

Car Rental Project

CSE 3330/5330

Database Systems 1

Project 2 part 3

Fall 2020

Team Members:

Mukka Himaneesh, Nudrat Nawal Saber

HONOR CODE

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work and honest effort in the pursuit of academic excellence. I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity and uphold the spirit of the Honor Code.

TASK 1

Query 1:

ALTER TABLE RENTAL

ADD Returned CHAR(1);

The screenshot displays a MySQL database management interface. At the top, the server is identified as '127.0.0.1' and the database as 'carrentalsystem'. A toolbar contains icons for Structure, SQL, Search, Query, Export, Import, Operations, and Privileges. A text box prompts the user to 'Run SQL query/queries on database carrentalsystem:'. Below this, the SQL query is entered in a text area:

```
1 ALTER TABLE RENTAL
2 ADD Returned CHAR(1);
3 |
```

The bottom section of the interface shows the query execution results. A green message bar states: 'MySQL returned an empty result set (i.e. zero rows). (Query took 0.0078 seconds.)'. Below this, the executed query is shown: 'ALTER TABLE RENTAL ADD Returned CHAR(1)'. At the bottom right, there are links for '[Edit inline]', '[Edit]', and '[Create PHP code]'.

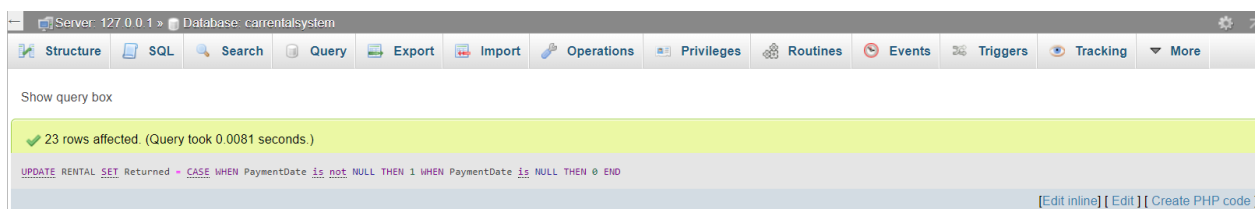
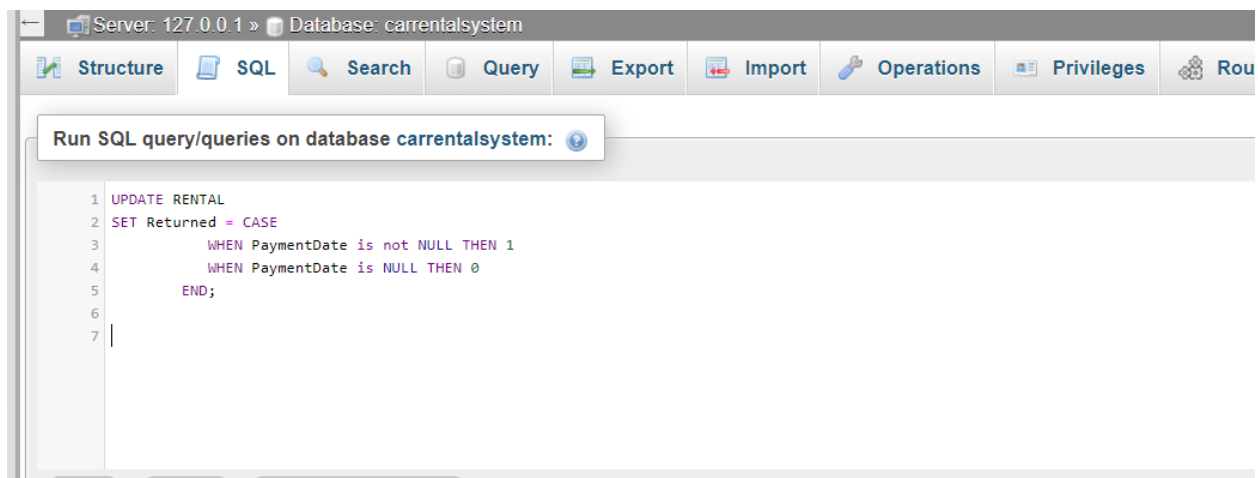
UPDATE RENTAL

SET Returned = CASE

WHEN PaymentDate is not NULL THEN 1

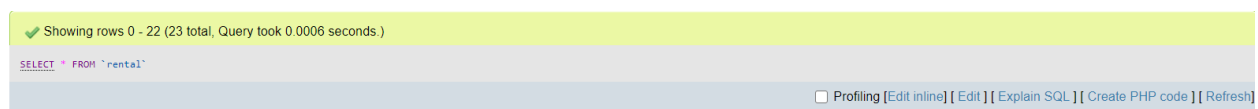
WHEN PaymentDate is NULL THEN 0

END;



Show the records rental Result Grid:

SELECT * from rental;



