Project Name:

CAR RIDE HAILING MANAGEMENT SYSTEM

Introduction:

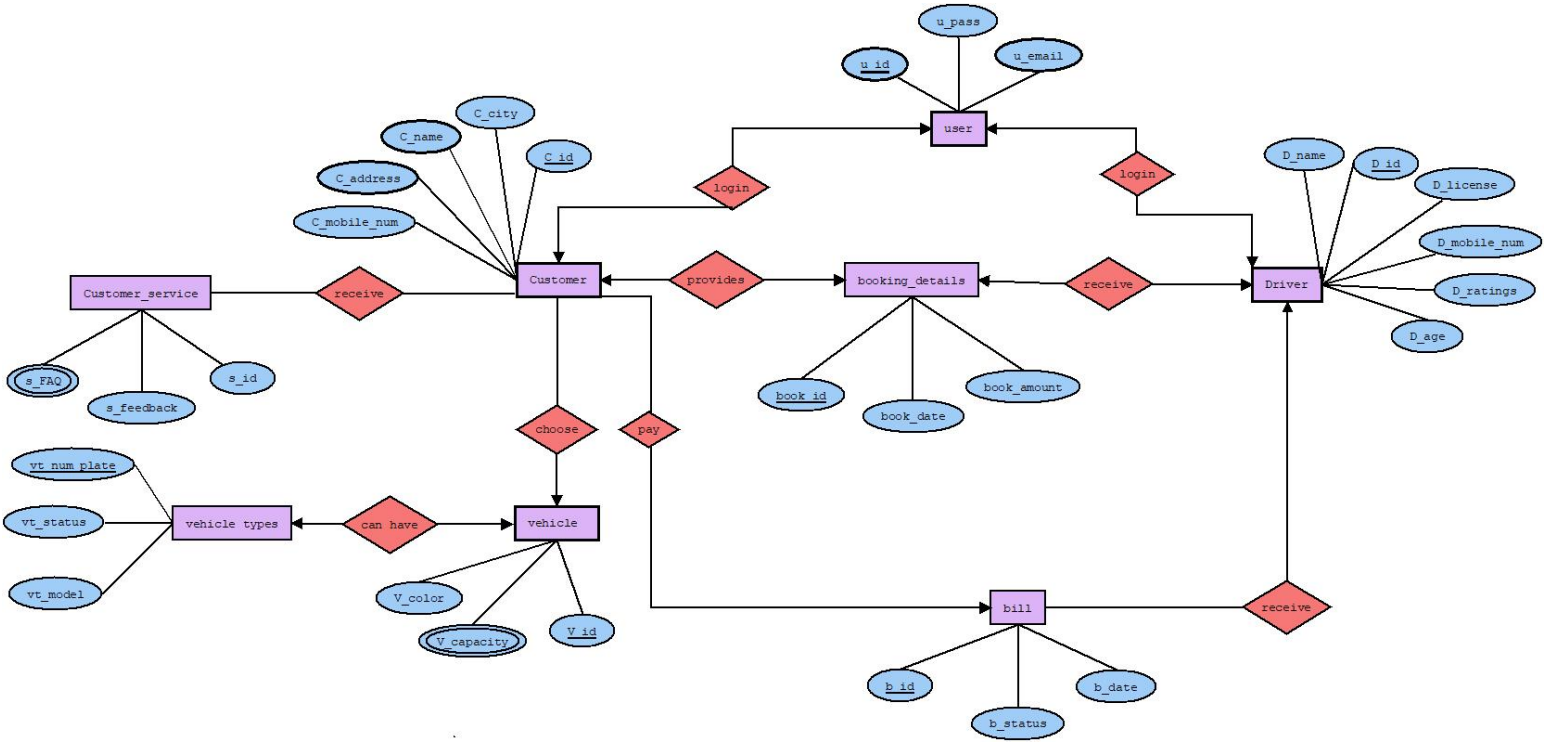
An online Car Ride Hailing Management System is developed to manage all Car hiring work online. It is helpful for car booking websites that are specialized in hiring cars for customers. Using this system, it is a straightforward process for customers to book a car online. It is an online system through which customers can view available cars, register the cars, view profiles, and book cars. Our online cab or car booking system's objective and scope are to record the details of users' various activities. Using this car ride-hailing management system, car owners can also become partners of car booking agencies by giving their car for booking. Online car, a ride-hailing management system, is a web-based application that allows users to book a car online. From this system, car rental companies can manage all car bookings and customer information.

Customers can book cars, and drivers can confirm the booking, and both can also cancel the ride. Car Ride Hailing Management System will be an android application that will work like Uber and Pathao. This application will provide services to passengers and drivers. This application enables passengers to order any vehicle with their smartphones. The main objective of this application is that drivers online register their vehicles by providing their essential information and then check their nearest pick and drop points. This application is managing passengers' booking quickly and easily with the shortest time possible. With one click on the button, you can request a vehicle if you are a passenger. With another click, you accept or decline request(s) if you are a driver. No third party is required to control this process.

