ABSTRACT

A cab or a taxi is a vehicle for hire with a driver, used by a single passenger or small group of passengers, often for a non-shared ride. Countries where public transportation is still in development phase, cabs play an important role. Usually a customer hires a cab by street hailing. In such countries the taxi drivers usually rely on their luck to increase their profits which usually includes first come first serve basis. These methods usually lead to problems like traffic congestion, long passenger waiting time, safety issues, and low cab utilization. With the introduction of online cab system, it became very easy for the drivers to find their customers and it also became so simple for the passengers to travel around the city using a cost-effective method.

The main aim of cab management project is to rent a cab and get payments from respective clients. We aim to demonstrate the use of create, read, update and delete operations through this project. The project starts by adding a cab and details of its driver .The owner provides cab to the drivers and ads their expenses on daily basis. Booking scene is where customers can book a taxi to get to the desired location. On booking, the passenger receives information like booking date & time , pick up & drop locations, and details of the cab & the driver to the registered mobile number and email address. At the end of the ride a bill will generated with the total distance traveled, time taken and total amount to be pain. The passenger can choose from any of the available payment methods .

TABLE OF CONTENTS

CHAPTER 1

Introduction

Software that handles the storage, retrieval and updating of data in a computer system is called Database Management System. A good example for this is Cab Management System. Cab Management System is a software system where the management of entire cab booking is computerized. The customers can even book their cab using an application. This Project is developed for smooth running and management of cab booking. It is based on new techniques and new ideas. This software is supported to eliminate and, in some cases, reduce the hardships faced without proper management of the system.

This Project is a fine thought to make the complex procedure of the cab management system to an easy manner which is systematic, modular designed, based on the user requirement. The modular design and constructed system is very much user oriented in which user can easily understand the tools and can do edit of his own choice. The system is not any tough more and does not possess many applications but it is made by focusing on the maintaining records driver’s actions in a computerized system rather than time taking and cumbersome manual system.

This project is a software application that can be easily handled by minimum educated and simple computer knowledge person without any option of error. The reports can be viewed completely and the head of the management can review it daily or weekly or even monthly. It will be more useful for company auditing. This proposed system will be interactive, faster and user-friendly for the end users.

CHAPTER 2

Requirement Specification

The hardware and software components of a computer system that are required to install and use software efficiently are specified in the Software Requirements Specification. The minimum system requirements need to be met for the program to run at all times on the system.

2.1 Hardware Requirements:

The hardware requirements specify the necessary hardware which provides us the platform to implement our programs.

- Processor: Intel dual core i5 or above
- 2gb RAM (System memory)
- 20gb of hard drive space

And necessary computer peripherals such as keyboard, monitor etc.

2.2 Software Requirements:

The software requirements specify the pre-installed software needed to run the code being implemented in this project.

- Windows Operating System 7 or above
- Oracle 10g express edition for SQL

CHAPTER 3

Project Description

3.1 SQL

SQL is a domain specific programming language designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDBMS). It is particularly useful in handling structured data, i.e. data incorporating relations among entities and variables. Originally based upon relational algebra and tuple relational calculus, SQL consists of a data definition language, data manipulation language, and a data control language. The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control. SQL stores each data item in its own fields. In SQL, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data called a record. Each record is made up of a number of fields. No two fields in a record can have the same field name.

3.2 Oracle Database 10g Express Edition

An Oracle Database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management, In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost-effective way to manage information and applications. Enterprise grid computing creates large pools of industry standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed.

3.3 ER Diagram

The Entity Relationship Model (ER model) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) band specifies relationships that can exist between entities (instances of those entity types).

