



Database Management System (CSE 304)

**HOUSEBOAT MANAGEMENT SYSTEM PROJECT**

---

**Kandasani Mamatha — AP20110010595**

# CONTENT OF THE PROJECT

1. Aim of the project.
2. Project Background.
3. Description of the Project.
4. ER Diagram Creation.
5. Description of ER diagram.
6. Conversion of ER diagram into Tables.
7. Description of Tables.
8. Normalization of tables up to 3-NF.
9. Creation of Data in the tables.
10. Few sql queries on the created tables.
11. Creation of 5 views using the tables.

## 1. Aim of the project.

A Houseboat agency keeps track of houseboats, its owners and the customers who rented it. It allows the owner of the company to efficiently monitor the booking history, accounts, check-in, checkout status of the boats. A customer can rent many houseboats and a houseboat can be rented by many customers on different dates. It is best suitable for managing your resources in a very simple and effective manner with minimal efforts and time so that it increases revenue. **Our aim is to design a database**

**system for the Agency to maintain all the data conveniently and efficiently.**

## 2. Project Background.

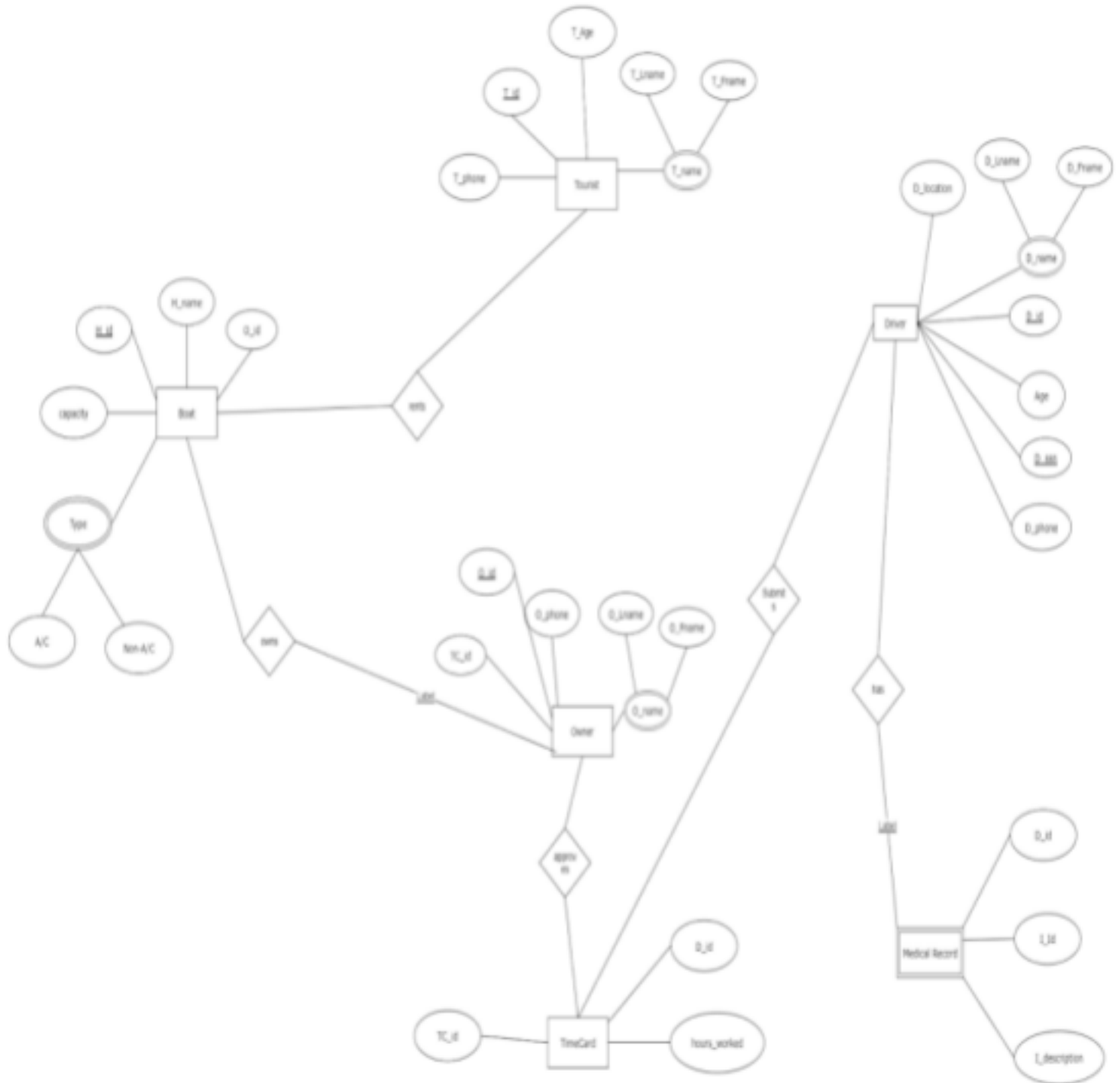
In this project, we explained how houseboat management works, described our project and we also created an ER diagram that represents how actually all the things in houseboat management are connected to each other and some description about that. Here we have converted the ER diagram that we have created into tables and described the tables. After conversion of ER diagram into tables, we have done normalization on those tables up to 3-NF for 4 tables. Coming to the SQL part, we have created tables that came after normalization and inserted some values into the tables. For those tables, we created some SQL queries and few views using those tables.