Server: 127.0.0.1 » Database: currentsystem » Table: rental												
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers												
+ Options												
		CustID	VehicleID	StartDate	OrderDate	RentalType	Qty	ReturnDate	TotalAmount	PaymentDate	Returned	
<input type="checkbox"/>	Edit Copy Delete	203	JM3KE4DY4F0441471	2019-09-09	2019-05-22	1	4	2019-09-13	460	2019-09-09	1	
<input type="checkbox"/>	Edit Copy Delete	210	19VDE1F3XEE414842	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	0	
<input type="checkbox"/>	Edit Copy Delete	210	JTHFF2C26F135BX45	2019-05-01	2019-04-15	7	1	2019-05-08	600	2019-05-08	1	
<input type="checkbox"/>	Edit Copy Delete	210	JTHFF2C26F135BX45	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	0	
<input type="checkbox"/>	Edit Copy Delete	210	WAUTFAFH0E0010613	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	0	
<input type="checkbox"/>	Edit Copy Delete	210	WBA3A9G51ENN73366	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	0	
<input type="checkbox"/>	Edit Copy Delete	210	WBA3B9C59EP458859	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	0	
<input type="checkbox"/>	Edit Copy Delete	210	WDCGG0EB0EG188709	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	0	
<input type="checkbox"/>	Edit Copy Delete	212	19VDE1F3XEE414842	2019-06-10	2019-04-15	7	3	2019-07-01	1800	2019-06-10	1	
<input type="checkbox"/>	Edit Copy Delete	216	1N6BF0KM0EN101134	2019-08-02	2019-03-15	7	4	2019-08-30	2740	2019-08-02	1	
<input type="checkbox"/>	Edit Copy Delete	216	1N6BF0KM0EN101134	2019-08-30	2019-03-15	1	2	2019-09-01	230	2019-08-02	1	
<input type="checkbox"/>	Edit Copy Delete	221	19VDE1F3XEE414842	2019-07-01	2019-06-12	7	1	2019-07-08	600	2019-07-01	1	
<input type="checkbox"/>	Edit Copy Delete	221	19VDE1F3XEE414842	2019-07-09	2019-06-12	1	2	2019-07-11	200	2019-07-01	1	
<input type="checkbox"/>	Edit Copy Delete	221	19VDE1F3XEE414842	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0	
<input type="checkbox"/>	Edit Copy Delete	221	JTHFF2C26F135BX45	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0	
<input type="checkbox"/>	Edit Copy Delete	221	WAUTFAFH0E0010613	2019-07-01	2019-06-12	7	1	2019-07-08	600	2019-07-01	1	
<input type="checkbox"/>	Edit Copy Delete	221	WAUTFAFH0E0010613	2019-07-09	2019-06-12	1	2	2019-07-11	200	2019-07-01	1	
<input type="checkbox"/>	Edit Copy Delete	221	WAUTFAFH0E0010613	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0	
<input type="checkbox"/>	Edit Copy Delete	221	WBA3A9G51ENN73366	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0	
<input type="checkbox"/>	Edit Copy Delete	221	WBA3B9C59EP458859	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0	
<input type="checkbox"/>	Edit Copy Delete	221	WDCGG0EB0EG188709	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0	
<input type="checkbox"/>	Edit Copy Delete	229	19VDE1F3XEE414842	2019-05-06	2019-04-12	1	4	2019-05-10	400	2019-05-06	1	
<input type="checkbox"/>	Edit Copy Delete	229	WAUTFAFH0E0010613	2019-05-06	2019-04-12	1	4	2019-05-10	400	2019-05-06	1	

Query 2:

CREATE VIEW:

```
DROP TABLE IF EXISTS vRentalInfo ;

DROP VIEW IF EXISTS vRentalInfo;

CREATE VIEW vRentalInfo AS

SELECT r.OrderDate, r.StartDate, r.ReturnDate, DATEDIFF(r.ReturnDate,
r.StartDate) as 'TotalDays',
r.VehicleID as 'VIN', v.Description as 'Vehicle',
CASE WHEN v.Type = 1 THEN "Compact"
      WHEN v.Type = 2 THEN "Medium"
      WHEN v.Type = 3 THEN "Large"
      WHEN v.Type = 4 THEN "SUV"
      WHEN v.Type = 5 THEN "Truck"
      WHEN v.Type = 6 THEN "VAN"
END AS Type,
CASE WHEN v.Category = 0 THEN "Basic"
      ELSE "Luxury"
END AS Category, r.CustID as 'CustomerID', c.Name as 'CustomerName',
TotalAmount as 'OrderAmount',
CASE WHEN r.Returned = 0 THEN TotalAmount
      ELSE 0
END AS RentalBalance
FROM RENTAL r , VEHICLE v , CUSTOMER c
WHERE r.VehicleID = v.VehicleID and r.CustID = c.CustID
ORDER BY StartDate ASC;
```

