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

ABHISHEK P
(80521003)

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* ***ABHISHEK P(80521003)****, 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

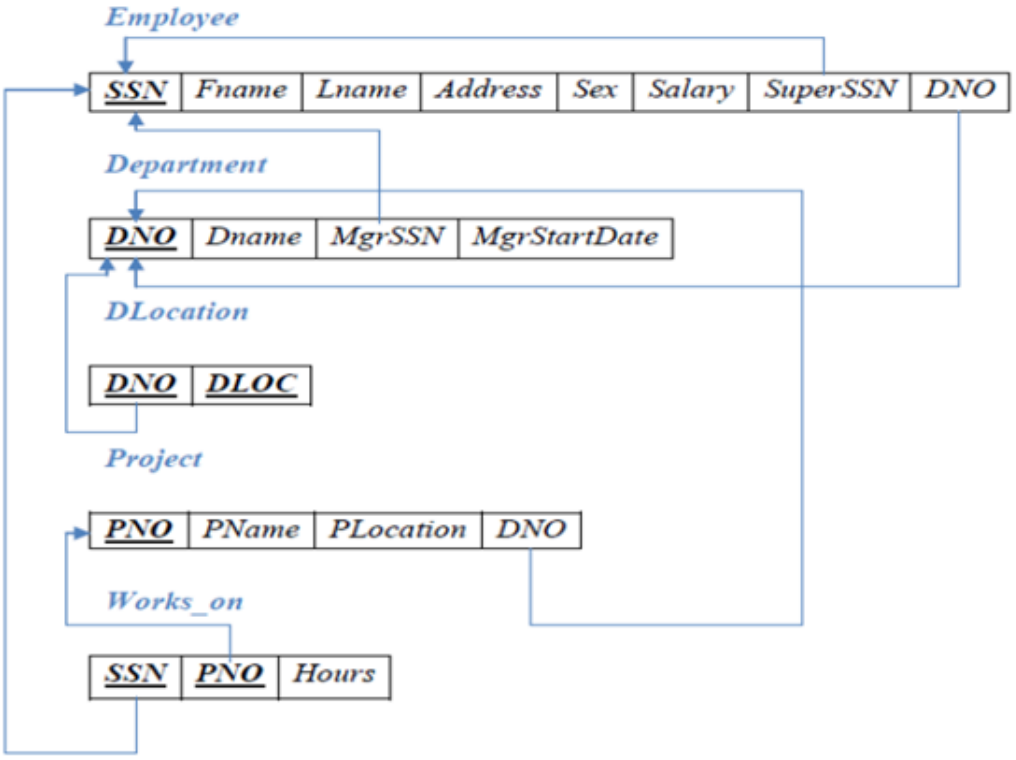
Sajmi Salam
Guest Faculty
Department of Computer Science
CUSAT

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

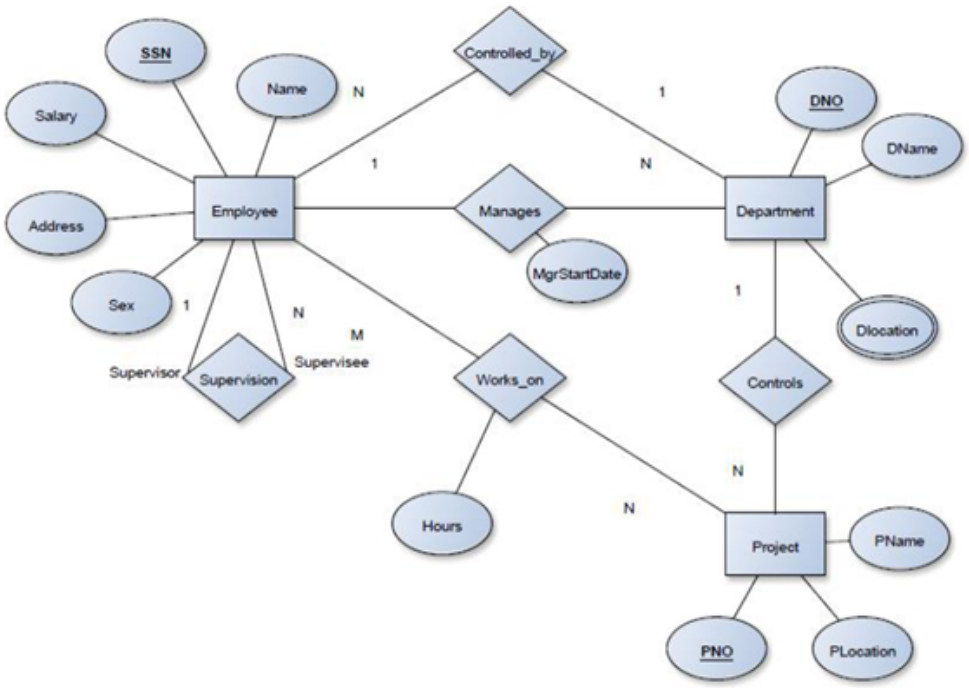
Table of Contents

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.	9
5	Queries to implement Group Functions	10
6	Program to implement Nested Queries	13
7	Program to implement Views	16
8	Programs of Functions And Procedures	18
9	Implementation of Cursor	22
10	Implementation of Trigger	24
11	Queries to implement TCL Commands	27
12	Operations on NoSQL systems	29
13	Simple Structure of GraphQL program	34
14	Programs demonstrating Java Database Connectivity	35
15	Project Report on Application Software	41

