**DB1 Project 2 - Part 3**

**Car Rental Application**

Karthikeyan Rajamani 1001267157

Vineel Krishna Vellanki 1001228753

[3. Database Transactions 1](#_Toc458650355)

[4. Web based User Interface 5](#_Toc458650356)

[5. Transactions tested in the database 9](#_Toc458650357)

# 3. Database Transactions

3.1. Add information about new Customer : (Procedure: **sp\_InsertCustomer)**

CREATE DEFINER=`db1`@`%` PROCEDURE `sp\_InsertCustomer`(cid int,cname varchar(45),dlno varchar(25), phone varchar(25))

BEGIN

if(cid=0) then

insert into customer(Customer\_Name,Customer\_DlNo,Phone) values (cname,dlno,phone);

else

update customer set Customer\_Name=cname, Customer\_DLNo=dlno, Phone=phone where Customer\_Id=cid;

end if;

END

3.2. Add information about new Car: (Procedure: **sp\_InsertCar)**

CREATE DEFINER=`db1`@`%` PROCEDURE `sp\_InsertCar`(vehicleid int,licenseplateno varchar(45),Model varchar(45), Year int,Dailyrate decimal, Weeklyrate decimal, cartype enum('COMPACT','MEDIUM','LARGE','SUV','TRUCK','VAN'))

BEGIN

DECLARE Vid INT;

if(vehicleid=0) then

begin

insert into car(LicensePlate\_no, Model, Year,Daily\_Rate,Weekly\_Rate, car\_type) values

(licenseplateno,Model,Year,Dailyrate,Weeklyrate, cartype);

end;

else

update car set LicensePlate\_no= licenseplateno,Model= Model,Year=Year,Daily\_Rate=Dailyrate,

Weekly\_Rate=Weeklyrate,car\_type=cartype where Vehicle\_id=vehicleid;

end if;

END

A trigger is written to add the newly entered car as available for rent.

CREATE DEFINER=`db1`@`%` TRIGGER `rentals`.`car\_AFTER\_INSERT` AFTER INSERT ON `car` FOR EACH ROW

BEGIN

declare vid int;

SET @vid = (SELECT max(Vehicle\_Id) from car);

insert into car\_availability(Vehicle\_Id,Period\_Id,Avail\_Flag) values (@vid,1,'Y');

insert into car\_availability (Vehicle\_Id,Period\_Id,Avail\_Flag) values (@vid,2,'Y');

insert into car\_availability (Vehicle\_Id,Period\_Id,Avail\_Flag) values (@vid,3,'Y');

insert into car\_availability (Vehicle\_Id,Period\_Id,Avail\_Flag) values (@vid,4,'Y');

insert into car\_availability (Vehicle\_Id,Period\_Id,Avail\_Flag) values (@vid,5,'Y');

insert into car\_availability (Vehicle\_Id,Period\_Id,Avail\_Flag) values (@vid,6,'Y');

END

We have assumed that all cars initially are available for a period of 3months. The three months are separated as six periods with 15days for each period.

3.3. Add information about new Rental: (Procedure: **sp\_** **GetAvailCardet, sp\_InsertRentals )**

To fetch available cars for a specified cartype and start date:

CREATE DEFINER=`db1`@`%` PROCEDURE `sp\_GetAvailCardet`(cartype enum ('COMPACT','MEDIUM','LARGE','SUV','TRUCK','VAN'),renttype char(1), stdate date,daysorweeks int)

BEGIN

declare stperiodid int;

declare endperiodid int;

declare rtdate date;

if(renttype='D') then

begin

set @stperiodid=(select period\_id from car\_availability\_periods where Stdate between st\_Date and End\_Date);

set @rtdate=(select DATE\_ADD(stdate, INTERVAL daysorweeks DAY));

set @endperiodid=(select period\_id from car\_availability\_periods where @rtdate between st\_Date and End\_Date);

select distinct c.Vehicle\_Id,replace(c.LicensePlate\_No,' ','') as LicensePlate\_No,c.car\_type,'Daily' as RentalType,replace(c.Model,' ','') as Model,c.Year,c.Daily\_Rate,

stdate, concat (daysorweeks,'','days') as days from

car c, car\_availability a

where c.Vehicle\_Id=a.Vehicle\_Id

and a.Period\_Id in (@stperiodid,@endperiodid)

and a.Avail\_Flag='Y'

and c.car\_Type=cartype;

end;

else

begin

set @stperiodid=(select period\_id from car\_availability\_periods where Stdate between st\_Date and End\_Date);

set @rtdate=(select DATE\_ADD(stdate, INTERVAL daysorweeks WEEK));

set @endperiodid=(select period\_id from car\_availability\_periods where @rtdate between st\_Date and End\_Date);

select distinct c.Vehicle\_Id,replace(c.LicensePlate\_No,' ','') as LicensePlate\_No,c.car\_type,'Weekly' as RentalType,replace(c.Model,' ','') as Model,c.Year,c.Weekly\_Rate,

stdate, concat (daysorweeks, '', 'week') as weeks from

car c, car\_availability a

where a.Vehicle\_Id=c.Vehicle\_Id

and a.Period\_Id in (@stperiodid,@endperiodid)

and a.Avail\_Flag='Y'

and c.car\_Type=cartype;

end;

end if;

END

To insert the rental details after a customer takes a car for rent:

CREATE DEFINER=`db1`@`%` PROCEDURE `sp\_InsertRentals`(customerid int,vehicleid int,dwflag char(1),stdate date, daysorweeks int)

BEGIN

DECLARE rtdate date;

DECLARE amtdue decimal(10,2);

declare dailyrate decimal(10,2);

declare weeklyrate decimal(10,2);

declare stperiodid int;

declare endperiodid int;

if(dwflag='D') then

begin

set @dailyrate=(select daily\_rate from car where Vehicle\_Id=vehicleid);

set @rtdate=(select DATE\_ADD(stdate, INTERVAL daysorweeks DAY));

set @amtdue=(select @dailyrate\*daysorweeks);

insert into rentals(R\_Customer\_Id,R\_Vehicle\_Id,Location, Daily\_Flag,D\_StDate,

D\_NoofDays,D\_RtDate,D\_AmtDue,Weekly\_Flag)

values(customerid,vehicleid,'Arlington','Y',stdate,daysorweeks,@rtdate,@amtdue,'N');

set @stperiodid=(select period\_id from car\_availability\_periods where stdate between st\_Date and End\_Date);

set @endperiodid=(select period\_id from car\_availability\_periods where @rtdate between st\_Date and End\_Date);

update car\_availability set Avail\_Flag='N' where Period\_Id in (@stperiodid,@endperiodid) and Vehicle\_Id=vehicleid;

end;

else

begin

set @weeklyrate=(select weekly\_rate from car where Vehicle\_Id=vehicleid);

set @rtdate=(select DATE\_ADD(stdate, INTERVAL daysorweeks WEEK));

set @amtdue=(select @weeklyrate\*daysorweeks);

insert into rentals(R\_Customer\_Id,R\_Vehicle\_Id,Location, Daily\_Flag,W\_StDate,