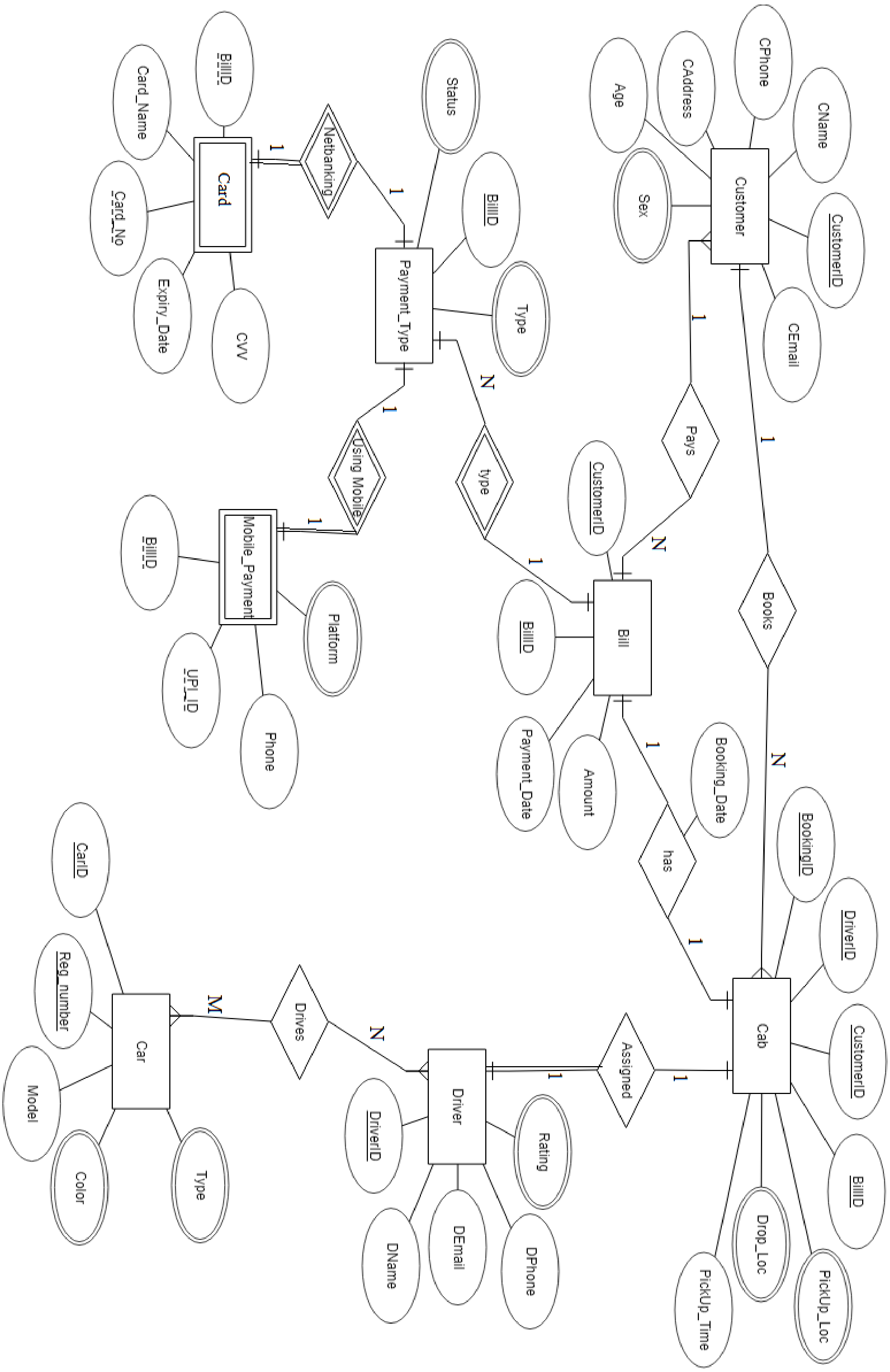
An ER model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. It does not define the business data schema in graphical form as boxes(entities) that are connected by lines (relationships) which express the associations and dependencies between entities. An ER model can also be expressed in a verbal form.

Entities may be characterized not only by relationships, but also by additional properties(attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute relationship diagram rather than Entity-Relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity.

A relationship captures how entities are related to one another. Relationships can be thought of as verbs, linking two or more nouns. Example: An owns relationship between the company and the computer, a supervises relationship between an artist and a song, a proves relationship between a mathematician and a conjecture. Entities and Relationships can both have the attributes. Example: an employee entity might have a Social Security Number (SSN) attribute; the proved relationship may have a date attribute.

The ER Diagram for this database is as shown:


Card
BillID
Card_Name
Card_No
Expiry_Date
CVV
1
Netbanking
1
Status
BillID
Payment_Type
Type
N
1
Using Mobile
type
1
Mobile_Payment
BillID
Platform
UPI_ID
Phone
CPhone
CAddress
Age
CName
Sex
Customer
CustomerID
CEmail
1
Pays
1
N
CustomerID
1
Bill
BillID
Payment_Date
Amount
Books
N
1
has
Booking_Date
BookingID
1
DriverID
Cab
CustomerID
BillID
Drop_Loc
PickUp_Loc
PickUp_Time
Assigned
1
1
Driver
DriverID
Rating
DName
DEmail
DPhone
Drives
N
M
Car
CarID
Reg_number
Model
Color
Type

3.4 Relational Schema

A relational schema is a named relation defined by a set of attributes; The term relational schema refers to a heading paired with a set of constraints defined in terms of that heading, A relation can thus be seen as an instantiation of a relation schema if it has the heading of that schema and it satisfies the applicable constraints.

