Project 2 Part 3

CSE 3330

Database system & File structures

Team members

Aashish Maharjan : 1001733603

Aayush Adhikari : 1001742258

Parichat Tan ai : 1001043176

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: Execute queries on the CarRental2019 database tables

Query 1:

Add an extra column ‘Returned’ to the RENTAL table. Values will be 0-for non-returned cars, and 1-for returned. Then update the ‘Returned’ column with '1' for all records that they have a payment date and with '0' for those that they do not have a payment date.

Submit:

(1)

```
CREATE TABLE RENTAL
(
        CustID          INT             NOT NULL,
        VehicleID       VARCHAR(20)     NOT NULL,
        StartDate       DATE            NOT NULL,
        OrderDate       DATE            NOT NULL,
        RentalType      INT             NOT NULL,
        Qty             INT             NOT NULL,
        ReturnDate      DATE            NOT NULL,
        TotalAmount     DECIMAL         NOT NULL,
        PaymentDate     DATE,
        FOREIGN KEY (VehicleID) REFERENCES VEHICLE(VehicleID),
        FOREIGN KEY(CustId) REFERENCES CUSTOMER(CustId)
);
ALTER TABLE RENTAL
ADD Returned INT NOT NULL;

UPDATE RENTAL
SET Returned = 1
WHERE PaymentDate IS NOT NULL;

UPDATE RENTAL
SET Returned = 0
WHERE PaymentDate IS NULL;

SELECT * FROM RENTAL;
```

(2)

Result Grid Filter Rows: Search Export:

CustID	VehicleID	StartDate	OrderDate	RentalType	Qty	ReturnDate	TotalAmount	PaymentDate	Returned
203	JM3KE4DY4F0441471	2019-09-09	2019-05-22	1	4	2019-09-13	460	2019-09-09	1
210	19VDE1F3XEE414842	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	0
210	JTHFF2C26F135BX45	2019-05-01	2019-04-15	7	1	2019-05-08	600	2019-05-08	1
210	JTHFF2C26F135BX45	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	0
210	WAUTFAFH0E0010613	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	0
210	WBA3A9G51ENN73366	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	0
210	WBA3B9C59EP458859	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	0
210	WDCGG0EB0EG188709	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	0
212	19VDE1F3XEE414842	2019-06-10	2019-04-15	7	3	2019-07-01	1800	2019-06-10	1
216	1N6BF0KM0EN101134	2019-08-02	2019-03-15	7	4	2019-08-30	2740	2019-08-02	1
216	1N6BF0KM0EN101134	2019-08-30	2019-03-15	1	2	2019-09-01	230	2019-08-02	1
221	19VDE1F3XEE414842	2019-07-01	2019-06-12	7	1	2019-07-08	600	2019-07-01	1
221	19VDE1F3XEE414842	2019-07-09	2019-06-12	1	2	2019-07-11	200	2019-07-01	1
221	19VDE1F3XEE414842	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0
221	JTHFF2C26F135BX45	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0
221	WAUTFAFH0E0010613	2019-07-01	2019-06-12	7	1	2019-07-08	600	2019-07-01	1
221	WAUTFAFH0E0010613	2019-07-09	2019-06-12	1	2	2019-07-11	200	2019-07-01	1
221	WAUTFAFH0E0010613	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0
221	WBA3A9G51ENN73366	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0
221	WBA3B9C59EP458859	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0
221	WDCGG0EB0EG188709	2020-01-01	2019-12-15	7	4	2020-01-29	2400	NULL	0
229	19VDE1F3XEE414842	2019-05-06	2019-04-12	1	4	2019-05-10	400	2019-05-06	1
229	WAUTFAFH0E0010613	2019-05-06	2019-04-12	1	4	2019-05-10	400	2019-05-06	1

(3)

Returned column (last column) added to the RENTAL table with the values of 0 and 1.

Query 2:

Create a view vRentalInfo that retrieves all information per rental.

Submit:

1.

```
CREATE VIEW vRentalInfo AS
SELECT DISTINCT OrderDate, StartDate, ReturnDate, (ReturnDate-StartDate) AS TotalDays,
VEHICLE.VehicleID AS VIN,
DESCRIPTION AS Vehicle,
CASE
        WHEN VEHICLE.Type = '1' THEN 'Compact'
        WHEN VEHICLE.Type = '2' THEN 'Medium'
        WHEN VEHICLE.Type = '3' THEN 'Large'
        WHEN VEHICLE.Type = '4' THEN 'SUV'
        WHEN VEHICLE.Type = '5' THEN 'Truck'
        WHEN VEHICLE.Type = '6' THEN 'VAN'
        END AS Type,
CASE
        WHEN VEHICLE.Category = '0' THEN 'Basic'
        WHEN VEHICLE.Category = '1' THEN 'Luxury'
        END AS Category,
CUSTOMER.CustID AS CustomerID,
NAME AS CustomerName,
RENTAL.TotalAmount AS OrderAmount,
CASE
        WHEN PaymentDate IS NULL THEN TotalAmount
        ELSE TotalAmount = 0
        END AS RentalBalance
        FROM CUSTOMER,VEHICLE,RENTAL,RATE
```

```
WHERE CUSTOMER.CustID = RENTAL.CustID
AND VEHICLE.VehicleID = RENTAL.VehicleID
ORDER BY StartDate ASC;
```

2.

```
SELECT * FROM vRentalInfo
```

Result Grid Filter Rows: Search Export:

OrderDate	StartDate	ReturnDate	TotalDays	VIN	Vehicle	Type	Category	CustomerID	CustomerName	OrderAmount	RentalBalance
2019-04-15	2019-05-01	2019-05-08	7	JTHFF2C26F135BX45	Lexus IS 250C	Compact	Luxury	210	G. Clarkson	600	0
2019-04-12	2019-05-06	2019-05-10	4	19VDE1F3XEE414842	Acura ILX	Compact	Luxury	229	D. Kirkpatrick	400	0
2019-04-12	2019-05-06	2019-05-10	4	WAUTFAFH0E0010613	Audi A5	Compact	Luxury	229	D. Kirkpatrick	400	0
2019-04-15	2019-06-10	2019-07-01	91	19VDE1F3XEE414842	Acura ILX	Compact	Luxury	212	H. Gallegos	1800	0
2019-06-12	2019-07-01	2019-07-08	7	19VDE1F3XEE414842	Acura ILX	Compact	Luxury	221	J. Brown	600	0
2019-06-12	2019-07-01	2019-07-08	7	WAUTFAFH0E0010613	Audi A5	Compact	Luxury	221	J. Brown	600	0
2019-06-12	2019-07-09	2019-07-11	2	19VDE1F3XEE414842	Acura ILX	Compact	Luxury	221	J. Brown	200	0
2019-06-12	2019-07-09	2019-07-11	2	WAUTFAFH0E0010613	Audi A5	Compact	Luxury	221	J. Brown	200	0
2019-03-15	2019-08-02	2019-08-30	28	1N6BF0KM0EN101134	Nissan NV	VAN	Basic	216	A. Hess	2740	0
2019-03-15	2019-08-30	2019-09-01	71	1N6BF0KM0EN101134	Nissan NV	VAN	Basic	216	A. Hess	230	0
2019-05-22	2019-09-09	2019-09-13	4	JM3KE4DY4F0441471	Mazda CX5	SUV	Basic	203	A. Hernandez	460	0
2019-10-28	2019-11-01	2019-11-15	14	19VDE1F3XEE414842	Acura ILX	Compact	Luxury	210	G. Clarkson	1200	1200
2019-10-28	2019-11-01	2019-11-15	14	JTHFF2C26F135BX45	Lexus IS 250C	Compact	Luxury	210	G. Clarkson	1200	1200
2019-10-28	2019-11-01	2019-11-15	14	WAUTFAFH0E0010613	Audi A5	Compact	Luxury	210	G. Clarkson	1200	1200
2019-10-28	2019-11-01	2019-11-15	14	WBA3A9G51ENN73366	BMW 3 Series	Compact	Luxury	210	G. Clarkson	1200	1200
2019-10-28	2019-11-01	2019-11-15	14	WBA3B9C59EP458859	BMW 3 Series	Compact	Luxury	210	G. Clarkson	1200	1200
2019-10-28	2019-11-01	2019-11-15	14	WDCGG0EB0EG1887...	Mercedes_B...	Compact	Luxury	210	G. Clarkson	1200	1200
2019-12-15	2020-01-01	2020-01-29	28	19VDE1F3XEE414842	Acura ILX	Compact	Luxury	221	J. Brown	2400	2400
2019-12-15	2020-01-01	2020-01-29	28	JTHFF2C26F135BX45	Lexus IS 250C	Compact	Luxury	221	J. Brown	2400	2400
2019-12-15	2020-01-01	2020-01-29	28	WAUTFAFH0E0010613	Audi A5	Compact	Luxury	221	J. Brown	2400	2400
2019-12-15	2020-01-01	2020-01-29	28	WBA3A9G51ENN73366	BMW 3 Series	Compact	Luxury	221	J. Brown	2400	2400
2019-12-15	2020-01-01	2020-01-29	28	WBA3B9C59EP458859	BMW 3 Series	Compact	Luxury	221	J. Brown	2400	2400
2019-12-15	2020-01-01	2020-01-29	28	WDCGG0EB0EG1887...	Mercedes_B...	Compact	Luxury	221	J. Brown	2400	2400

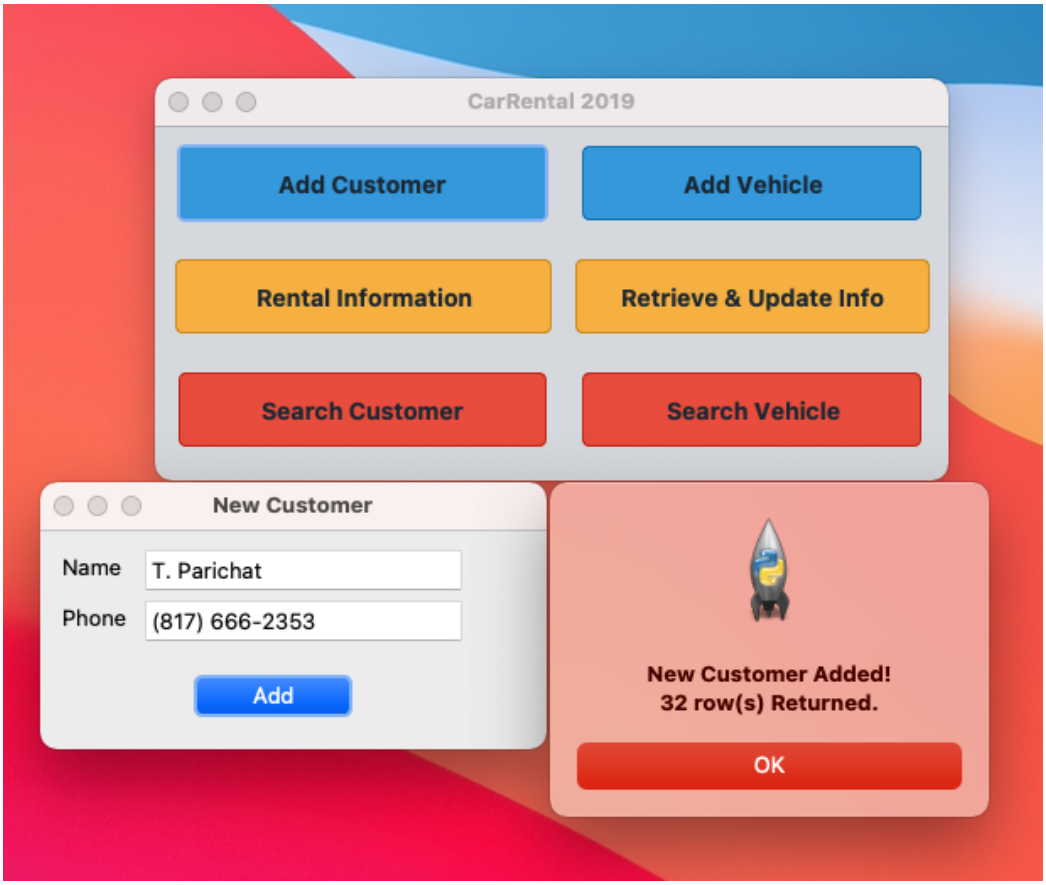
3.