Server: 127.0.0.1 » Database: carrentalsystem

Structure SQL Search Query Export Import Operations Priv

Run SQL query/queries on database carrentalsystem:

```

1 DROP TABLE IF EXISTS vRentalInfo ;
2 DROP VIEW IF EXISTS vRentalInfo;
3 CREATE VIEW vRentalInfo AS
4 SELECT r.OrderDate, r.StartDate, r.ReturnDate, DATEDIFF(r.ReturnDate,
5 r.StartDate) as 'TotalDays',
6 r.VehicleID as 'VIN', v.Description as 'Vehicle',
7 CASE WHEN v.Type = 1 THEN "Compact"
8 WHEN v.Type = 2 THEN "Medium"
9 WHEN v.Type = 3 THEN "Large"
10 WHEN v.Type = 4 THEN "SUV"
11 WHEN v.Type = 5 THEN "Truck"
12 WHEN v.Type = 6 THEN "VAN"
13 END AS Type,
14 CASE WHEN v.Category = 0 THEN "Basic"
15 ELSE "Luxury"
16 END AS Category, r.CustID as 'CustomerID', c.Name as 'CustomerName',
17 TotalAmount as 'OrderAmount',
18 CASE WHEN r.Returned = 0 THEN TotalAmount
19 ELSE 0
20 END AS RentalBalance
21 FROM RENTAL r , VEHICLE v , CUSTOMER c
22 WHERE r.VehicleID = v.VehicleID and r.CustID = c.CustID
23 ORDER BY StartDate ASC;
24

```

Server: 127.0.0.1 » Database: carrentalsystem

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking More

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0016 seconds.)

```

DROP TABLE IF EXISTS vRentalInfo

```

[Edit inline] [Edit] [Create PHP code]

⚠ Note: #1051 Unknown table 'carrentalsystem.vrentalinfo'

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0006 seconds.)

```

DROP VIEW IF EXISTS vRentalInfo

```

[Edit inline] [Edit] [Create PHP code]

⚠ Note: #4092 Unknown VIEW: 'carrentalsystem.vrentalinfo'

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0131 seconds.)

```

CREATE VIEW vRentalInfo AS SELECT r.OrderDate, r.StartDate, r.ReturnDate, DATEDIFF(r.ReturnDate, r.StartDate) as 'TotalDays', r.VehicleID as 'VIN', v.Description as 'Vehicle', CASE WHEN v.Type = 1 THEN "Compact"
WHEN v.Type = 2 THEN "Medium" WHEN v.Type = 3 THEN "Large" WHEN v.Type = 4 THEN "SUV" WHEN v.Type = 5 THEN "Truck" WHEN v.Type = 6 THEN "VAN" END AS Type, CASE WHEN v.Category = 0 THEN "Basic" ELSE "Luxury" END AS
Category, r.CustID as 'CustomerID', c.Name as 'CustomerName', TotalAmount as 'OrderAmount', CASE WHEN r.Returned = 0 THEN TotalAmount ELSE 0 END AS RentalBalance FROM RENTAL r , VEHICLE v , CUSTOMER c WHERE
r.VehicleID = v.VehicleID and r.CustID = c.CustID ORDER BY StartDate ASC

```

[Edit inline] [Edit] [Create PHP code]

SELECT VIEW:

select * from vRentalInfo;