ENTITY

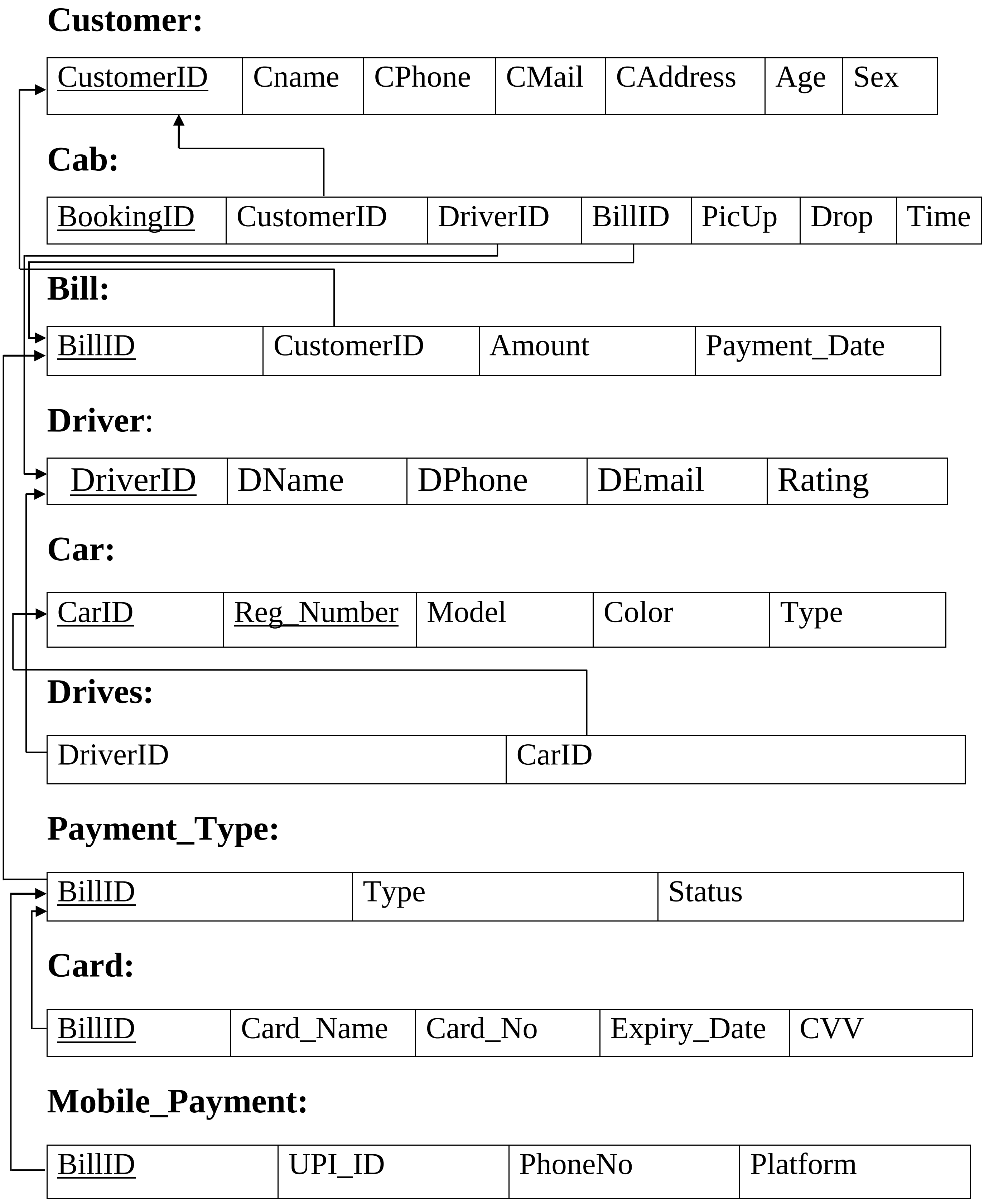
Basic object that the ER model represent is an entity, which is a thing in the real world with an independent existence. An entity may be an object with a physical existence (example: a particular person, car, house, employee) or it may be an object with a conceptual existence (example: a company, a job or a university course).

ATTRIBUTE

Each entity has attributes. It is a particular property that describes entity, For example, an employee entity may be described by the employees name, age, address, salary and job. A particular entity will have a value for each of its attributes. The attribute values that describe each entity become a major part of the data stored in the database.

RELATION

Relation is sometimes used to refer to a table in a relational database but is more commonly used to describe the relationships that can be created between those tables in a relational database. In relational databases, a relationship exists between two tables when one of them has a foreign key that references the primary key of the other table. This single fact allows relational databases to split and store data in different tables, yet still link the disparate data items together. It is one of the features that makes relational databases such powerful and efficient stores of information. The ability to define relationships is so fundamental and so important that this is what differentiates relational databases from other types of databases, such as flat-file databases. Relation may also be known as relationship**.**



Customer:

CustomerID	Cname	CPhone	CMail	CAddress	Age	Sex

Cab:

BookingID	CustomerID	DriverID	BillID	PicUp	Drop	Time

Bill:

BillID	CustomerID	Amount	Payment_Date

Driver:

DriverID	DName	DPhone	DEmail	Rating

Car:

CarID	Reg_Number	Model	Color	Type

Drives:

DriverID	CarID

Payment_Type:

BillID	Type	Status

Card:

BillID	Card_Name	Card_No	Expiry_Date	CVV

Mobile_Payment:

BillID	UPI_ID	PhoneNo	Platform

CHAPTER 4

CODING

4.1 Creation of Tables

```
CREATE TABLE Customer

(

CustomerID int primary key,

Cname varchar2(100),

Cemail varchar2(20),

Cphone varchar2(10),

Caddress varchar2(255)

Sex Char,

Age int

);

CREATE TABLE Cab

(

BookingID int primary key,

CustomerID int references Customer(CustomerID),

DriverID int references Driver(DriverID),

BillID int references Bill(BillID),

PickUp_Loc varchar2(255),

Drop_Loc varchar2(255),

Time varchar2(20)

);
```

```
CREATE TABLE Bill
(
BillID int primary key,
CustomerID int references Customer(CustomerID),
Amount Number,
Payment_Date date
);
CREATE TABLE Driver
(
DriverID int primary key,
Dname varchar2(20),
DEmail varchar2(20),
Dphone varchar2(10),
Rating int
);
CREATE TABLE Car
(
CarID int primary key;
Reg_Number varchar2(10) unique,
Model varchar2(50),
Color varchar2(30),
Type varchar2(20)
);
```

```
CREATE TABLE Payment_Type
(
BillID int primary key ,
type varchar2(20) not null,
status varchar(20) not null
);
```

```
CREATE TABLE Drives
(
DriverID int references Driver (DriverID),
CarID int references Driver (DriverID)
);
```

```
CREATE TABLE card
(
BillID int references Bill ( BillID),
card_no int unique,
Name_on_card varchar2(50),
Expiry Date not null,
CVV int not null
);
```

```
CREATE TABLE Mobile_Payment

 (

BillID int references Payment_Type(BillID),

UPI_ID varchar2(20) unique,

phone varchar2(10),

Platform varchar2(20)

);
```