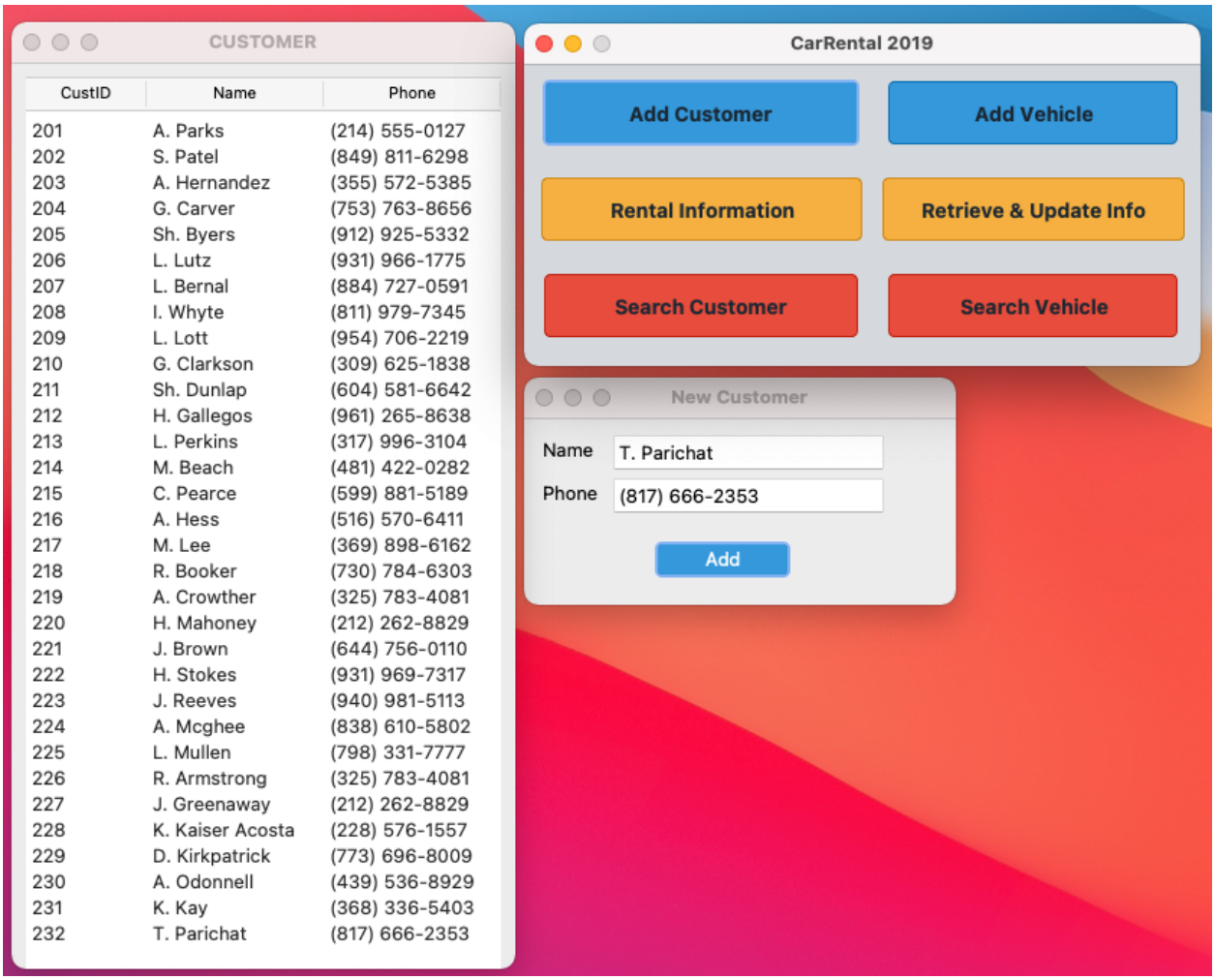
```
23 row(s) returned
```

Task 2: Create a GUI for the CarRental2019 database

1. The first requirement is to add information about a new customer. Do not provide the customer ID in your query. Submit your editable SQL query that your code executes.

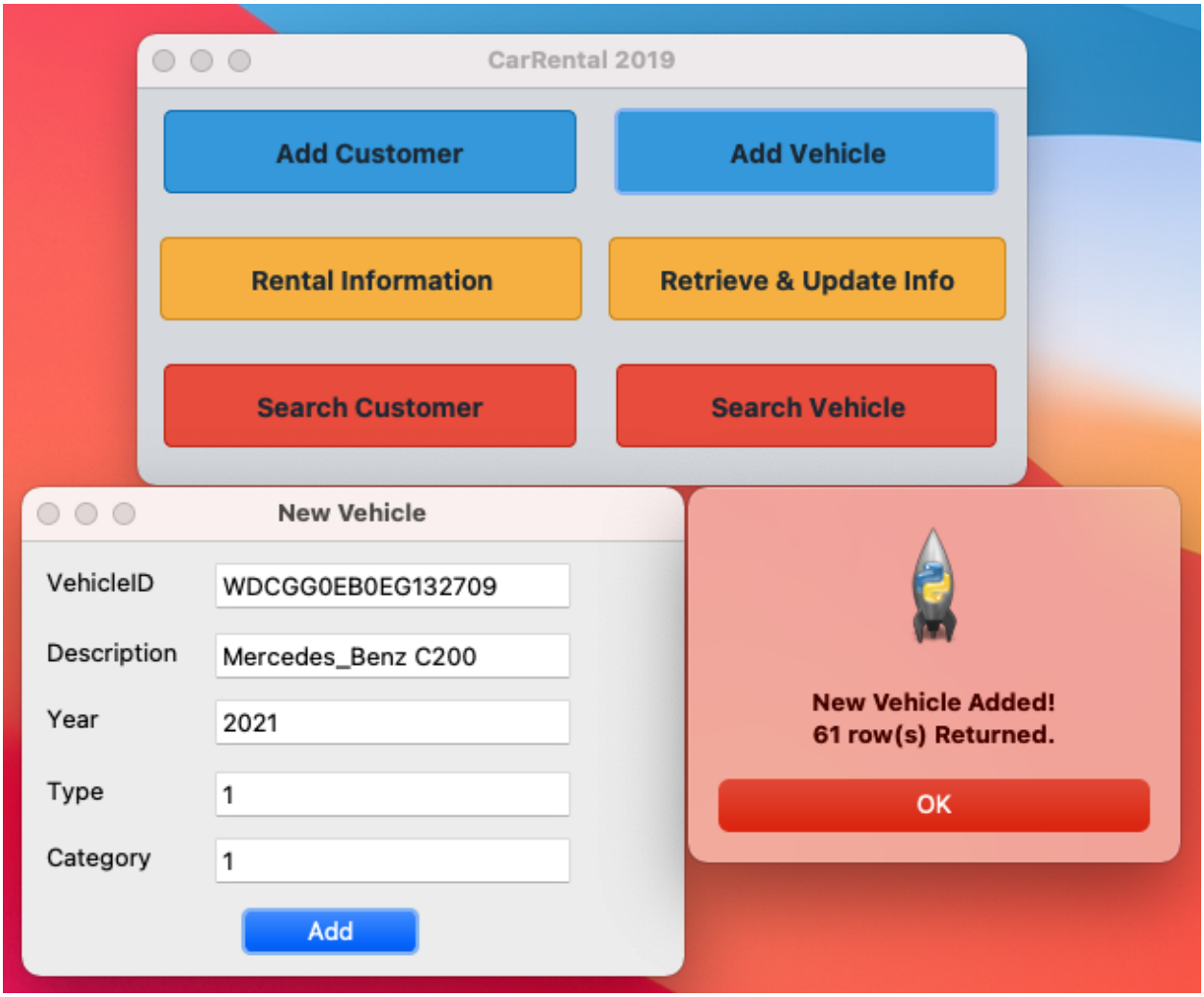
```
data = "INSERT INTO customer(Name, Phone) Values (%s, %s)"
value = (Name, Phone)
my_cursor.execute(data, value)
```

