phase3

Group members:
Johnathan Murad(sid: 861227537) queries(1,2,3,9,10) jmura003@ucr.edu
Haasith Sanka (sid:861215756) queries(4,5,6,7,8) hsank001@ucr.edu
12/14/2018
CS166 DBMS
Instructor V.Tsotras
Section 022

In completing this assignment, we consulted no one.

All the important code is original.

Any assumptions made:

addCustomer()

- No assumptions, simply request input from customers for their name,email, phone number and address and insert in to the database.

addMechanic()

- No assumption, also simply request input from the mechanic name and year of experience

addCar()

- One assumption i made was after adding a car in to the database, assign an owner of the car to a customer.

InsertServiceRequest()

- No assumptions made, simply followed the specs as provided.

closeServiceRequest()

- Assumption that the open date is before the closed date

Implementation descriptions:

1. public static void AddCustomer(MechanicShop esql) // completed by Johnathan Implementation
    a. For Primary Key “id”, i SELECT id FROM Customer and store it in List<List<String>>to get the size() of the id and use that as the unique value for the id.
    b. Take the users input for their First name, Last name, phone number, and email and check for possible cases where the input may be invalid
    c. INSERT INTO the customer table with the values of id, and the user inputs of lname, fname, phone number and address

2. public static void AddMechanic(MechanicShop esql) // Completed by Johnathan

Implementation:

a. For Primary Key “id” SELECT id FROM Mechanic and store the result in List<List<String>> and get the size of the List to use as the new id for the mechanic
b. Get the users input for fname, lname and years of experience
c. Check that the fname, lname are valid in length.
d. Check that years of experience is between 0 and 100
e. Then insert into the mechanic table the values selected

3. public static void AddCar(MechanicShop esql) // Completed by Johnathan
   Implementation:
   a. For the Primary key VIN, We ask the user for the VIN of their vehicle checking to make sure that the VIN entered is unique and follows the format of 6 LETTERS followed by 10 integers as prompted when asked for input
   b. Make,model, and year and user inputs from the user.
   c. Year is checked to be sure it is greater than or equal to 1970.
   d. Then in this query, i also create the relationship between the car added and the customer existing within the database. I implemented this by asking for the fname and lname of the Customer to be added as an owner. Then i insert the ownership_id, vin, and customer id. I get the ownership_id by storing the result of SELECT ownership_id FROM Owns; in to a List<List<String>> and then get the size of the string.
   e. I then insert in to the car table the values VIn, make, model, year and then i insert in to owns the values ownership_id, customer_id, car_vin.

4. public static void InsertServiceRequest(MechanicShop esql) // Completed by Haasith
   Implementation:
   a. We get the primary By selection the rids from the service request table in to a List<List<String>> and getting the size of the List.
   b. We then ask the user for their last name and print the tuples of Customers sharing the same last name
   c. If the customer doesn’t exist, add a new customer.
   d. The user then is asked to pick their cid of the existing customers
   e. The VINs of a customer are then displayed and the user has the choice of selecting which of their cars they want to initiate
   f. The user is then asked for their odometer reading, making sure it greater than the previous visit,the complaint on the car, and the date initiated. All these values (rid,customer_id, car_vin, date, odometer, complain) are then inserted in the table.

5. public static void CloseServiceRequest(MechanicShop esql) throws Exception // Completed by Haasith
   Implementation:

    a. Wid is set by selecting all of the closing id's and storing them in to a List<List<String>> then picking values greater than the size.
    b. We then ask the user to input their rid and we make sure it is a valid rid.
    c. We then ask the user for the mechanic ID and make sure that it is also a valid mechanic id
    d. We then search for the open date of the service request, and ask the mechanic to enter a date for the closing request and make sure the date is valid based on the open date.
    e. The mechanic then enters the bill and any comments they have
    f. The values wid, rid, mid, date, comment, bill are entered in to the table.

6. public static void ListCustomersWithBillLessThan100(MechanicShop esql) // completed by haasith
    a. Simply select the date, comment and bill where teh bill is less than 100
7. public static void ListCustomersWithMoreThan20Cars(MechanicShop esql) // completed by haasith
    a. Select the fname and lname from the customer and from the count of 20 or more cars and connecting the customer_id to the ones who have a count of greater than 20.
8. public static void ListCarsBefore1995With50000Milles(MechanicShop esql) // completed by haasith
    a. Selecting the make, model, and year from car and service request to see if the year is less than 1995 and has an odometer less than 50000. Assuming no duplicates so selection is distinct
9. public static void ListKCarsWithTheMostServices(MechanicShop esql) // Completed by Johnathan
    a. Selecting the make, model and the number of service requests being a selection of the car_vin and the count of the rids from service request
10. public static void ListCustomersInDescendingOrderOfTheirTotalBill(MechanicShop esql)//completed by Johnathan
    a. Selecting the fname, lname and the total cost being a selection of the sum of the bill from closed request and the id's of the service request making sure the id's from service request and closed request match and then making sure the customer ids match from the service request id.

Also created indexes in create.sql