✓ Showing rows 0 - 22 (23 total. Query took 0.0019 seconds.)

```

select * from vRentalInfo

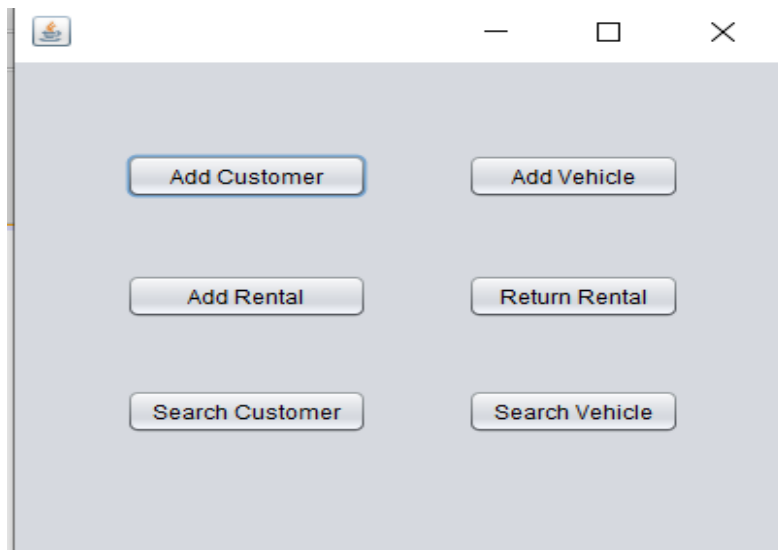
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Options															
		▼	OrderDate	StartDate	ReturnDate	TotalDays	VIN	Vehicle	Type	Category	CustomerID	CustomerName	OrderAmount	RentalBalance	
<input type="checkbox"/>				2019-04-15	2019-05-01	2019-05-08	7	JTHFF2C26F135BX45	Lexus IS 250C	Compact	Luxury	210	G. Clarkson	600	0
<input type="checkbox"/>				2019-04-12	2019-05-06	2019-05-10	4	WAUTFAFH0E0010613	Audi A5	Compact	Luxury	229	D. Kirkpatrick	400	0
<input type="checkbox"/>				2019-04-12	2019-05-06	2019-05-10	4	19VDE1F3XEE414842	Acura ILX	Compact	Luxury	229	D. Kirkpatrick	400	0
<input type="checkbox"/>				2019-04-15	2019-06-10	2019-07-01	21	19VDE1F3XEE414842	Acura ILX	Compact	Luxury	212	H. Gallegos	1800	0
<input type="checkbox"/>				2019-06-12	2019-07-01	2019-07-08	7	WAUTFAFH0E0010613	Audi A5	Compact	Luxury	221	J. Brown	600	0
<input type="checkbox"/>				2019-06-12	2019-07-01	2019-07-08	7	19VDE1F3XEE414842	Acura ILX	Compact	Luxury	221	J. Brown	600	0
<input type="checkbox"/>				2019-06-12	2019-07-09	2019-07-11	2	WAUTFAFH0E0010613	Audi A5	Compact	Luxury	221	J. Brown	200	0
<input type="checkbox"/>				2019-06-12	2019-07-09	2019-07-11	2	19VDE1F3XEE414842	Acura ILX	Compact	Luxury	221	J. Brown	200	0
<input type="checkbox"/>				2019-03-15	2019-08-02	2019-08-30	28	1N6BF0KM0EN101134	Nissan NV	VAN	Basic	216	A. Hess	2740	0
<input type="checkbox"/>				2019-03-15	2019-08-30	2019-09-01	2	1N6BF0KM0EN101134	Nissan NV	VAN	Basic	216	A. Hess	230	0
<input type="checkbox"/>				2019-05-22	2019-09-09	2019-09-13	4	JM3KE4DY4F0441471	Mazda CX5	SUV	Basic	203	A. Hernandez	460	0
<input type="checkbox"/>				2019-10-28	2019-11-01	2019-11-15	14	19VDE1F3XEE414842	Acura ILX	Compact	Luxury	210	G. Clarkson	1200	1200
<input type="checkbox"/>				2019-10-28	2019-11-01	2019-11-15	14	JTHFF2C26F135BX45	Lexus IS 250C	Compact	Luxury	210	G. Clarkson	1200	1200
<input type="checkbox"/>				2019-10-28	2019-11-01	2019-11-15	14	WBA3B9C59EP458859	BMW 3 Series	Compact	Luxury	210	G. Clarkson	1200	1200
<input type="checkbox"/>				2019-10-28	2019-11-01	2019-11-15	14	WAUTFAFH0E0010613	Audi A5	Compact	Luxury	210	G. Clarkson	1200	1200
<input type="checkbox"/>				2019-10-28	2019-11-01	2019-11-15	14	WBA3A9G51ENN73366	BMW 3 Series	Compact	Luxury	210	G. Clarkson	1200	1200
<input type="checkbox"/>				2019-10-28	2019-11-01	2019-11-15	14	WDCGG0EB0EG188709	Mercedes-Benz GLK	Compact	Luxury	210	G. Clarkson	1200	1200
<input type="checkbox"/>				2019-12-15	2020-01-01	2020-01-29	28	19VDE1F3XEE414842	Acura ILX	Compact	Luxury	221	J. Brown	2400	2400
<input type="checkbox"/>				2019-12-15	2020-01-01	2020-01-29	28	WAUTFAFH0E0010613	Audi A5	Compact	Luxury	221	J. Brown	2400	2400
<input type="checkbox"/>				2019-12-15	2020-01-01	2020-01-29	28	WBA3A9G51ENN73366	BMW 3 Series	Compact	Luxury	221	J. Brown	2400	2400
<input type="checkbox"/>				2019-12-15	2020-01-01	2020-01-29	28	WBA3B9C59EP458859	BMW 3 Series	Compact	Luxury	221	J. Brown	2400	2400
<input type="checkbox"/>				2019-12-15	2020-01-01	2020-01-29	28	WDCGG0EB0EG188709	Mercedes-Benz GLK	Compact	Luxury	221	J. Brown	2400	2400
<input type="checkbox"/>				2019-12-15	2020-01-01	2020-01-29	28	JTHFF2C26F135BX45	Lexus IS 250C	Compact	Luxury	221	J. Brown	2400	2400

TASK 2

The figure given below is a simple Home interface for the car rental System.



1. Our first requirement is adding information of a new customer by NOT providing the customer id .Whenever a new customer is added to the table ,the customer id automatically gets incremented.

The query for is given below:

```
ps = con.prepareStatement("insert into customer (name,phone) values (?,?)");
```

```
ps.setString(1,addcustomername);
```

```
ps.setString(2,addcustomerphone);
```

The top screenshot shows a web application window with a 'Home' button at the top. Below it, there are two input fields: 'Name *' with the value 'Nudrat' and 'Phone *' with the value '(214)-892-6389'. A 'Submit' button is located below the phone field.

The bottom screenshot shows the same web application window, but with a 'Message' dialog box overlaying the form. The dialog box has a title bar 'Message' and a close button 'X'. Inside the dialog, there is a blue information icon 'i' and the text 'Success'. An 'OK' button is at the bottom right of the dialog. The 'Submit' button from the form is still visible below the dialog.

A total of 32 rows are returned.