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(10) primary key,Name varchar(20)
     ,Address varchar(30),Sex varchar(10),salary integer,
      SuperSSN varchar(10),DNo varchar(10));

create table department(DNo varchar(10) primary key,DName varchar(20),
MgrSSN varchar(10),MgrStartDate date);

create table DLocation(DNo varchar(10),DLoc varchar(20));

create table project(PNo varchar(10) primary key,PName varchar(20),
        PLocation varchar(20),DNo varchar(10));

create table works_On(SSN varchar(10),PNo varchar(10),hours integer);

alter table employee add foreign key (DNo) references department(DNo);
alter table department add foreign key (MgrSSN) references employee(SSN);
```

DATABASE TABLES

Field	Type	Null	Key	Default	Extra
SSN	varchar(10)	NO	PRI	NULL	
Name	varchar(20)	YES		NULL	
Address	varchar(30)	YES		NULL	
Sex	varchar(10)	YES		NULL	
salary	int	YES		NULL	
SuperSSN	varchar(10)	YES		NULL	
DNo	varchar(10)	YES	MUL	NULL	

Field	Type	Null	Key	Default	Extra
DNo	varchar(10)	NO	PRI	NULL	
DName	varchar(20)	YES		NULL	
MgrSSN	varchar(10)	YES	MUL	NULL	
MgrStartDate	date	YES		NULL	

Field	Type	Null	Key	Default	Extra
PNo	varchar(10)	NO	PRI	NULL	
PName	varchar(20)	YES		NULL	
PLocation	varchar(20)	YES		NULL	
DNo	varchar(10)	YES		NULL	

Field	Type	Null	Key	Default	Extra
SSN	varchar(10)	YES		NULL	
PNo	varchar(10)	YES		NULL	
hours	int	YES		NULL	

Field	Type	Null	Key	Default	Extra
DNo	varchar(10)	YES		NULL	
DLoc	varchar(20)	YES		NULL	

Questions : 2

Add another column Age with datatype integer in employee table

QUERY

```
alter table employee add column Age integer;
```

DATABASE TABLES

Field	Type	Null	Key	Default	Extra
SSN	varchar(10)	NO	PRI	NULL	
Name	varchar(20)	YES		NULL	
Address	varchar(30)	YES		NULL	
Sex	varchar(10)	YES		NULL	
salary	int	YES		NULL	
SuperSSN	varchar(10)	YES		NULL	
DNo	varchar(10)	YES	MUL	NULL	
Age	int	YES		NULL	

Questions : 3

Drop a table named Project

QUERY

```
DROP TABLE project;
```

DATABASE TABLES

19:32:36	desc project	Error Code: 1146. Table 'company.project' doesn't exist

Questions : 4

Truncate a table named WORKS_ON

QUERY

```
truncate table works_On;
```

DATABASE TABLES

Field	Type	Null	Key	Default	Extra
SSN	varchar(10)	YES		NULL	
PNo	varchar(10)	YES		NULL	
hours	int	YES		NULL	

Questions : 5

View the structure of the table Department

QUERY

```
desc department;
```

DATABASE TABLES

Field	Type	Null	Key	Default	Extra
DNo	varchar(10)	NO	PRI	NULL	
DName	varchar(20)	YES		NULL	
MgrSSN	varchar(10)	YES	MUL	NULL	
MgrStartDate	date	YES		NULL	

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

```
insert into employee values
("S101","David","Calicut","Male",100000,"SS10","D1001",36),
("S102","Dongli","Kochi","Male",50000,"SS11","D1002",24),
    ("S103","Vilasini","Kannur","Female",70000,"SS14","D1005",17),
    ("S104","Dasan","Wayanad","Male",1200000,"SS15","D1004",45),
    ("S105","Arundhathi","Kollam","Female",50000,"SS16","D1002",39);

insert into department values
("D1001","Chemistry","S103",'2014-02-28'),
("D1002","Physics","S102",'2014-07-5'),
    ("D1003","Integrated Studies","S104",'2014-10-3'),
    ("D1004","Computer Science","S101",'2017-10-3'),
    ("D1005","Mathematics","S105",'2017-4-1');

insert into DLocation values
("D1001","South Kalamassery"),
("D1001","Edappaly"),
    ("D1002","South Kalamassery"),
("D1004","Pathadi Palam"),
    ("D1002","M.G.Road");

insert into project values
("P1001","Computer Vision","Calicut","D1004"),
("P1002","Quantum Mechanics","Kochi","D1002"),
("P1003","Linear Algebra","Alappuzha","D1005");

insert into works_On values
("S101","P1001",5),
    ("S103","P1002",6),
    ("S105","P1003",4);
```

Questions : 2

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

QUERY

```
select * from employee;
select * from department;
select * from DLocation;
select * from project;
select * from works_On;
```

DATABASE TABLES

SSN	Name	Address	Sex	salary	SuperSSN	DNo	Age
S101	David	Calicut	Male	100000	SS10	D1001	36
S102	Dongli	Kochi	Male	50000	SS11	D1002	24
S103	Vilasini	Kannur	Female	70000	SS14	D1005	17
S104	Dasan	Wayanad	Male	1200000	SS15	D1004	45
S105	Arundhathi	Kollam	Female	50000	SS16	D1002	39
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

DNo	DName	MgrSSN	MgrStartDate
D1001	Chemistry	S103	2014-02-28
D1002	Physics	S102	2014-07-05
D1003	Integrated Studies	S104	2014-10-03
D1004	Computer Science	S101	2017-10-03
D1005	Mathematics	S105	2017-04-01
NULL	NULL	NULL	NULL

DNo	DLoc
D1001	South Kalamassery
D1001	Edappaly
D1002	South Kalamassery
D1004	Pathadi Palam
D1002	M.G.Road

PNo	PName	PLocation	DNo
P1001	Computer Vision	Calicut	D1004
P1002	Quantum Mechanics	Kochi	D1002
P1003	Linear Algebra	Alappuzha	D1005
NULL	NULL	NULL	NULL

SSN	PNo	hours
S101	P1001	5
S103	P1002	6
S105	P1003	4

Questions : 3

Modify the salary of the employee as 25000 whose SSN is s101

QUERY

```
update employee set salary=25000 where SSN="S101";
```

DATABASE TABLES

SSN	Name	Address	Sex	salary	SuperSSN	DNo	Age
S101	David	Calicut	Male	25000	SS10	D1001	36
S102	Dongli	Kochi	Male	50000	SS11	D1002	24
S103	Vilasini	Kannur	Female	70000	SS14	D1005	17
S104	Dasan	Wayanad	Male	1200000	SS15	D1004	45
S105	Arundhathi	Kollam	Female	50000	SS16	D1002	39
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Questions : 4

Delete the details of the employee whose SSN is ”s102”

QUERY