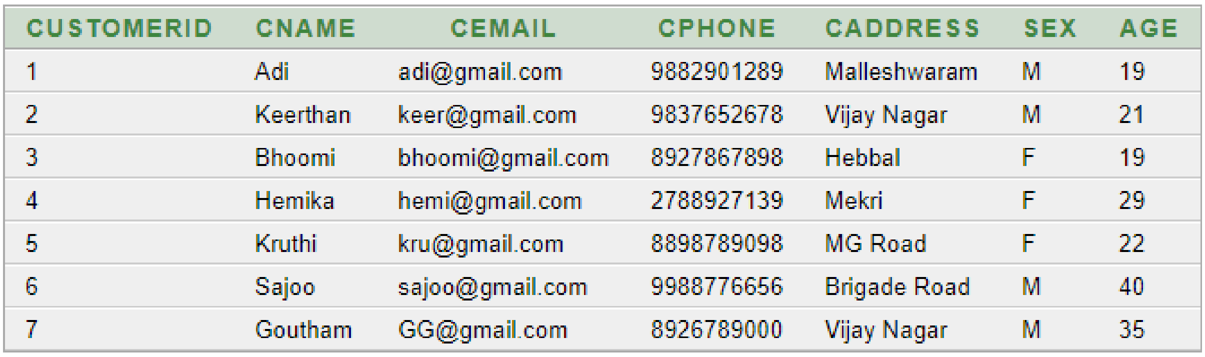
4.2 Insertion of Tuples

```
insert into Customer values(1,'Adi','adi@gmail.com','9882901289','Malleshwaram',’M’,19);

insert into Customer values(2,'Keerthan','keer@gmail.com','9837652678','VijayNagar,’M’, 21);

insert into Customer values(3,Shashi,Shashi@gmail.com','8927867898','Hebbal',’F’,19);

insert into Customer values(4,'Hemika','hemi@gmail.com','2788927139','Mekri',’F’,29);

insert into Customer values(5,'Kruthi','kru@gmail.com','8898789098','MG Road',’F’,22);

insert into Customer values(6,'Sajoo','sajoo@gmail.com','9988776656','BrigadeRoad',’M’, 40);

insert into Customer values (7,'Goutham','GG@gmail.com','8926789000','Vijay Nagar',’M’,35);
```

```
insert into Driver values(10,'Ravi','ravi@gmail.com','9878987890',4);

insert into Driver values(11,'Ramu','ram@gmail.com','892029890',10);

insert into Driver values(12,'Rajeev','raj@gmail.com','2139876678',7);

insert into Driver values(13,'Sam','sam@gmail.com','9878982720',8);

insert into Driver values(14,'Jhon','jhon@gmail.com','9873987890',9);

insert into Driver values(15,'Shiv','shiv@gmail.com','9878987890',3);
```

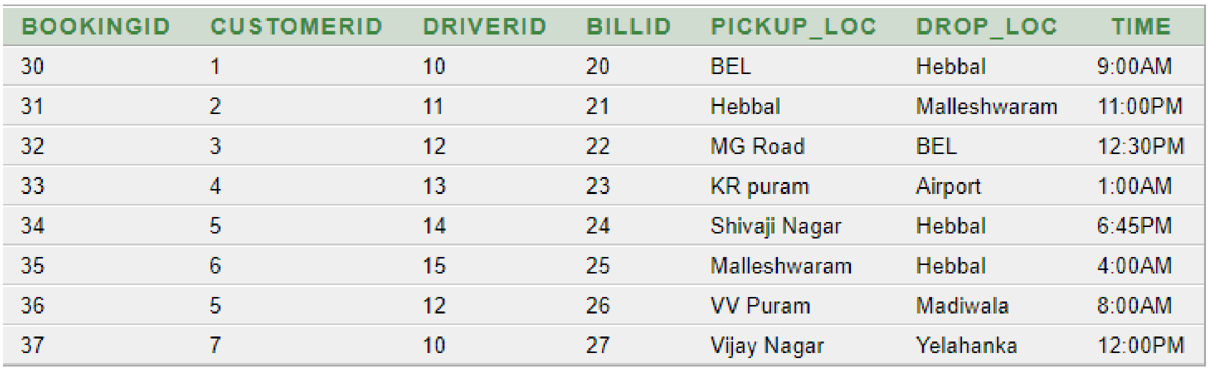
```
insert into Bill values(20,1,1000,'01-NOV-19');

insert into Bill values(21,2,1500,'01-MAR-19');

insert into Bill values(22,3,1250,'11-DEC-19');

insert into Bill values(23,4,750,'15-JUL-19');

insert into Bill values(24,5,670,'21-JUN-19');

insert into Bill values(25,6,450,'18-NOV-19');

insert into  bill values ( 26,5,900,'10-DEC-2019');

insert into Bill values(27,7,300,'11-DEC-19');
insert into Cab values(30,1,10,20,'BEL','Hebbal',’9:00AM’);

insert into Cab values(31,2,11,21,'Hebbal','Malleshwaram',’11:00PM’);

insert into Cab values(32,3,12,22,'MG Road','BEL',’12:30PM’);

insert into Cab values(33,4,13,23,'KR puram','Airport',’1:00AM’);

insert into Cab values(34,5,14,24,'Shivaji Nagar','Hebbal',’6.45PM’);

insert into Cab values(35,6,15,25,'Malleshwaram','Hebbal',’4:00AM’);

insert into cab values (36,5,12,26,'VV Puram','Madiwala',’8:00AM’);

insert into cab values (37,7,10,27,'Vijay Nagar','Yelahanka','12:00PM');


insert into car values(40,'KA04JT8990','Swift','Red','Hatchback');

insert into car values(41,'KA06MF990','WagonR','Brown','Hatchback');

insert into car values(42,'KA23JM4555','Compass','Blue','SUV');

insert into car values(43,'KA56JN4521','Swift','Maroon','Hatchback');

insert into car values(44,'KA07MN6322','Harrier','Black','SUV');

insert into car values(45,'KA08JM1234','Seltos','Red','SUV');

insert into car values(46,'KA07LZ6666','Ciaz','Black','Sedan');

insert into car values(47,'KA02JJ9843','Accord','White','Sedan');

insert into Car values(49,'KA11ZT1121','Etios','Grey','Hatchback');

insert into Car values(48,'KA01MF8999','Ecosport','White','SUV');


insert into Drives values(10,40);

insert into Drives values(11,41);

insert into Drives values(12,42);

insert into Drives values(13,43);

insert into Drives values(14,44);

insert into Drives values(15,47);

insert into Drives values(12,49);
insert into Payment_type values(20,'Card','Successful');

insert into Payment_type values(21,'BHIM','Successful');

insert into Payment_type values(22,'Card','Waiting for Payment');

insert into Payment_type values(23,'Cash','Successful');

insert into Payment_type values (24,'Cash','Successful');

insert into Payment_type values(25,'BHIM','Waiting for Payment');

insert into Payment_Type values(26,'BHIM','Successful');

insert into Payment_Type values(27,'Cash','Successful');


insert into Card values(20,'4600367890786543','Aditya','11-NOV-20',433);

insert into Card values(22,'4232445662345111','Bhoomika','11-JAN-21',123);


insert into Mobile_Payment values (21,'keer@ybl','9837652678','Google Pay');

insert into Mobile_Payment values (25,'saj@ybl','9988776656','PhonePe');

insert into Mobile_Payment values (26,'kru@ybl','8898789098','PayTM');
```