```
SELECT COUNT(CustID) FROM customer;
```

✓ Showing rows 25 - 31 (32 total, Query took 0.0009 seconds.)

`SELECT * FROM `customer``

<< < 2 | ☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

				CustID	Name	Phone
<input type="checkbox"/>		Edit		Copy		Delete
				226	R. Armstrong	(325) 783-4081
<input type="checkbox"/>		Edit		Copy		Delete
				227	J. Greenaway	(212) 262-8829
<input type="checkbox"/>		Edit		Copy		Delete
				228	K. Kaiser Acosta	(228) 576-1557
<input type="checkbox"/>		Edit		Copy		Delete
				229	D. Kirkpatrick	(773) 696-8009
<input type="checkbox"/>		Edit		Copy		Delete
				230	A. Odonnell	(439) 536-8929
<input type="checkbox"/>		Edit		Copy		Delete
				231	K. Kay	(368) 336-5403
<input type="checkbox"/>		Edit		Copy		Delete
				232	Nudrat	(214)-892-6389

2. Now for this query we are adding the information of a new vehicle.

```
ps = con.prepareStatement("insert into vehicle values (?, ?, ?, ?, ?)");
```

```
ps.setString(1, vehicleid);
```

```
ps.setString(2, description);
```

```
ps.setString(3, String.valueOf(year));
```

```
ps.setString(4, type);
```

```
ps.setString(5, category);
```

Home

Vehicle ID *

Description *


Year *

Type ▼

Category ▼

Home

Message

 Success

Category ▼

A total of 61 rows are returned for this query.

`SELECT COUNT(VehicleID) FROM vehicle;`

✓ Showing rows 0 - 24 (61 total, Query took 0.0008 seconds.)

```
SELECT * FROM `vehicle`
```

☐ Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refre](#)

<input type="checkbox"/>				TVVCH7A3XEC037969	Volkswagen Passat	2014	2	1
<input type="checkbox"/>				2HGFB2F94FH501940	Honda Civic	2015	1	0
<input type="checkbox"/>				2T3DFREV0FW317743	Toyota RAV4	2015	4	0
<input type="checkbox"/>				3MZBM1L74EM109736	Mazda 3	2014	1	0
<input type="checkbox"/>				3N1CE2CP0FL409472	Nissan Versa Note	2015	1	0
<input type="checkbox"/>				3N1CN7APXEK444458	Nissan Versa	2014	1	0
<input type="checkbox"/>				3VW2A7AU1FM012211	Volkswagen Golf	2015	1	0
<input type="checkbox"/>				4S4BRCFC1E3203823	Subaru Outback	2014	4	0
<input type="checkbox"/>				4S4BSBF39F3261064	Subaru Outback	2015	4	0
<input type="checkbox"/>				4S4BSELC0F3325370	Subaru Outback	2015	4	0
<input checked="" type="checkbox"/>				5cfrdcfg	Honda	2018	1	0
<input type="checkbox"/>				5J6RM4H90FL028629	Honda CR-V	2015	4	0
<input type="checkbox"/>				5N1AL0MM8EL549388	Infiniti JX35	2014	4	1
<input type="checkbox"/>				5NPDH4AE2FH565275	Hyundai Elantra	2015	1	0
<input type="checkbox"/>				5TDBKRFH4ES26D590	Toyota Highlander	2014	4	0

3. Now we need to add all the information about a new rental reservation and the condition is those vehicles must be free from any rental .And the customer can pay either at the order or the return date.

```
String sql="SELECT Description FROM Vehicle WHERE VehicleID NOT IN (SELECT DISTINCT VehicleID FROM Rental WHERE (StartDate >= '"+startdate+"' and returndate <= '"+returndate+"' ) or (startDate< '"+returndate+"' and returndate>= '"+startdate+"' )) and Type = '"+type+"' and Category = '"+category+"' UNION SELECT Description FROM Vehicle WHERE Type = '"+type+"' and Category = '"+category+"' and VehicleID NOT IN (SELECT DISTINCT VehicleID FROM Rental)";
```

```
String sql2="SELECT phone from customer where custid ='"+customerid+"'";
```

```
String sql3="SELECT VehicleID FROM Vehicle WHERE Description = '"+selectedmodel+"' and Type = '"+type+"' and Category = '"+category+"'";
```

```
String query4="select * from rate where (type = '"+type+"' and category = '"+category+"' );
```

