

# Table of contents:

Overview:.....	- 3 -
Project Scope & Description:.....	- 3 -
Short presentation of our proproject: .....	- 4 -
Requirements: .....	- 4 -
User Transactions: .....	- 7 -
ER Digram .....	- 9 -
Database Schema .....	- 10 -
Creating Tables: .....	- 11 -
Constraint: .....	- 14 -
Insertion: .....	- 16 -
Our project answered these question (Query) :.....	- 18 -
Normalization: .....	- 22 -
Conclusion: .....	- 22 -
References :.....	- 23 -

# CAR RENTAL SYSTEM

## Overview:

The primary goal of this project is to provide an easy and user-friendly software to manage a Car Rental system.

Management decisions need to be based on information of the system. While there may be many efforts to monitor, and manage the company, a system wide overview of the information available is essential for management of the system as whole unit. Collecting all of the relevant information, and compiling it into one easily accessible database, is the goal of the Car Rental System.

The design of the database is based on the information and requirements that is to be collected, or has been collected in the past.

## Project Scope & Description:

The Car Rental Management System is a project for the Database Management System Course at Arab American University.

- The design of the database is based on information that is to be collected, or has been collected in the past.
- This design is to allow easy data entry, access and querying to the complex entities and data.

The driving philosophy behind the database design is to have an efficient database that would be easy to maintained and expanded, as well as allow easy data entry and access.



## **Short presentation of our project:**

We have decided to deliver a Car Rental system. In our framework, Customer can lease a vehicle dependent on brand and model. Our framework gives client to have distinctive get and drop-off areas and will force late expense if the rental vehicle is returned past the return date and time. Clients who have membership will be given a rebate in their bill.

## **Requirements:**

1. car rental office ought to have assortment of car . Every car ought to have a place with a specific Car Category and every car will have a place with a particular location, and has colour, brand, model ,date of procurement ,plate number ,VIN, remark of outside , comment of return.
2. car category incorporates classification number, cost per day, number of luggages, number of individual.
3. Customer can be an individual from the rental company or a nonmember ,if a member, he/she has an enrollment ID, salary and occupation title , every client dependent on his name, ID, cell phone, address (city ,road number), driving permit number and vehicle classification inclinations , rents one or more car.
4. list of car accessible to lease will be appeared for the current day alongside accessible date and time (from and to).
5. Customer will choose a vehicle from the proposals and ought to be able to reserve it for rent.
6. If a client is an individual from the vehicle rental office and has an enrollment ID then he/she will be given a default 20% markdown .

7. Billing is paid (money) as per number of days the client leases a car, and created when a car is returned.

8. At the point when a client restores a vehicle, a bill will be produced on the specific booking. Billing has credits like Bill ID, charge date, charge status (cash or check).

9. Customer can restore the car before the due date, on the due date or he/she can restore it late also, but if a client restores a vehicle after the due date, extra late charge is determined and added to the bill.

10. A default 7% expense is applied on the sum which is incorporated inside the late charge.

11. When the vehicle is returned it opens up for the booking.

12. Every vehicle reservation will be observed in the element called booking. Booking will have attributes like booking id, from date and time of booking and due return date and time and actual return date and time of the booking, and booking status. This booking amount might also include rental insurance and discount code.

13. A booking can be dropped until 5 days before the genuine get.

14. Company may have a few discount, discounts incorporate code, start date, end date and types like end of the week markdown, organization rebate, enrollment rebate and so on.

15. Total expense that should be covered = (car cost per day \* number of days vehicle was reserving including overbid days) + total late fee + (cost per day of insurance \* number of days with insurance) – (rebate amount \* number of days car was reserving including overbid days).

note: discount sum = (percentage of discount \* cost every day)

16. Location signifies the pickup and drop off area of the car . Client can get the car from the specific Location and can have same or different drop off location. It has Location id and address(street , city).