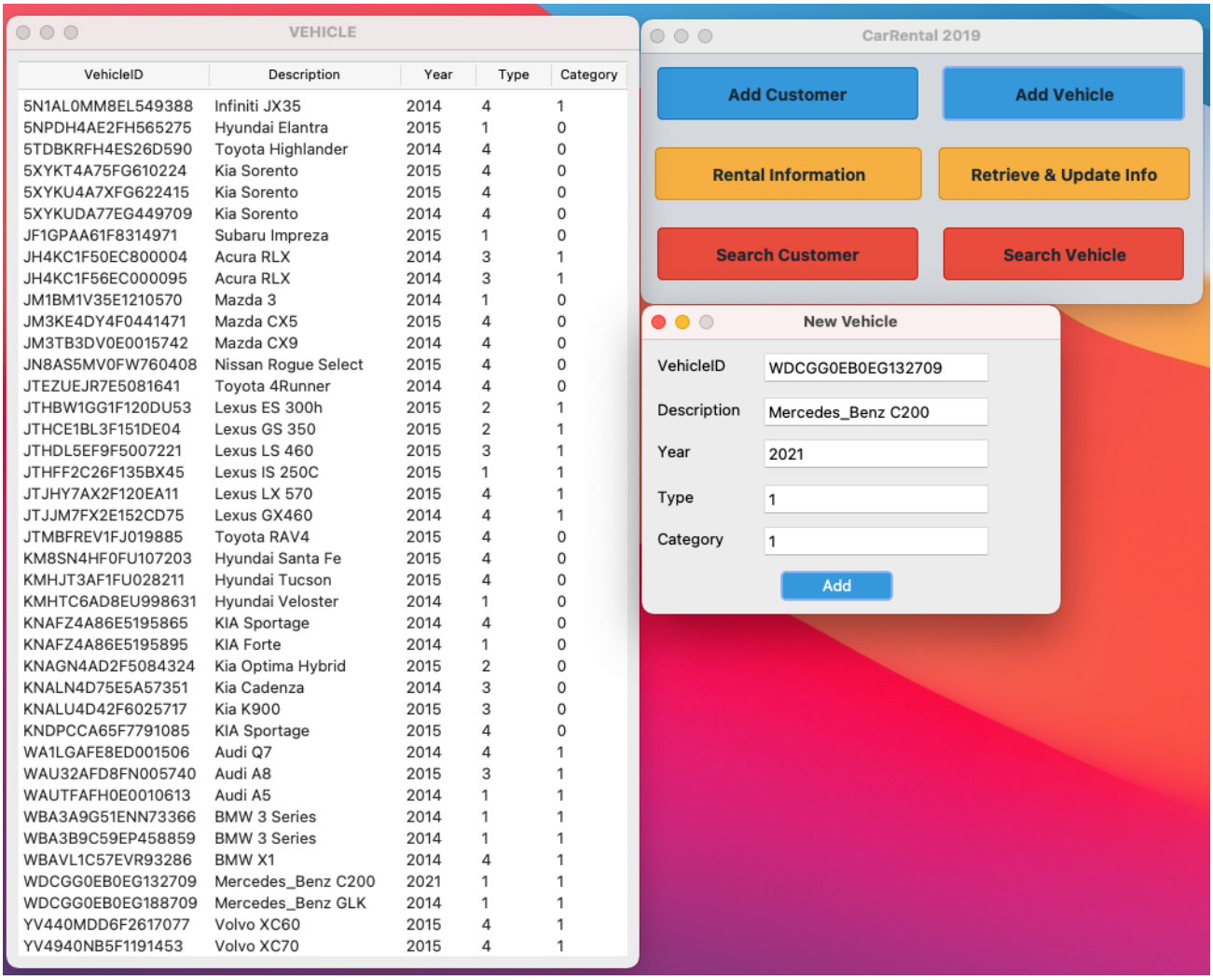



CUSTOMER
CustID Name Phone
201 A. Parks (214) 555-0127
202 S. Patel (849) 811-6298
203 A. Hernandez (355) 572-5385
204 G. Carver (753) 763-8656
205 Sh. Byers (912) 925-5332
206 L. Lutz (931) 966-1775
207 L. Bernal (884) 727-0591
208 I. Whyte (811) 979-7345
209 L. Lott (954) 706-2219
210 G. Clarkson (309) 625-1838
211 Sh. Dunlap (604) 581-6642
212 H. Gallegos (961) 265-8638
213 L. Perkins (317) 996-3104
214 M. Beach (481) 422-0282
215 C. Pearce (599) 881-5189
216 A. Hess (516) 570-6411
217 M. Lee (369) 898-6162
218 R. Booker (730) 784-6303
219 A. Crowther (325) 783-4081
220 H. Mahoney (212) 262-8829
221 J. Brown (644) 756-0110
222 H. Stokes (931) 969-7317
223 J. Reeves (940) 981-5113
224 A. Mcghee (838) 610-5802
225 L. Mullen (798) 331-7777
226 R. Armstrong (325) 783-4081
227 J. Greenaway (212) 262-8829
228 K. Kaiser Acosta (228) 576-1557
229 D. Kirkpatrick (773) 696-8009
230 A. Odonnell (439) 536-8929
231 K. Kay (368) 336-5403
232 T. Parichat (817) 666-2353
CarRental 2019
Add Customer
Add Vehicle
Rental Information
Retrieve & Update Info
Search Customer
Search Vehicle
New Customer
Name T. Parichat
Phone (817) 666-2353
Add

