Project 2 (Part – 3)

Task 1: Execute the following queries on the CarRental2019 database tables:

-- Query 1:

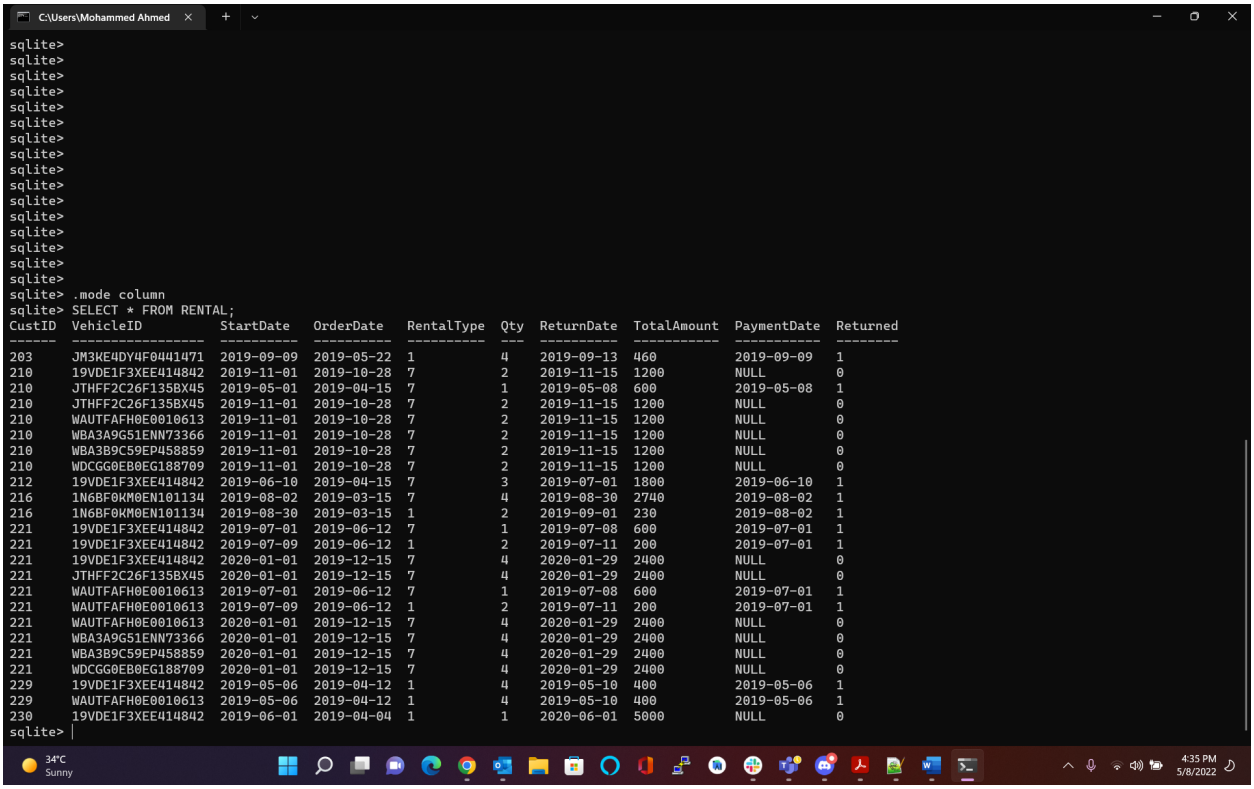
(1)

```
ALTER TABLE RENTAL
ADD Returned DEFAULT 0;


UPDATE RENTAL
SET Returned = 1
WHERE PaymentDate < > 'NULL';
```

(2)

```
SELECT * FROM RENTAL;
```



(3)

The update statement was used to ensure that the values were set accurately over here, and the alter command was used to modify or add a column to the rental table. 0 for cars not returned (payment not made) and 1 for cars returned (Payment completed).