Scenario:

In a car hailing management system a user can login as a customer. And a customer can log in as one user. A customer is identified by customer id. The system also stores customer name, city, address, mobile number. A user is identified by user id. User email and user password of a user are also stored. A user can also login as a driver. And a driver can log in as one user. A driver is identified by driver id. The system also stores driver name, license, age, ratings, mobile number. A customer can receive many customer service. And a customer service can be received by many customers. A customer service is identified by service id. A customer service may have multiple service frequently asked questions. Service feedback of a customer service is also stored. customer can choose a vehicle. A vehicle can be chosen by many customers. A vehicle is identified by its vehicle id. During choosing vehicle its vehicle color and capacity are also stored. There may have multiple vehicle capacity of a vehicle. A vehicle can have many vehicle types. A vehicle types has a vehicle. Vehicle types are identified by vehicle number plate. The system also stores vehicle status and model. To get a customer service a customer provides one booking details. A booking details is provided by one customer. A booking details is identified by booking id. The system also stores booking amount and date of booking details. A driver receives a booking details. And a booking details can be received by one driver. To complete the ride a customer needs to pay a bill. And a bill is paid by many customers. A bill is identified by bill id. While paying bill, bill status and bill date are also stored by system. A driver receives many bill. A bill is received by a driver.

ER Diagram:



Normalization:

Log in as customer:

UNF: login_C (u_id , u_email, u_password, c_id, c_name, c_address, c_city, c_mobile_num)

1NF: There is no multivalued attribute.

1) u_id , u_email, u_password, c_id, c_name, c_addresses, c_city, c_mobile_num.

2NF:

1) u_id, u_email, u_password.
2) c_id, c_name, c_addresses, c_city, c_mobile_num.

3NF:

1) u_id , u_email, u_password
3) c_id, c_name, c_mobile_num,
4) c_addresses, c_city

Table creation:

1) u_id, u_email, u_password

2) c_id, c_name, c_mobile_num, **a_id, u_id**.
3) a_id, c_addresses, c_city.

Log in as driver:

UNF: login_D(u_id , u_email, u_password, d_id, d_name, d_age, d_license, d_mobile_number ,d_ratings

1NF: There is no multivalued attribute.

1) u_id, u_email, u_password, d_id, d_name, d_age, d_license, d_mobile_number, d_ratings

2NF:

1) u_id , u_email, u_password.
2) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings

3NF: There is no transitive dependency.

1) u_id , u_email, u_password.
2) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings.

Table creation:

1) u_id , u_email, u_password
2) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings, **u_id**.

Receive Bill:

UNF: Receive (d_id, d_name, d_age, d_license, d_mobile_number, d_ratings, b_id, b_status, b_date)

1NF: There is no multivalued attribute

1) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings, b_id, b_status, b_date.

2NF:

1) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings.
2) b_id, b_status, b_date.

3NF: There is no transitive dependency.

1) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings.
2) b_id, b_status, b_date.

Table creation:

1) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings.
2) b_id, b_status, b_date, **d_id**.

Pay Bill:

UNF: Pay (c_id, c_name, c_address, c_city, c_mobile_num, b_id, b_status, b_date)

1NF: There is no multivalued attribute

1) c_id, c_name, c_address, c_city, c_mobile_num, b_id, b_status, b_date.

2NF:

1) c_id, c_name, c_address, c_city, c_mobile_num.
2) b_id, b_status, b_date

3NF:

1) c_id, c_name, c_mobile_num.
2) c_address, c_city
3) b_id, b_status, b_date

Table Creation:

1) c_id, c_name, c_mobile_num,**a_id, b_id**.
2) a_id, c_address, c_city.
3) b_id, b_status, b_date.

Customer provides booking details:

UNF: Provides (c_id, c_name, c_address, c_city, c_mobile_num, book_id, book_date, book_amount)

1NF: There is no multivalued attribute.

1) c_id, c_name, c_address, c_city, c_mobile_num, book_id, book_date, book_amount.

2NF:

1) c_id, c_name, c_address, c_city, c_mobile_num.
2) book_id, book_date, book_amount.

3NF:

1) c_id, c_name, c_mobile_num.
2) c_address, c_city.
3) book_id, book_date, book_amount

Table creation:

1) c_id, c_name, c_mobile_num, **a_id, book_id**.
2) a_id, c_address, c_city.
3) book_id, book_date, book_amount.

Driver receive booking details:

UNF: Receive (d_id, d_name, d_age, d_license, d_mobile_number, d_ratings, book_id, book_date, book_amount)

1NF: There is no multivalued attribute

1) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings, book_id, book_date, book_amount

2NF:

1) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings
2) book_id, book_date, book_amount

3NF: There is no transitive dependency

1) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings
2) book_id, book_date, book_amount.

Table creation:

1) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings, book_id
2) book_id, book_date, book_amount.

Customer choose vehicle:

UNF: Choose (c_id, c_name, c_address, c_city, c_mobile_num, v_id, v_color, v_capacity)

1NF: v_ capacity is a Multivalued attribute.

1) c_id, c_name, c_address, c_city, c_mobile_num, v_id, v_color, v_capacity.

2NF:

1) c_id, c_name, c_address, c_city, c_mobile_num.
2) v_id, v_color, v_capacity

3NF:

1) c_id, c_name, c_mobile_num.
2) c_address, c_city.
3) v_id, v_color, v_capacity.

Table creation:

1) c_id, c_name, c_mobile_num, **a_id, v_id**.
2) a_id, c_address, c_city.
3) v_id, v_color, v_capacity.

Receive Customer service:

UNF (c_id, c_name, c_address, c_city, c_mobile_num, s_id, s_FAQ, s_feedback)

1NF: FAQ is a multivalued attribute.

1) c_id, c_name, c_address, c_city, c_mobile_num, s_id, s_FAQ, s_feedback.

2NF:

1) c_id, c_name, c_address, c_city, c_mobile_num.
2) s_id, s_FAQ, s_feedback.

3NF:

1) c_id, c_name, c_mobile_num.
2) c_address, c_city.
3) s_id, s_FAQ, s_feedback

Table creation:

1) c_id, c_name, c_mobile_num, **a_id, s_id**.
2) a_id, c_address, c_city.
3) s_id, s_FAQ, s_feedback.
4) s_id, c_id.