2. The second requirement is to add all the information about a new vehicle. Submit your editable SQL query that your code executes.

```
data = "INSERT INTO vehicle(VehicleID, Description, Year,Type,Category) Values (%s, %s, %s, %s,%s)"
value = (VehicleID, Description, Year, Type, Category)
my_cursor.execute(data, value)
```

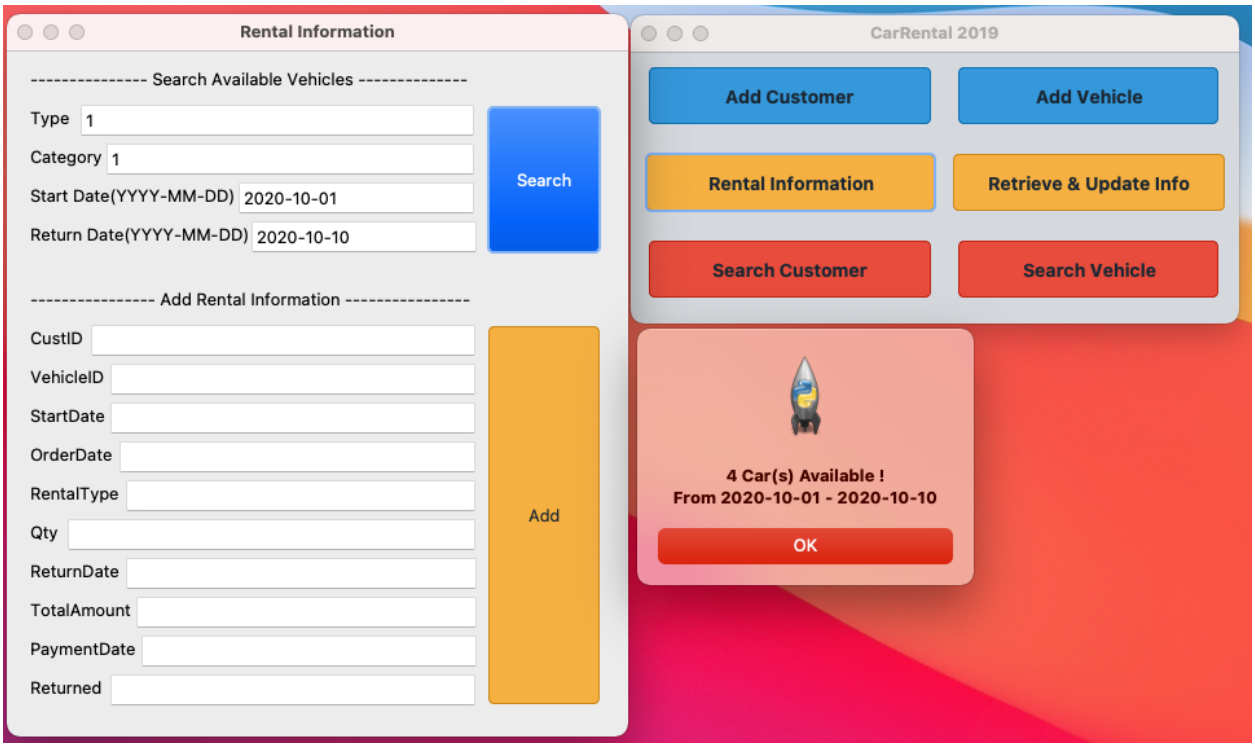


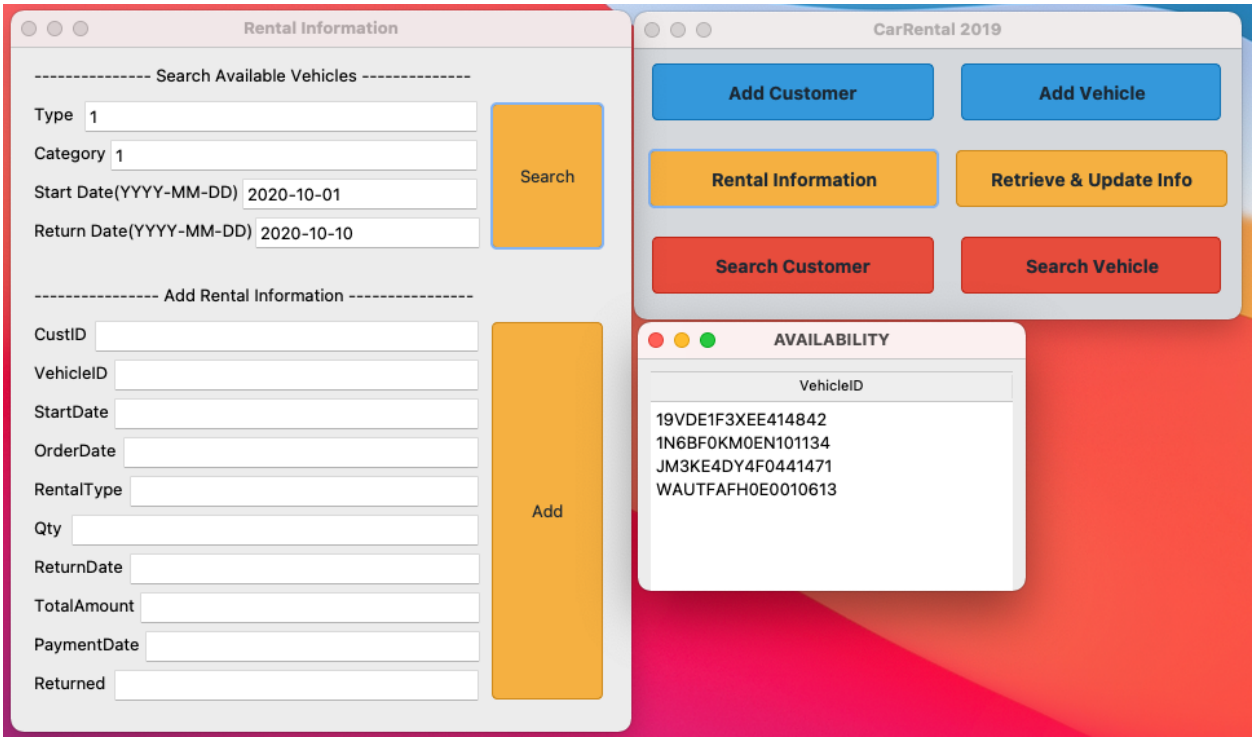

VEHICLE
VehicleID Description Year Type Category
5N1AL0MM8EL549388 Infiniti JX35 2014 4 1
5NPDH4AE2FH565275 Hyundai Elantra 2015 1 0
5TDBKRFH4ES26D590 Toyota Highlander 2014 4 0
5XYKT4A75FG610224 Kia Sorento 2015 4 0
5XYKU4A7XFG622415 Kia Sorento 2015 4 0
5XYKUDA77EG449709 Kia Sorento 2014 4 0
JF1GPAA61F8314971 Subaru Impreza 2015 1 0
JH4KC1F50EC800004 Acura RLX 2014 3 1
JH4KC1F56EC000095 Acura RLX 2014 3 1
JM1BM1V35E1210570 Mazda 3 2014 1 0
JM3KE4DY4F0441471 Mazda CX5 2015 4 0
JM3TB3DV0E0015742 Mazda CX9 2014 4 0
JN8AS5MV0FW760408 Nissan Rogue Select 2015 4 0
JTEZUEJR7E5081641 Toyota 4Runner 2014 4 0
JTHBW1GG1F120DU53 Lexus ES 300h 2015 2 1
JTHCE1BL3F151DE04 Lexus GS 350 2015 2 1
JTHDL5EF9F5007221 Lexus LS 460 2015 3 1
JTHFF2C26F135BX45 Lexus IS 250C 2015 1 1
JTJHY7AX2F120EA11 Lexus LX 570 2015 4 1
JTJJM7FX2E152CD75 Lexus GX460 2014 4 1
JTMBFREV1FJ019885 Toyota RAV4 2015 4 0
KM8SN4HF0FU107203 Hyundai Santa Fe 2015 4 0
KMHJT3AF1FU028211 Hyundai Tucson 2015 4 0
KMHTC6AD8EU998631 Hyundai Veloster 2014 1 0
KNAFZ4A86E5195865 KIA Sportage 2014 4 0
KNAFZ4A86E5195895 KIA Forte 2014 1 0
KNAGN4AD2F5084324 Kia Optima Hybrid 2015 2 0
KNALN4D75E5A57351 Kia Cadenza 2014 3 0
KNALU4D42F6025717 Kia K900 2015 3 0
KNDPCCA65F7791085 KIA Sportage 2015 4 0
WA1LGAFE8ED001506 Audi Q7 2014 4 1
WAU32AFD8FN005740 Audi A8 2015 3 1