W\_NoofWeeks,W\_RtDate,W\_AmtDue,Weekly\_Flag)

values(customerid,vehicleid,'Arlington','N',stdate,daysorweeks,@rtdate,@amtdue,'Y');

set @stperiodid=(select period\_id from car\_availability\_periods where stdate between st\_Date and End\_Date);

set @endperiodid=(select period\_id from car\_availability\_periods where @rtdate between st\_Date and End\_Date);

update car\_availability set Avail\_Flag='N' where Period\_Id in (@stperiodid,@endperiodid) and Vehicle\_Id=vehicleid;

end;

end if;

END

3.4. Returning back a rented car: (Procedure: **sp\_GetRentals**)

Shows all rented cars:

CREATE DEFINER=`db1`@`%` PROCEDURE `sp\_GetRentals`(vehicleid int,custid int, vfrom char(1))

BEGIN

declare stdate date;

declare enddate date;

declare stperiodid int;

declare endperiodid int;

declare dwflag char(1);

if(vehicleid=0) then

begin

if(vfrom='T') then

select R\_Vehicle\_Id, R\_Customer\_Id,cr.Customer\_Name,cr.Customer\_DLNo, c.Car\_Type as CarType,c.LicensePlate\_No,

'Daily' as RentalType,

D\_StDate as StartDate,concat(D\_NoofDays,' ','days') as Period,

D\_RtDate as ReturnDate,concat('$',D\_AmtDue) as AmtDue

from rentals r, customer cr, car c

where r.R\_Vehicle\_Id=c.Vehicle\_Id and

r.R\_Customer\_Id=cr.Customer\_Id and

r.Daily\_Flag='Y'

union

select R\_Vehicle\_Id, R\_Customer\_Id,cr.Customer\_Name,cr.Customer\_DLNo, c.Car\_Type as CarType,c.LicensePlate\_No,

'Weekly' as RentalType,

W\_StDate as StartDate,concat(W\_NoofWeeks,' ','weeks') as Period,

W\_RtDate as ReturnDate,concat('$',W\_AmtDue) as AmtDue

from rentals r, customer cr, car c

where r.R\_Vehicle\_Id=c.Vehicle\_Id and

r.R\_Customer\_Id=cr.Customer\_Id and

r.Weekly\_Flag='Y'

order by Customer\_Name;

end if;

end;

The returned car details are deleted from the rentals table and moved to the rentalshistory table and the cars are available for rent again.

begin

set @stdate=(select W\_Stdate from rentals where R\_Vehicle\_Id=vehicleid and R\_Customer\_Id=custid);

set @enddate=(select W\_Rtdate from rentals where R\_Vehicle\_Id=vehicleid and R\_Customer\_Id=custid);

set @stperiodid=(select period\_id from car\_availability\_periods where @stdate between St\_Date and End\_Date);

set @endperiodid=(select period\_id from car\_availability\_periods where @enddate between St\_Date and End\_Date);

insert into rentalshistory

select rentals.\* from rentals where R\_Vehicle\_Id=vehicleid and R\_Customer\_Id=custid;

delete from rentals where R\_Vehicle\_Id=vehicleid and R\_Customer\_Id=custid;

update car\_availability set Avail\_Flag='Y' where Vehicle\_Id=vehicleid and Period\_Id in (@stperiodid,@endperiodid);

select R\_Vehicle\_Id, R\_Customer\_Id,cr.Customer\_Name,cr.Customer\_DLNo, c.Car\_Type as CarType,c.LicensePlate\_No,'Daily' as RentalType,D\_StDate as StartDate,concat(D\_NoofDays,' ','days') as Period,D\_RtDate as ReturnDate,concat('$',D\_AmtDue) as AmtDue

from rentals r, customer cr, car c

where r.R\_Vehicle\_Id=c.Vehicle\_Id and r.R\_Customer\_Id=cr.Customer\_Id and r.Daily\_Flag='Y'

end;

3.5. To update the daily rate and weekly rate of any car.

update car set LicensePlate\_no= licenseplateno,Model= Model,Year=Year,Daily\_Rate=Dailyrate,

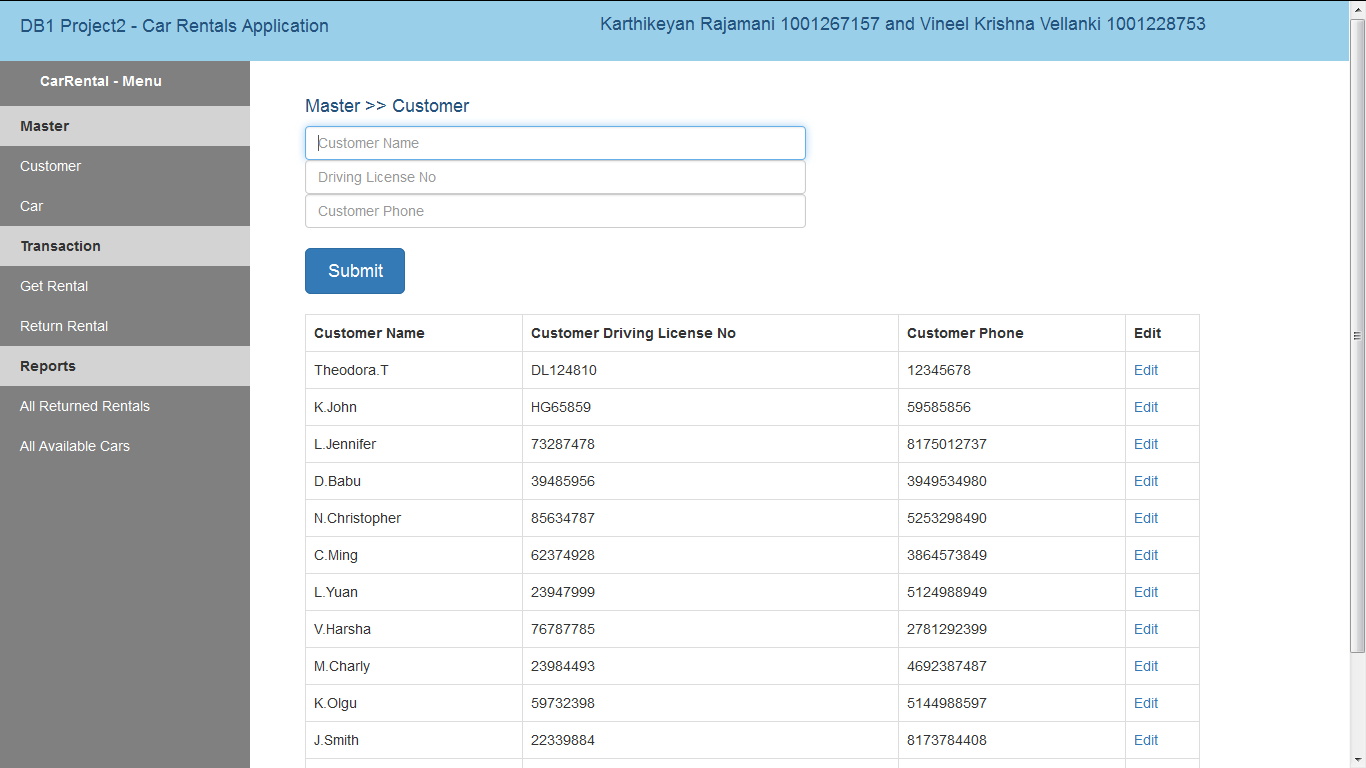
Weekly\_Rate=Weeklyrate,car\_type=cartype where Vehicle\_id=vehicleid;

# 4. Web based User Interface

The interface is designed in Python for all adding Car, Customer details and Rental Transactions.