```
delete from employee where SSN="S102";o
```

DATABASE TABLES

SSN	Name	Address	Sex	salary	SuperSSN	DNo	Age
S101	David	Calicut	Male	25000	SS10	D1001	36
S103	Vilasini	Kannur	Female	70000	SS14	D1005	17
S104	Dasan	Wayanad	Male	1200000	SS15	D1004	45
S105	Arundhathi	Kollam	Female	50000	SS16	D1002	39
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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 'hlo';
GRANT SELECT ON employee TO user1@'localhost';
```

DATABASE TABLES

Grants for user1@localhost
GRANT USAGE ON *.* TO `user1`@`localhost`
GRANT SELECT ON `company`.`employee` TO `user1`@`localhost`

Questions : 2

Revoking a privilege to all users in a table

QUERY

```
REVOKE SELECT on employee from user1@'localhost';
```

DATABASE TABLES

Grants for user1@localhost
GRANT USAGE ON *.* TO `user1`@`localhost`

GROUP FUNCTION OR AGGREGATE FUNCTION

AIM

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

Questions : 1

List the Name 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
Dasan

Questions : 2

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

QUERY

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

DATABASE TABLES

Sex	count(*)
Male	2
Female	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)
60000.0000

Questions : 4

Calculate the sum of salaries of male employees.

QUERY

```
select sum(salary) from employee where Sex="Male";
```

DATABASE TABLES

sum(salary)
1225000

Questions : 5

Display the maximum and minimum salaries of male employees.

QUERY

```
select max(salary),min(salary) from employee where Sex="Male";
```

DATABASE TABLES

max(salary)	min(salary)
1200000	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
S101	David	Calicut	Male	25000	SS10	D1001	36
S105	Arundhathi	Kollam	Female	50000	SS16	D1002	39
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Questions : 7

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

QUERY

```
select Name,salary from employee where salary in (30000,40000,50000);
```

DATABASE TABLES

Name	salary
Arundhathi	50000

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 'Alappuzha'.

QUERY

```
update employee,project set employee.salary=employee.salary*0.25
    where employee.DNo=project.DNo and PLocation="Alappuzha";
```

DATABASE TABLES

SSN	Name	Address	Sex	salary	SuperSSN	DNo	Age
S101	David	Chennai	Male	25000	SS10	D1001	36
S103	Vilasini	Kannur	Female	17500	SS14	D1005	17
S104	Dasan	Wayanad	Male	1200000	SS15	D1004	45
S105	Arundhathi	Kollam	Female	50000	SS16	D1002	39
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Questions : 2

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

QUERY

```
select employee.Name,project.PLocation from project,employee,works_On
    where employee.SSN=works_On.SSN and project.PNo=works_On.PNo and
     works_On.hours>5;
```

DATABASE TABLES

Name	PLocation
Vilasini	Kochi

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
S101	David	Chennai	Male	25000	SS10	D1001	36	S101	P1001	5
S103	Vilasini	Kannur	Female	17500	SS14	D1005	17	S103	P1002	6
S104	Dasan	Wayanad	Male	1200000	SS15	D1004	45	NULL	NULL	NULL
S105	Arundhathi	Kollam	Female	50000	SS16	D1002	39	S105	P1003	4

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
S101	P1001	5	S101	David	Chennai	Male	25000	SS10	D1001	36
S103	P1002	6	S103	Vilasini	Kannur	Female	17500	SS14	D1005	17
NULL	NULL	NULL	S104	Dasan	Wayanad	Male	1200000	SS15	D1004	45
S105	P1003	4	S105	Arundhathi	Kollam	Female	50000	SS16	D1002	39

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
S105	P1003	4	S101	David	Chennai	Male	25000	SS10	D1001	36
S103	P1002	6	S101	David	Chennai	Male	25000	SS10	D1001	36
S101	P1001	5	S101	David	Chennai	Male	25000	SS10	D1001	36
S105	P1003	4	S103	Vilasini	Kannur	Female	17500	SS14	D1005	17
S103	P1002	6	S103	Vilasini	Kannur	Female	17500	SS14	D1005	17
S101	P1001	5	S103	Vilasini	Kannur	Female	17500	SS14	D1005	17
S105	P1003	4	S104	Dasan	Wayanad	Male	1200000	SS15	D1004	45
S103	P1002	6	S104	Dasan	Wayanad	Male	1200000	SS15	D1004	45
S101	P1001	5	S104	Dasan	Wayanad	Male	1200000	SS15	D1004	45
S105	P1003	4	S105	Arundhathi	Kollam	Female	50000	SS16	D1002	39
S103	P1002	6	S105	Arundhathi	Kollam	Female	50000	SS16	D1002	39
S101	P1001	5	S105	Arundhathi	Kollam	Female	50000	SS16	D1002	39

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

SSN	Name	Address	Sex	salary	SuperSSN	DNo	Age
S101	David	Calicut	Male	25000	SS10	D1001	36
S103	Vilasini	Kannur	Female	70000	SS14	D1005	17
S104	Dasan	Wayanad	Male	1200000	SS15	D1004	45
S105	Arundhathi	Kollam	Female	50000	SS16	D1002	39

Questions : 2

Create another view VW_SSN contains SuperSSN and Dno of female employees

QUERY

```
create view VW_SSN as select SuperSSN,DNo from employee where Sex="Female";
```

DATABASE TABLES

SuperSSN	DNo
SS14	D1005
SS16	D1002

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="S101";
```

DATABASE TABLES

SSN	Name	Address	Sex	salary	SuperSSN	DNo	Age
S101	David	Chennai	Male	25000	SS10	D1001	36
S103	Vilasini	Kannur	Female	70000	SS14	D1005	17
S104	Dasan	Wayanad	Male	1200000	SS15	D1004	45
S105	Arundhathi	Kollam	Female	50000	SS16	D1002	39

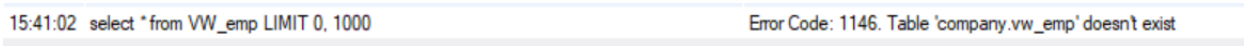
Questions : 4

Delete the view VW_emp

QUERY

