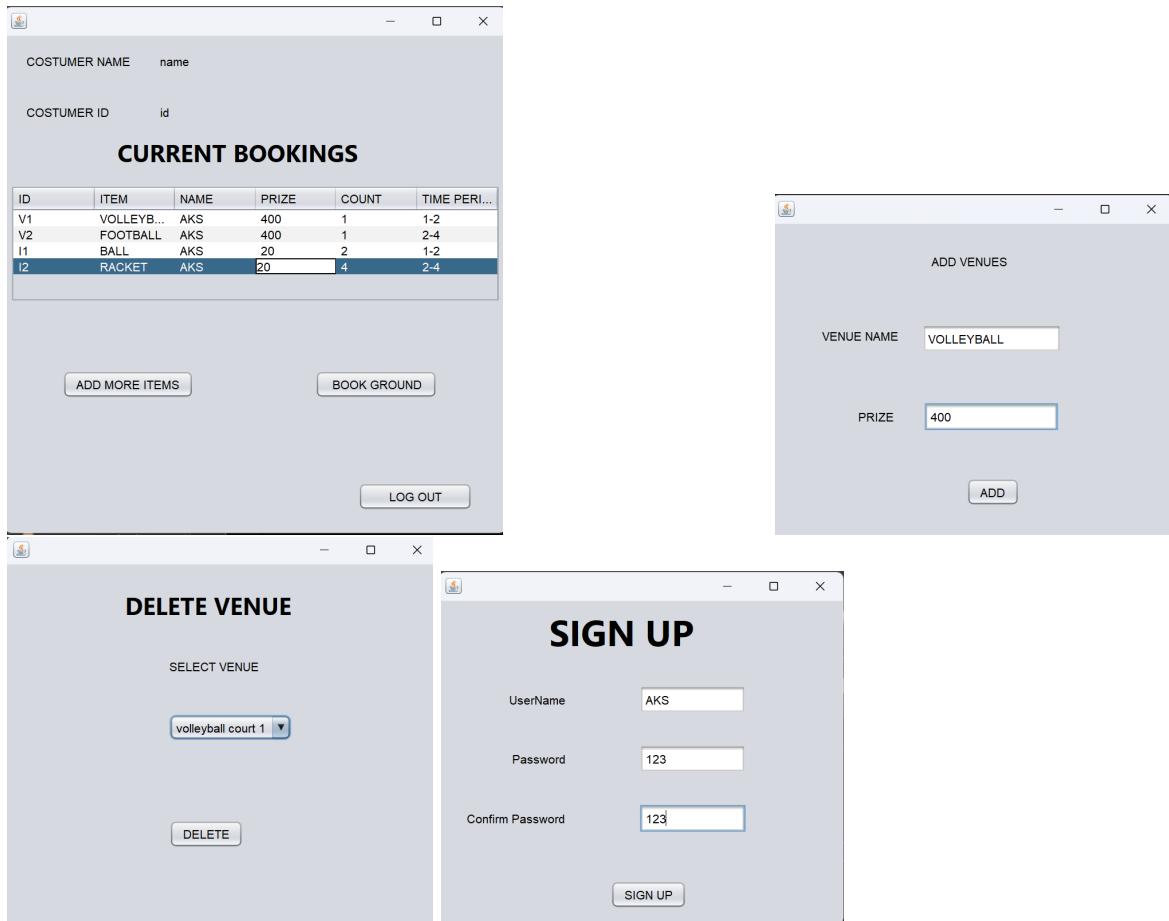
Java

IMPLEMENTATION

GANTT CHART



RESULT AND OUTPUT



CRITICAL EVALUATION

The software has been tested for various use cases and works properly.

CONCLUSION

It is an effective method for proper allocation of sports items and venues in a sports complex.

REFERENCES

MySQL
JAVA Database Connectivity
JAVA Swing