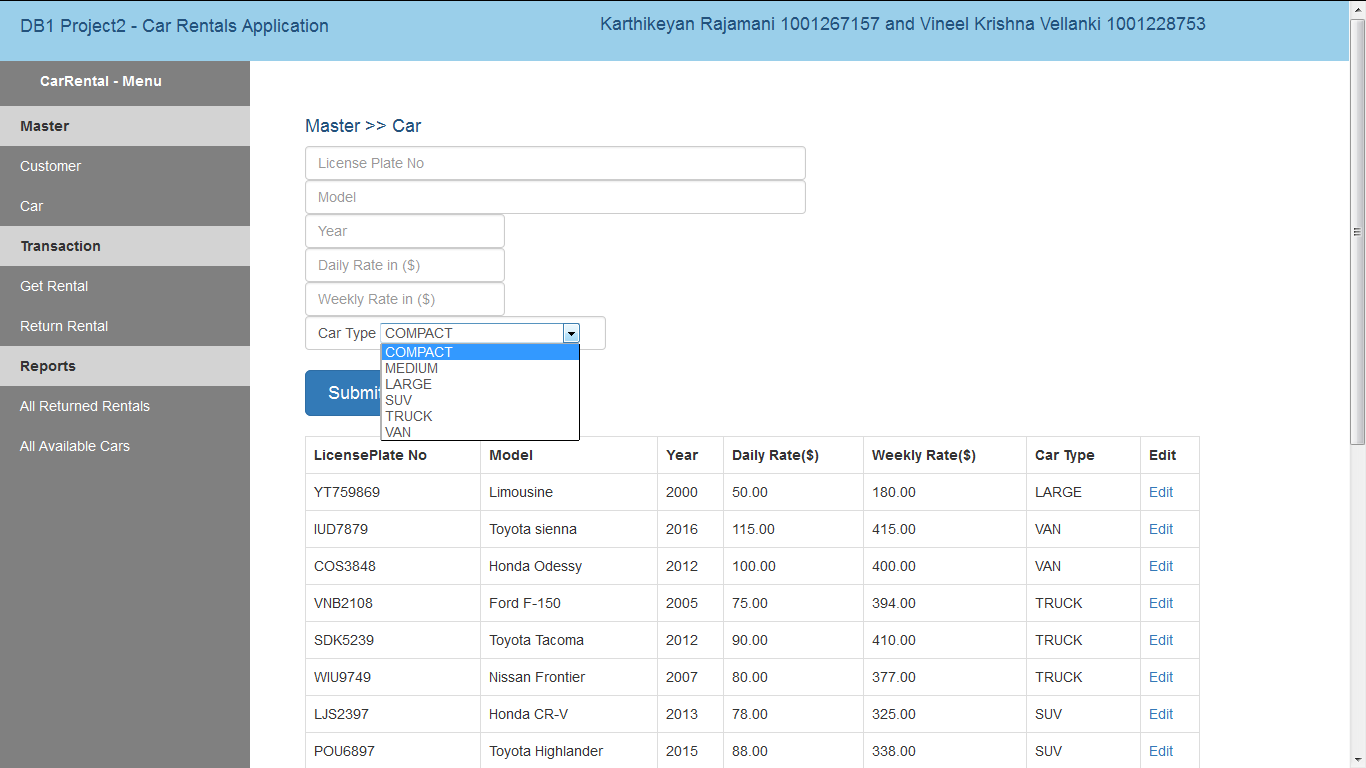
a). Add information about new Customer:

Customer Details like Customer Name, Customer DLNo and Phone can be added. The same details can also be edited when required.



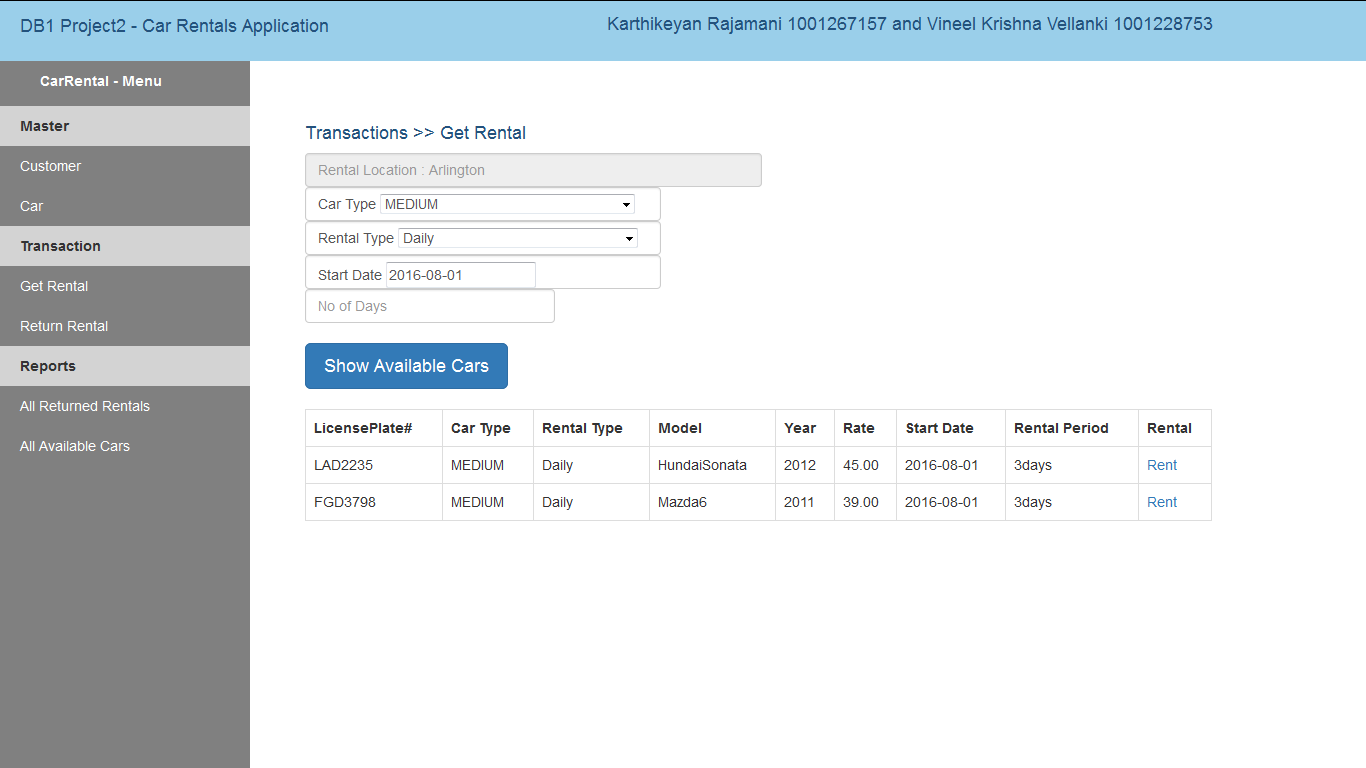
b). Add information about new Car:

Car Details like LicensePlateNo, Model, Year, Daily Rate, Weekly Rate and Car Type can be added. The Car Type can be any one in COMPACT, MEDIUM,LARGE,SUV,TRUCK,VAN .The same details can also be edited when required.