```
ps = con.prepareStatement("insert into rental values (?,?,?,?,?,?,?,?,?)");
```

The screenshot shows a web application interface for car rental reservation. The form is organized into two columns of input fields. The left column contains 'Type' (dropdown), 'Category' (dropdown), 'Start Date *' (text input), 'Return Date *' (text input), and 'Rental Type' (dropdown). The right column contains 'Customer ID *' (text input), 'Quantity *' (text input), 'Order Date *' (text input), and 'Payment Date' (text input). Below these fields is a 'Submit' button. At the bottom of the form, there is a 'Description' text area and a 'Selected Model *' dropdown menu, followed by a 'Rent' button. The interface has a light blue background and standard web form elements.

Home

Type: 1 Customer ID *: 231

Category: 0 Quantity *: 1

Start Date *: 2019-02-20 Order Date *: 2019-01-10

Return Date *: 2019-02-2 Payment Date: 2019-02-29

Rental Type: 1

Submit

Description: Chevrolet Optra, Nissan Sentra, Honda Civic, Mazda 3

Selected Model *: Chevrolet Optra

Rent

Home

Type: 1 Customer ID *: 231

Category: 0 Quantity *: 1

Start Date *: 2020-02-20 Order Date *: 2020-01-10

Return Date *: 2020-02-2 Payment Date: 2020-02-29

Rental Type: 1

Submit

Description: Mazda 3, Nissan Versa Note, Nissan Versa, Volkswagen Golf

Selected Model *: Chevrolet Optra

Message: Success

OK

Total of a 24 rows are returned.

Showing rows 0 - 23 (24 total, Query took 0.0006 seconds.)

SELECT * FROM `rental`

<input type="checkbox"/>				221	WDCGG0E80EG188709	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0
<input type="checkbox"/>				229	19VDE1F3XEE414842	2019-05-06	2019-04-12	1	4	2019-05-10	400	2019-05-06	1
<input type="checkbox"/>				229	WAUTFAFH0E0010613	2019-05-06	2019-04-12	1	4	2019-05-10	400	2019-05-06	1
<input type="checkbox"/>				231	1G1JD5SB3E4240835	2020-02-20	2020-01-10	1	1	2020-02-29	720	2020-02-29	0

4.

By inputting the return date, phone number and vehicle info we will search for the customer and his total due for a particular vehicle. If there is a due, we can clear the due by clicking the "Pay Due" button. The payment will be cleared for that vehicle for the customer.

```
String sql = "SELECT DISTINCT v.VehicleID, v.Description FROM Vehicle v, Rental r WHERE v.VehicleID = r.VehicleID and Returned = '0'";
```

```
String sql2 = "SELECT v.VehicleID FROM Vehicle v, rental r WHERE v.VehicleID = r.VehicleID and v.Description = '"+selectedmodel+"' and r.ReturnDate = '"+returndate+"'";
```

```
String sql3 = " SELECT CustID FROM Customer WHERE Phone = '"+phonenumbeber+"'";
```

```
String sql4 = "SELECT c.Name as 'Name', c.Phone as 'Phone', r.ReturnDate as 'ReturnDate', SUM(r.TotalAmount) as 'PaymentDue' FROM customer c, rental r, Vehicle v WHERE c.CustID = '"+custid+"' and c.CustID = r.CustID and r.VehicleID = v.VehicleID and v.VehicleID = '"+vehid+"'";
```

```
if(due==0)
{
    String query = "UPDATE rental SET Returned = '1' WHERE CustID = ? and ReturnDate = ? and VehicleID = ?";
    ps = con.prepareStatement(query);
    ps.setString(1, custid);
    ps.setString(2, returndate);
    ps.setString(3, vehid);
    System.out.println(ps);
    int i = ps.executeUpdate();
    if (i > 0) {
        JOptionPane.showMessageDialog(null, "Vecihle returned!");
    } else {
        JOptionPane.showMessageDialog(null, "Failed");
    }
}
```

```

    }
    else
    {
        String query = "UPDATE rental SET Returned = '1', TotalAmount = '0' WHERE CustID = ? and
ReturnDate = ? and VehicleID = ?";

        ps = con.prepareStatement(query);

        ps.setString (1, custid);

        ps.setString (2, returndate);

        ps.setString (3, vehid);

        System.out.println(ps);

        int i = ps.executeUpdate();

        if (i > 0) {

            JOptionPane.showMessageDialog(null, "Payment successful");

        } else {

            JOptionPane.showMessageDialog(null, "Failed");

        }

    }
}

```

Home

Return Date * 2019-11-15

Phone Number * (309) 625-1838

Vehicle Info BMW 3 Series ▼

Search

Name	Phone	Return Date	Total Due
G. Clarkson	(309) 625-1838	2019-11-15	1200

Pay Due

Home

Message

Payment successful

OK

2019-11-15

(309) 625-1838

BMW 3 Series

Search

Name	Phone	Return Date	Total Due
G. Clarkson	(309) 625-1838	2019-11-15	1200

Pay Due

Home

Return Date *

2019-11-15

Phone Number *

(309) 625-1838

Vehicle Info

BMW 3 Series

Search

Name	Phone	Return Date	Total Due
G. Clarkson	(309) 625-1838	2019-11-15	0

Pay Due

5. Now we have to return the view's results.

(I) For this question ,we have to list every customer ID,name if there is any remaining balance.User can search either by a customer's id ,name ,part of the name ,or also without any filter or criteria.For the customers with zero balance it will return 0 dollars(\$0.00).