WAUTFAFH0E0010613 Audi A5 2014 1 1
WBA3A9G51ENN73366 BMW 3 Series 2014 1 1
WBA3B9C59EP458859 BMW 3 Series 2014 1 1
WBAVL1C57EVR93286 BMW X1 2014 4 1
WDCGG0EB0EG132709 Mercedes_Benz C200 2021 1 1
WDCGG0EB0EG188709 Mercedes_Benz GLK 2014 1 1
YV440MDD6F2617077 Volvo XC60 2015 4 1
YV4940NB5F1191453 Volvo XC70 2015 4 1
CarRental 2019
Add Customer
Add Vehicle
Rental Information
Retrieve & Update Info
Search Customer
Search Vehicle
New Vehicle
VehicleID WDCGG0EB0EG132709
Description Mercedes_Benz C200
Year 2021
Type 1
Category 1
Add

3. The third requirement is to add all the information about a new rental reservation (this must find a free vehicle of the appropriate type and category for a specific rental period). We assume that the customer has the right either to pay at the order or return date. Submit your editable SQL queries (select available vehicles & insert rental) that your code executes.

Available rental:

```
data = "select distinct VehicleID from RENTAL as R1 where ((StartDate NOT BETWEEN %s AND %s ) AND (ReturnDate NOT BETWEEN %s AND %s)  and RentalType = %s)"
value = (search3.get(), search4.get(), search3.get(), search4.get(), search1.get())
my_cursor.execute(data, value)
```




Rental Information
---------- Search Available Vehicles ----------
Type 1
Category 1
Start Date(YYYY-MM-DD) 2020-10-01
Return Date(YYYY-MM-DD) 2020-10-10
Search
---------- Add Rental Information ----------
CustID
VehicleID
StartDate
OrderDate
RentalType
Qty
ReturnDate
TotalAmount
PaymentDate
Returned
Add
CarRental 2019
Add Customer
Add Vehicle
Rental Information
Retrieve & Update Info
Search Customer
Search Vehicle
AVAILABILITY
VehicleID
19VDE1F3XEE414842
1N6BF0KM0EN101134
JM3KE4DY4F0441471
WAUTFAFH0E0010613