select * from Customer;



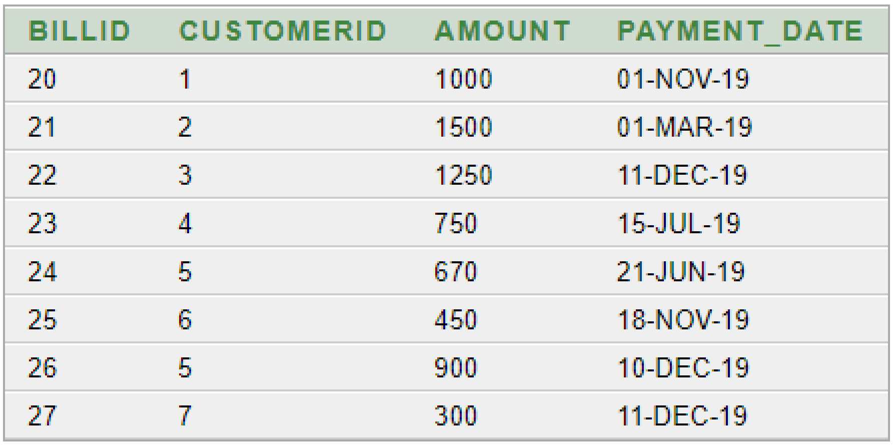
CUSTOMERID	CNAME	CEMAIL	CPHONE	CADDRESS	SEX	AGE
1	Adi	adi@gmail.com	9882901289	Malleshwaram	M	19
2	Keerthan	keer@gmail.com	9837652678	Vijay Nagar	M	21
3	Bhoomi	bhoomi@gmail.com	8927867898	Hebbal	F	19
4	Hemika	hemi@gmail.com	2788927139	Mekri	F	29
5	Kruthi	kru@gmail.com	8898789098	MG Road	F	22
6	Sajoo	sajoo@gmail.com	9988776656	Brigade Road	M	40
7	Goutham	GG@gmail.com	8926789000	Vijay Nagar	M	35

Select * from Cab;



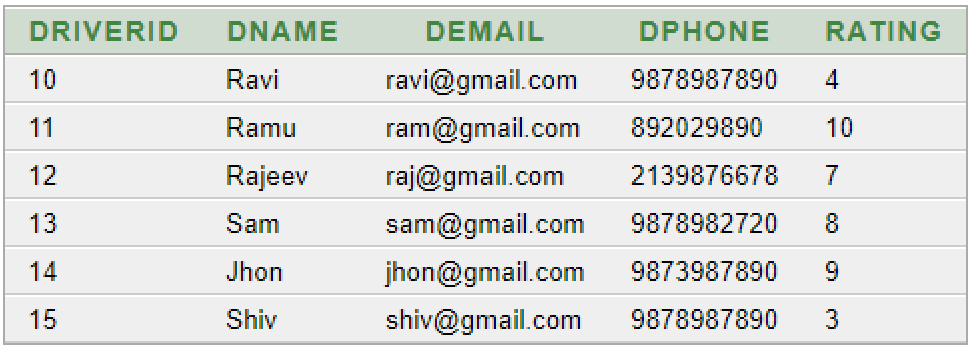
BOOKINGID	CUSTOMERID	DRIVERID	BILLID	PICKUP_LOC	DROP_LOC	TIME
30	1	10	20	BEL	Hebbal	9:00AM
31	2	11	21	Hebbal	Malleshwaram	11:00PM
32	3	12	22	MG Road	BEL	12:30PM
33	4	13	23	KR puram	Airport	1:00AM
34	5	14	24	Shivaji Nagar	Hebbal	6:45PM
35	6	15	25	Malleshwaram	Hebbal	4:00AM
36	5	12	26	VV Puram	Madiwala	8:00AM
37	7	10	27	Vijay Nagar	Yelahanka	12:00PM