```
if(custid.length()==0 && custname.length()==0)
{
```

```
// order by amount
```

```
String sql = "SELECT CustomerID, CustomerName, RentalBalance FROM vRentalInfo";
ResultSet rs = stmt.executeQuery(sql);
DefaultTableModel tblmodel=(DefaultTableModel)searchcustomertable.getModel();
while(rs.next())
{
    String data[]={rs.getString(1),rs.getString(2),"$" +rs.getString(3)};
    tblmodel.addRow(data);
}
}
else
{
    if(custid.length()!=0)
    {
        String sql = "SELECT CustomerID, CustomerName, RentalBalance FROM vRentalInfo WHERE
CustomerID = '"+custid+"'";
        ResultSet rs = stmt.executeQuery(sql);
        DefaultTableModel tblmodel=(DefaultTableModel)searchcustomertable.getModel();
        while(rs.next())
        {
            String data[]={rs.getString(1),rs.getString(2),"$" +rs.getString(3)+".00"};
            tblmodel.addRow(data);
        }
    }
    else
    {
        String sql = "SELECT CustomerID, CustomerName, RentalBalance FROM vRentalInfo WHERE
CustomerName Like '%" +custname+"%'";
```



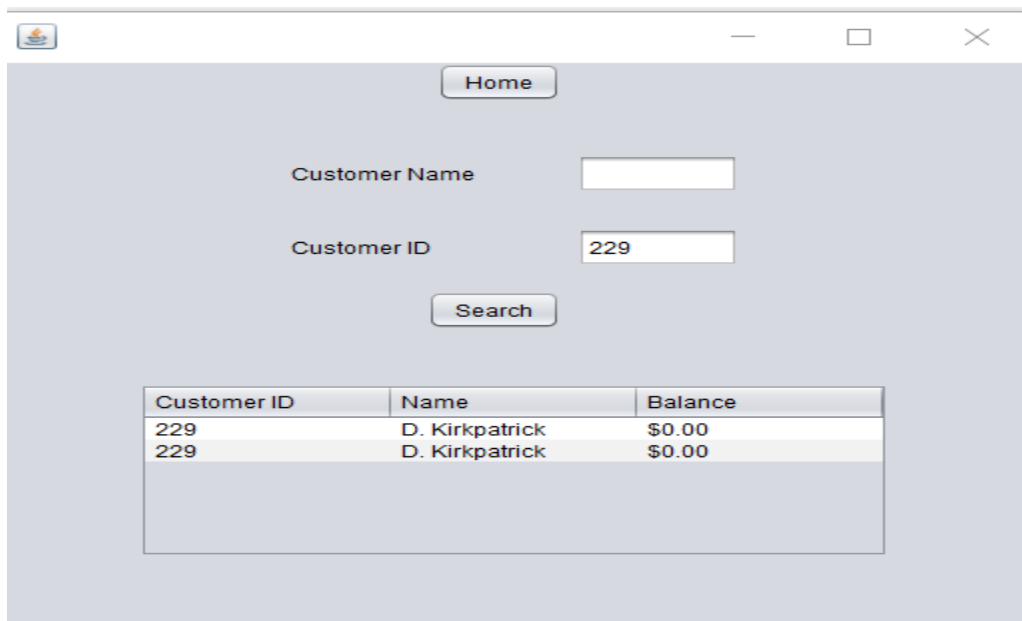
```

ResultSet rs = stmt.executeQuery(sql);

DefaultTableModel tblmodel=(DefaultTableModel)searchcustomertable.getModel();

while(rs.next())
{
    String data[]={rs.getString(1),rs.getString(2),"$" +rs.getString(3)+".00"};
    tblmodel.addRow(data);
}
}
}

```



The screenshot shows a Java Swing window with a title bar containing a Home icon, a minus sign, a maximize button, and a close button. The window has a light gray background. At the top center is a button labeled "Home". Below it, there are two labels: "Customer Name" and "Customer ID". The "Customer Name" label is followed by an empty text input field. The "Customer ID" label is followed by a text input field containing the value "229". Below these fields is a button labeled "Search". At the bottom of the window is a table with three columns: "Customer ID", "Name", and "Balance". The table contains two rows of data, both showing "229" for the Customer ID and "D. Kirkpatrick" for the Name, with a Balance of "\$0.00".

Customer ID	Name	Balance
229	D. Kirkpatrick	\$0.00
229	D. Kirkpatrick	\$0.00

(II) This query retrieves a list for every vehicle the VIN, the description, and the average DAILY price. And The user has the right either to search by the VIN, vehicle's description, part of the description, or to run the query with no filters/criteria.