```
drop view VW_emp;
```

DATABASE TABLES



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

```
 set serveroutput on
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;
@XEfactorial.sql
```

DATABASE TABLES

```
GET_FACTORIAL(5)

Factorial is 120
```

Questions : 2

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

QUERY

```
set serveroutput on
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;
@XEmax.sql
```

DATABASE TABLES

```
MAXIMUM(4,9)

Maximum is 9
```

Questions : 3

Write a PL/SQL procedure to print the prime

QUERY

```
set serveroutput on
edit@prime.sql

declare
n number;
i number;
temp number;
begin

n := &n;
i := 2;
temp := 1;

    for i in 2..n/2
        loop
```

```
            if mod(n, i) = 0
                then
                temp := 0;
                exit;
            end if;
    end loop;

    if temp = 1
    then
        dbms_output.put_line('Prime');
    else
        dbms_output.put_line('Not Prime');
    end if;
end;
@XEprime.sql
```

DATABASE TABLES

```
Enter value for n: 5
Prime
```

```
Enter value for n: 20
Not Prime
```

Questions : 4

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

QUERY

```
set serveroutput on
edit@numbers.sql
DECLARE
     i INTEGER := 1;
BEGIN
     WHILE i <= 10 LOOP
         DBMS_OUTPUT.PUT_LINE(i);
     i := i+1;
     END LOOP;
END;

@XEnumbers.sql
```

DATABASE TABLES

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

```
declare cursor employee_cur is
 2  select SSN,Salary from Employee where DNO = 'd01'
 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  /
```

DATABASE TABLES

```
mysql> select * from Employee;
+------+----------+--------+---------+--------+--------+----------+------+------+
| SSN  | Fname    | Lname  | Address | Sex    | Salary | SuperSSN | DNO  | Age  |
+------+----------+--------+---------+--------+--------+----------+------+------+
| e001 | Arun     | Kumar  | Kochi   | Male   |  28750 | s1001    | d01  |   25 |
| e002 | Ann      | Susan  | Chennai | Female |  25000 | s1002    | d02  |   23 |
| e003 | Anu      | Priya  | Chennai | Female |  18000 | s1003    | d03  |   23 |
| e004 | Sidharth | Shukla | Pune    | Male   |  17000 | s1004    | d04  |   22 |
| e005 | Ali      | Khan   | Mumbai  | Male   |  30000 | s1005    | d04  |   22 |
+------+----------+--------+---------+--------+--------+----------+------+------+
5 rows in set (0.00 sec)
```

Questions : 2

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

QUERY

```
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 :  d01  :Dname :  Sales
DNO :  d02  :Dname :  Finance
DNO :  d03  :Dname :  Marketing
DNO :  d04  :Dname :  HR
DNO :  d05  :Dname :  Designing
```

Trigger

AIM

Develop and execute a Trigger before and after Update/Delete/Insert operations on a table

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_sal_changes
BEFORE DELETE OR INSERT OR UPDATE ON employee
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;
/

update employeee set salary = 50000 where ssn = 'e1001';
```

Output

```
+------------+
| Old salary |
+------------+
|    34000   |
|    28000   |
|    32000   |
|    20000   |
|    45000   |
```

```
+------------+
```

```
+------------+
| New salary |
+------------+
|    34000   |
|    28000   |
|    32000   |
|    20000   |
|    50000   |
+------------+
```

```
+------------+
| Sal diff |
+----------+
|     0    |
|     0    |
|     0    |
|     0    |
|    5000  |
+----------+
```

Question : 2

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

Query

```
CREATE OR REPLACE TRIGGER display_hour_updates
BEFORE DELETE OR INSERT OR UPDATE ON works_on
FOR EACH ROW
WHEN (NEW.HOURS > 0)
DECLARE
   hour_diff number;
BEGIN
   hour_diff := :NEW.HOURS  - :OLD.HOURS;
   dbms_output.put_line('Old HOURS: ' || :OLD.HOURS);
   dbms_output.put_line('New HOURS: ' || :NEW.HOURS);
```

```
   dbms_output.put_line('HOUR difference: ' || hour_diff);
END;
/

update works_on set hours = 20 where ssn = 'e1001';
```

Output

```
+------------+
|  Old hours |
+------------+
|     10     |
+------------+

+------------+
|  New hours |
+------------+
|     20     |
+------------+

+------------+
|  hour diff |
+------------+
|     10     |
+------------+
```

Transaction Control

AIM

Develop SQL Queries to understand the concept of Transaction Control Language

Question : 1

Creating Check points in the program

Query

```
begin transaction;

savepoint s1;
```

Question : 2

Rollback to a previously created Checkpoint in the program

Query

```
update employee set salary where = 50000 ssn = 'e1001';

select ssn, salary from employee;

rollback to s1;

select ssn, salary from employee;
```

Database Tables

```
+--------+--------+
| ssn    | salary |
+--------+--------+
| 8052A1 |  34000 |
| 8052A2 |  28000 |
| 8052A3 |  32000 |
```

```
| 8052A4 |  20000 |
| e1001  |  50000 |
+--------+--------+

+--------+--------+
| ssn    | salary |
+--------+--------+
| 8052A1 |  34000 |
| 8052A2 |  28000 |
| 8052A3 |  32000 |
| 8052A4 |  20000 |
| e1001  |  45000 |
+--------+--------+
```

