CSE2/4DBF 2021

Assignment 2 (20%) 19th

Due date: 11:59pm F~~ri~~day, October ~~15~~th 2021

Tuesday

AIMS AND OBJECTIVES:

- ✓ to perform queries on a relational database system using SQL;
- ✓ to demonstrate an advanced knowledge of stored procedures, stored functions and triggers.

This is an individual Assignment. You are not permitted to work as a group when writing this assignment.

Copying, Plagiarism: Plagiarism is the submission of somebody else's work in a manner that gives the impression that the work is your own. The Department of Computer Science and Information Technology treats plagiarism very seriously. When it is detected, penalties are strictly imposed. Students are referred to the Department of Computer Science and Information Technology's Handbook and policy documents with regard to plagiarism and assignment return, and also to the section of 'Academic Integrity' on the subject learning guide.

No extensions will be given: Penalties are applied to late assignments (5% of total assignment mark given is deducted per day, accepted up to 5 days after the due date only). If there are circumstances that prevent the assignment being submitted on time, an application for special consideration may be made. See Student Handbook for details. Note that delays caused by computer downtime cannot be accepted as a valid reason for a late submission without penalty. Students must plan their work to allow for both scheduled and unscheduled downtime.

SUBMISSION GUIDELINES:

Task 1 should be saved to a file named task1.txt.
Task 2 should be saved to a file named task2.txt.
Task 3 should be saved to a file named task3.txt.

All the tasks above are to be submitted in soft-copy format using the CSE2/4DBF submission link provided on LMS by 11:59pm, October 15th, 2021.

SUBMISSION CHECKLIST:

- ✓ The relevant SQL queries for the 'ServiceMatch' Database System.
- ✓ The required stored procedures, stored function, and triggers.

NOTE: No built-in ORACLE column numbering (such as ROWNUM) or other ORACLE ranking facilities (such as RANK) can be used in this assignment.

Implement the following tasks using ***ORACLE Application Express (APEX)***.

Download the file `ArchieSchema.sql` from the LMS site and run it on APEX. This file contains all the CREATE and INSERT statements you will need for this assignment.

Run the schema file on APEX following the same process we followed to run a schema during the labs (watch week 7 lab recording).

NOTE: YOU DO NOT NEED TO INSERT MORE DATA INTO THE TABLES.

The list of tables available for this assignment is the following:

Manufacturer(manufacturerID, name, region)

Model(modelNo, name, type, *previousModel, manufacturerID*)

Car(VIN, dateAcquired, yearBuilt, purchasedPrice, askingPrice, currentMileage, color, *modelNo*)

Feature(featureID, desc, category)

CarFeature(*VIN, featureID*)

Customer(custID, name, DOB, streetAddress, suburb, postcode, gender, phoneNo, email, type)

CustomerPreference(*custID, featureID*)

SalesAgent(agentID, name, DOB)

SeniorAgent(*agentID*, yearPromoted)

JuniorAgent(*agentID*, *supervisor*)

SalesTransaction(*VIN, custID, agentID*, dateOfSale, agreedPrice)

ViewingParty(viewingPartyID, contactNo, email)

Organisation(*viewingPartyID*, name)

InternationalGuests(*viewingPartyID*, country)

CarsViewed(_VIN, viewingPartyID,_ dateViewed, amountPaid)

NOTE: PK is printed underlined, and FK is printed in italics.

Task 1 [50 marks]

Using the tables provided above, provide **SQL** statements for the following queries.

a. Display the name of the customer who has purchased the most cars from Archie's Luxury Motors.

b. For each sales agent, display their ID and name, along with the total number of sales they have made thus far. Order by number of sales decreasing.

c. For each month in 2020, display the total profit (i.e., using the attributes *purchasedPrice* and *agreedPrice*) generated from car sales only. Do not consider any discounts.

d. Display the details (i.e., Manufacturer name, model name, type, and the number of times it was sold) of the top selling European car model. Hint: use the manufacturer region information.

e. Display the average number of sales transactions (i.e., car sales) per month. Hint: count the number of sales for each month, then divide the count by the number of years the dealership has been making transactions for.

f. Display the total profit to date for Archie's Luxury Motors. Note that you must take into consideration all car sales (ignoring unsold cars as they may be sold later) and viewing party shows, and any VIP discounts that may apply. Do not subtract any commission owed to the senior sales agents.

   - Note: A VIP customer gets a 5% discount for any car purchase greater than or equal to $50,000 AUD. However, the agreed price stored in the database is without the discount. For each car sold to a VIP customer with an agreed price of $50,000 AUD or more, you should subtract this 5% discount from the agreed price when calculating the profit. Also note that the profit from each viewing party show is equal to the amount paid for the show.

[a – e: 8 marks each, f: 10 marks – 50%]

Task 2 [35 marks]

Provide the implementation of the following stored procedures and function. For submission, please include both the PL/SQL code and code to execute (for the procedure)/SQL statement (for the function) to demonstrate the functionality.

a. Write a **stored procedure** that accepts a particular year as input, and as output displays the number of cars sold grouped by the 3 mileage groups (Low Mileage: <50000km, Medium Mileage: >=50000km & <150000km, High Mileage: >=150000km). Also display the total number of cars sold overall.

b. Write a **stored function** that uses a senior agent's ID as input and calculates the total commission owed to date for that agent. You also need to show an SQL statement to display the total amount of commission (i.e., the sum) owed to all senior agents in the database.

- Note: the commission policy states that a senior agent receives an additional 1% commission multiplier for each year since they became a senior agent, *rounding down to the nearest full year*. This means that a senior agent who was promoted 4.25 years before the date of sale, would have their commission calculated as (agreedPrice – askingPrice) × 4%. Also, a senior agent receives the commission only when the agreedPrice is greater than the askingPrice stored in the database. The asking price in the database is not visible to customers.

[a: 20 marks, b: 15 marks – 35%]

Task 3 [15 marks]

Provide the implementation of the following trigger. For submission, please include both the PL/SQL code and an insert statement to demonstrate the trigger functionality.

a. A **Trigger** which automatically stores in a separate table called 'ExcellentSale' the Sales Agent name, car model and manufacturer name, each time the agreed price of a `SalesTransaction` is more than 20% above the car's asking price. (Note: You need to create the 'ExcellentSale' table before implementing this trigger. To create the primary key, use a sequence that starts at 1 and increments by 1).

[15 marks – 15%]