Select * from Bill;



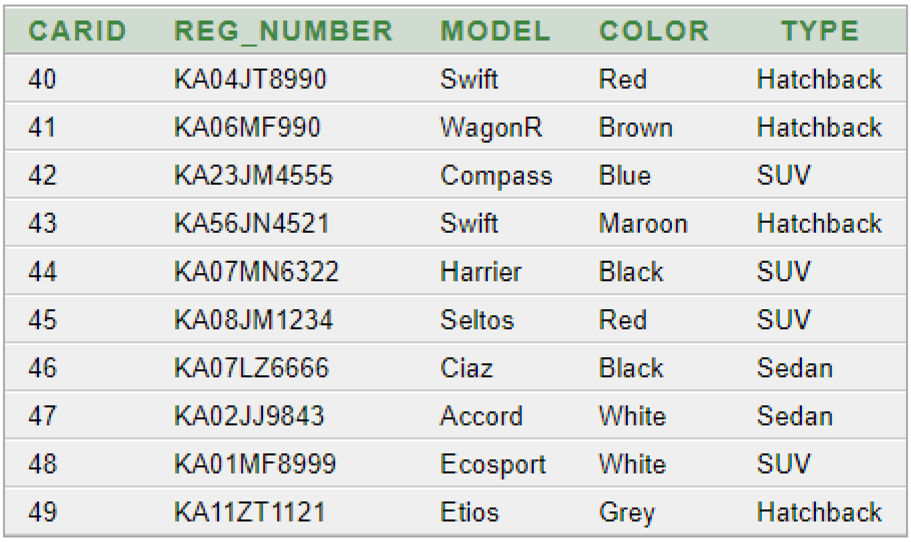
BILLID	CUSTOMERID	AMOUNT	PAYMENT_DATE
20	1	1000	01-NOV-19
21	2	1500	01-MAR-19
22	3	1250	11-DEC-19
23	4	750	15-JUL-19
24	5	670	21-JUN-19
25	6	450	18-NOV-19
26	5	900	10-DEC-19
27	7	300	11-DEC-19

select * from Driver;



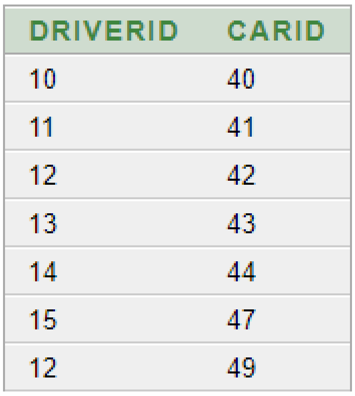
DRIVERID	DNAME	DEMAIL	DPHONE	RATING
10	Ravi	ravi@gmail.com	9878987890	4
11	Ramu	ram@gmail.com	892029890	10
12	Rajeev	raj@gmail.com	2139876678	7
13	Sam	sam@gmail.com	9878982720	8
14	Jhon	jhon@gmail.com	9873987890	9
15	Shiv	shiv@gmail.com	9878987890	3

Select * from Car;



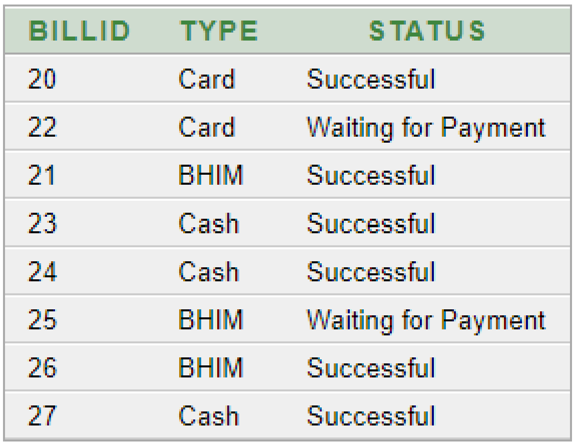
CARID	REG_NUMBER	MODEL	COLOR	TYPE
40	KA04JT8990	Swift	Red	Hatchback
41	KA06MF990	WagonR	Brown	Hatchback
42	KA23JM4555	Compass	Blue	SUV
43	KA56JN4521	Swift	Maroon	Hatchback
44	KA07MN6322	Harrier	Black	SUV
45	KA08JM1234	Seltos	Red	SUV
46	KA07LZ6666	Ciaz	Black	Sedan
47	KA02JJ9843	Accord	White	Sedan
48	KA01MF8999	Ecosport	White	SUV
49	KA11ZT1121	Etios	Grey	Hatchback

Select * from Drives;