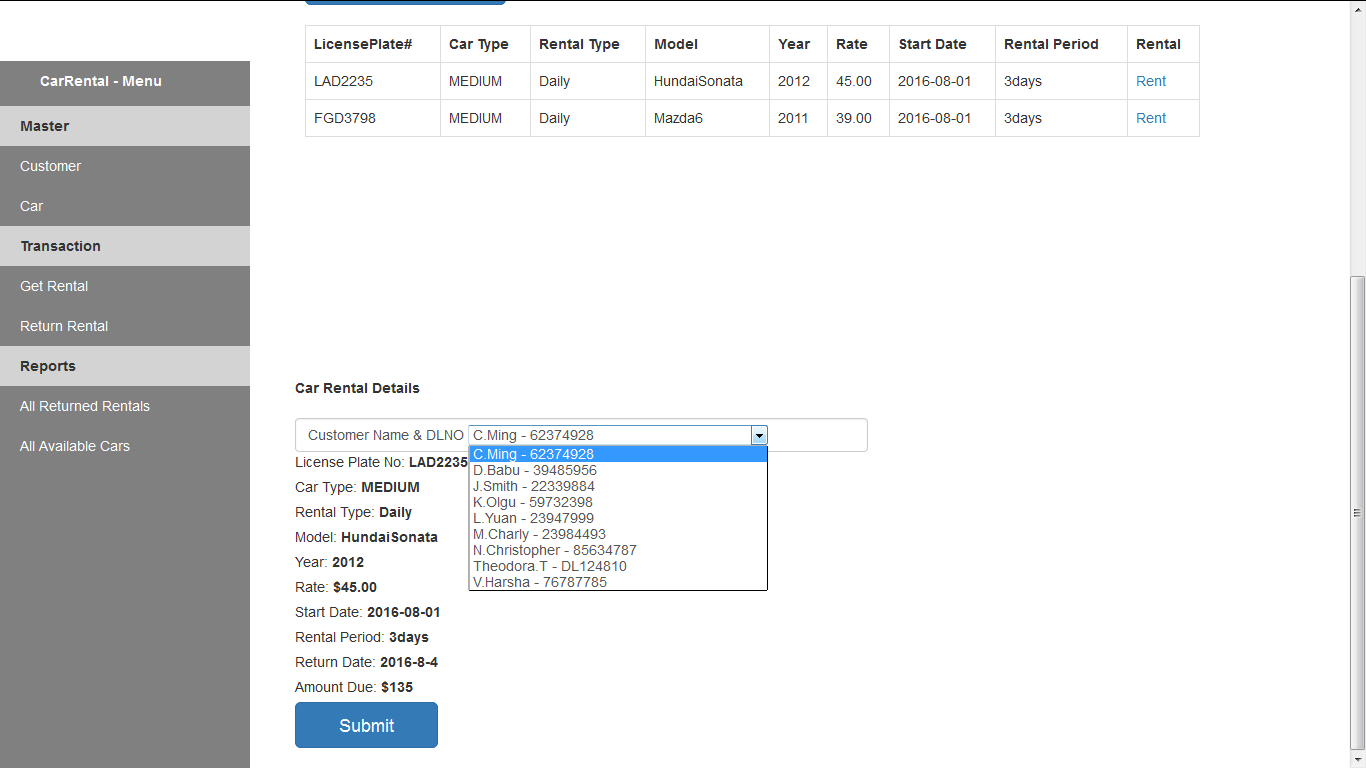


c). Adding information about New Rental:

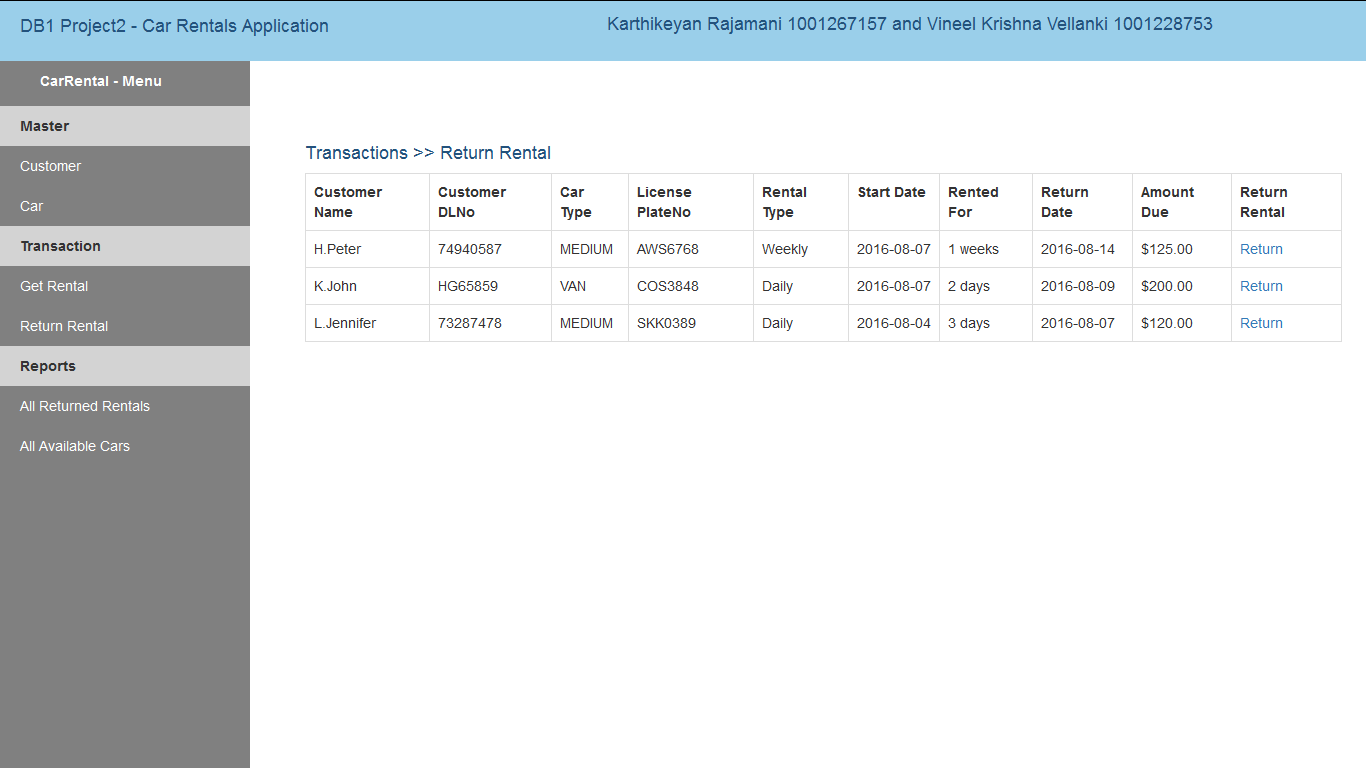
A new rental reservation can be added, by choosing a car type, Rental Type(Daily or Weekly), Start Date, and the NoofDays or Noof Weeks, based on the Rental Type. This transaction first shows the list of available cars with their rental rates and vehicle details; for the specified car type and start date.