```

des = tfsearchvehdes.getText();

vin = tfsearchvehvin.getText();

if(vin.length()==0 && des.length()==0)
{

```

```
String sql = "SELECT VIN, Vehicle, CONCAT('$', round((OrderAmount/TotalDays),2)) as  
'DailyPrice' FROM vRentalInfo UNION select v.VehicleID, v.Description, 'N/A' AS 'DailyPrice' from vehicle  
v, rate r where v.Type = r.Type and v.Category = r.Category and v.VehicleID NOT IN (select VehicleID  
from rental)";
```

```
ResultSet rs = stmt.executeQuery(sql);
```

```
DefaultTableModel tblmodel=(DefaultTableModel)searchvehtable.getModel();
```

```
while(rs.next())
```

```
{
```

```
String data[]={rs.getString(1),rs.getString(2),rs.getString(3)};
```

```
tblmodel.addRow(data);
```

```
}
```

```
}
```

```
else
```

```
{
```

```
if(vin.length()!=0)
```

```
{
```

```
String sql = "SELECT VIN, Vehicle, CONCAT('$', round((OrderAmount/TotalDays),2)) as  
'DailyPrice' FROM vRentalInfo WHERE VIN = '"+vin+"' UNION select v.VehicleID, v.Description, 'N/A' AS  
'DailyPrice' from vehicle v, rate r where v.Type = r.Type and v.Category = r.Category and v.VehicleID =  
 '"+vin+"' and v.VehicleID NOT IN (select VehicleID from rental)";
```

```
ResultSet rs = stmt.executeQuery(sql);
```

```
DefaultTableModel tblmodel=(DefaultTableModel)searchvehtable.getModel();
```

```
while(rs.next())
```

```
{
```

```
String data[]={rs.getString(1),rs.getString(2),rs.getString(3)};
```

```
tblmodel.addRow(data);
```

```
}
```

```
}
```

```
else
```

```
{
```

```
String sql = "SELECT VIN, Vehicle, CONCAT('$', round((OrderAmount/TotalDays),2)) as  
'DailyPrice' FROM vRentalInfo WHERE Vehicle Like '%" + des + "%' UNION select v.VehicleID, v.Description,  
'N/A' AS 'DailyPrice' from vehicle v, rate r where v.Type = r.Type and v.Category = r.Category and  
v.Description like '%" + des + "%' and VehicleID NOT IN (select VehicleID from rental)";
```

```
ResultSet rs = stmt.executeQuery(sql);
```

```
DefaultTableModel tblmodel=(DefaultTableModel)searchvehtable.getModel();
```

```
while(rs.next())
```

```
{
```

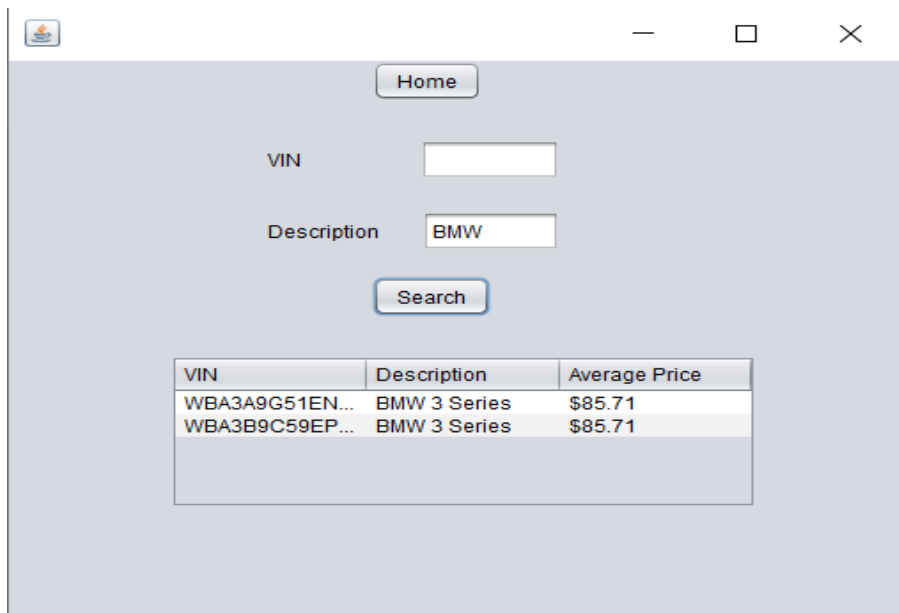
```
String data[]={rs.getString(1),rs.getString(2),rs.getString(3)};
```

```
tblmodel.addRow(data);
```

```
}
```


```
}
```

```
}
```



The screenshot shows a Java Swing window with a title bar containing a Home icon and standard window controls (minimize, maximize, close). The window has a light gray background. At the top center is a button labeled "Home". Below it are two text input fields: the first is labeled "VIN" and is empty; the second is labeled "Description" and contains the text "BMW". Below these fields is a button labeled "Search". At the bottom of the window is a table with three columns: "VIN", "Description", and "Average Price". The table contains two rows of data.

VIN	Description	Average Price
WBA3A9G51EN...	BMW 3 Series	\$85.71
WBA3B9C59EP...	BMW 3 Series	\$85.71



—

□

×

Home

VIN

Description

Search

VIN	Description	Average Price	
WAUTFAFH0E...	Audi A5	\$100.00	▲
WBA3B9C59E...	BMW 3 Series	\$85.71	
1FDEE3FL6ED...	Ford E 350	N/A	
1FDRF3B61FE...	Ford Super Dut...	N/A	
1FTNF1CF2EK...	Ford F Series P...	N/A	▼