-- Query 2:

(1)

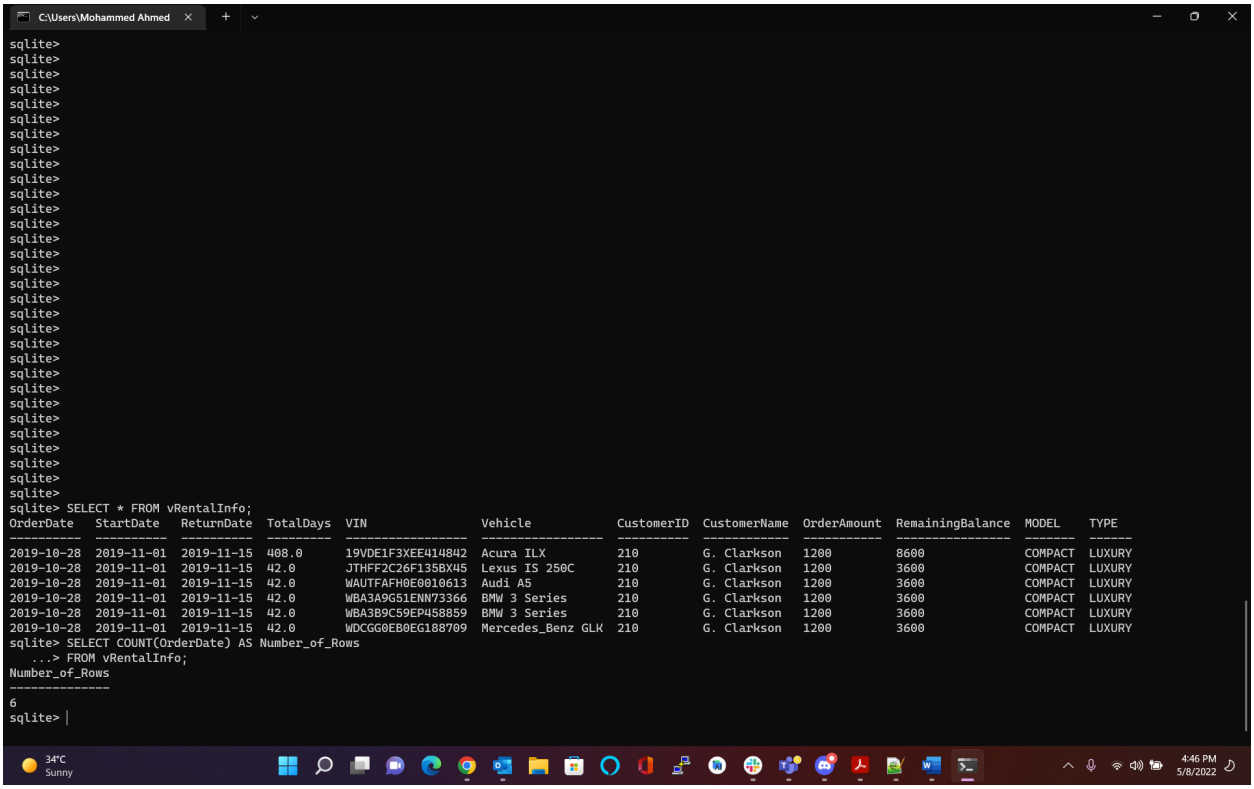
```
CREATE VIEW vRentalInfo AS
SELECT R.ORDERDATE, R.STARTDATE, R.RETURNDATE,
SUM(JULIANDAY(R.RETURNDATE) - JULIANDAY(R.STARTDATE)) AS 'TotalDays',
V.VehicleID AS 'VIN', V.Description AS 'Vehicle', C.CustID AS 'CustomerID',
C.Name AS 'CustomerName',
R.TotalAmount AS 'OrderAmount', SUM(R.TOTALAMOUNT) AS 'RemainingBalance',
CASE V.TYPE
WHEN 1 THEN 'COMPACT'
WHEN 2 THEN 'MEDIUM'
WHEN 3 THEN 'LARGE'
WHEN 4 THEN 'SUV'
WHEN 5 THEN 'TRUCK'
WHEN 6 THEN 'VAN'
END AS MODEL,
CASE V.CATEGORY
WHEN 1 THEN 'LUXURY'
ELSE 'BASIC'
END AS TYPE
FROM CUSTOMER AS C JOIN RENTAL AS R ON C.CustID = R.CustID
JOIN VEHICLE AS V ON R.VehicleID = V.VehicleID
WHERE R.PAYMENTDATE IN
(SELECT PAYMENTDATE FROM RENTAL WHERE PAYMENTDATE = 'NULL')
```

