

## 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;
```

```
C:\Users\Mohammed Ahmed > + ^ v
```

```
sqlite>  
sqlite>  
sqlite>  
sqlite>  
sqlite>  
sqlite>  
sqlite>  
sqlite>  
sqlite>  
sqlite>  
sqlite>  
sqlite>  
sqlite>.mode column  
sqlite> SELECT * FROM RENTAL;
```

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	0
210	19VDE1F3XEE414842	2019-11-01	2019-10-28	7	2	2019-11-15	1200	NULL	1
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
230	19VDE1F3XEE414842	2019-06-01	2019-04-04	1	1	2020-06-01	5000	NULL	0

```
sqlite>
```

4:35 PM  
5/8/2022

(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')
```

ORDER BY R.STARTDATE;

(2)

```
SELECT * FROM vRentalInfo;
```

```
C:\Users\Mohammed Ahmed > + ^
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite>
sqlite> SELECT * FROM vRentalInfo;
OrderDate      StartDate      ReturnDate     TotalDays    VIN           Vehicle          CustomerID   CustomerName   OrderAmount   RemainingBalance  MODEL        TYPE
-----
2019-10-28      2019-11-01      2019-11-15      408.0       19VDE1F3XEE414842 Acuna ILX         210         G. Clarkson    1200          8600             COMPACT      LUXURY
2019-10-28      2019-11-01      2019-11-15      42.0        JTHFF2C26F135B445 Lexus IS 250C     210         G. Clarkson    1200          3600             COMPACT      LUXURY
2019-10-28      2019-11-01      2019-11-15      42.0        WAUTFAFH0E0010613 Audi A5           210         G. Clarkson    1200          3600             COMPACT      LUXURY
2019-10-28      2019-11-01      2019-11-15      42.0        WBA3A9G51ENN73366 BMW 3 Series      210         G. Clarkson    1200          3600             COMPACT      LUXURY
2019-10-28      2019-11-01      2019-11-15      42.0        WBA3B9G59EP458859 BMW 3 Series      210         G. Clarkson    1200          3600             COMPACT      LUXURY
2019-10-28      2019-11-01      2019-11-15      42.0        WDCGG0EB0EG188709 Mercedes_Benz GLK 210         G. Clarkson    1200          3600             COMPACT      LUXURY
sqlite> SELECT COUNT(OrderDate) AS Number_of_Rows
...> FROM vRentalInfo;
Number_of_Rows
-----
6
sqlite>
```

(3)

SELECT COUNT(OrderDate) AS Number of Rows

FROM vRentalInfo;

**Task 2:** Create a GUI for the CarRental2019 database:

The screenshot shows a window titled "Car Rental Database" with a standard macOS-style title bar (red, yellow, green buttons). The window content includes a welcome message, a prompt to select an option, and seven buttons arranged vertically: "Add Customer", "Add Vehicle", "Rental Info", "Return and Pay", "Look Up By Customer", and "Look Up By Vehicle".

**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

The screenshot shows a small dialog box titled "Update" with a standard macOS-style title bar. It contains a single button labeled "Ok".

Customer Added -- Rows Modified (1)  
A. Sakallah 1001846204

The screenshot shows a window titled "Car Rental Database" with a standard macOS-style title bar. The window contains two text input fields: "Name" with the value "A. Sakallah" and "Phone" with the value "1001846204". Below the fields are two buttons: "Add Customer" and "Exit Window".

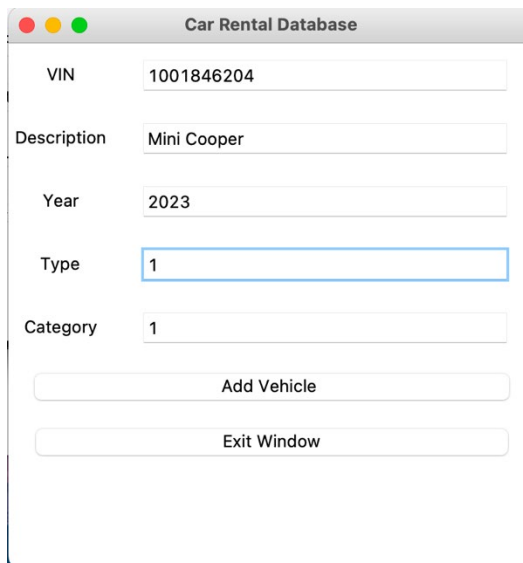
### Requirement 2:

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;
```



Car Rental Database

VIN: 1001846204

Description: Mini Cooper

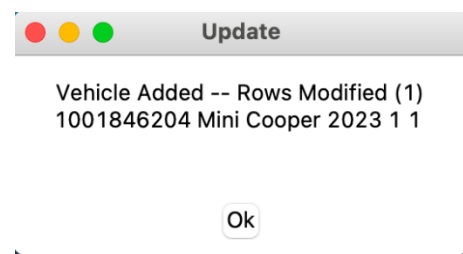
Year: 2023

Type: 1

Category: 1

Add Vehicle

Exit Window



Update

Vehicle Added -- Rows Modified (1)  
1001846204 Mini Cooper 2023 1 1

Ok

### 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 (?, ?, ?, ?, ?, ?, ?, ?, ?);

Car Rental Database

Customer ID: 229

Vehicle ID: 1001846204

Car Category: 1

Car Type: 1

Start Date: 2022-09-09

Return Date: 2022-09-11

Order Date: 2022-09-09

Rental Type: 1

Qty: 1

Total Amount: 420

Payment Date: 2022-09-11

Return Status: 1

Show Cars available

Add Rental

Available Cars

Available Cars -- Rows Modified (6):-

- Acura ILX
- Audi A5
- Lexus IS 250C
- Mazda CX5
- Mini Cooper
- Nissan NV

Exit

Update

Rental Added -- Rows Modified(1)

229	1001846204	2022-09-09	2022-09-09	1	1	2022-09-11	420	2022-09-11	1
-----	------------	------------	------------	---	---	------------	-----	------------	---

Ok

#### Requirement 4:

This requirement retrieves the user information from the CarRental2019 database and prompts 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 = ?;

Car Rental Database

Customer Name: J. Brown

Customer ID: 221

VehicleID: WDCGG0EB0EG188709

Return Date: 2020-01-29

Payment Due

Payment

The payment amount due is 2400

Pay

Exit

Payment

The payment amount due is 0

Pay

Exit

### 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;
```

The image displays three sequential screenshots of a 'Car Rental Database' application window, illustrating the 'Look up Customer' process. Each window has a title bar with three colored buttons (red, yellow, green) and the text 'Car Rental Database'.

**First Screenshot:** The 'Customer Name' and 'Customer ID' fields are empty. The 'Look up Customer' button is visible at the bottom.

**Second Screenshot:** The 'Customer ID' field contains the value '210'. The 'Look up Customer' button is visible at the bottom.

**Third Screenshot:** The 'Customer Name' field contains the value 'G. Clarkson'. The 'Look up Customer' button is visible at the bottom.

**Customer Book Dialog:** A dialog box titled 'Customer Book' is shown at the bottom. It displays the result of the lookup: '--Customers--Rows Modified (1)--' followed by '210 G. Clarkson 26600'. An 'Okay' button is at the bottom of the dialog.



**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