Question : 3

Commit the program

Query

```
commit
```

MongoDB

AIM

Develop program to perform operations in MongoDB

Question : 1

Create a database emp

Query

```
use emp;
```

Database Tables

```
switched to db emp
```

Question : 2

Create new Collection

Query

```
db.createCollection("employee")
show collections;
```

Database Tables

```
employee
```

Question : 3

Check the collection list created and drop collection

Query

```
show collections;
db.employee.drop()
```

Database Tables

```
employee
```

Question : 4

Insert document in selected Collection

Query

```
db.employee.insertOne({
    ssn : 'e1001',
    firstname : 'Raj',
    lastname : 'Kiran',
    sex : 'M',
    Address : 'Bangalore',
    salary : 25000,
    department : 'Department of Computer Science'
})

db.employee.find();
```

Database Tables

```
{
```

```
  acknowledged: true,
  insertedId: ObjectId("63a09189b2a462e8c1bcb151")
}

[
  {
    _id: ObjectId("63a09189b2a462e8c1bcb151"),
    ssn: 'e10001',
    firstname: 'Raj',
    lastname: 'Kiran',
    sex: 'M',
    Address: 'Bangalore',
    salary: 25000,
    department: 'Department of Computer Science'
  }
]
```

Question : 5

To get the list documents in Collection

Query

```
db.employee.find();
[
```

Database Tables

```
[
  {
    _id: ObjectId("63a09189b2a462e8c1bcb151"),
    ssn: 'e10001',
    firstname: 'Raj',
    lastname: 'Kiran',
    sex: 'M',
    Address: 'Bangalore',
```

```
    salary: 25000,
    department: 'Department of Computer Science'
  }
]
```

Question : 6

Update the document in Collection

Query

```
db.employee.updateOne({ssn : 'e10001'}, {$set : {salary : 30000}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Database Tables

```
emp> db.employee.find()
[
  {
    _id: ObjectId("63a09189b2a462e8c1bcb151"),
    ssn: 'e10001',
    firstname: 'Raj',
    lastname: 'Kiran',
    sex: 'M',
    Address: 'Bangalore',
    salary: 30000,
    department: 'Department of Computer Science'
  }
]
```

Question : 7

Delete the document in selected Collection

Query

```
db.employee.deleteOne({ssn : 'e10001'})
```

Database Tables

```
{ acknowledged: true, deletedCount: 1 }
```

Question : 8

Projection using find() method

Query

```
db.employee.find({},{firstname : 1, lastname : 1, salary : 1})
[
  {
    _id: ObjectId("63a09363b2a462e8c1bcb152"),
    firstname: 'Deepthy',
    lastname: 'Varyar',
    salary: 35000
  }
]
```

Question : 9

Drop database emp

Query

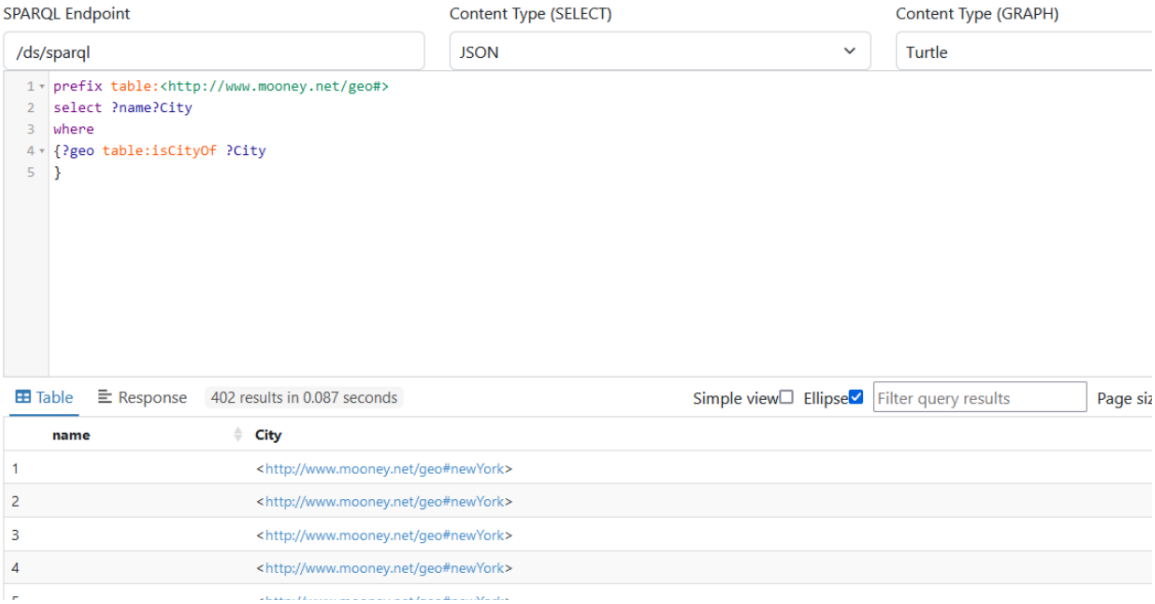
```
db.dropDatabase("emp")
```

GraphQL

AIM

Develop a GraphQL program to perform different operations in created ontology

Output



Database Connectivity

AIM

Develop program to implement Java Database Connectivity

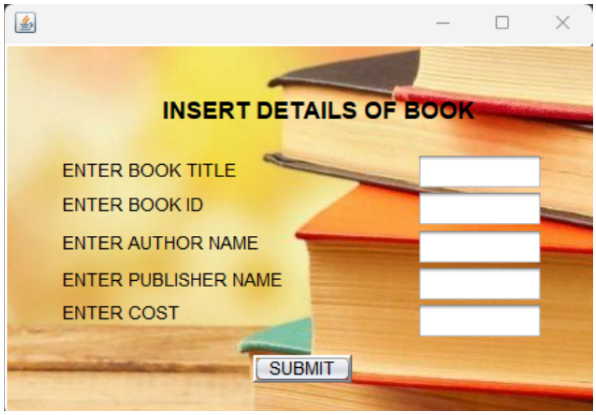
Question : 1

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

Program