GROUP BY V.VEHICLEID

ORDER BY R.STARTDATE;

(2)

SELECT * FROM vRentalInfo;

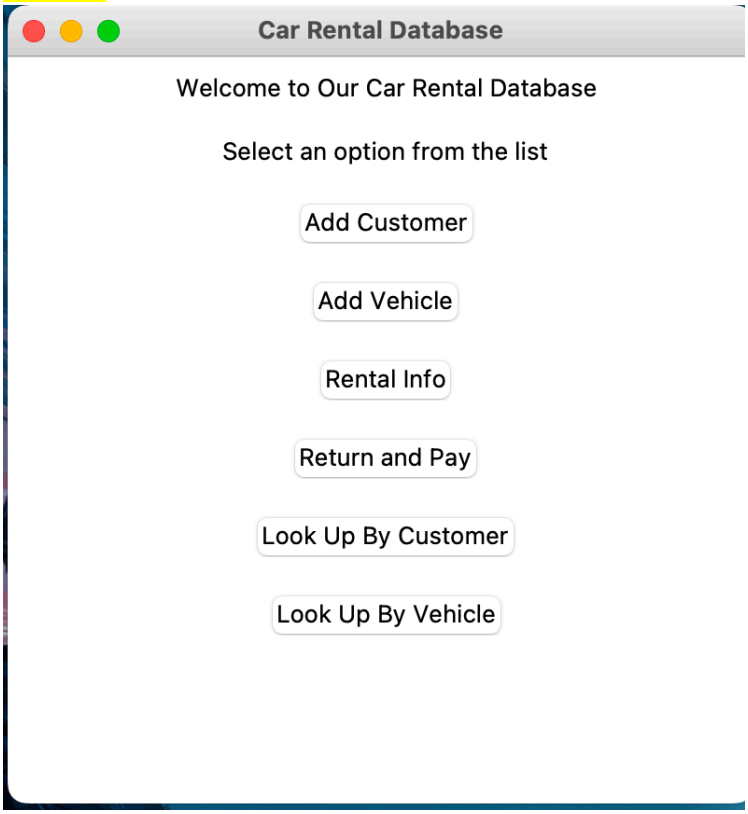


(3)

SELECT COUNT(OrderDate) AS Number_of_Rows

FROM vRentalInfo;

Task 2: Create a GUI for the CarRental2019 database:

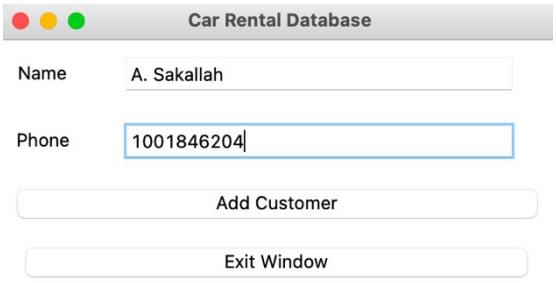


Requirement 1:

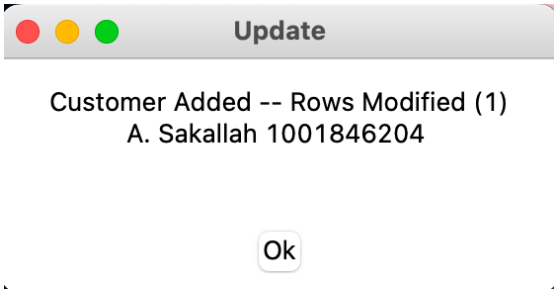
The first task is to enter information for a new customer. The user must provide his full name and a valid phone number. Once he has entered all his information he would receive a confirmation that his information had been successfully entered to the database.