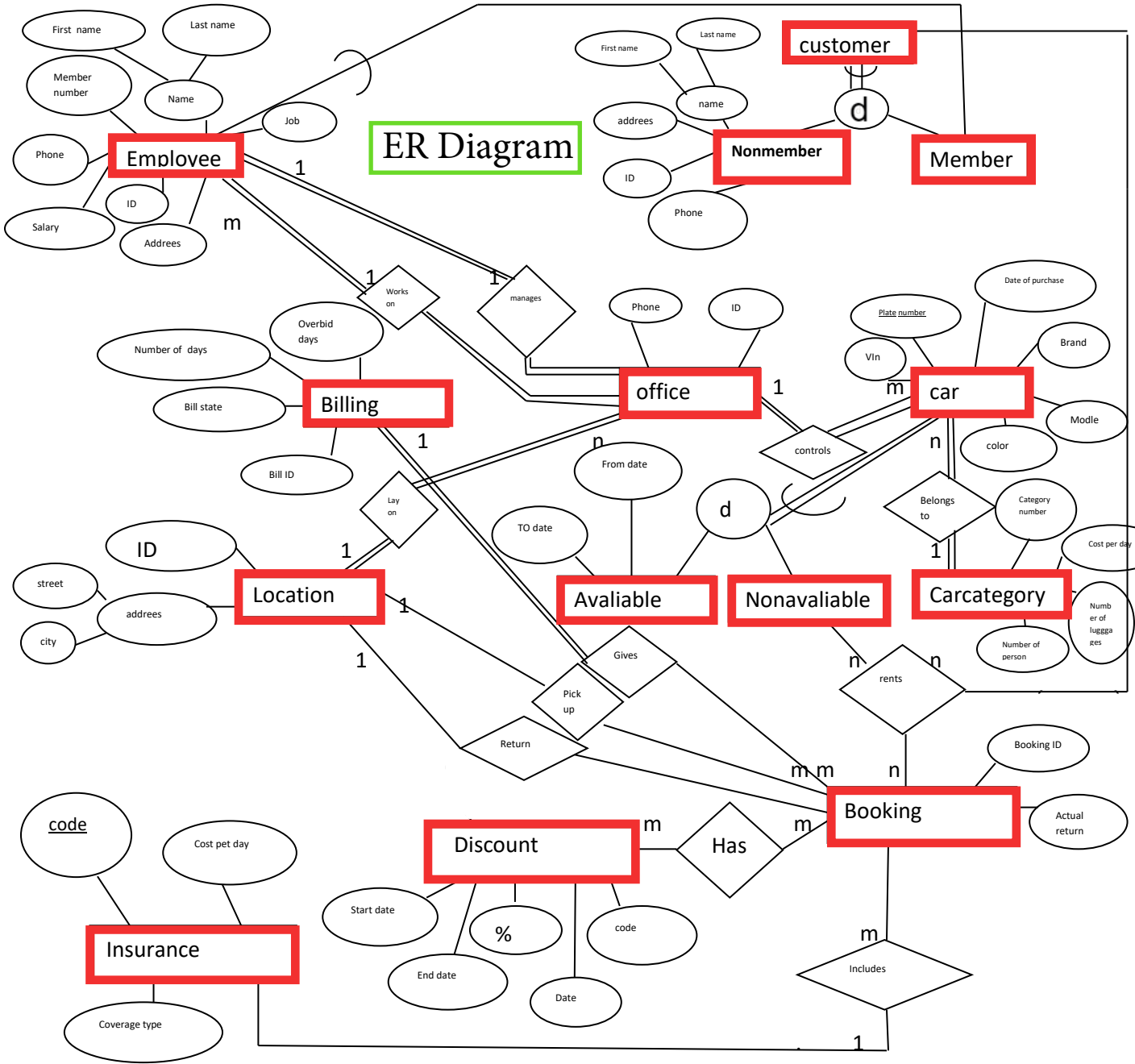
17.car rental company may have many offices each has ID ,location(ID,street,city),phone number .

18. Client can purchase a insurance while booking the car . car rental insurance will have credits like insurance code, inclusion type and cost every day.

## **User Transactions:**

1. Show the available cars with the date they available on .
2. Show the cars that belong to each office.
3. Show information about each employee and the office they work in.
4. Show the manager and location of each office?
5. Is the customer a member of the company or not?
6. What is the car category of a chosen car?
7. Show the non-available cars with the pick-up date , supposed return date and actual return date?
8. Show the bill status and the total amount of the bill for each customer by licence\_id
9. List the licence\_id of customers who insured the cars they rent?
10. show the discounts available for each customer member
11. Show the cars with its insurance and discounts if exist.
12. Show the salary and office ID of employees who is managers

13. Show information of customers who is non member and rented a car in 18/4/2020.
14. Find the average salary for employees who are not customers.
15. Find the employees who work in an office located in Ramallah.
16. Find the cars controlled by office number 2 and available for renting.
17. find the discount code ,percentage and type that the customer which licence # is 4987532 has.
18. Find the coverage type of insurance of car which has plate number 6215894
19. Find information about cars which return on 12/11/2020 in Nablus.
20. Show when to when the car which has plate number 7713863 is available.