The required vehicle can be rented using the “Rent” link, which shows the details of the vehicle with the return date and amount. A customer must be chosen who is renting the car and the rental details are saved by clicking the Submit button.



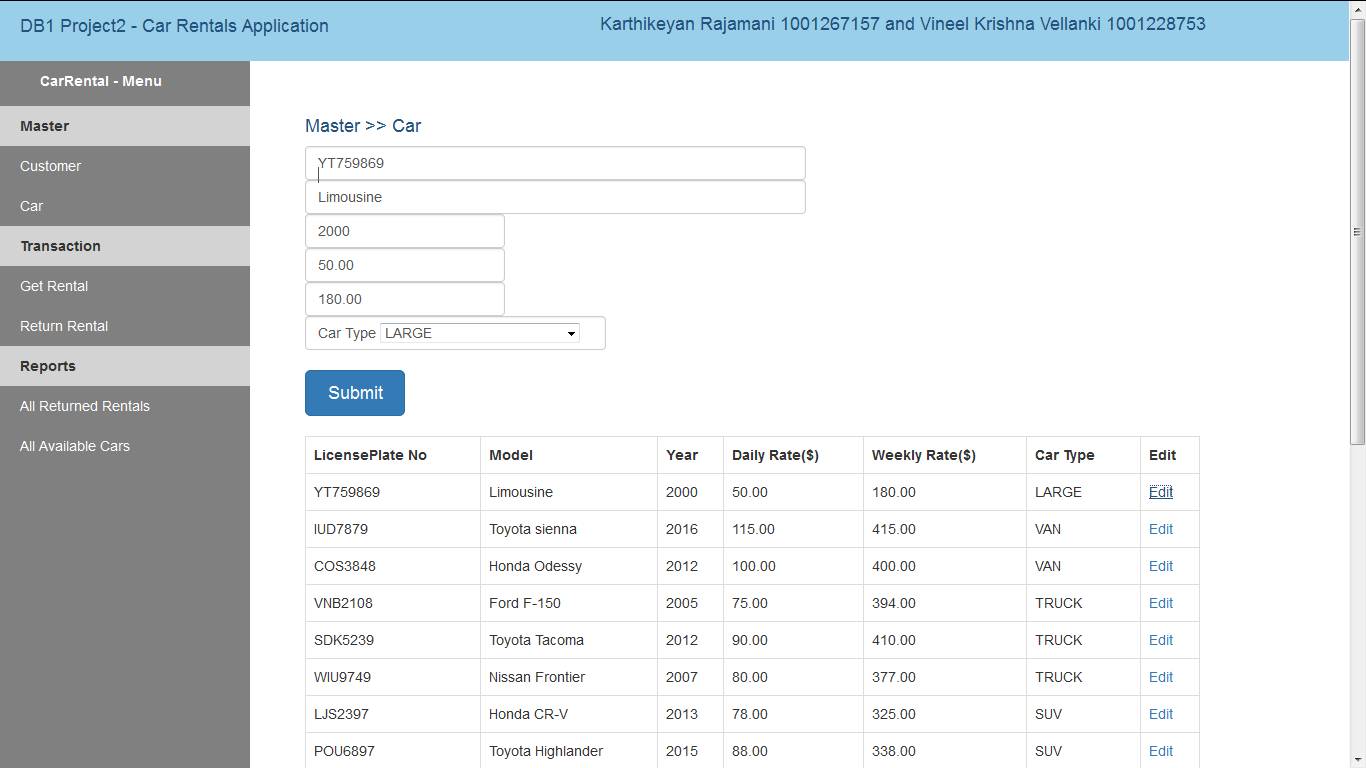
d) Return of a rental car:



All the rented vehicles will appear under this transaction. This shows the details of the rented car, customer, rental period and the amount due. The car is returned back using “Return” link and is available for other customers to rent.

e). Update rental rates:

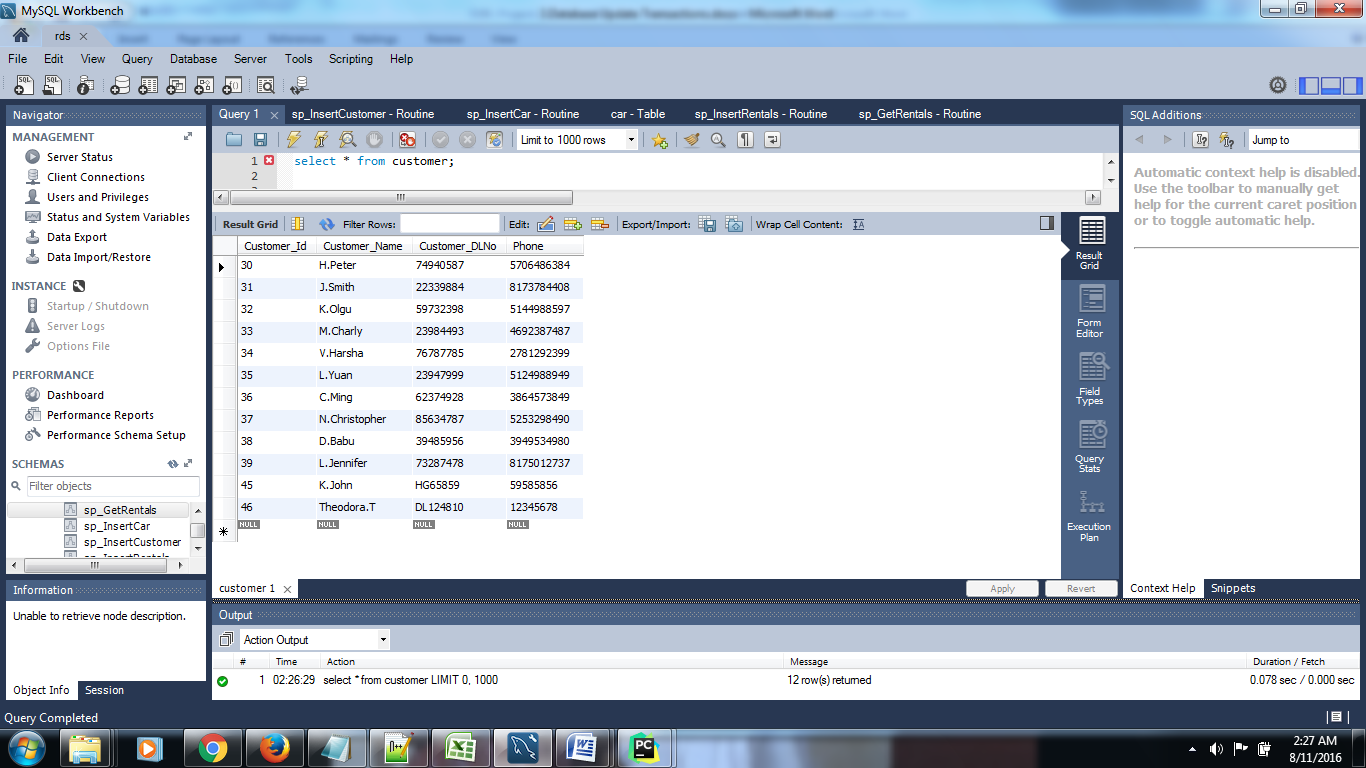
The rental rates(both Daily and Weekly) can be edited for any car at any time.