## 3. Description of the Project.

In the Houseboat agency, there are two scenarios of Tourist and Owner. A Tourist provides his details such as his Name, Age and his Contact Number at the time of booking. A Tourist is uniquely identified by a Tourist\_id. A Tourist can book a single boat per booking. She/He can make any number of bookings. A Houseboat is uniquely identified by Houseboat number. Every Houseboat offers some set of services such as AC, Cuisine, Wifi etc based on its type. Every Houseboat is owned by an owner.

An Owner can have any number of boats. Owner details and his id is recorded. A client is allowed to choose a single route for a specific booking. A route is uniquely identified by a route\_id. A booking is uniquely identified by tourist\_id, houseboat\_number. An amount for every booking is based on the type of boat and number of days of booking.

## 4. ER Diagram Creation.



ER diagram for **HOUSEBOAT MANAGEMENT SYSTEM**

## 5. Description of ER diagram.

ERD-> Entity relationship diagram

Represents relation between entities in database.

ER diagram on Houseboat management system.

Main entities are - Boat,owner,driver,medical card,time ,tourist,time card.

## 6. Conversion of ER diagram into Tables.

Tourist(T\_id, TFirst\_name, TLast\_name, Age,phone,T\_location)

Driver(D\_id, DFirst\_name, DLast\_name, D\_salary, D\_age,Dlocation,D\_phone)

Medical\_Record(D\_id, I\_id, I\_description)

Time Card(TC\_id, Hours\_worked, D\_id)

Owner(O\_id, OFirst\_name, OLast\_name, O\_phone,O\_phone, TC\_id)

Boat(capacity,H\_id ,H\_name,O\_id,Type,price,noofrooms)

where:

Primary Keys are represented by : \_\_\_\_ and

Foreign Keys are represented by : grey

## 7. Description of Tables.

The table tourist contains the details of the tourist - name, age and phone number.

The entities BOAT and tourist form an aggregate which results in the formation of relation rents, which will be having the primary keys of both tourist and Boat entities. If the driver is unavailable, tourists won't be able to rent a houseboat.

The driver entity contains details of the driver such as his id and age. It also contains his medical records, which is a separate entity and is treated as a weak entity.

- The Boat entity can be differentiated on the basis of type as A/C and non-A/C. •
- The time card entity will have details related to the driver, such as his id and the number of hours worked by him. Each driver will have his time card which has to be approved by the respective owner.

The relation between entities:

- One to one relationship: Time card and Driver, Driver and Medical records.
- One to many relationships: Owner and Boat, owner and Time card.
- Many to many relationships: Tourist and Boat.

We used aggregation between Boat and Tourist entities.

- Total participation is included through the Medical records entity.
- Added new entities such as Medical records, Time cards . Medical records are a weak entity.