INSERT INTO CUSTOMER (CustID, Name, Phone)

VALUES (NULL, 'A. Sakallah', '1001846204');




```
CustID  Name          Phone
------  ------------  ----------
232     A. Sakallah   1001846204
```



Requirement 2:

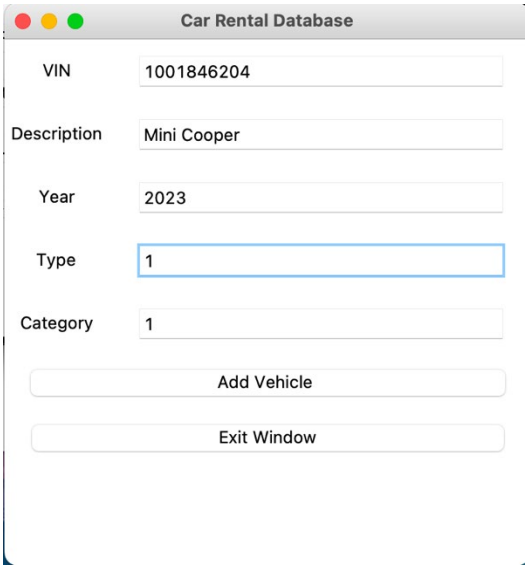
The new vehicle's information must be added as the second requirement. To enter new vehicle information, the user must click the chosen button, which will open a new window where he must fill in the relevant information, which includes VehicleID, Description, Year, Type, and Category. We can use the SELECT statement to check the information once it has been entered into the database.

INSERT INTO VEHICLE (VehicleID, Description, year, type, category)

VALUES(?, ?, ?, ?, ?);

To check if the data that the user has stored into the database

SELECT * FROM VEHICLES;



Requirement 3:

The third requirement is to update the details of a new rental reservation. As a result, this query will locate a free vehicle of the type and category specified by the user. We assume that the user must pay on the order date or on the return date.

(a)

SELECT V.Description

FROM VEHICLE AS V, RENTAL AS R

WHERE R.VehicleID = V.VehicleID AND R.Returned = 1

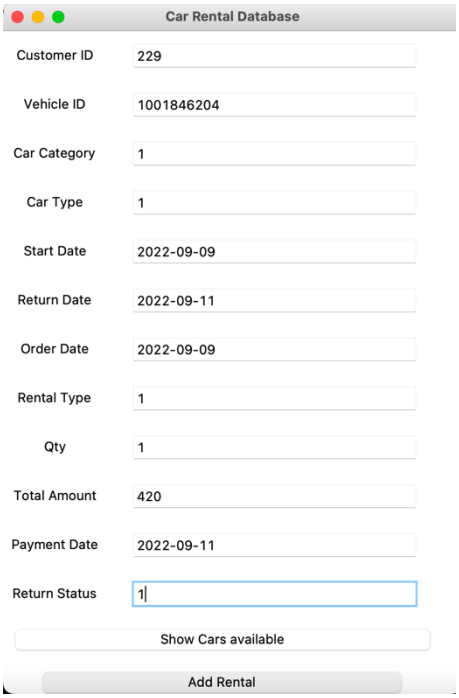
GROUP BY V.Description;

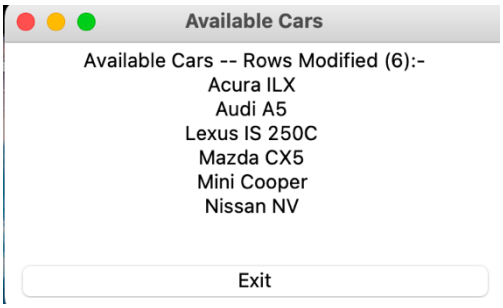
(b)