```
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 succesfully");
    }
    catch(HeadlessException | NumberFormatException | SQLException e){
        JOptionPane.showMessageDialog(this,e);
    }
}
```

Output



Question : 2

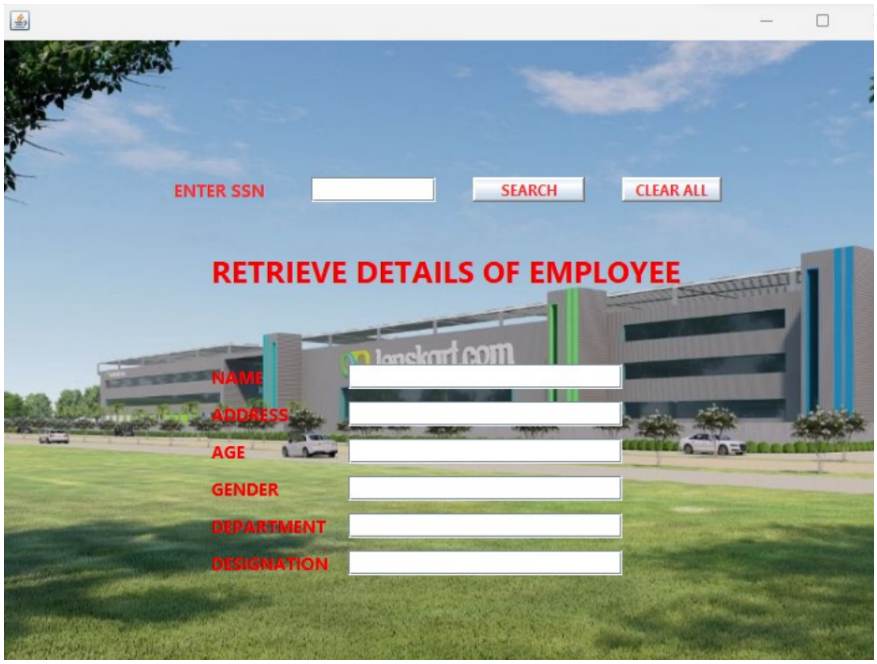
Write a program which connects to an online Employee database and retrieve the details of the employees in the database as per the schema

Program

```
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);
        }
}
```

Output



Question : 3

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

Program

```
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 Succesfully");
        }
        else{
            JOptionPane.showMessageDialog(this,"data Not Found");
        }
    }
    catch(HeadlessException | NumberFormatException | SQLException e){
        JOptionPane.showMessageDialog(this,e);
    }
}
```

Output

Question : 4

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

Program