Vehicle:

UNF: can have (v_id, v_color, v_capacity, vt_numplate, vt_status, vt_model)

1NF: v_capacity is a multivalued attribute.

1) v_id, v_color, v_capacity, vt_numplate, vt_status, vt_model.

2NF:

1) v_id, v_color, v_capacity
2) vt_numplate, vt_status, vt_model.

3NF:

1) v_id, v_color, v_capacity.
2) vt_numplate, vt_status, vt_model.

Table creation:

1) v_id, v_color, v_capacity.
2) vt_numplate, vt_status, vt_model, **v_id**.

Temporary Table:

1) u_id, u_email, u_password
2) c_id, c_name, c_mobile_num, **a_id, u_id**.
3) a_id, c_addresses, c_city.
4) ~~u_id , u_email, u_password~~
5) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings, **u_id**.
6) ~~d_id, d_name, d_age, d_license, d_mobile_number, d_ratings.~~
7) b_id, b_status, b_date, **d_id**.
8) c_id, c_name, c_mobile_num,**a_id, b_id**.
9) ~~a_id, c_address, c_city~~.
10) ~~b_id, b_status, b_date.~~
11) c_id, c_name, c_mobile_num, **a_id, book_id**.
12) ~~a_id, c_address, c_city.~~
13) book_id, book_date, book_amount.
14) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings, book_id
15) ~~book_id, book_date, book_amount.~~
16) c_id, c_name, c_mobile_num, **a_id, v_id**.
17) ~~a_id, c_address, c_city.~~
18) v_id, v_color, v_capacity.
19) c_id, c_name, c_mobile_num, **a_id, s_id**.
20) ~~a_id, c_address, c_city.~~
21) s_id, s_FAQ, s_feedback.
22) s_id, c_id.
23) ~~v_id, v_color, v_capacity.~~
24) vt_numplate, vt_status, vt_model, **v_id**.

Final Table:

1) u_id, u_email, u_password
2) c_id, c_name, c_mobile_num, **a_id, u_id**.
3) a_id, c_addresses, c_city.
4) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings, **u_id**.
5) b_id, b_status, b_date, **d_id**.
6) c_id, c_name, c_mobile_num,**a_id, b_id**.
7) c_id, c_name, c_mobile_num, **a_id, book_id**.
8) book_id, book_date, book_amount.
9) d_id, d_name, d_age, d_license, d_mobile_number, d_ratings, book_id
10) c_id, c_name, c_mobile_num, **a_id, v_id**.
11) v_id, v_color, v_capacity.
12) c_id, c_name, c_mobile_num, **a_id, s_id**.
13) s_id, s_FAQ, s_feedback.
14) s_id, c_id.
15) vt_numplate, vt_status, vt_model, **v_id**.

Schema Diagram:

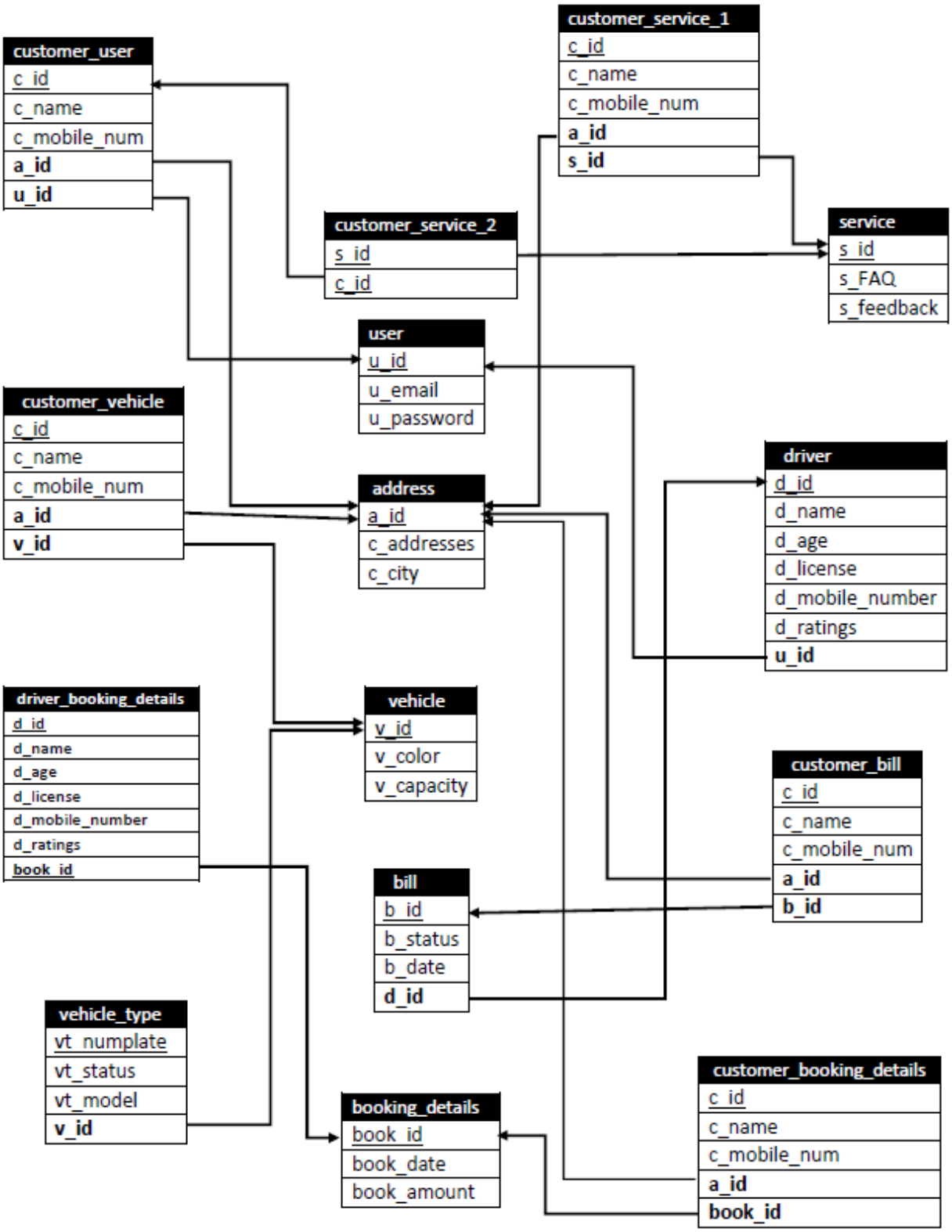


Table Creation:

Creating table Users:

create table Users(u_id number(5) primary key, u_email varchar2(50), u_password varchar2(20));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
USERS	U_ID	Number	-	5	0	1	-	-	-
	U_EMAIL	Varchar2	50	-	-	-	✓	-	-
	U_PASSWORD	Varchar2	20	-	-	-	✓	-	-
									1 - 3

Creating table Address:

create table Address(a_id number(5) primary key, c_addresses varchar2(20), c_city varchar2(15));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ADDRESS	A_ID	Number	-	5	0	1	-	-	-
	C_ADDRESSES	Varchar2	20	-	-	-	✓	-	-
	C_CITY	Varchar2	15	-	-	-	✓	-	-
									1 - 3

Creating table Customer_User:

create table Customer_User(c_id number(5) primary key, c_name varchar2(20), c_mobile_num number(11), a_id number(5), foreign key(a_id) references Address(a_id), u_id number(5), foreign key(u_id) references Users(u_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_USER	C_ID	Number	-	5	0	1	-	-	-
	C_NAME	Varchar2	20	-	-	-	✓	-	-
	C_MOBILE_NUM	Number	-	11	0	-	✓	-	-
	A_ID	Number	-	5	0	-	✓	-	-
	U_ID	Number	-	5	0	-	✓	-	-
									1 - 5

Creating table Service:

create table Service(s_id number(5) primary key, s_FAQ1 varchar2(50), s_FAQ2 varchar2(50), s_FAQ3 varchar2(50), s_feedback varchar2(50));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SERVICE	S_ID	Number	-	5	0	1	-	-	-
	S_FAQ1	Varchar2	50	-	-	-	✓	-	-
	S_FAQ2	Varchar2	50	-	-	-	✓	-	-
	S_FAQ3	Varchar2	50	-	-	-	✓	-	-
	S_FEEDBACK	Varchar2	50	-	-	-	✓	-	-
									1 - 5

Creating table Customer_Service_1:

create table Customer_Service_1(c_id number(5) primary key, c_name varchar2(20), c_mobile_num number(11), a_id number(5), foreign key(a_id) references Address(a_id), s_id number(5), foreign key(s_id) references Service(s_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_SERVICE_1	C_ID	Number	-	5	0	1	-	-	-
	C_NAME	Varchar2	20	-	-	-	✓	-	-
	C_MOBILE_NUM	Number	-	11	0	-	✓	-	-
	A_ID	Number	-	5	0	-	✓	-	-
	S_ID	Number	-	5	0	-	✓	-	-
									1 - 5

Creating table Customer_Service_2:

create table Customer_Service_2(s_id number(5), foreign key(s_id) references Service(s_id), c_id number(5), foreign key(c_id) references Customer_User(c_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_SERVICE_2	S_ID	Number	-	5	0	-	✓	-	-
	C_ID	Number	-	5	0	-	✓	-	-
									1 - 2

Creating table Vehicle:

create table Vehicle(v_id number(5) primary key, v_color varchar2(10), v_capacity1 varchar2(30), v_capacity2 varchar2(30), v_capacity3 varchar(30));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VEHICLE	V_ID	Number	-	5	0	1	-	-	-
	V_COLOR	Varchar2	10	-	-	-	✓	-	-
	V_CAPACITY1	Varchar2	30	-	-	-	✓	-	-
	V_CAPACITY2	Varchar2	30	-	-	-	✓	-	-
	V_CAPACITY3	Varchar2	30	-	-	-	✓	-	-

1 - 5

Creating table Customer_Vehicle:

create table Customer_Vehicle(c_id number(5) primary key, c_name varchar2(20), c_mobile_num number(11), a_id number(5), foreign key(a_id) references Address(a_id), v_id number(5), foreign key(v_id) references Vehicle(v_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_VEHICLE	C_ID	Number	-	5	0	1	-	-	-
	C_NAME	Varchar2	20	-	-	-	✓	-	-
	C_MOBILE_NUM	Number	-	11	0	-	✓	-	-
	A_ID	Number	-	5	0	-	✓	-	-
	V_ID	Number	-	5	0	-	✓	-	-

1 - 5

Creating table Driver:

create table Driver(d_id number(5) primary key, d_name varchar2(20), d_age number(2), d_license varchar2(20), d_mobile_num number(11), d_ratings number(1), u_id number(5), foreign key(u_id) references Users(u_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DRIVER	D_ID	Number	-	5	0	1	-	-	-
	D_NAME	Varchar2	20	-	-	-	✓	-	-
	D_AGE	Number	-	2	0	-	✓	-	-
	D_LICENSE	Varchar2	20	-	-	-	✓	-	-
	D_MOBILE_NUM	Number	-	11	0	-	✓	-	-
	D_RATINGS	Number	-	1	0	-	✓	-	-
	U_ID	Number	-	5	0	-	✓	-	-