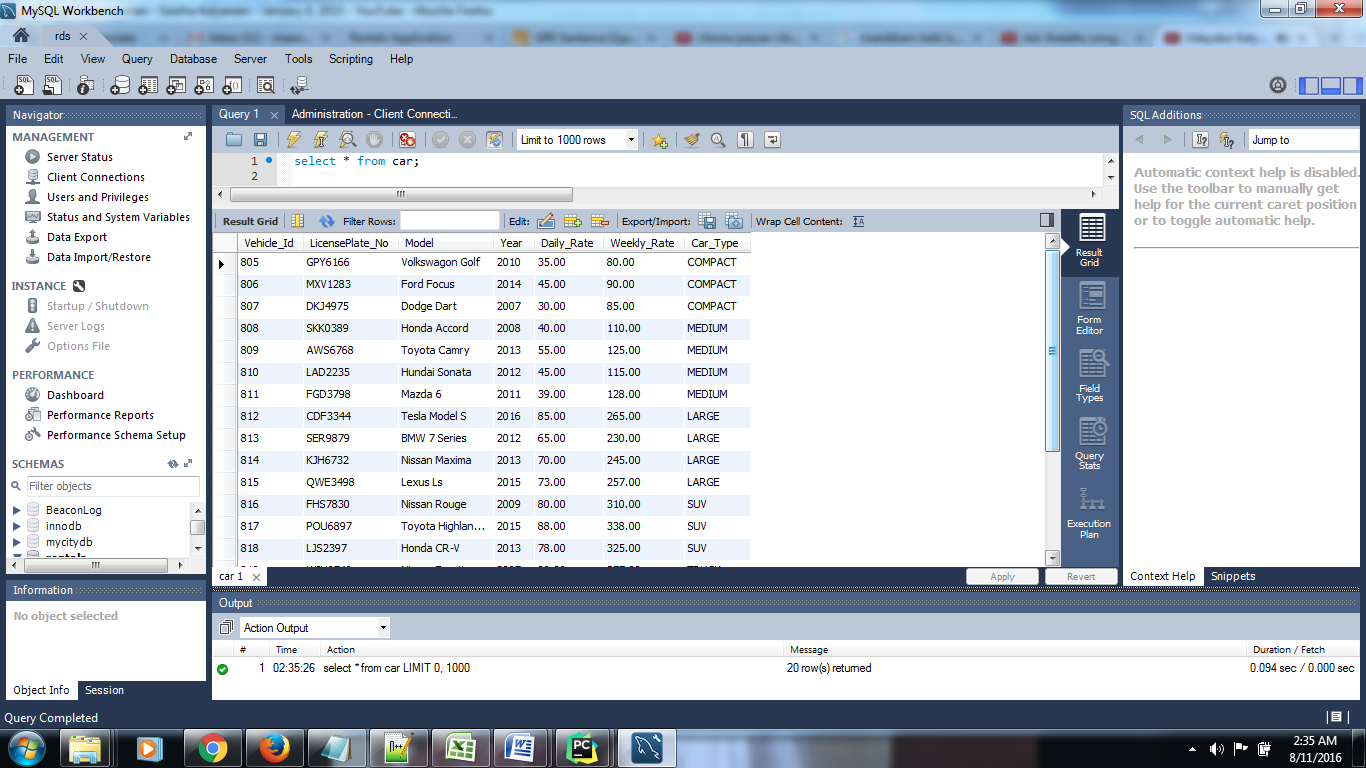
The “Edit” link populates the fields in the UI and their details can be edited.

# 5. Transactions tested in the database

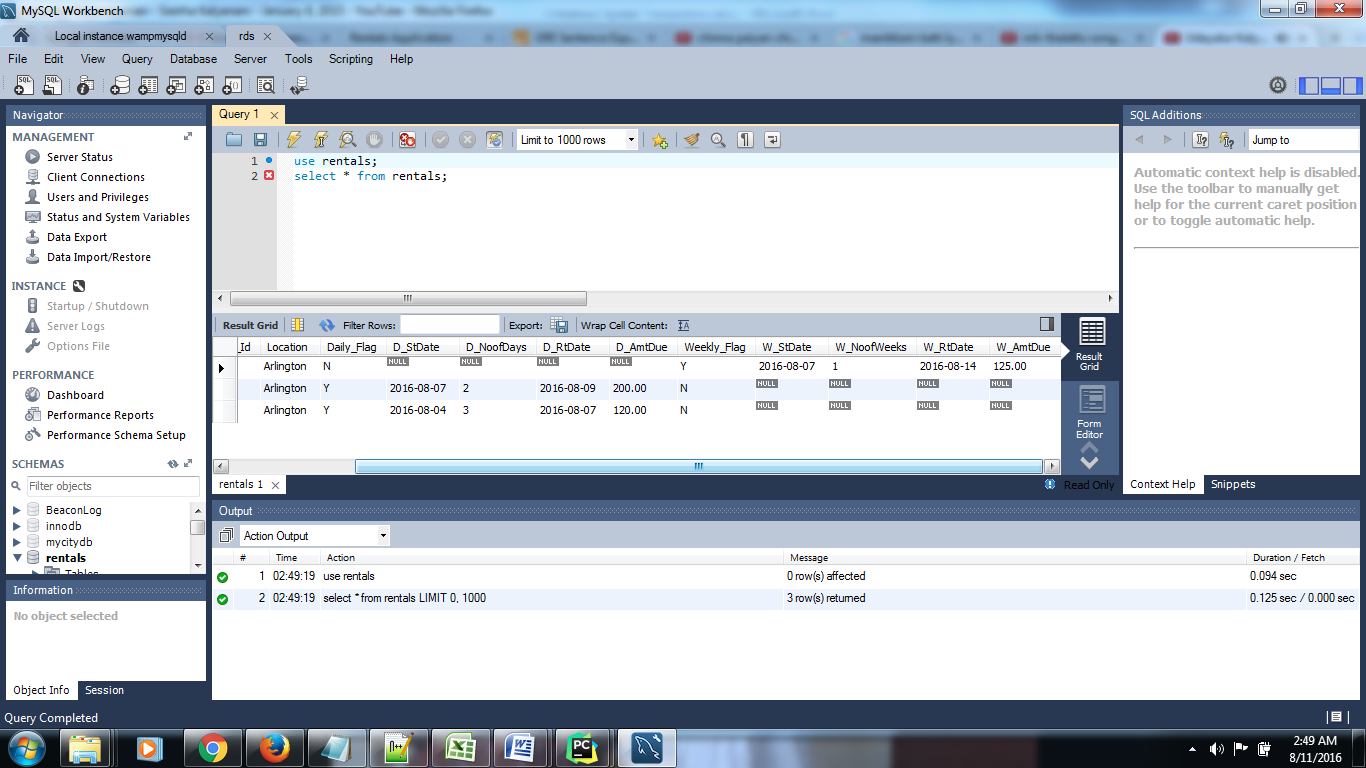
a) Added customers in the **customer** table:



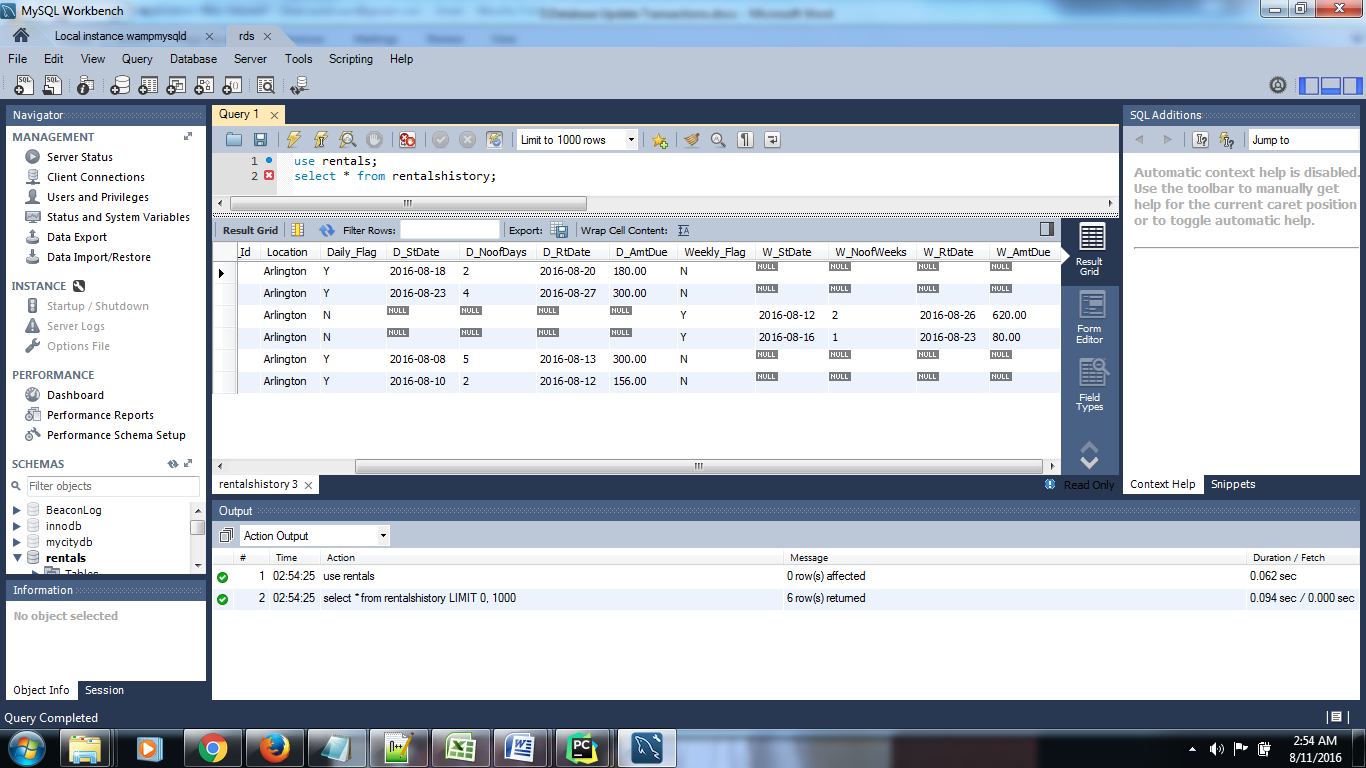
b) Added cars in the **car** table:



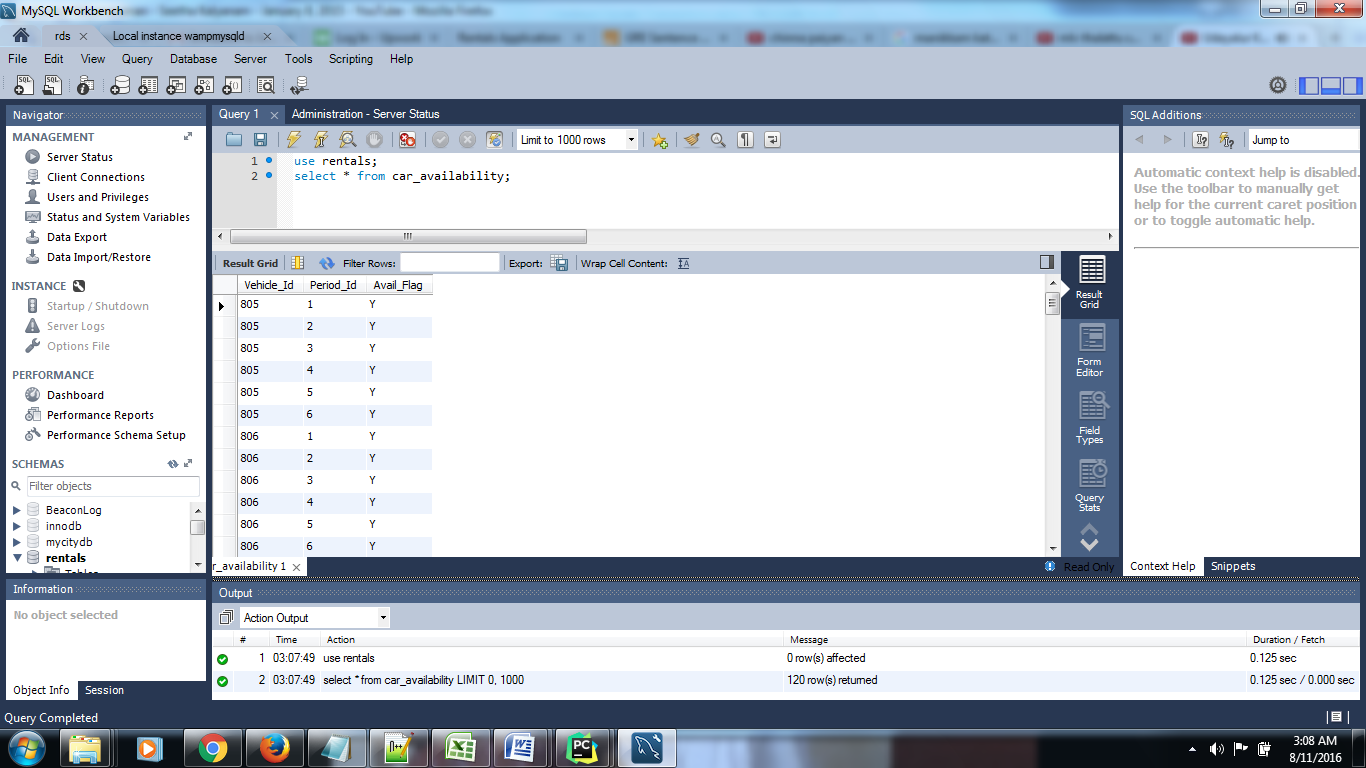
c) Added rental details in the **rentals** table:



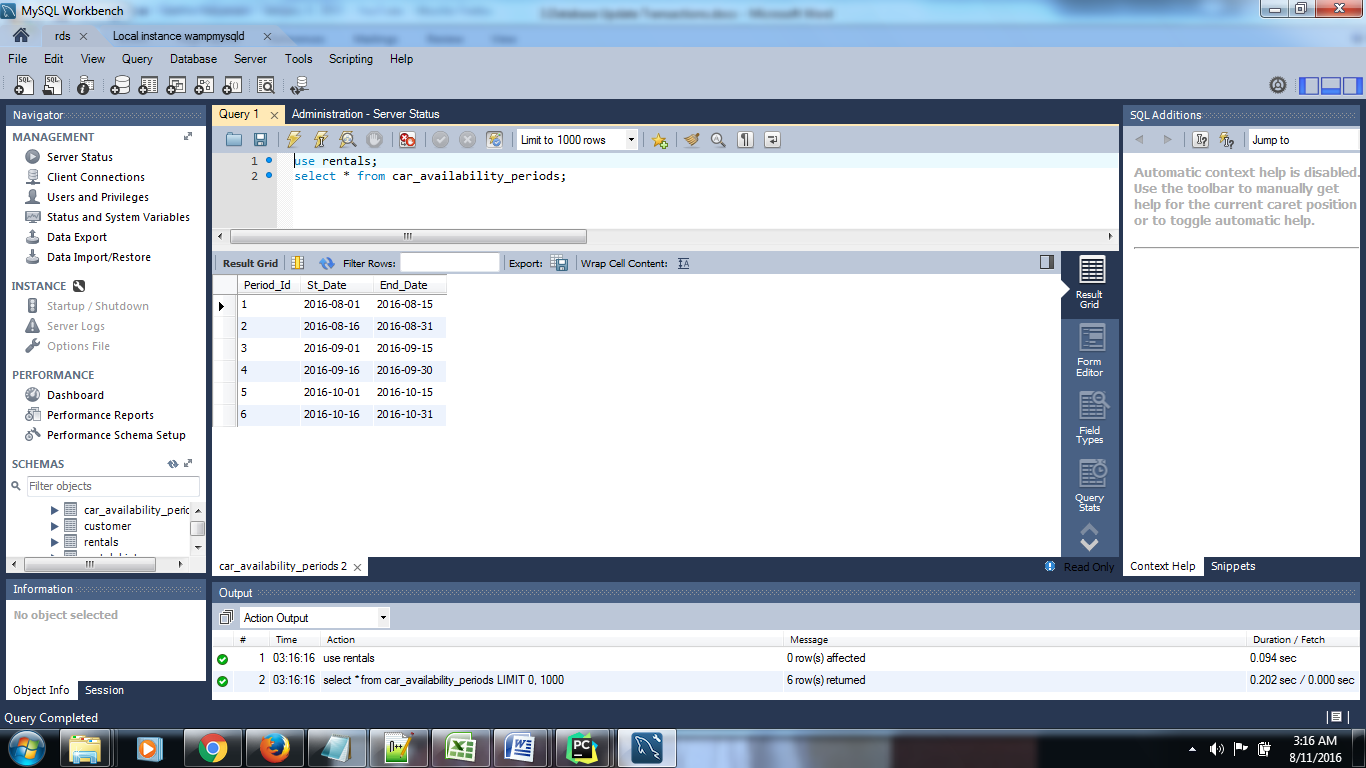
d) The returned cars are inserted in the rentalshistory table.



e) The returned cars are made available in the **car\_availability** table.



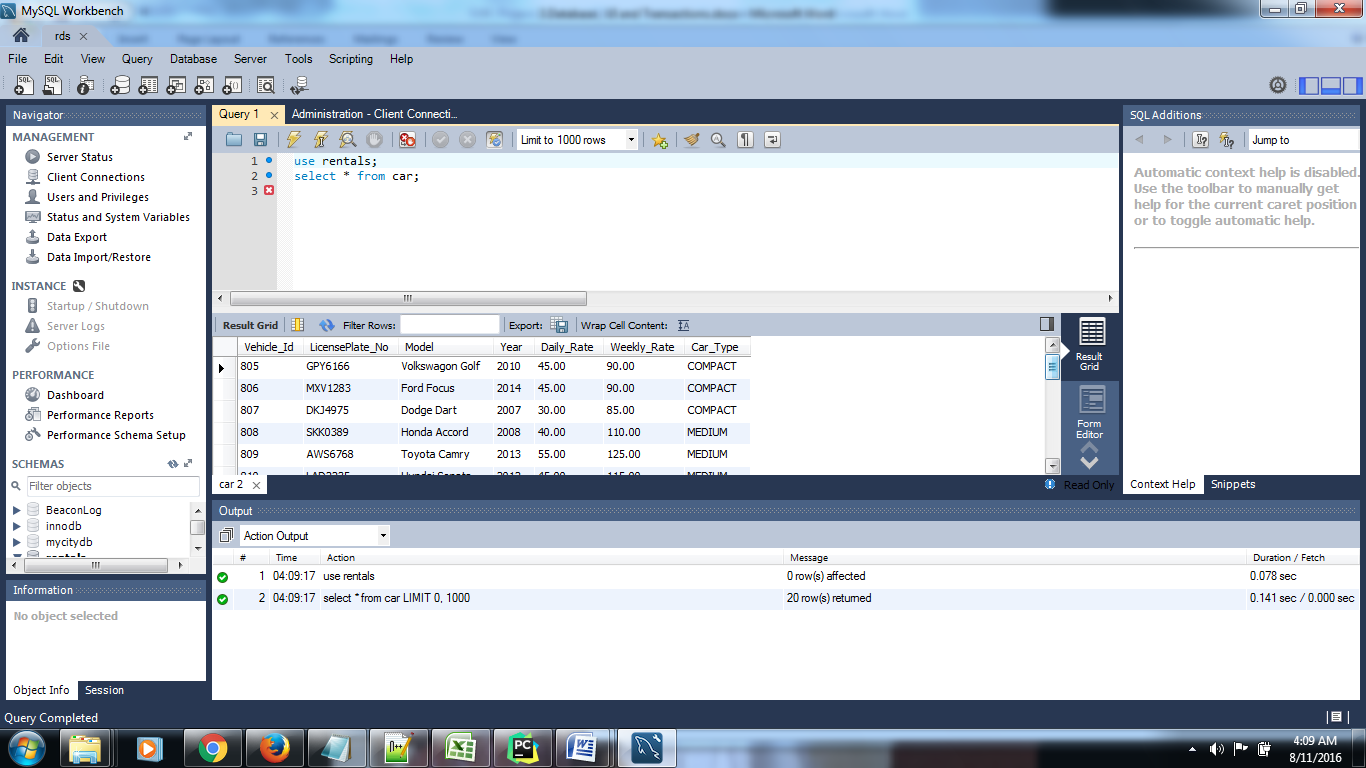
f) We have assumed all cars are available initially for a period of three months. The three months period are inserted in **car\_available\_periods** as period of 15days.



g) The daily and weekly rental rates are updated in the UI.

Before Updation:

The Vehicle\_Id with value ‘805’ has a Daily Rate of $45 and WeeklyRate of $90.



After Updation:

The Vehicle\_Id with value ‘805’ is now updated with Daily Rate of $75 and WeeklyRate of $120.