```
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 succesfully");
        con.close();
        statement.close();
        result.close();
        ps.close();
        }
    }
    catch(HeadlessException | NumberFormatException | SQLException e){
        JOptionPane.showMessageDialog(this,e);
    }
}
```

Output

Project

AIM

Develop an Application Software using Java and Mysql for Information Management purpose.

Project Description

A place to rent Bikes in a Campus. We know people who don't own a bike on their own and also people who have bikes which they only use for a few hours in a day. This helps the both parties to enjoy having a bike. The one who don't own one can enjoy a bike by renting it whenever he needs one and the other own can make profits from the bike(s) when he is not using it.This is a better model as the students can have frequent transactions and more renting as it has lower price than the corporate renters as the owners of the bikes are students.

Users And Functionalities

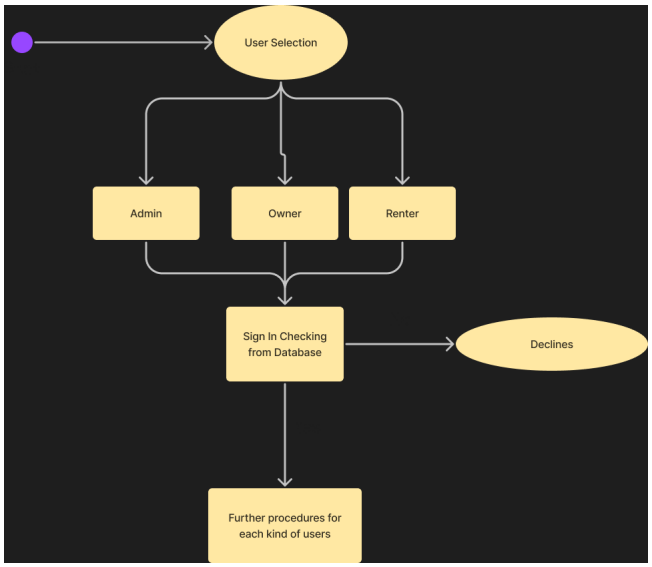
Admin User :-
One who monitor all transactions, owners of bikes and renters. Can check for the owners, the bookings done as far as now, Ongoing to be completed.

Owner :-
Can give their bikes for renting purpose. Can check their previous bookings, Can register more bikes if any, the bookings done as far as now.

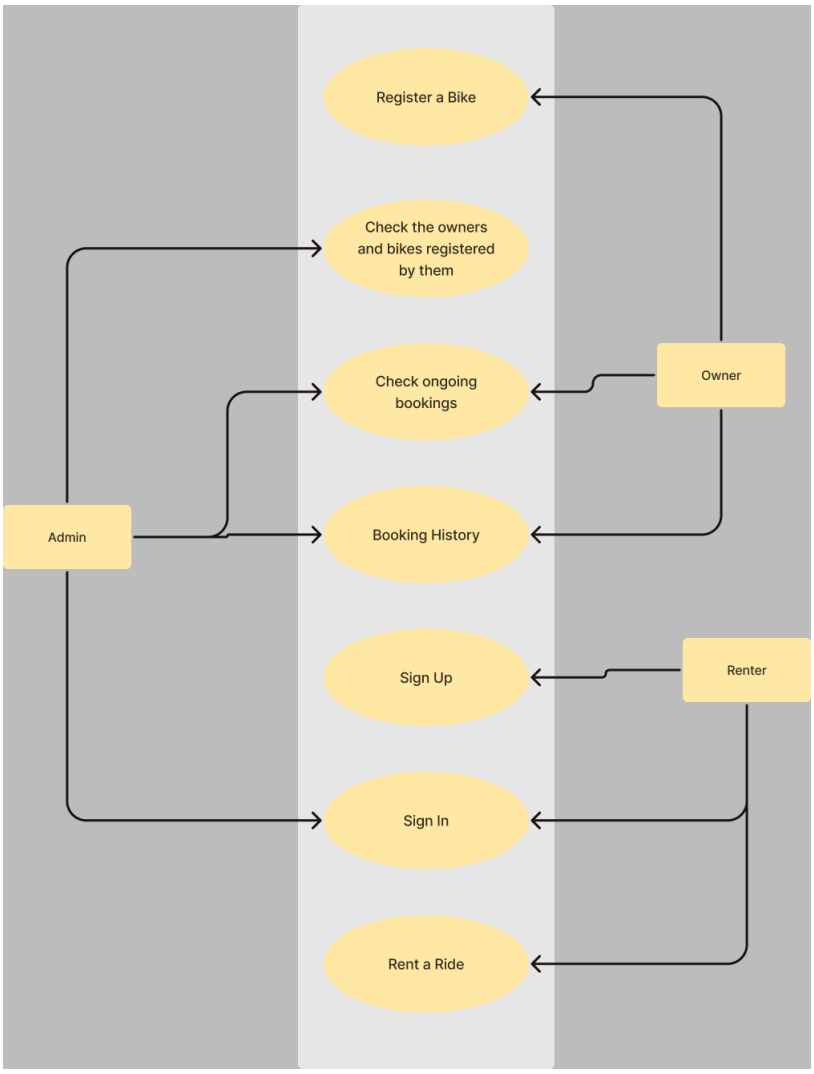
End Users :-
Can rent bikes, can check on their recent rides.

Reference Design

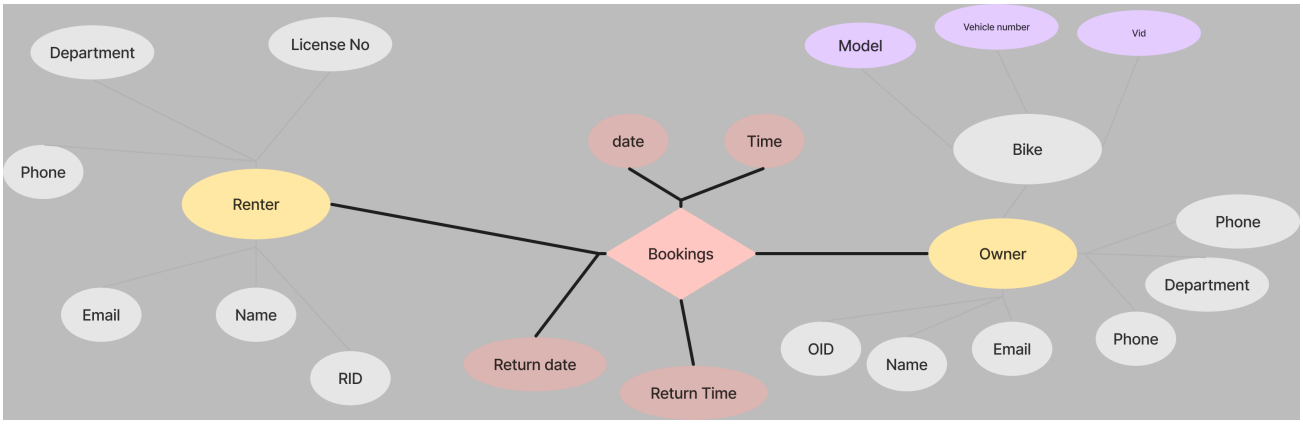
Activity Diagram



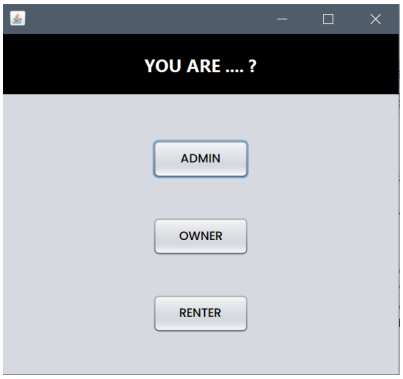
UML Diagram

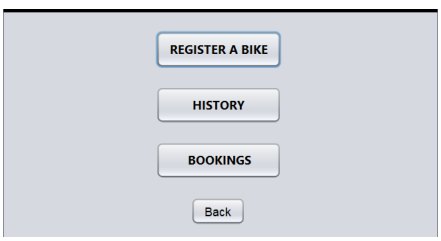


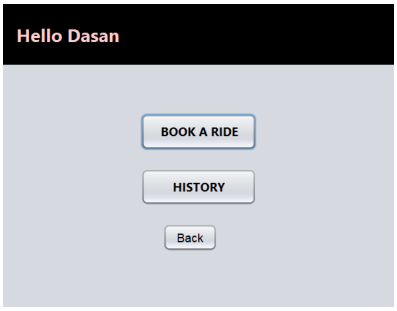
ER Diagram



GUI



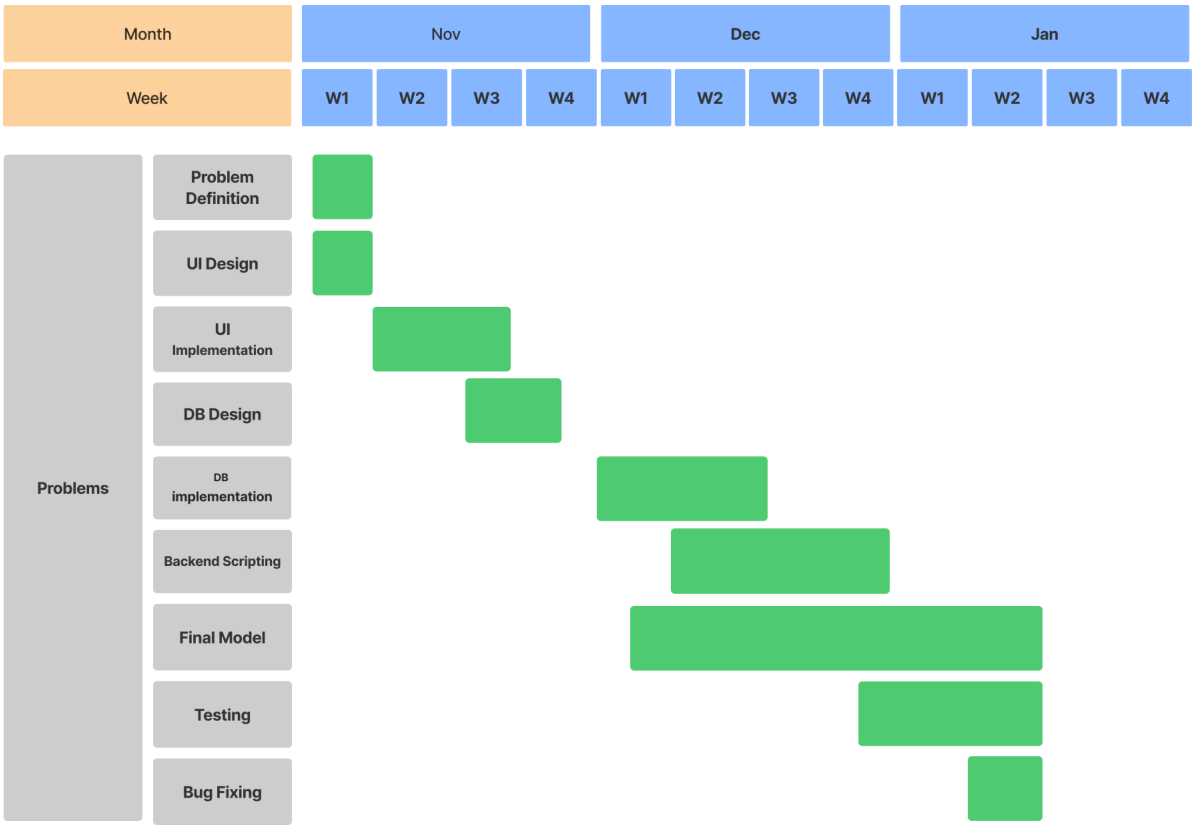




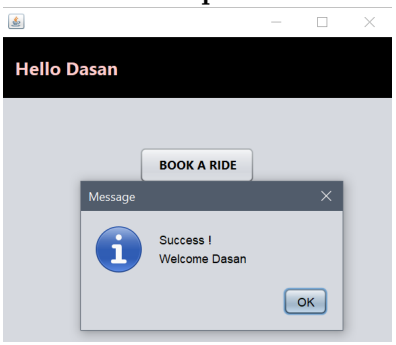
Software Tools

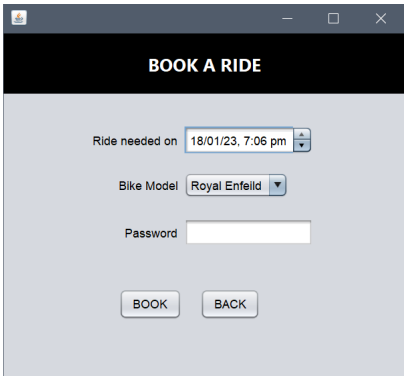
MySQL
