**DATABASE SYSTEMS AND FILE STRUCTURES**

**CSE 3330-003**

**Fall 2019**

**Car Rental Project Phase 3**

**Group Members:**

Sudeep Bhadel

Diwakar Parajuli

**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.**

Sudeep, Diwakar

**Task 1**

**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.

1. **SQL Commands:**
   * To add an extra column ‘Returned’:

ALTER TABLE rental

ADD COLUMN Returned CHAR(1);

* + To update the Returned column

UPDATE rental

SET

Returned = CASE

WHEN PaymentDate IS NULL THEN '0'

ELSE '1'

END;

1. **Result Grid Screenshot:**

SELECT \* FROM carrental2019.rental;

**A screenshot of a cell phone

Description automatically generated**

1. **Action Output Response**

Add column ‘Returned’ and select table

**A screenshot of a cell phone

Description automatically generated**

Update Returned attribute in rental table

**A screenshot of a social media post

Description automatically generated**

**Query 2:**

Create a view vRentalInfo that retrieves all information per rental. The view should have the following attributes:

• OrderDate

• StartDate – in an ascending order

• ReturnDate

• Total Days per Rental as 'TotalDays'– you need to change weeks to days

• Vehicle’s ID as 'VIN' • Vehicle’s Description as 'Vehicle'

• Vehicle’s Type as 'Type'– you need to use the description of the type

• Vehicle’s Category as 'Category' – you need to use the description of the category

• Customer’s ID as 'CustomerID'

• Customer’s Name as 'CustomerName'

• Order Total Amount as 'OrderAmount',

• Order Remaining Amount as 'RentalBalance',If there is no remaining balance return ‘0’

1. **SQL Commands:**

CREATE VIEW vRentalInfo AS

SELECT

R.OrderDate,

R.StartDate,

R.ReturnDate,

CASE R.RentalType

WHEN 1 THEN Qty

WHEN 7 THEN Qty \* 7

END AS TotalDays,

R.VehicleId AS VIN,

V.Description AS Vehicle,

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 Type,

CASE V.Category

WHEN 0 THEN 'Basic'

WHEN 1 THEN 'Luxury'

END Category,

C.CustID AS CustomerID,

C.Name AS CustomerName,

R.TotalAmount AS OrderAmount,

CASE

WHEN R.PaymentDate IS NULL THEN R.Totalamount

ELSE 0

END AS RentalBalance

FROM

customer AS C,

rental AS R,

vehicle AS V

WHERE C.CustID = R.CustID

AND R.VehicleID = V.VehicleID

ORDER BY R.StartDate ASC

1. **Result Grid Screenshot:**

SELECT \* FROM carrental2019.vrentalinfo;

A screenshot of a computer

Description automatically generated

1. **Action Output Response**

**A screenshot of a social media post

Description automatically generated**

**Task 2**

**Requirement 1: Add new Customer**

**SQL Query that code executes:**

INSERT INTO customer (Name,Phone) VALUES ('$name','$phone');

**GUI Screenshot:**

* Type in the required fields and Click ‘Add Customer’ to Add Customer

A screenshot of a cell phone

Description automatically generated

* After you click ‘Add Customer’, it’s prints out that the new customer is added to database

A screenshot of a cell phone

Description automatically generated

**Requirement 2: Add New Vehicle**

**SQL Query that code executes:**

INSERT INTO vehicle VALUES ('$vin','$description','$year','$type','$category');

**GUI Screenshot:**

* Type in the required information about vehicle and then click on ‘Add Vehicle’

A screenshot of a cell phone

Description automatically generated

* After that it prints out the message that new vehicle is added to database

A screenshot of a cell phone

Description automatically generated

**Requirement 3: Add information about new rental reservation**

**SQL Query that code executes:**

1. **To show available vehicles**

(SELECT Description

FROM vehicle AS V, rental AS R

WHERE

V.Type = $type AND V.Category = $category AND V.VehicleID = R.VehicleID

AND ((V.VehicleID NOT IN (SELECT

VehicleID

FROM

rental

WHERE

StartDate BETWEEN '$startDate' AND '$mrDate'))

AND (V.VehicleID NOT IN (SELECT

VehicleID

FROM

rental

WHERE

ReturnDate BETWEEN '$msDate' AND '$returnDate')))

GROUP BY V.VehicleID)

UNION

(SELECT Description

FROM

vehicle AS V

WHERE

V.VehicleID NOT IN (SELECT

VehicleID

FROM

rental)

AND V.Type = $type

AND V.Category = $category)

1. **To insert into rental**

INSERT INTO rental (CustID,VehicleID, StartDate,OrderDate, RentalType, Qty, ReturnDate,TotalAmount,PaymentDate) VALUES('$id','$vin\_no','$startDate','$orderDate',$rentalType,$qty,'$returnDate',$amount,'$pdate');

**GUI Screenshots:**

* Type the required information and click ‘Find available Vehicles’

A screenshot of a social media post

Description automatically generated

* Then, the dropdown menu will show the available vehicles for rental

A screenshot of a social media post

Description automatically generated

* After that make the choice and click on ‘Rent’ to rent that vehicle

**A screenshot of a social media post

Description automatically generated**

**Requirement 4: Return Vehicle**

**SQL Query that code executes:**

1. **To retrieve rental**

SELECT C.CustID,R.TotalAmount FROM rental AS R, customer AS C

WHERE C.Name = '$name'AND C.CustID = R.CustID

AND R.ReturnDate = '$return' AND R.VehicleID = '$vin'

1. **To update rental**

UPDATE rental

SET PaymentDate = '$payDate' , Returned = 1

WHERE CustID = $ID and rental.ReturnDate = '$returnDate'

and rental.VehicleID= '$vin';

**GUI Screenshots:**

* Type in the required fields to retrieve your rental information and click ‘Search’

A screenshot of a cell phone

Description automatically generated

* You can then see your due amount, click on ‘Pay’ to clear your balance

**A screenshot of a cell phone

Description automatically generated**

**Requirement 5(i): View Customer Info with remaining balance**

**SQL Query that code executes:**

SELECT customer.CustID, Name, SUM(IF(PaymentDate is null,TotalAmount, 0)) as TotalAmount

FROM customer LEFT OUTER JOIN rental ON (customer.CustID = rental.CustID)

WHERE customer.Name LIKE '$cust\_name'

AND customer.CustID LIKE '$id'

Group by CustID

ORDER BY TotalAmount

**GUI Screenshot:**

* You can filter your results by typing in the fields or check all the customer if the fields are empty

A screenshot of a social media post

Description automatically generated

**Requirement 5(ii): View Vehicle Information with Average Daily Price**

**SQL Query that code executes:**

SELECT V.VehicleID,V.Description,ROUND(AVG(TotalAmount / (R.RentalType \* R.Qty)),2) AS AverageDaily

FROM vehicle AS V LEFT OUTER JOIN rental AS R ON (V.VehicleID = R.VehicleID)

WHERE V.VehicleID LIKE '$vin' and V.Description LIKE '$description'

GROUP BY VehicleID

ORDER BY AverageDaily

**GUI Screenshot:**

* You can filter your results by typing in the fields or check all the vehicles if the fields are empty

A screenshot of a social media post

Description automatically generated