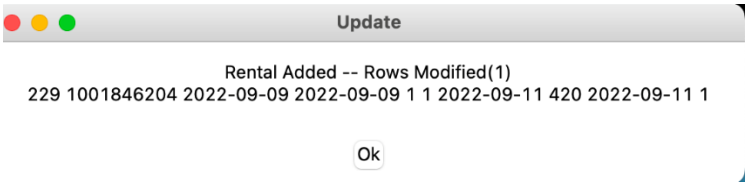
INSERT INTO RENTAL

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

VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?);







Requirement 4:

This requirement retrieves the user information from the CarRental2019 database and prompt the user to pay his/her bill with offering an option to pay or exit. It issues the payment for the user and updates the database with the new balance.

(a)

UPDATE RENTAL SET Returned = 1, PaymentDate = '2022-05-08', TotalAmount = 0

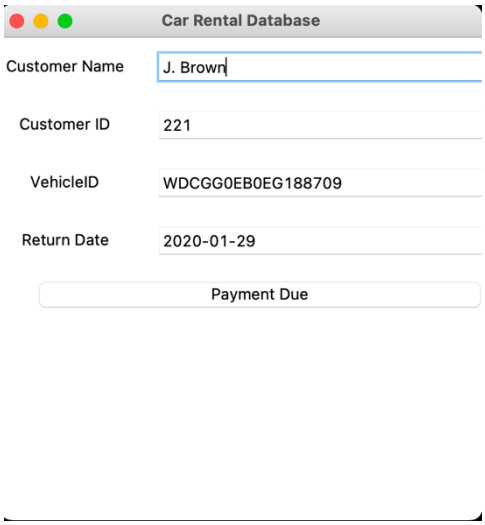
WHERE CustID = ? AND VehicleID = ?;

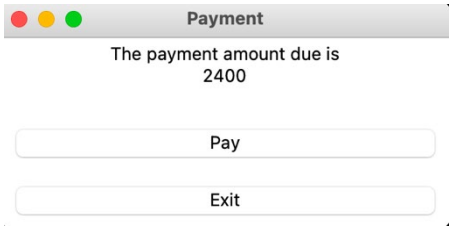
(b)

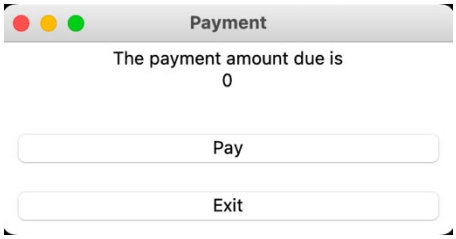
SELECT R.TotalAmount

FROM RENTAL AS R JOIN CUSTOMER AS C ON R.CustID = C.CustID

WHERE R.ReturnDate = ? AND C.Name = ? AND R.VehicleID = ?;







Requirement 5:

5A: -

This requirement prompts the user to enter information about a customer to retrieve by a customer ID or a customer name or neither. It displays the customer information for a corresponding answer based on what the user entered.

(a)

SELECT CustomerID, CustomerName, SUM(RemainingBalance)

FROM vRentalInfo WHERE CustomerName LIKE ?

GROUP BY CustomerName

ORDER BY COUNT(RemainingBalance) DESC;

(b)

SELECT CustomerID, CustomerName, SUM(RemainingBalance)

FROM vRentalInfo WHERE CustomerID LIKE ? AND CustomerName LIKE ?

GROUP BY CustomerID

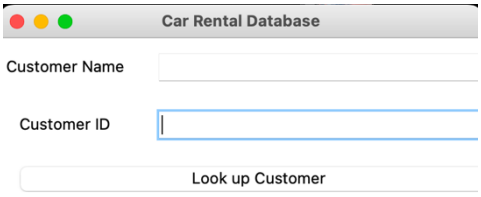
ORDER BY COUNT(RemainingBalance) DESC;

