Car Rental Project - Phase 3: Due December 16th 2020 @ 11.59 pm

This is the last part of the Car Rental Project.
In this phase, you need to start with a clean installation and to load your database with the Car Rental data from Phase 2. If your database has any extra or altered data, you will lose points! For this phase, you will have to implement some more complicated queries, create a user interface, and create a video to upload to for

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

Query 1: [5 points]
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) Type the appropriate SQL commands in your submission report file.
(2) Type the select the RENTAL table command and paste the "Result Grid" screenshot. (3) Provide the action output response for all the above actions.

Query 2: [10 points]
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 zero '0'*

Submit:

1. Type the "**create view**" command text in your submission report file. Use the above names for your attributes' names. Do NOT change any name.
2. Type the "**select view**" command and paste the "result grid" screenshot after the 'Create view' SQL code.
3. Finally, type the "action output response", how many row(s) returned.

Task 2: Create a GUI for the CarRental2019 database:

Car Rental Application: [20 points]
Create a simple and friendly GUI interface that would be able to perform the following tasks. You may use JAVA programming using JDBC, or C/C++/C# programming with ODBC/Oracle or Python programming or PHP/MySQL or other programming languages to develop a GUI interface. You can always create a simple interface like a command prompt! For each query, you need to provide some info to the user about the query purpose. The user will have to type or select the query's input parameters and post the question to your program. The program needs to return all result's rows. You also need to print the count of rows returned.

Requirements:

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. **[3 points]**
2. The second requirement is to add all the information about a new vehicle. Submit your editable SQL query that your code executes. **[3 points]**
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. **[8 points]**
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. **[6 points]**
5. The fifth requirement is to return the **view's results** by applying the following criteria:
    a. List for every customer the ID, name, and if there is any remaining balance. The user has the right to search either by a customer's ID, name, part of the name, or to run the query with no filters/criteria. The amount needs to be in US dollars. For customers with zero (0) or NULL balance, you need to return zero dollars ($0.00). Make sure that your query returns meaningful attribute names. In the case that the user decides not to provide any filters, order the results based on the balance amount. Make sure that you return all records. Submit your editable SQL query that your code executes. **[10 points]**
    b. List for every vehicle the VIN, the description, and the average DAILY price. The user has the right either to search by the VIN, vehicle's description, part of the description, or to run the query with no filters/criteria. An example criterion would be all 'BMW' vehicles. The amount needs to be in US dollars. The average DAILY price derives from the rental table, and the amount needs to have two decimals as well as the dollar '$' sign. For vehicles that they do not have any rentals, you need to substitute the NULL value with a 'Non-Applicable' text. Make sure that your query returns meaningful attribute names. In the case that the user decides not to provide any filters, order the results based on the average daily price. Submit your editable SQL query that your code executes. **[10 points]**

Note: You can test your code by adding a few new customers, vehicles, rentals, or by changing some rental reservations dates; but make sure before you submit your phase 3 answers and before your demo presentation that your database has only the phase 2 data. Failing to do so will result in losing points.

Task 3: Demo Presentation

Record a video demo of your GUI Application and load it to the oneDrive Folder (link posted on Canvas). Thevideo recording must be named "teammate1lastname_teammate2lastname"**[15 points]**

Canvas Submission:
Create a folder with your code, named "Code" Inside the "Code" folder, create a Readme.docx file with step-by-step instructions and screenshots on how to install and run your program from scratch. Also, place a database backup (self-contained file), named as CarRental2019.sql inside your "Code" folder,

Create your submission file.
In the front-page, type a title for your project submission and specify your name, and your team member's name.
On the second page, include your honor code. Failing to do so will cost **[10 points].**
Next, you need to follow the guidelines from Task 1 and submit your work.
Edit your submission file and append (after the Task 1) tasks queries from Task 2, screenshots from your GUI for each task, along with some explanation or a meaningful label.

Save your submission file as "Report.docx" or "Report.pdf". Make sure that all commands are editable!

Create a folder named as "teammate1lastname_teammate2lastname" and place inside the "Code" folder and your "Report.docx" file.

Zip your folder and submit it on Canvas any time before the deadline.

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.

Students are required to not share any of the project related documents and solution with others in any way or form even after the completion of the project. Students may, however, show their projects to interviewers.

Late penalty: -5 points per hour