1 - 7

Creating table Bill:

create table Bill(b_id number(5) primary key, b_status varchar2(12), b_date varchar2(10), d_id number(5), foreign key(d_id) references Driver(d_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BILL	B_ID	Number	-	5	0	1	-	-	-
	B_STATUS	Varchar2	12	-	-	-	✓	-	-
	B_DATE	Varchar2	10	-	-	-	✓	-	-
	D_ID	Number	-	5	0	-	✓	-	-
									1 - 4

Creating table Customer_Bill:

create table Customer_Bill(c_id number(5) primary key, c_name varchar2(20), c_mobile_num number(11), a_id number(5), foreign key(a_id) references Address(a_id), b_id number(5), foreign key(b_id) references Bill(b_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_BILL	C_ID	Number	-	5	0	1	-	-	-
	C_NAME	Varchar2	20	-	-	-	✓	-	-
	C_MOBILE_NUM	Number	-	11	0	-	✓	-	-
	A_ID	Number	-	5	0	-	✓	-	-
	B_ID	Number	-	5	0	-	✓	-	-
									1 - 5

Creating table Booking_Details:

create table Booking_Details(book_id number(5) primary key, book_date varchar2(10), book_amount number(4));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BOOKING_DETAILS	BOOK_ID	Number	-	5	0	1	-	-	-
	BOOK_DATE	Varchar2	10	-	-	-	✓	-	-
	BOOK_AMOUNT	Number	-	4	0	-	✓	-	-
									1 - 3

Creating table Vehicle_Type:

create table Vehicle_Type (vt_numplate varchar2(50) primary key, vt_status varchar2(50), vt_model varchar2(50), v_id number(5), foreign key(v_id) references Vehicle(v_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VEHICLE_TYPE	VT_NUMPLATE	Varchar2	50	-	-	1	-	-	-
	VT_STATUS	Varchar2	50	-	-	-	✓	-	-
	VT_MODEL	Varchar2	50	-	-	-	✓	-	-
	V_ID	Number	-	5	0	-	✓	-	-

1 - 4

Creating table Customer_Booking_Details:

create table Customer_Booking_Details(c_id number(5) primary key, c_name varchar2(20), c_mobile_num number(11), a_id number(5), foreign key(a_id) references Address(a_id), book_id number(5), foreign key(book_id) references Booking_Details(book_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_BOOKING_DETAILS	C_ID	Number	-	5	0	1	-	-	-
	C_NAME	Varchar2	20	-	-	-	✓	-	-
	C_MOBILE_NUM	Number	-	11	0	-	✓	-	-
	A_ID	Number	-	5	0	-	✓	-	-
	BOOK_ID	Number	-	5	0	-	✓	-	-

1 - 5

Creating table Driver_Booking_Details:

create table Driver_Booking_Details(d_id number(5) primary key, d_name varchar2(20), d_age number(2), d_license varchar2(20), d_mobile_num number(11), d_ratings number(1), book_id number(5), foreign key(book_id) references Booking_Details(book_id));

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DRIVER_BOOKING_DETAILS	D_ID	Number	-	5	0	1	-	-	-
	D_NAME	Varchar2	20	-	-	-	✓	-	-
	D_AGE	Number	-	2	0	-	✓	-	-
	D_LICENSE	Varchar2	20	-	-	-	✓	-	-
	D_MOBILE_NUM	Number	-	11	0	-	✓	-	-
	D_RATINGS	Number	-	1	0	-	✓	-	-
	BOOK_ID	Number	-	5	0	-	✓	-	-

1 - 7

Data Insertion:

Inserting into Users:

insert into Users values('101','labibaibnat17@gmail.com','labi1234');

insert into Users values('102','rubyeazannat2833@gmail.com','rubu1234');

insert into Users values('103','khaled0207@gmail.com','khaled1234');

insert into Users values('104','tanzimidris@gmail.com','farabi1234');

insert into Users values('105','Moinulislam@gmail.com','moinul1234');

insert into Users values('106','taseen17@gmail.com','samiha1234');

insert into Users values('107','nuhafadila2833@gmail.com','afrin1234');

insert into Users values('108','shadhin0207@gmail.com','zia1234');

insert into Users values('109','tanzinabinte@gmail.com','tanzina1234');

insert into Users values('110','Moinulasajid@gmail.com','moinulyya1234');

U_ID	U_EMAIL	U_PASSWORD
101	labibaibnat17@gmail.com	labi1234
102	rubyeazannat2833@gmail.com	rubu1234
103	khaled0207@gmail.com	khaled1234
104	tanzimidrees@gmail.com	farabi1234
105	Moinulislam@gmail.com	moinul1234
106	taseen17@gmail.com	samiha1234
107	nuhafadila2833@gmail.com	afrin1234
108	shadhin0207@gmail.com	zia1234
109	tanzinabinte@gmail.com	tanzina1234
110	Moinulasajid@gmail.com	moinulyya1234

Insert into Address:

insert into Address values('1','60/9 Mannanline','Dhaka');

insert into Address values('2','17/a Badda','Dhaka');

insert into Address values('3','68/c Patenga','Chittagong');

insert into Address values('4','24/a Mohammadpur','Dhaka');

insert into Address values('5','35/z kolabagan','Dhaka');