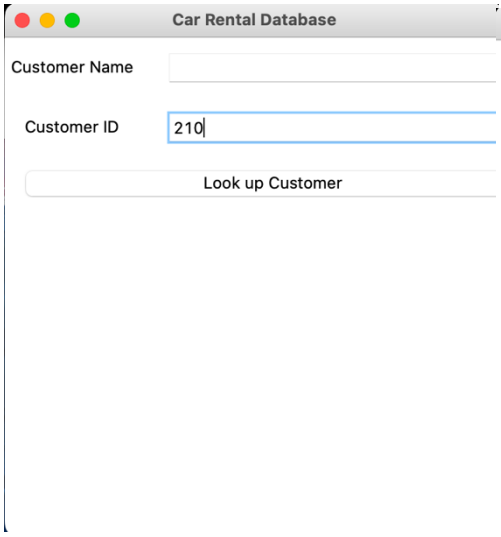
(c)

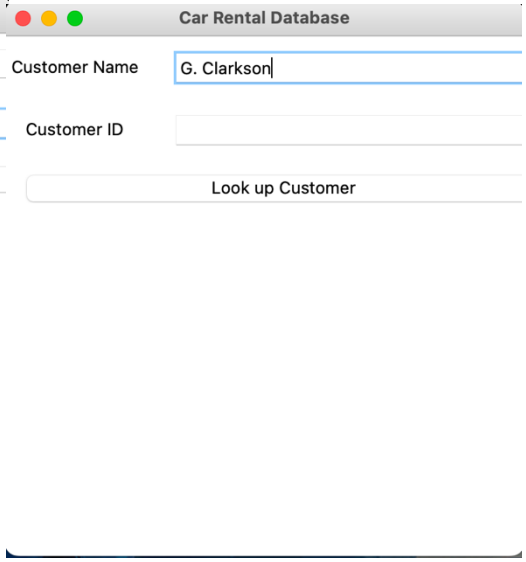
```
SELECT CustomerID, CustomerName, SUM(RemainingBalance)
FROM vRentalInfo
WHERE CustomerID LIKE ?
GROUP BY CustomerID ORDER BY COUNT(RemainingBalance) DESC;
```

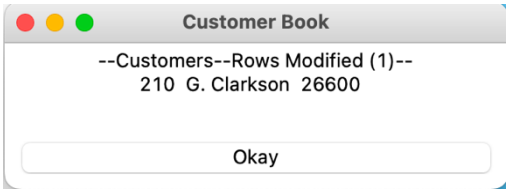
(d)

```
SELECT CustomerID, CustomerName, SUM(RemainingBalance)
FROM vRentalInfo
GROUP BY CustomerName
ORDER BY COUNT(RemainingBalance) DESC;
```









5B: -

This requirement prompts the user to enter information about a vehicle to retrieve by a Vehicle Description or the VIN or neither. It displays the vehicle information for a corresponding answer based on what the user entered.

(a)

```
SELECT VIN, Vehicle, (SUM(OrderAmount) / SUM (TotalDays)) AS DAILYPRICE
FROM vRentalInfo
WHERE Vehicle LIKE ?
GROUP BY VIN
ORDER BY (SUM(OrderAmount)/SUM(TotalDays)) ASC;
```

(b)

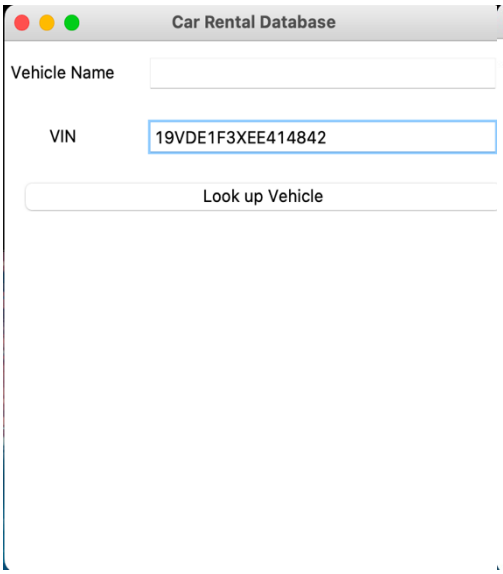
```
SELECT VIN, Vehicle, (SUM(OrderAmount) / SUM (TotalDays)) AS DAILYPRICE
FROM vRentalInfo
WHERE VIN LIKE ? AND Vehicle LIKE ?
GROUP BY VIN
ORDER BY (SUM(OrderAmount)/SUM(TotalDays)) ASC;
```