# Database Schema

## Employee

f_name	l_name	id	address	phone	<u>membership_id</u>	job	salary	office_id
--------	--------	----	---------	-------	----------------------	-----	--------	-----------

## Office

<u>ID</u>	Phone	Manager_id	Location_id
-----------	-------	------------	-------------

## Location

<u>Id</u>	City	Street
-----------	------	--------

## Customer

<u>License id</u>
-------------------

## Nonmember

F_name	L_name	Address	Phone	<u>Id</u>	License_id
--------	--------	---------	-------	-----------	------------

## Member

<u>Member id</u>	License_id
------------------	------------

## Car

<u>Plate num</u>	VIN	color	brand	model	date_of_purchase	category_num	office_ID
------------------	-----	-------	-------	-------	------------------	--------------	-----------

## Available

To_date	from_date	<u>plate_num</u>
---------	-----------	------------------

## Nonavailable

<u>plate_num</u>
------------------

## Category

<u>Category num</u>	Num_of_persons	Num_of_luggages	cost_per_day
---------------------	----------------	-----------------	--------------

## Rents

To_date	from_date	<u>plate num</u>	<u>book ID</u>	<u>license id</u>
---------	-----------	------------------	----------------	-------------------

## Booking

<u>Booking id</u>	Actual_return_date	Insurance_code	Pick_id_location	Return_id_location	Bill_id
-------------------	--------------------	----------------	------------------	--------------------	---------

## Insurance

<u>Code</u>	Coverage_type	Cost_per_day
-------------	---------------	--------------

## Billing

<u>Bill id</u>	Bill_date	Bill_status	Num_days	Overbid_days
----------------	-----------	-------------	----------	--------------

## Discount

<u>Code</u>	Type	Percentage	Start_date	End_date
-------------	------	------------	------------	----------

## Has\_Discount

<u>Discount_Code</u>	<u>Booking_id</u>
----------------------	-------------------

## Creating Tables:

```
Create table employee (  
  f_name varchar (20) not null,  
  l_name varchar (20) not null,  
  id char(9) unique not null,  
  address varchar (30),  
  phone char(10) not null,  
  membership_id number primary key,  
  job varchar (15),  
  salary decimal (10,2) not null,  
  office_id number (9) not null);
```

```
Create table office (  
  ID number(9) primary key,  
  Phone char(10) not null,  
  Manager_id number not null,  
  Location_id number(9) not null);
```

```
Create table location(  
  Id number(9) primary key,  
  City varchar(20) not null,  
  Street varchar(20) not null);
```

```
Create table customer (  
  License_id char(7)primary key);
```

```
Create table member(  
  Member_id number primary key,  
  License_id char(7) not null);
```

```
Create table nonmember(  
  F_name varchar(20) not null,  
  L_name varchar(20) not null,  
  Address varchar(30),  
  Phone char(10) not null,  
  Id char(9) primary key,  
  License_id char(7) not null);
```

```
Create table car(  
Plate_num char(7) primary key,  
VIN char(17) unique,  
color varchar(15),  
brand varchar(15),  
model number,  
date_of_purchase date,  
category_num number,  
office_ID number(9) not null);
```

```
Create table nonavailable (plate_num char(7) primary key);
```

```
Create table available (To_date date ,  
from_date date,  
plate_num char(7) primary key);
```

```
Create table rents (To_date date not null ,  
from_date date not null,  
plate_num char(7),  
book_ID varchar(9),  
license_id char(7) ,  
primary key(plate_num, book_id, license_id));
```

```
Create table category (Category_num number primary key,  
Num_of_persons number ,  
Num_of_luggages number,  
cost_per_day decimal not null);
```

```
Create table booking (  
Booking_id varchar(9) primary key,  
Actual_return_date date not null ,  
Insurance_code varchar(20),  
Pick_id_location number(9),  
Return_id_location number(9),  
Bill_id varchar(9));
```

```
Create table insurance(  
Code varchar(20) primary key,  
Coverage_type varchar(30),  
Cost_per_day decimal not null);
```

```
Create table billing(  
Bill_id varchar(9) primary key,  
Bill_date date ,  
Bill_status varchar(15),  
Num_days number not null,  
Overbid_days number );
```

```
Create table discount(  
Code varchar(20) primary key,  
Type varchar (30),  
Percentage decimal(5,2) not null,  
Start_date date,  
End_date date);
```

```
Create table has_discount (  
Discount_Code varchar(20),  
Booking_id varchar(9),  
Primary key(discount_code,booking_id));
```