A_ID	C_ADDRESSES	C_CITY
1	60/9 Mannanline	Dhaka
2	17/a Badda	Dhaka
3	68/c Patenga	Chittagong
4	24/a Mohammadpur	Dhaka
5	35/z kolabagan	Dhaka

Insert into Customer_User:

insert into Customer_User values('1001','Labiba Ibnat','01715043631','1','101');

insert into Customer_User values('1002','Rubyea zannat','01915043678','2','102');

insert into Customer_User values('1003','khaled Hossain','01815043645','3','103');

insert into Customer_User values('1004','Tanzim Idris','01515043666','4','104');

insert into Customer_User values('1005','Moinul Islam','01615043678','5','105');

C_ID	C_NAME	C_MOBILE_NUM	A_ID	U_ID
1001	Labiba Ibnat	1715043631	1	101
1002	Rubyea zannat	1915043678	2	102
1003	khaled Hossain	1815043645	3	103
1004	Tanzim Idris	1515043666	4	104
1005	Moinul Islam	1615043678	5	105

Insert into Service:

insert into service values ('2001', 'How do I request a ride?', 'How will I know if I can get the ride I requested?', 'Who is eligible for Ride Connection Services?', 'Well skilled driver');

insert into service values ('2002', 'What is the cost of the service?', 'What are the hours of services? ', 'What areas do you serve?', 'Well behavior');

insert into service values ('2003', 'Where can Ride Connection take me?', 'Who will drive me?', 'How will I get home?', 'vehicle condition was satisfied');

insert into service values ('2004', 'What if my ride is late?', 'What if I need to cancel my ride?', 'Why sometimes other names of driver pick me up?', ' Driver took more time for my ride');

insert into service values ('2005', 'Who is eligible for Ride Connection Services?', 'What is the cost of the service?', 'How will I know if I can get the ride I requested?', ' He asked for extra money');

S_ID	S_FAQ1	S_FAQ2	S_FAQ3	S_FEEDBACK
2001	How do I request a ride?	How will I know if I can get the ride I requested?	Who is eligible for Ride Connection Services?	Well skilled driver
2002	What is the cost of the service?	What are the hours of services?	What areas do you serve?	Well behavior
2003	Where can Ride Connection take me?	Who will drive me?	How will I get home?	vehicle condition was satisfied
2004	What if my ride is late?	What if I need to cancel my ride?	Why sometimes other name of driver pick me up?	Driver took more time for my ride
2005	Who is eligible for Ride Connection Services?	What is the cost of the service?	How will I know if I can get the ride I requested?	He asked for extra money

Insert into Customer_Service_1:

insert into Customer_Service_1 values('1001','Labiba Ibnat','01715043631','1','2001');

insert into Customer_Service_1 values('1002','Rubyea zannat','01915043678','2','2002');

insert into Customer_Service_1 values('1003','khaled Hossain','01815043645','3','2003');

insert into Customer_Service_1 values('1004','Tanzim Idris','01515043666','4','2004');

insert into Customer_Service_1 values('1005','Moinul Islam','01615043678','5','2005');

C_ID	C_NAME	C_MOBILE_NUM	A_ID	S_ID
1001	Labiba Ibnat	1715043631	1	2001
1002	Rubyea zannat	1915043678	2	2002
1003	khaled Hossain	1815043645	3	2003
1004	Tanzim Idris	1515043666	4	2004
1005	Moinul Islam	1615043678	5	2005

Insert into Customer_Service_2:

insert into Customer_Service_2 values('2001', '1001');

insert into Customer_Service_2 values('2002', '1002');

insert into Customer_Service_2 values('2003', '1003');

insert into Customer_Service_2 values('2004', '1004');

insert into Customer_Service_2 values('2005', '1005');

S_ID	C_ID
2001	1001
2002	1002
2003	1003
2004	1004
2005	1005

Insert into Vehicle:

insert into Vehicle values('1111', 'Sky Blue', 'Seat-5', ' Gross Weight-4400', 'Net capacity-2200');

insert into Vehicle values('1112', 'Yellow', 'Seat-3', ' Gross Weight-2000', 'Net capacity-1100');

insert into Vehicle values('1113', 'Green', 'Seat-8', ' Gross Weight-4000', 'Net capacity-2000');

insert into Vehicle values('1114', 'Sky Blue', 'Seat-10', ' Gross Weight-5000', 'Net capacity-3000');

insert into Vehicle values('1115', 'Sky Blue', 'Seat-9', ' Gross Weight-4500', 'Net capacity-2100');

V_ID	V_COLOR	V_CAPACITY1	V_CAPACITY2	V_CAPACITY3
1111	Sky Blue	Seat-5	Gross Weight-4400	Net capacity-2200
1112	Yellow	Seat-3	Gross Weight-2000	Net capacity-1100
1113	Green	Seat-8	Gross Weight-4000	Net capacity-2000
1114	Sky Blue	Seat-10	Gross Weight-5000	Net capacity-3000
1115	Sky Blue	Seat-9	Gross Weight-4500	Net capacity-2100

Insert into Customer_Vehicle:

insert into Customer_Vehicle values('1001', 'Labiba Ibnat', '01715043631', '1','1111');

insert into Customer_Vehicle values('1002', 'Rubyea Zannat','01915043678','2','1112');

insert into Customer_Vehicle values('1003', 'khaled Hossain','01815043645','3','1113');

insert into Customer_Vehicle values('1004', 'Tanzim Idris', '01515043666','4','1114');

insert into Customer_Vehicle values('1005', 'Moinul Islam', '01615043678','5','1115');