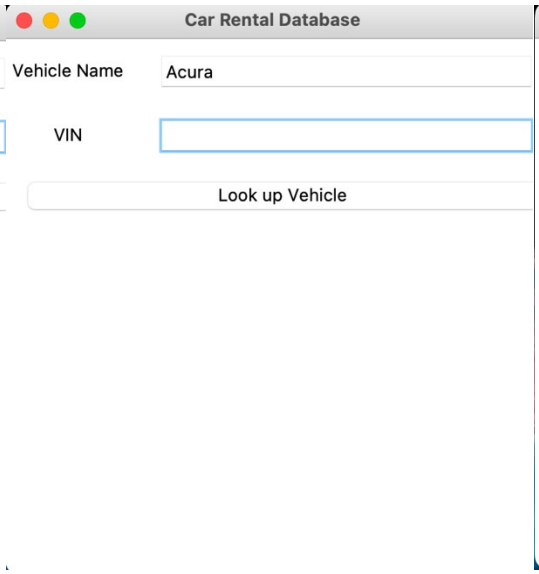
(c)

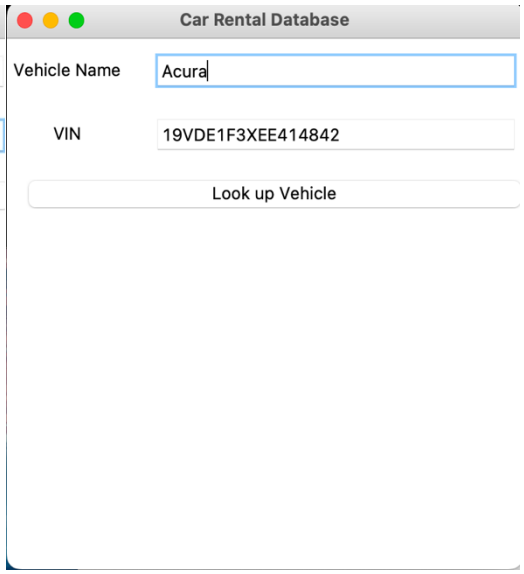
```
SELECT VIN, Vehicle, (SUM(OrderAmount) / SUM (TotalDays)) AS DAILYPRICE
FROM vRentalInfo
WHERE VIN LIKE ?
GROUP BY VIN
ORDER BY (SUM(OrderAmount)/SUM(TotalDays)) ASC;
```

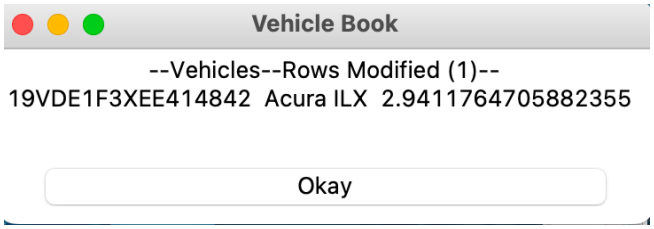
(d)

```
SELECT VIN, Vehicle, (SUM(OrderAmount) / SUM (TotalDays)) AS DAILYPRICE
FROM vRentalInfo
GROUP BY VIN
ORDER BY (SUM(OrderAmount)/SUM(TotalDays)) ASC;
```


Car Rental Database
Vehicle Name
VIN
19VDE1F3XEE414842
Look up Vehicle


Car Rental Database
Vehicle Name
Acura
VIN
Look up Vehicle


Car Rental Database
Vehicle Name
Acura
VIN
19VDE1F3XEE414842
Look up Vehicle


Vehicle Book
--Vehicles--Rows Modified (1)--
19VDE1F3XEE414842 Acura ILX 2.9411764705882355
Okay