## **Constraint:**

Alter table employee add constraint c1 foreign key (office\_id) references office (id) on delete set null;

Alter table office add constraint c2 foreign key (manager\_id) references employee (membership\_id) on delete set null;

Alter table office add constraint c3 foreign key (location\_id) references location (id) on delete set null;

Alter table member add constraint c4 foreign key (member\_id) references employee (membership\_id) on delete set null;

Alter table member add constraint c5 foreign key (license\_id) references customer (license\_id) on delete set null;

Alter table nonmember add constraint c6 foreign key (license\_id) references customer (license\_id) on delete set null;

Alter table car add constraint c7 foreign key (category\_num) references category (category\_num) on delete set null;

Alter table car add constraint c8 foreign key (office\_id) references office (id) on delete set null;

Alter table nonavailable add constraint c9 foreign key (plate\_num) references car (plate\_num) on delete set null;

Alter table available add constraint c10 foreign key (plate\_num) references car (plate\_num) on delete set null;

Alter table rents add constraint c11 foreign key (plate\_num) references nonavailable (plate\_num) on delete set null;

Alter table rents add constraint c12 foreign key (book\_id) references booking (booking\_id) on delete set null;

Alter table rents add constraint c13 foreign key (license\_id) references customer (license\_id) on delete set null;

Alter table booking add constraint c14 foreign key (insurance\_code) references insurance (code) on delete set null;

Alter table booking add constraint c15 foreign key (pick\_id\_location) references location (id) on delete set null;

Alter table booking add constraint c16 foreign key (return\_id\_location) references location (id) on delete set null;

Alter table booking add constraint c17 foreign key (bill\_id) references billing (bill\_id) on delete set null;

Alter table has\_discount add constraint c18 foreign key (discount\_code) references discount (code) on delete set null;

Alter table has\_discount add constraint c19 foreign key (booking\_id) references booking (booking\_id) on delete set null;

## Insertion:

insert into employee values ('Soltan', 'Alshaikh', '147258369','Nablu', '0599248862',  
16, 'Accountant', 7624, 1);

Insert into employee values ( 'Wissam', 'Alahmad', '741852963', 'Hebron' ,  
'0594621786' , 13, 'Receptionist', 3496, 2);

insert into employee values ('Anas', 'Ibrahim', '159753654', 'Ramallah', '0598621852',  
19, 'Programmer', 2891, 3);

Insert into office values(1, '0598425896', 16, 5);

Insert into office values(2, '0599619375', 13, 6);

Insert into office values( 3, '0592761486', 19, 7);

Insert into location values ( 5 , 'Jenin' , 'AbuBaker street');

Insert into location values ( 6 , 'Nablu', 'Academy street');

Insert into location values ( 7, 'Ramallah' , 'Tira street');

Insert into customer values ( '9647531');

Insert into customer values ('5486215');

Insert into customer values ( '4987532');

Insert into member values(16, '4987532');

Insert into nonmember values( 'Islam','Mousa','Jenin', '0592331594' , '456321897',  
'9647531');