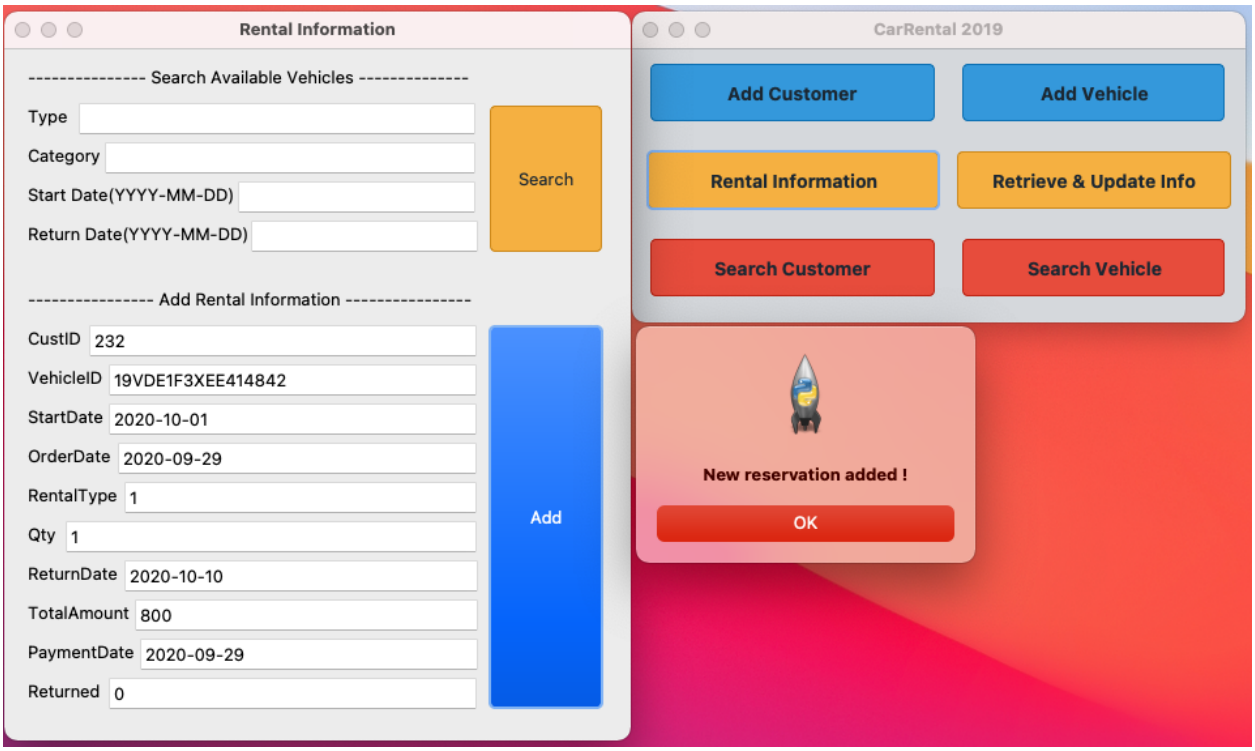
Insert rental:

data = “INSERT INTO Rental (CustID,VehicleID,StartDate,OrderDate,RentalType,Qty,ReturnDate,TotalAmount,PaymentDate, Returned) Values (%s, %s, %s, %s, %s, %s, %s, %s,%s,%s)”

value = (

CustID, VehicleID, StartDate, OrderDate, RentalType, Qty, ReturnDate, TotalAmount, PaymentDate, Returned)

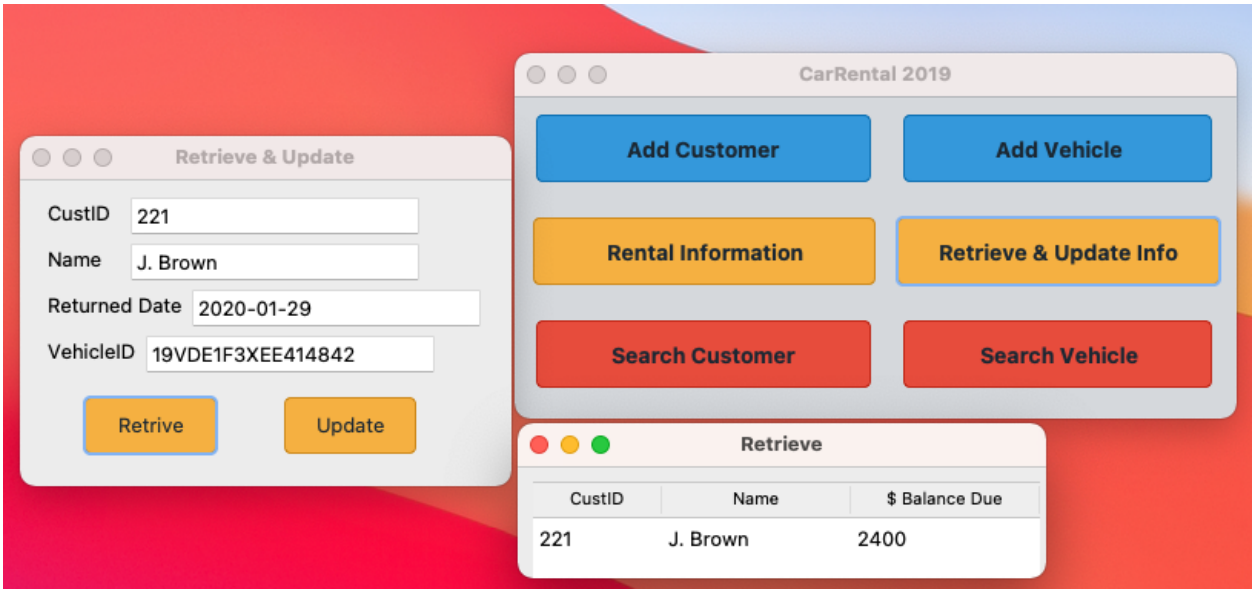
my_cursor.execute(data, value)



4. The fourth requirement is to handle the return of a rented car. This transaction should print the total customer payment due for that rental, enter it in the database and update the returned attribute accordingly. You need to be able to retrieve a rental by the return date, customer name (the table needs the id), and vehicle info. Submit your editable SQL queries (retrieve & update rental) that your code executes.