## 8. Normalization of tables up to 3-NF.

### 8.1 Normalization:

It is used to make the tables into smaller relations by breaking the bad relations. To reduce the redundancy and minimize the insertion, deletion and update anomalies. Normal form is a framework for analyzing relation schemas based on their keys and the FDs among their Attributes.

In the taken houseboating data base in the relational scheme there are violations of normal form. There are violations in the tables Tourist, Driver, Boat, Owner.

### 8.2 Normalization for Tourist table:

In the Tourist table the FD's and the violations of the normal forms.

TOURIST(T\_ID, T\_FIRSTNAME, T\_LASTNAME, T\_AGE, T\_PHONE, T\_LOCATION)

As it violates the 1NF as it has the multivalued attributes for the name section .

We change them into the T\_name

FD's are : FD1: {T\_ID} → {T\_name, T\_AGE, T\_phone, T\_location}

FD2: {T\_phone} → {T\_Location}

The taken FD's violates the 3 NF as where it has transitive relation in the FD2 we should remove it

After the 3NF:

Tourist 1 : (T\_ID, T\_NAME, T\_AGE, T\_PHONE)

Tourist 2: (T\_Phone, T\_location)

### 8.3 Normalization for Driver table:

in the Driver table the FD's and the violations of the normal forms

Driver(D\_ID, D\_SSN, D\_FIRSTNAME, D\_LASTNAME, D\_SALARY, D\_AGE, D\_PHONE, D\_LOCATION)

As it violates the 1NF as it has the multivalued attributes for the name section

We change them into the D\_name after the first normal form

FD's are : FD1: {D\_ID,D\_SSN} → {D\_name,D\_salary,D\_age,D\_phone,D\_location}

FD 2: {D\_ID} → {D\_name}

FD 3 : {D\_SSN} → {D\_salary}

FD 4: {D\_phone} → {D\_Location}

The taken Fd's violates the 2 Nf from FD2 and FD3 as it has the partial dependence

After the 2NF

(D\_ID,D\_SSN,D\_AGE,D\_PHONE,D\_LOCATION)

(D\_ID,D\_name)

(D\_SSN, D\_salary)

The taken FD's violates the 3 NF as where it has transitive relation in the FD2 we should remove it

After the 3NF:

Driver 1 :(D\_ID,D\_SSN,D\_AGE,D\_PHONE)

Driver 2 : (D\_ID, D\_name)

Driver 3: (D\_SSN, D\_salary)

Driver 4:(D\_Phone,D\_location)

## 8.4 Normalization for Boat Table:

In the **Boat table** the FD's and the violations of the normal forms

Boat(H\_ID,O\_id,H\_name,Capacity,Type ,Noofbeds,price )



As it violates the 1NF as it has the multivalued attributes for the ac,non/ac section

We change them into the Type after the first normal form

FD's are : FD1: {H\_ID, O\_id} → {Capacity, Type ,Noofbeds )

FD 2: {H\_ID} → {H\_name}

FD 3 :{TYPE} → {price}

The taken Fd's violates the 2 Nf from FD2 as it has the partial dependence

After the 2NF

(H\_ID,O\_id,Type ,Noofbeds,price,capacity)

(H\_ID,H\_name)

The taken FD's violates the 3 NF as where it has transitive relation in the FD2 we should remove it

After the 3NF:

Boat :(H\_ID,O\_id,Capacity,Type,No.of.beds)

Boat 1 : (H\_ID,H\_name)

Boat 2: (TYPE ,Price)

### **8.5 Normalization for Owner table:**

In the Owner table the FD's and the violations of the normal forms

Owner(O\_ID,O\_FIRSTNAME,O\_LASTNAME,O\_PHONE,TC\_id)

As it violates the 1NF as it has the multivalued attributes of the first name,lastname section

We change them into the O\_name after the first normal form

FD's are : FD1: {O\_ID} → {O\_name,O\_phone,TC\_id)

FD 2: {O\_name} → {TC\_id}

The taken FD's violates the 3 NF as where it has transitive relation in the FD2 we should remove it

After the 3NF:

Owner 1 :(O\_ID,O\_name,D\_PHONE)

Owner 2 : (O\_name,TC\_id)

## 8.6 Final Tables after Normalization are

- Owner 1
- Owner 2
- Boat
- Boat 1
- Boat 2
- Driver 1
- Driver 2
- Driver 3
- Driver 4
- Tourist 1
- Tourist 2

## 9. Creation of Data in the tables.

### 9.1 creation of tables in mysql:

#### SQL Query for owner1 table creation

```
create table owner1(O_id varchar(100),O_name varchar(100),O_phone numeric(20,2),primary key(O_id));
```

#### SQL Query for owner2 table creation

```
create table owner2(O_name varchar(100),TC_id varchar(102), foreign key (TC_id )REFERENCES time_card (TC_id));
```

#### **SQL Query for boat table creation**

```
create table boat(H_id varchar(70),O_id varchar(70),Capacity numeric(30),Type  
varchar(100),Noofbed numeric(30), primary key(H_id),foreign key(O_id) references  
owner1 (O_id));
```

#### **SQL Query for boat1 table creation**

```
create table boat1(Type varchar(100),price numeric(30,5));
```

#### **SQL Query for boat2 table creation**

```
create table boat2(H_id varchar(70),H_name varchar(100) ,foreign key (H_id)  
REFERENCES boat(H_id));
```

#### **SQL Query for driver1 table creation**

```
create table driver1(D_id varchar(100),D_ssn numeric(20,2),D_age  
numeric(20,2),D_phone varchar(100),primary key(D_id,D_ssn));
```

#### **SQL Query for driver2 table creation**

```
create table driver2(D_phone varchar(100),D_location varchar(100));
```

#### **SQL Query for driver3 table creation**

```
create table driver3(D_id varchar(100),D_name varchar(100),foreign key(D_id)  
references driver1(D_id));
```

#### **SQL Query for driver4 table creation**

```
create table driver4(D_salary numeric(20,2),D_ssn numeric(20,2));
```

#### **SQL Query for tourist1 table creation**

```
create table tourist1(T_id varchar(100),T_name varchar(100),T_age
```

numeric(20,2),T\_phone varchar(100),primary key(T\_id));

### **SQL Query for tourist2 table creation**

create table tourist2(T\_phone varchar(100),T\_location varchar(100) );

## **9.2 insertion of data into tables:**

### **SQL Query for owner1 table insertion**

insert into owner1 values('o1','pullaya',789456123), ('o2','rangaya',123456789),  
('o3','brahmendra',456123789), ('o4','yellaya',741852963),  
('o5','sahayam',147852963);

### **SQL Query for owner2 table insertion**

insert into owner2 values('pullaya','TC-101'),('rangaya','TC-101'),('mamatha','TC-103'),  
('nehal','TC-104'), ('brahmi','TC-105');

### **SQL Query for boat table insertion**

insert into boat values('H-201','o1',120,'non/ac',40), ('H-206','o1',60,'ac',5),  
('H-203','o2',80,'non/ac',10), ('H-205','o3',10,'non/ac',3),  
('H-204','o4',12,'ac',4), ('H-202','o1',120,'ac',70);

### **SQL Query for boat1 table insertion**

insert into boat1 values('AC',1500),('nonac',650),('AC',6050),('nonac',1250),('AC',1650);

### **SQL Query for boat2 table insertion**

insert into boat2 values('H-201','jatayu'), ('H-202','varaha'), ('H-203','garuda'),  
('H-201','vennela'), ('H-204','radham');

### **SQL Query for driver1 table insertion**

insert into driver1

```
values('d1',782,23,'+91293382631'),('d2',712,33,'+91293382632'),('d3',7122,24,'+91293382231'),('d4',982,28,'+913232382631'),('d5',7123,23,'+9129335182631');
```

#### **SQL Query for driver2 table insertion**

```
insert into driver2 values('+91293382631','amalapuram'),  
('+91293382632','athreypuram'), ('+91293382231','nouru'),('+913232382631','karachi'),  
('+9129335182631','california');
```