Insert into nonmember values('Moath','Ali','Hebron', '0564561893' , '556674219' ,  
'5486215');

insert into car values( '6215894','DFHW5498KLN78541',  
'Grey','GMC',2020,to\_date('19/10/2020','dd/mm/yyyy'),3, 1);

insert into car values( '3578426','LACK8521WER15943','Black','SKODA  
Superb',2019, to\_date('26/11/2020','dd/mm/yyyy'),2, 2);

insert into car values( '4569812','OPDF3791FVT64825','White','Seat Leon',2021,  
to\_date('22/1/2021','dd/mm/yyyy'),1, 3);

insert into car values( '7713863','DFHW5488KLJ78541', 'Red','Rang  
rover',2021,to\_date('19/10/2021','dd/mm/yyyy'),3, 1);

insert into category values(1,8,4,80);

insert into category values(2,6,5,90);

insert into category values(3,7,3,70);

```

insert into rents values
(to_date('10/11/2020','dd/mm/yyyy'),to_date('8/11/2020','dd/mm/yyyy'), '6215894',
'47', '9647531');
insert into rents values(to_date('20/4/2020','dd/mm/yyyy'),to_date('18/4/2020'
,'dd/mm/yyyy'), '3578426','48', '5486215');
insert into rents values(to_date('7/3/2021','dd/mm/yyyy'),
to_date('2/3/2021','dd/mm/yyyy') , '4569812' , '49', '4987532');

```

```

insert into nonavailable values('6215894');
insert into nonavailable values('3578426');
insert into nonavailable values('4569812');

```

```

insert into available values(to_date('20/12/2020','dd/mm/yyyy'),
to_date('22/10/2020','dd/mm/yyyy'),'7713863');

```

```

Insert into booking values('47',to_date('12/11/2020','dd/mm/yyyy'),'20', 5, 6,'26');
Insert into booking values('48',to_date('22/4/2020','dd/mm/yyyy'),'21', 6, 7,'27');
Insert into booking values('49', to_date('9/3/2021','dd/mm/yyyy'),'22', 7, 7 , '28');

```

```

Insert into insurance values('20','Liability',15);
Insert into insurance values('21','Medical payments',20);
Insert into insurance values('22','Collision' ,25);

```

```

Insert into billing values('26', to_date('12/11/2020','dd/mm/yyyy'),'Cash',2,0);
Insert into billing values('27', to_date('23/4/2020','dd/mm/yyyy'),'Cheques',6,2);
Insert into billing values('28', to_date('9/3/2021','dd/mm/yyyy'),'Credit card',7,3);

```

```

Insert into discount values ('8f6j191','Membership',0.2 ,null,null);
Insert into discount values
('9r3y61tk','Weekend',0.7,to_date('8/11/2020','dd/mm/yyyy'),
to_date('10/11/20','dd/mm/yyyy'));
Insert into discount values ('9518hzu','Company_Discount',0.3,
to_date('18/4/2020','dd/mm/yyyy'),to_date('20/4/2020','dd/mm/yyyy'));

```

```

Insert into has_discount values('8f6j191', '49');
Insert into has_discount values('9518hzu', '48');
Insert into has_discount values('9r3y61tk', '47');

```



## **Our project answered these question (Query) :**

1. Show the available cars with the date they available on .

```
Select plate_num,from_date ,To_date  
from Available;
```

2. Show the cars that belong to each office.

```
Select office_ID,Plate_num  
from Car  
order by office_ID ;
```

3. Show information about each employee and the office they work in.

```
Select membership_id ,f_name ,l_name , salary ,office_id  
from Employee;
```

4. Show the manager and location of each office?

```
Select s.id ,e.f_name as Manager ,l.City, l.Street  
from employee e join Office s on (s.Manager_id=e.membership_id)  
Join Location l on(l.ID =s.Location_id);
```

5. Is the customer a member of the company or not?

```
Select Member_id, License_id  
from Customer natural join Member ;
```

6. What is the car category of a chosen car?

```
Select Plate_num ,Category_num ,Num_of_persons, Num_of_luggages ,cost_per_day  
from Car natural join Category  
where Plate_num = 6215894;
```

7. Show the non-available cars with the pick-up date , supposed return date and actual return date?

```
Select r.plate_num,r.from_date,r.To_date,b.Actual_return_date  
from Rents r join Booking b  
on (r.book_ID=b.Booking_id);
```

8. Show the bill status and the total amount of the bill for each customer by licence\_id  
?

```
Select r.license_id,l.Bill_status ,
((y.cost_per_day*I.Num_days)
+(y.cost_per_day*1.07 *I.Overbid_days))
+(i.Cost_per_day*(I.Num_days+I.Overbid_days))
-(d.Percentage*y.cost_per_day*(I.Num_days+I.Overbid_days)) as "total amount of the
bill"
from rents r Join Booking b on(r.book_ID = b.Booking_id)
Join Billing l on (b.Bill_id=l.Bill_id)
Join Car c on (r.Plate_num =c.Plate_num)
Join Category y on (c.Category_num=y.Category_num)
LEFT OUTER JOIN Insurance i on (i.Code=b.Insurance_code)
Join has_discount h on(h.booking_id=b.booking_id)
Join discount d on (d.code=h.discount_code)
order by r.license_id ;
```