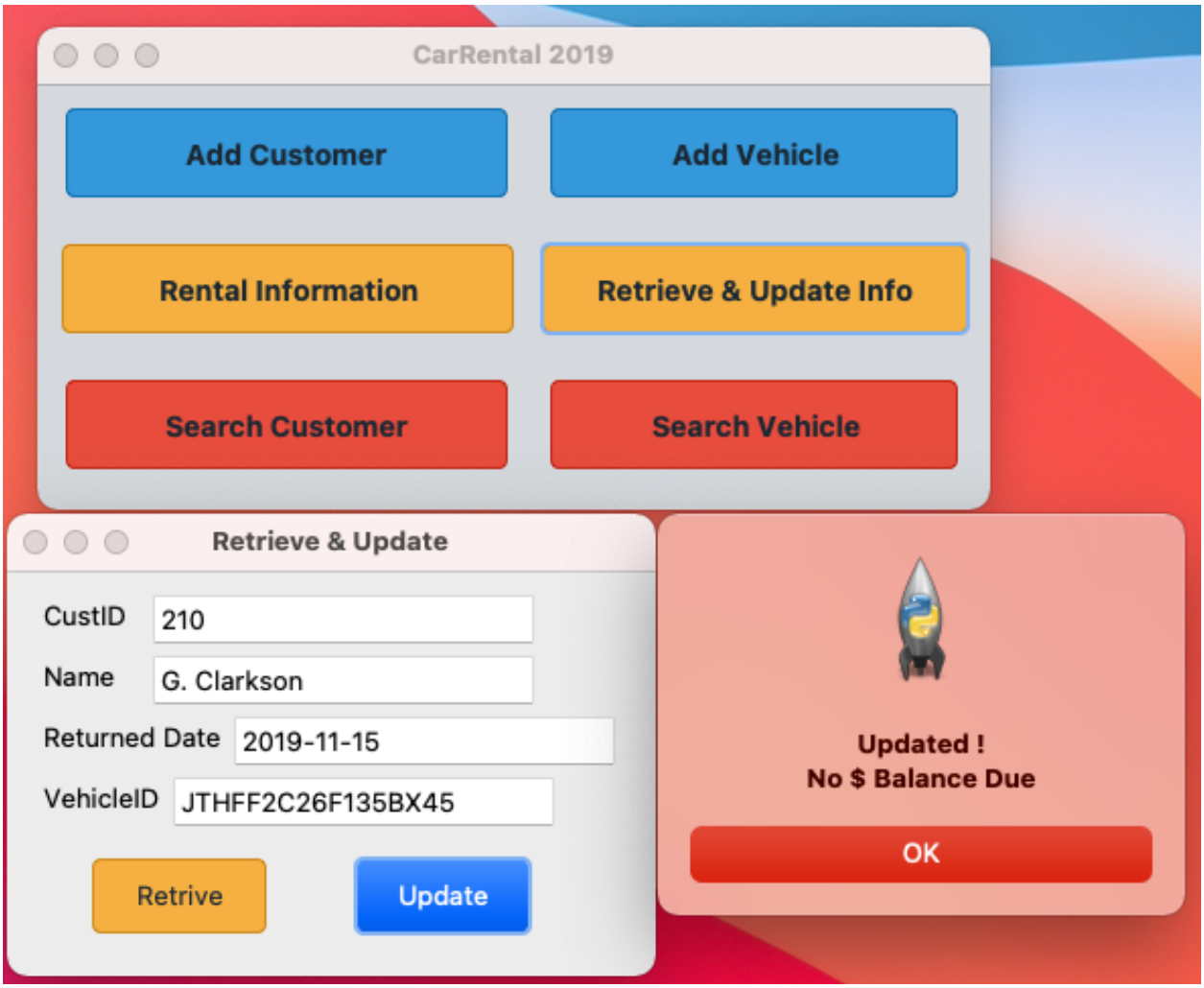
Retrieve rental:

```
data = """SELECT Rental.CustID, Name, TotalAmount FROM rental,  customer WHERE
Rental.CustID=Customer.CustID AND Rental.CustID=%s AND Name=%s AND
ReturnDate=%s AND VehicleID= %s"""
value = (return1.get(), return2.get(), return3.get(), return4.get())
my_cursor.execute(data, value)
```



Update rental:

```
data = """UPDATE RENTAL, CUSTOMER  SET Returned='1', TotalAmount=0  WHERE
Rental.CustID=Customer.CustID AND Rental.CustID=%s AND Name=%s AND
ReturnDate=%s AND VehicleID= %s"""
value = (return1.get(), return2.get(), return3.get(), return4.get())
my_cursor.execute(data, value)
```



5. The fifth requirement is to return the view's results by applying the following criteria:

a.

ID:

```
data = "SELECT CustID, Name FROM CUSTOMER WHERE Customer.CustID='%s' "
%view1.get()
my_cursor.execute(data)
data2 = "SELECT  FORMAT( SUM(TotalAMount ),2) FROM customer, rental WHERE
Rental.CustID= Customer.CustID AND customer.Name='%s' " % view1.get()
my_cursor.execute(data2)
```

Name:

```
data = "SELECT  Rental.CustID, Name , FORMAT( SUM(TotalAMount ),2) FROM customer,
rental WHERE Rental.CustID= Customer.CustID AND PaymentDate IS NULL AND
customer.Name='%s' " % view2.get()
my_cursor.execute(data)
```

Part of name:

```
data = "SELECT  Rental.CustID, Name , FORMAT( SUM(TotalAMount ),2) FROM customer,
Rental WHERE Rental.CustID= Customer.CustID AND PaymentDate IS NULL AND
customer.Name LIKE '%%%s%%' " % view3.get()
my_cursor.execute(data)
```

Without:

```
data = "SELECT  customer.CustID, Name , FORMAT( SUM(TotalAMount ),2) FROM
customer, Rental WHERE PaymentDate IS NULL AND Rental.CustID=customer.CustID
GROUP BY TotalAMount ORDER BY TotalAmount ASC"
my_cursor.execute(data)
```

b.

VIN:

```
data = "SELECT  Rental.VehicleID, Description, FORMAT( ((AVG(TotalAMount))/7),2)
FROM Vehicle, Rental WHERE Rental.VehicleID = Vehicle.VehicleID AND PaymentDate IS
NULL AND Rental.VehicleID='%s' " % v1.get()
my_cursor.execute(data)
```

Description:

```
data = "SELECT  Rental.VehicleID, Description, FORMAT( ((AVG(TotalAMount))/7),2)
FROM Vehicle, Rental WHERE Rental.VehicleID= Vehicle.VehicleID AND PaymentDate IS
NULL AND vehicle.Description='%s' " % v2.get()
my_cursor.execute(data)
```

Part of Description:

```
data = "SELECT  Rental.VehicleID, Description, FORMAT( ((AVG(TotalAMount))/7),2)
FROM Vehicle, Rental WHERE Rental.VehicleID = Vehicle.VehicleID AND PaymentDate IS
NULL AND Vehicle.Description LIKE '%%%s%%' " % v3.get()
my_cursor.execute(data)
```

Without:

```
data = "SELECT  Rental.VehicleID, Description, FORMAT( ((AVG(TotalAMount))/7),2)
FROM Vehicle, Rental WHERE Rental.VehicleID = Vehicle.VehicleID GROUP BY
Description"
my_cursor.execute(data)
```

Contributions:

Aashish Maharjan: SQL queries, Report

Aayush Adhikari: SQL queries, Report

Parichat Tan ai: tkinter in Python, SQL queries