C_ID	C_NAME	C_MOBILE_NUM	A_ID	V_ID
1001	Labiba Ibnat	1715043631	1	1111
1002	Rubyea Zannat	1915043678	2	1112
1003	khaled Hossain	1815043645	3	1113
1004	Tanzim Idris	1515043666	4	1114
1005	Moinul Islam	1615043678	5	1115

Insert into Driver:

insert into Driver values('1121','taseen','25', 'DHK00032C74521V','0192374957','1','106');

insert into Driver values('1122','nuhafadila','27', 'CTG000215KL3654','0173926452','2','107');

insert into Driver values('1123','shadhin','29','NK00003214LP697','01823476589','1','108');

insert into Driver values('1124','tanzinabinte','30','BAR001254Z36414','01918726354','2','109');

insert into Driver
values('1125','Moinulasajid','40','KHU0125M365V125','01531542741','2','110');

D_ID	D_NAME	D_AGE	D_LICENSE	D_MOBILE_NUM	D_RATINGS	U_ID
1121	taseen	25	DHK00032C74521V	192374957	1	106
1122	nuhafadila	27	CTG000215KL3654	173926452	2	107
1123	shadhin	29	NK00003214LP697	1823476589	1	108
1124	tanzinabinte	30	BAR001254Z36414	1918726354	2	109
1125	Moinulasajid	40	KHU0125M365V125	1531542741	2	110

Insert into Bill:

insert into Bill values('221', 'Paid','05-08-2021','1121');

insert into Bill values('222', 'Paid','06-08-2021','1122');

insert into Bill values('223', 'Paid','07-08-2021','1123');

insert into Bill values('224', 'Paid','08-08-2021','1124');

insert into Bill values('225', 'Paid','09-08-2021','1125');

B_ID	B_STATUS	B_DATE	D_ID
221	Paid	05-08-2021	1121
222	Paid	06-08-2021	1122
223	Paid	07-08-2021	1123
224	Paid	08-08-2021	1124
225	Paid	09-08-2021	1125

Insert into Customer_Bill:

insert into Customer_Bill values('101', 'Labiba Ibnat','01715043631','1','221');

insert into Customer_Bill values('102', 'Rubyea Zannat','01815043645','2','222');

insert into Customer_Bill values('103','khaled Hossain', '01815043645','3','224');

insert into Customer_Bill values('104', 'Tanzim Idris','01515043666','4','225');

insert into Customer_Bill values('105', 'Moinul Islam', '01615043678','5','225');

C_ID	C_NAME	C_MOBILE_NUM	A_ID	B_ID
101	Labiba Ibnat	1715043631	1	221
102	Rubyea Zannat	1815043645	2	222
103	khaled Hossain	1815043645	3	224
104	Tanzim Idris	1515043666	4	225
105	Moinul Islam	1615043678	5	225

Insert into Booking_Details:

insert into Booking_Details values('5001','14/8/2021','5000');

insert into Booking_Details values('5002','13/8/2021','500');

insert into Booking_Details values('5003','12/8/2021','4380');

insert into Booking_Details values('5004','14/8/2021','327');

insert into Booking_Details values('5005','13/8/2021','5000');

BOOK_ID	BOOK_DATE	BOOK_AMOUNT
5001	14/8/2021	5000
5002	13/8/2021	500
5003	12/8/2021	4380
5004	14/8/2021	327
5005	13/8/2021	5000

Insert into Booking_Details:

insert into Vehicle_Type values('D-Ka 501','Available','Toyota SUV','1111');

insert into Vehicle_Type values('D-Ga 601','Available','BMW SEDEN','1112');

insert into Vehicle_Type values('D-Gha 701','Not available','Toyota Hilux','1113');

insert into Vehicle_Type values('D-Kha 801','Available','Mazda MX5','1114');

insert into Vehicle_Type values('D-Ka 901','Not available','AUDI A8','1115');

VT_NUMPLATE	VT_STATUS	VT_MODEL	V_ID
D-Ka 501	Available	Toyota SUV	1111
D-Ga 601	Available	BMW SEDEN	1112
D-Gha 701	Not available	Toyota Hilux	1113
D-Kha 801	Available	Mazda MX5	1114
D-Ka 901	Not available	AUDI A8	1115

Insert into Customer_Booking_Details:

insert into Customer_Booking_Details values('1001','Labiba Ibnat','01715043631','1', '5001');

insert into Customer_Booking_Details values('1002','Rubyea zannat','01915043678','2', '5002');

insert into Customer_Booking_Details values('1003','khaled Hossain','01815043645','3', '5003');

insert into Customer_Booking_Details values('1004','Tanzim Idris','01515043666','4', '5004');

insert into Customer_Booking_Details values('1005','Moinul Islam','01615043678','5', '5005');

C_ID	C_NAME	C_MOBILE_NUM	A_ID	BOOK_ID
1001	Labiba Ibnat	1715043631	1	5001
1002	Rubyea zannat	1915043678	2	5002
1003	khaled Hossain	1815043645	3	5003
1004	Tanzim Idris	1515043666	4	5004
1005	Moinul Islam	1615043678	5	5005

Insert into Driver_Booking_Details:

insert into Driver_Booking_Details
values('1121','taseen','25','DHK00032C74521V','0192374957','1','5001');

insert into Driver_Booking_Details
values('1122','nuhafadila','27','CTG000215KL3654','0173926452','2','5002');

insert into Driver_Booking_Details
values('1123','shadhin','29','NK00003214LP697','01823476589','1','5003');

insert into Driver_Booking_Details
values('1124','tanzinabinte','30','BAR001254Z36414','01918726354','2','5004');

insert into Driver_Booking_Details
values('1125','Moinulasajid','25','KHU0125M365V125','01531542741','2','5005');