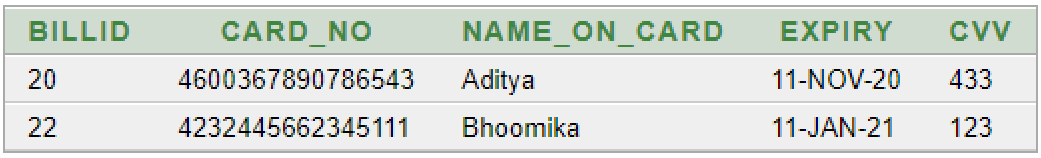
DRIVERID	CARID
10	40
11	41
12	42
13	43
14	44
15	47
12	49

Select * from Payment_Type;



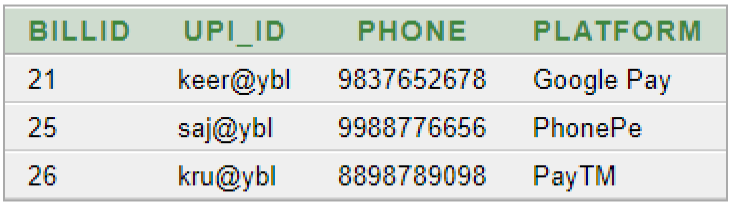
BILLID	TYPE	STATUS
20	Card	Successful
22	Card	Waiting for Payment
21	BHIM	Successful
23	Cash	Successful
24	Cash	Successful
25	BHIM	Waiting for Payment
26	BHIM	Successful
27	Cash	Successful

Select * from Card;



BILLID	CARD_NO	NAME_ON_CARD	EXPIRY	CVV
20	4600367890786543	Aditya	11-NOV-20	433
22	4232445662345111	Bhoomika	11-JAN-21	123

Select * from Mobile_Payment;

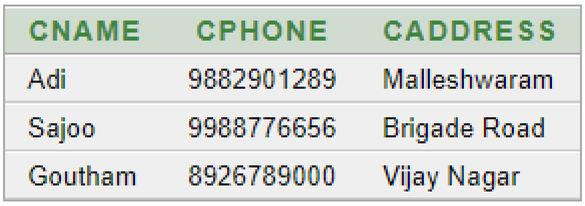


BILLID	UPI_ID	PHONE	PLATFORM
21	keer@ybl	9837652678	Google Pay
25	saj@ybl	9988776656	PhonePe
26	kru@ybl	8898789098	PayTM

4.3 Queries

1. Retrieve the name, phone number and address of all the customers who are allotted drivers with rating below 5.

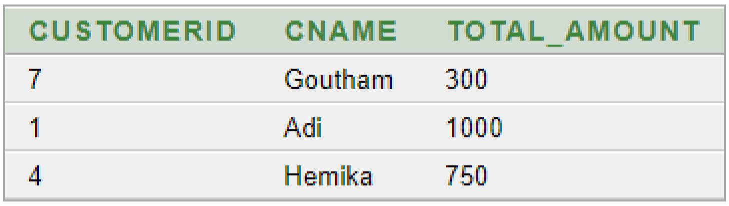
```
Query:  SELECT cus.CName,CPhone ,CAddress

        FROM Customer cus,Cab ca

        WHERE cus.CustomerID=ca.CustomerID

        AND  ca.DriverID  IN (   SELECT DriverID

                                 FROM Driver

                                WHERE Rating<5  );
```



CNAME	CPHONE	CADDRESS
Adi	9882901289	Malleshwaram
Sajoo	9988776656	Brigade Road
Goutham	8926789000	Vijay Nagar

2. Display the names of all the customers along with their total amount who have been allotted ‘swift’.

```
Query: SELECT c.CustomerID,c.CName,B.Amount as Total_Amount

        FROM Customer c, Cab ca,Bill B

       WHERE c.CustomerID=ca.CustomerID AND b.BillID= ca.BillID AND

       ca.DriverID  IN  ( SELECT  DriverID

                          FROM Drives

                          WHERE CarID  IN  (SELECT CarID

                                            FROM car

                                             WHERE Model='Swift'));
```

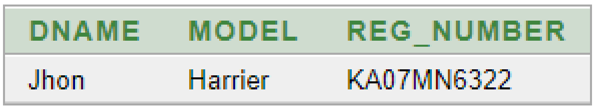


CUSTOMERID	CNAME	TOTAL_AMOUNT
7	Goutham	300
1	Adi	1000
4	Hemika	750

3.Retrieve the names of all drivers who drives a 'black' car along with its registration number and Model.

Query:

```
SELECT d.Dname,c.Model,Reg_Number
FROM driver d,  drives di,car c
WHERE (  d.DriverID=di.DriverID AND c.CarID=di.CarID   )
              AND c.color='Black' ;
```

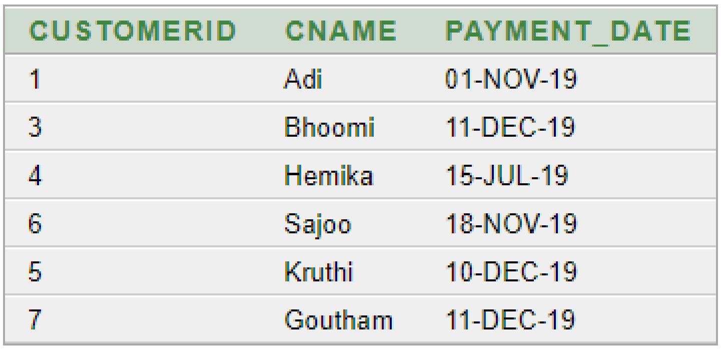


DNAME	MODEL	REG_NUMBER
Jhon	Harrier	KA07MN6322

4.List the names of Customers who have booked after July 11,2019.