9. List the licence\_id of customers who insured the cars they rent?

```
Select r.license_id,i.Coverage_type
from rents r Join Booking b on(r.book_ID = b.Booking_id)
JOIN Insurance i on (i.Code=b.Insurance_code);
```

10. show the discounts available for each customer member

```
Select e.f_name,m.license_id,d.type
from employee e join member m on (e.membership_id=m.member_id)
Join rents r on(m.license_id=r.license_id)
Join Booking o on(r.book_ID = o.Booking_id)
Join has_discount b on(b.booking_id=o.booking_id)
Join discount d on (d.code=b.discount_code);
```

11. Show the cars with its insurance and discounts if exist.

```
select r.plate_num,r.book_id ,i.coverage_type ,d.type from rents r join booking b on
(r.book_id=b.booking_id)
Join insurance l on(i.code=b.insurance_code)
Join has_discount bd on (bd.booking_id=b.booking_id)
Join discount d on (d.code=bd.discount_code);
```

12. Show the salary and office ID of employees who is managers

```
select office_id,salary ,membership_id from employee
where membership_id in(
Select manager_id from office);
```

13. Show information of customers who is non member and rented a car in 18/4/2020.

```
select n.f_name,n.l_name,n.license_id from rents r join customer c
on(c.license_id=r.license_id)
Join nonmember n on(n.license_id=c.license_id)
Where r.from_date = '18-april -2020';
```

14. Find the average salary for employees who are not customers.

```
select membership_id,avg(salary)
from employee
where membership_id not in(
select m.member_id
from member m join customer c
on(c.license_id=m.license_id)) group by membership_id;
```

15. Find the employees who work in an office located in Ramallah.

```
Select e.f_name,e.l_name,l.city
from employee e join office o on (e.office_id=o.id)
Join location l on (l.id=o.location_id)
Where l.city ='Ramallah';
```

16. Find the cars controlled by office number 2 and available for renting.

```
Select c.office_id,c.plate_num ,c.model,c.color
From car c join available a on(c.plate_num=a.plate_num)
Where c.office_id=2;
```

17. find the discount code ,percentage and type that the customer which licence # is 4987532 has.

```
select d.code,d.type ,d.percentage from discount d join has_discount bd
on(bd.discount_code=d.code)
Join booking b on (b.booking_id=bd.booking_id)
Join rents r on (r.book_id=b.booking_id)
Join customer c on (c.license_id=r.license_id)
Where c.license_id ='4987532';
```

18. Find the coverage type of insurance of car which has plate number 6215894

```
select i.Coverage_type
from rents r Join Booking b on(r.book_ID = b.Booking_id)
JOIN Insurance i on (i.Code=b.Insurance_code)
Where r.plate_num = 6215894;
```

19. Find information about cars which return on 12/11/2020 in Nablus.

```
select c.plate_num,c.color,c.model,c.brand,c.category_num
from car c join office o on (c.office_id=o.id)
join location l on (o.location_id=l.id)
join booking b on (b.return_id_location=l.id )
where b.actual_return_date = '12/November/2020' and l.city='Nablus';
```

20. Show when to when the car which has plate number 7713863 is available.

```
select from_date ,To_date
from car c join Available a on (c.plate_num=a.plate_num)
where c.Plate_num ='7713863';
```

## **Normalization:**

We have built a database that adheres to the correct rules and standards between tables, and we transformed the relationships according to the rules to prevent duplication in tables and make the database more flexible and efficient.

in our project every table doesn't contain multiple values and every nonprime attribute depends on all primary and candidate keys, in addition to no transitive dependencies within them .

so our project are in 1st and 2nd and 3rd normal form , no need to normalization..

## **Conclusion:**

While chipping away at this task, we figured out how to make an database, the principles to build a RE diagram, How to come up with relational schema mapping from the ER diagram, determining the practical conditions and how to normalize the relational schema. . We figured out how to plan a framework from Database point of view and how to proficiently store and control information.