D_ID	D_NAME	D_AGE	D_LICENSE	D_MOBILE_NUM	D_RATINGS	BOOK_ID
1121	taseen	25	DHK00032C74521V	192374957	1	5001
1122	nuhafadila	27	CTG000215KL3654	173926452	2	5002
1123	shadhin	29	NK00003214LP697	1823476589	1	5003
1124	tanzinabinte	30	BAR001254Z36414	1918726354	2	5004
1125	Moinulasajid	25	KHU0125M365V125	1531542741	2	5005

Query Writing:

Sub Query

1. Display the driver names who have the same ratings as driver taseen.

=>

select d_name from Driver_Booking_Details where d_ratings = (select d_ratings from Driver_Booking_Details where d_name='taseen');

D_NAME
taseen
shadhin

2. Display the vehicle model which has the same status as Toyota SUV.

=>

select vt_model from Vehicle_Type where vt_status = (select vt_status from Vehicle_Type where vt_model='Toyota SUV');

VT_MODEL
Toyota SUV
BMW SEDEN
Mazda MX5

Joining query

1. Write a query to display the customer name, customer mobile number, and email address for all employees.

```
select Customer_User.c_name, Customer_User.c_mobile_num, Users.u_email
from Customer_User,Users
where Customer_User.u_id=Users.u_id;
```

C_NAME	C_MOBILE_NUM	U_EMAIL
Labiba Ibnat	1715043631	labibaibnat17@gmail.com
Rubyea zannat	1915043678	rubyeazannat2833@gmail.com
khaled Hossain	1815043645	khaled0207@gmail.com
Tanzim Idris	1515043666	tanzimidrees@gmail.com
Moinul Islam	1615043678	Moinulislam@gmail.com

2. Display the Customer names and Customer id of only those Customer whose booking amount is greater than 3000.

```
Select Customer_Booking_Details.c_name,Customer_Booking_Details.c_id,
Booking_Details.book_date
from Customer_Booking_Details,Booking_Details
where Customer_Booking_Details.book_id=Booking_Details.book_id and
Booking_Details.book_amount>'3000';
```

C_NAME	C_ID	BOOK_DATE
Labiba Ibnat	1001	14/8/2021
khaled Hossain	1003	12/8/2021
Moinul Islam	1005	13/8/2021

View

1. Create a view called Customer_User101view based on customer name and monile number from Customer_User table.

=>

```
CREATE VIEW Customer_User101view
  AS SELECT  c_name,c_mobile_num
  FROM Customer_User;
```

2. Display the contents of the Customer_User101view view.

=>

```
select * from Customer_User101view;
```

C_NAME	C_MOBILE_NUM
Labiba Ibnat	1715043631
Rubyea zannat	1915043678
khaled Hossain	1815043645
Tanzim Idris	1515043666
Moinul Islam	1615043678

3. Create a view called VehicleType102View based on the model and vehicle id from the VehicleType table.

=>

```
CREATE VIEW  VehicleType102View
  AS SELECT  vt_model,v_id
  FROM       Vehicle_Type;
```

4. Display the contents of the VehicleTypeView view.

=>

select * from VehicleType102View;

VT_MODEL	V_ID
Toyota SUV	1111
BMW SEDEN	1112
Toyota Hilux	1113
Mazda MX5	1114
AUDI A8	1115

Relational Algebra:

1. Find the name of the customer whose user id is 103?
   $\pi_{customer\_name}(\sigma_{user\_id="103"}(\boldsymbol{Customer\_user}))$

2. Find the book id where the booking amount is greater than $4000?
   $\pi_{book\_id}(\sigma_{book\_amount>4000}(\boldsymbol{Booking\_Details}))$

3. Find the driver's name and mobile number where the driver's age is less than 29?
   $\pi_{driver\_name,driver\_mobile\_num}(\sigma_{driver\_age<29}(\boldsymbol{Driver\_user}))$

4. Find the feedback of the service where service id is 2004?
   $\pi_{service\_feedback}(\sigma_{service\_id="2004"}(\boldsymbol{Service}))$

5. Find the vehicle number plate and vehicle id where vehicle status is available?
   $\pi_{vehicle\_numplate,vehicle\_id}(\sigma_{vehicle\_status="available"}(\boldsymbol{Vehicle\_Type}))$

Conclusion:

The Car Ride-Hailing Management System project was done with the provided guidelines. During the project work, we learned the way to implement our theoretical knowledge into a real-life structure. We made a mind map where we structured our scenario, and following that, we drew the Entity-relationship model. After that, we normalized and got our final table to implement it on a database. We drew the schema diagram to visualize the tables and relations in a graphical format. Then we created tables, manipulated the attributes where needed, and inserted data. Additionally, we learned important topics such as subqueries, joining, view, user creation, role creation, and most importantly, relational algebra, which is language independent.

We tried to make the Car Ride-Hailing Management System in the best possible way so that this management system can be used as a role model in the city area of most of the countries in the world. At the same time, there are some lacking's such as we could include entities such as discount, offer, emergency security, VIP service, and maybe more. We faced several problems during the work period, but we solved them together in a group discussion where we got a clear view of every topic and the foundation of the database. Group work has helped us to learn to co-operate with each other more efficiently. We hope in the future; we will perform more efficiently while doing this type of project using our learning and experience. As a whole, this project has helped us to learn, demonstrate and apply the theoretical knowledge to solve real-world problems with the help of Oracle Database.