#### **SQL Query for driver3 table insertion**

```
insert into driver3
```

```
values('d5','raju'),('d3','rani'),('d1','king'),('d2','queen'),('d5','raja');
```

#### **driver4 table insertion**

```
insert into driver4 values(787,1250.5),(7855,12710.5),(7817,12050.5),  
(7287,12250.5),(1787,19250.5),(782,78952);
```

#### **SQL Query for tourist1 table insertion**

```
insert into tourist1 values('T1','mammu',50,'+9182134631'),  
('T2','mamu',30,'+9182134633'),  
('T3','qwerty',10,'+91822134631'),  
('T4','masd',12,'+9183214631'),  
('T5','mqammu',21,'+91812214631');
```

#### **SQL Query for tourist2 table insertion**

```
insert into tourist2 values('+9182134631','amaravathi'),  
('+9182134531','guntur'),  
('+9182135631','addanki'),  
('+9182174631','gudivada'),  
('+182134631','florida');
```

## 10. Few sql queries on the created tables

(1) Display the names of table driver3?

Sql Query

select d\_name from driver3 order by D\_name;

OUTPUT

```
mysql> select d_name from driver3 order by D_name;
+-----+
| d_name |
+-----+
| king   |
| queen  |
| raja   |
| raju   |
| rani   |
+-----+
5 rows in set (0.00 sec)
```

(2) Display Location of driver whose age is 23?

Sql Query

select d2.d\_LOCATION from driver2 d2,driver1 d1 where d1.D\_phone = d2.d\_phone and d1.d\_age=23;

OUTPUT

```
mysql> select d2.d_LOCATION from driver2 d2,driver1 d1 where d1.D_phone = d2.d_phone and d1.d_age=23;
+-----+
| d_LOCATION |
+-----+
| amalapuram |
| california |
+-----+
2 rows in set (0.00 sec)
```

**(3) Display the maximum salary and d\_id of the driver4?**

**Sql Query**

```
select max(d4.d_salary),d1.d_id from driver4 d4,driver1 d1 where d1.d_ssn=d4.d_ssn;
```

**OUTPUT**

```
mysql> select max(d4.d_salary),d1.d_id from driver4 d4,driver1 d1 where d1.d_ssn=d4.d_ssn;
+-----+-----+
| max(d4.d_salary) | d_id |
+-----+-----+
|          78952.00 | d1    |
+-----+-----+
1 row in set (0.00 sec)
```

**(4) Display the average salary of the drivers?**

**Sql Query**

```
select avg(D_salary) from driver4;
```

**OUTPUT**

```
mysql> select avg(D_salary) from driver4;
+-----+
| avg(D_salary) |
+-----+
|  22764.925000 |
+-----+
```

**(5) Display all entries from driver4**

**Sql Query**

select \* from driver4;

**OUTPUT**

```
mysql> select *from driver4;
+-----+-----+
| D_salary | D_ssn |
+-----+-----+
| 787.00 | 1250.50 |
| 7855.00 | 12710.50 |
| 7817.00 | 12050.50 |
| 7287.00 | 12250.50 |
| 1787.00 | 19250.50 |
| 782.00 | 78952.00 |
+-----+-----+
6 rows in set (0.00 sec)
```

**(6) Display the updated salary of driver4 if 10% increment is done and ssn is equal to 787?**

**Sql query**

UPDATE driver4 SET d\_Salary = d\_Salary + (d\_Salary \* 10/100) where d\_ssn =787;

**OUTPUT**

```
+-----+-----+
| D_ssn | D_salary |
+-----+-----+
| 787.00 | 1375.55 |
| 7855.00 | 12710.50 |
| 7817.00 | 12050.50 |
| 7287.00 | 12250.50 |
| 1787.00 | 19250.50 |
| 782.00 | 78952.00 |
+-----+-----+
6 rows in set (0.00 sec)
```



**(7) Display capacity and price of different houseboat?**

**Sql Query**

select b0.capacity,b1.price from boat b0,boat1 b1 where b0.type=b1.type;