JAVA Development Kit(JDK)
Integrated Development Kit(IDE): NetBeans
MySQL Connector/J

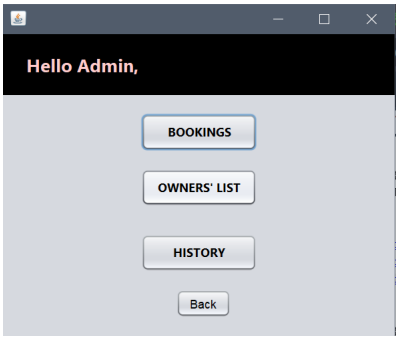
Implementation

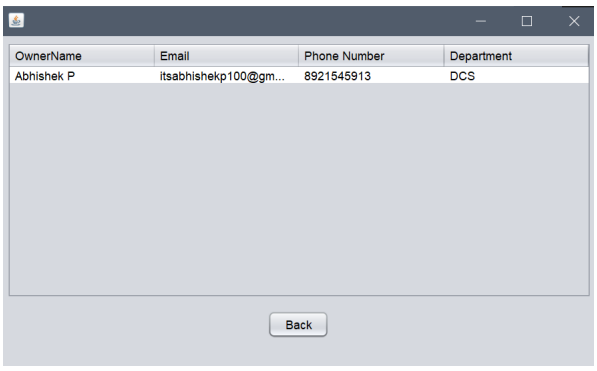


Result And Output









Critical Evaluation

Login

No	Input	Expected Outcome	Actual Outcome	Success/Failure
1	username=correct password=correct	Able to login	Able to login	Success
2	username=incorrect password=correct	Unable to login	Unable to login	Success
3	username=null password=null	Show a dailogue box telling the input field is empty	Show a dailogue box telling the input field is empty	Success

Book a ride

No	Input	Expected Outcome	Actual Outcome	Success/Failure
1	Empty Input fields	Asks the user to give entries	Asks the user to give entries	Success
2	Incorrect password given by the user	Unable to book the ride	Unable to book the ride	Success
3	Bike model combo Box	Display bike models only from the Bike table	Display bike models only from the Bike table	Success

Sign Up

No	Input	Expected Outcome	Actual Outcome	Success/Failure
1	Details are empty	Show a dailogue box telling the input field is empty	Show a dailogue box telling the input field is empty	Success
2	confirmPassword not equal to password	Show a dailogue box telling the confirmPassword not equal to password	Show a dailogue box telling the confirmPassword not equal to password	Success
3	Details filled and ConfirmPassword and password are same	Able to login	Able to login	Success

Conclusion

The Rental Management System is a Application Software that records the details of bikes,owners and renters. It keeps a record of renters and owners of the bikes in rental system. It helps the owners to generate a income from the bike which will be laying in the garage. Also helps the renters to use bikes without paying a high amount to buy one.

References

MySQL
JAVA Database Connectivity
JAVA Swing