Query:

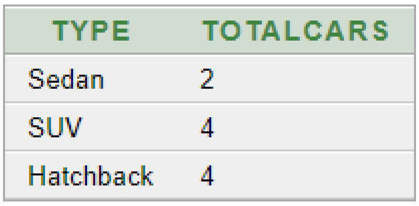
```
SELECT c.CustomerID,c.Cname, b.payment_Date
FROM Customer c, Bill b
WHERE  c.CustomerID=b.CustomerID
          AND b.payment_Date > '11-JULY-19';
```



CUSTOMERID	CNAME	PAYMENT_DATE
1	Adi	01-NOV-19
3	Bhoomi	11-DEC-19
4	Hemika	15-JUL-19
6	Sajoo	18-NOV-19
5	Kruthi	10-DEC-19
7	Goutham	11-DEC-19

5. Count the number of cars of each type in the company and arrange them in Ascending order.

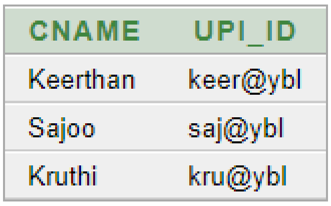
```
Query:  SELECT Type, count(*) AS TotalCars

           FROM car

           GROUP BY Type

           ORDER BY TotalCars;
```



TYPE	TOTALCARS
Sedan	2
SUV	4
Hatchback	4

6.Retrieve the names of all the Customers who have opted for mobile Payment.

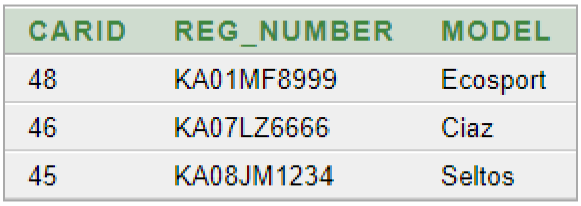
```
Query: SELECT c.Cname, p.UPI_ID

           FROM bill b , Mobile_Payment p , Customer c

           WHERE ( b.CustomerID= c.CustomerID  )

                          AND ( b.billID = p.billID );
```



CNAME	UPI_ID
Keerthan	keer@ybl
Sajoo	saj@ybl
Kruthi	kru@ybl

7.Retrieve the names of the cars that have not been assigned a driver.

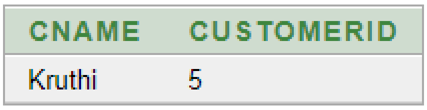
```
Query: SELECT CarId, REG_Number , Model

           FROM car

           WHERE carid  NOT IN (SELECT CarID

                                               FROM Drives);
```



CARID	REG_NUMBER	MODEL
48	KA01MF8999	Ecosport
46	KA07LZ6666	Ciaz
45	KA08JM1234	Seltos

8.Retrieve the names of the Customers who have used the service more than once.

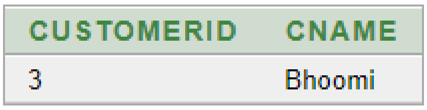
```
Query:  SELECT c.CName,C.CustomerID

            FROM customer c

            WHERE c.CustomerID IN (Select b.CustomerID

                                                FROM Bill b

                                               GROYP BY b.CustomerID

                                               HAVING count(CustomerID)>1);
```



CNAME	CUSTOMERID
Kruthi	5

9.Retrieve the names of all the customers who have booked the cab on the same day ‘Goutham’ booked.

```
Query: (SELECT c.CustomerID,c.cname

            FROM customer c,bill b

            WHERE c.CustomerID = b.CustomerID AND

            b.payment_Date IN ( SELECT b1.payment_Date

                                        FROM Customer c2 ,bill b1

                                        WHERE c2.Cname='Goutham' AND
                                        b1.customerID=c2.CustomerID))

            MINUS (SELECT CustomerID,Cname

                        FROM Customer

                         WHERE CName='Goutham');
```



CUSTOMERID	CNAME
3	Bhoomi

CONCLUSION

This project presents an investigate view of presenting the cab management system including the history of taxis. In this project, we have developed a simple database to manage a cab system for a single city. As a stepping stone into a database design, we have included the primarily possible transactions and services in the system by efficiently using the features provided by the Relational Database Management System. I further look forward to improvise my work by adding features of Object Oriented Database Management.

Its function basically meets the demand, to achieve the management of the site, driver customer information, booking info and payment methods. Of course, it is essential to insert, modify, delete and query the above data. We have tried to eliminate all the shortcomings of the present cab system and tried to ease the owner as well as the customers to manage the system in an efficient way.

Extension to this work shall be done by taking into consideration of taxis which runs in multiple cities followed by maintenance, illegal, insurance, accident and other major aspects of the management function.

Bibliography

Books:

1. Fundamentals of Database System,7th edition,Ramez Elmasri and Navathe .
2. Database System Concepts by Silberschatz, Korth and Sudharshan 5th edition.
3. Database System Concepts by Avi Silberschatz, Henry F. Korth, and S. Sudarshan.
4. SQL Cookbook by Anthony Molinaro.

Websites:

1. www.oracle.com
2. www.w3schools.com
3. www.projecttopics.info
4. www.erplus.com
5. www.wiki.com