**OUTPUT**

```
mysql> select b0.capacity,b1.price from boat b0,boat1 b1 where b0.type=b1.type;
+-----+-----+
| capacity | price |
+-----+-----+
| 120      | 1650.00000 |
| 120      | 6050.00000 |
| 120      | 1500.00000 |
| 12       | 1650.00000 |
| 12       | 6050.00000 |
| 12       | 1500.00000 |
| 60       | 1650.00000 |
| 60       | 6050.00000 |
| 60       | 1500.00000 |
+-----+-----+
9 rows in set (0.00 sec)
```

**(8) Display all the entries in tourist1 whose name contains 'amu' as sub string**

**Sql Query**

select \* from tourist1 where T\_name like '%amu%';

**OUTPUT**

```
mysql> select * from tourist1 where T_name like '%amu%';
+-----+-----+-----+-----+
| T_id | T_name | T_age | T_phone |
+-----+-----+-----+-----+
| T2   | mamu   | 30.00 | +9182134633 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

**(9) Display the contents in an order of price from boat**

**Sql Query**

select \* from boat1 group by price order by price;

**OUTPUT**

```
mysql> select * from boat1 group by price order by price;
+-----+-----+
| Type | price |
+-----+-----+
| nonac | 650.00000 |
| nonac | 1250.00000 |
| AC    | 1500.00000 |
| AC    | 1650.00000 |
| AC    | 6050.00000 |
+-----+-----+
5 rows in set (0.00 sec)
```

## **11. Creation of 5 views using the tables**

**(1) Create a view 'newdriver' with id,age and name, then display the drivers whose age is less than 24**

**Sql Query**

**Creating the view "newdriver"**

create view newdriver as select d1.d\_id,d1.d\_age,d\_name from driver1 d1,driver3;

select distinct (D\_id) from newdriver where d\_age<24;

**OUTPUT**

```
mysql> select distinct (D_id) from newdriver where d_age<24;
+-----+
| d_id |
+-----+
| d5   |
| d1   |
+-----+
2 rows in set (0.00 sec)
```

**(2) Create a view 'room' and display id of the boad,typeand number of beds with capacity greater than 50**

#### Sql Query

create view room AS select h\_id,type,noofbed from boat where capacity >

50; select \* from room;

#### OUTPUT

```
mysql> select * from room;
+-----+-----+-----+
| h_id | type  | noofbed |
+-----+-----+-----+
| H-201 | non/ac | 40      |
| H-202 | ac     | 70      |
| H-203 | non/ac | 10      |
| H-206 | ac     | 5       |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

**(3) create view 'updatedtimecard' and select id and number of hours worked should be equal to 23.5**

create view updatedTimecard AS select TC\_id,hours\_worked as hours from time\_card where hours\_worked = 23.5;

select \* from updatedTimecard;

#### OUTPUT

```
mysql> select * from updatedTimecard;
+-----+-----+
| TC_id  | hours |
+-----+-----+
| TC-101 | 23.50 |
| TC-103 | 23.50 |
+-----+-----+
2 rows in set (0.00 sec)
```

(4) create a view of 'hospitalstaff' and display the id and medical condition of the driver.

#### Sql Query

create view hospitalstaff AS select I\_id,I\_description from medical;

select \* from hospitalstaff;

#### OUTPUT

```
mysql> select * from hospitalstaff;
+-----+-----+
| I_id  | I_description |
+-----+-----+
| 124.00 | good          |
| 124.00 | good          |
| 224.00 | should check monthly |
| 15.00  | bad           |
| 123.00 | good          |
+-----+-----+
5 rows in set (0.00 sec)
```

(5) Create view tourister

create view tourister AS select t\_name,T\_age,T\_phone from

tourist1; **Sql Query**

select \* from tourister;

```
mysql> select * from tourister;
+-----+-----+-----+
| t_name | T_age | T_phone      |
+-----+-----+-----+
| mammu  | 50.00 | +9182134631  |
| mamu   | 30.00 | +9182134633  |
| qwerty | 10.00 | +91822134631 |
| masd   | 12.00 | +9183214631  |
| mqammu | 21.00 | +91812214631 |
+-----+-----+-----+
5 rows in set (0.00 sec)
```