COEN 178 Intro. To Database Management Systems Winter 2019

Project (Up to three people per group allowed) Total: 200 pts

1.0 Introduction

The Repair department of the *Meteor Car Dealership* is planning to automate its record-maintenance system. The application, ***Meteor RepairRecord System***, to be designed is to assist the management of the *Meteor dealership* with the following tasks:

a. Maintain a detailed list of the customer records with information of the repair jobs done. The management uses the customer records to schedule the routine maintenance dates for its regular customers.
b. Maintain a record of the repair jobs to assist the management in customer billing, compute the mechanic's pay.
c. Maintain a record of all the mechanics, the repair-jobs they are responsible for and their work hours. This information helps the management compute the total pay of each mechanic. Additionally, the information helps the management assess the future hiring needs and allocation of duties to the mechanics.
d. All the information in the database helps the management assess their monthly and yearly revenues.

1.1 Functional Requirements

The enterprise-specific details of the repair shop which are relevant to designing the database are given below.

A **car** that is brought to the repair shop with one or more problems, where each problem is identified by a **problem type (**transmission problem, wheel alignment **etc)** and a problem id *number*. A car has a *car_licence-number*, *model of the car*, *phone* and/or an *email address of the customer*. A customer is identified with a name, phone (or email), and address. Phone or address is regarded as unique.

When a car is brought in for repair, a **repair job** is created for that car, with a *repair job id*. A single **mechanic** is assigned to each repair job. The information that is maintained for each mechanic includes the *employee_Id, name, phone* and *hourly pay rate*.

The data that is maintained for a **repair job** includes the following information: the *car_licence-number*, *customer contact info* (phone or email), *time_in* (the time car is brought in), *time_out* (the time when the car is ready to be picked up), the *problems ids associated with the car*, *mechanic_Id (*s), *labor hours*, name and the cost of the parts used.

When a RepairJob is completed, a Customer Bill is created.

A mechanic is paid by the hour ($20.00 an hour). A customer is billed a minimum of $30.00 service fee for the repair job + cost of parts used and labor ($25 dollars an hour).

Note: You are free to include any additional attributes you may require for the entities described in this document.

1.2 Integrity constraints and business rules to be enforced

a. When RepairJob is completed on a car and before the information is deleted, it should be saved into a separate log table.

b. A customer who has brought in a car for repair (within the same year) is given a 10% discount on subsequent repair jobs (need not be on the same car).

1.3 The system should support the queries to do the following

a. Create a **RepairJob** for a car when it is brought in. This will have information for all the fields except for *time_out* (the datetime when the car is ready to be picked up), the *problems ids associated with the car*, *labor hours*, name and the cost of the parts used.

b. Create a customer's bill based on the RepairJob. This should show the customer's information (name, telephone(s), address), model of the car, time (and date) the car brought in, datetime the car is ready, billing information which includes the

name and description of each repair or service performed, charge for each service, names and price of each part used, labor and the service fee and the total amount due.

c. Show a listing of all the repair jobs done between two given dates. The listing should include the model of the car, id and the description of the repair job and the mechanic in charge of the repair.

d. Show a listing of the mechanics (employeeId and the name) who worked the most number of hours, the mechanic who worked the least number of hours and the average number of hours each mechanic worked.

e. A listing showing the amount of money generated from the customer billing (between two specific dates)**.**

2.0 Implementation

Implement the application, **Meteor RepairRecord System** using the requirements given below:

a) The application, **Meteor RepairRecord System,** is an on-line, **web-based (client-server) application** with a relational database backend.

b) The **relational database** is designed to maintain the records of customers, mechanics, and repair jobs for *Meteor dealership*, keeping in mind that the database should support the data requirements to serve the car-dealership in its daily tasks of creating repair jobs, generating customer bills and provide support for queries to find mechanics that worked the most number of hours per week, their earnings etc.

c) The front end of the application uses the browser and an end-user interacts with the application via the browser. When the application starts, the browser displays the home page of the dealership with a list of options that the user can select from. Each of the options refers to a query that you have implemented for this system.

For example, you may show options to a) Create Repair Job b) Generate customer Bill c) Show repair jobs between dates d) Find Mechanic's pay etc. Use the queries in 1.3 to create your option list.

d) The option to create a repair job should display a web form to be filled by the user and submit it via the browser. The information that is submitted is processed on the server side using a PHP program that accesses the relevant tables in a relational database.

e) When the option to generate a customer bill is selected, the bill is displayed in the browser.

2.1 Tools and Technologies

- You are required to use **Oracle** as the relational database, **HTML** and **PHP** as the Web technologies. You are free to use any other additional tools you may be familiar with.

- **You must use at least one PLSQL procedure or function and at least one trigger.**

2.2 Requirements by Team size

Three people teams are required to implement all the queries in 1.3 and items, **a** to **e** in 2.0 as part of a web application.

Two people teams are required to implement all the queries in 1.3. and items, **a, b, d** and **e** in 2.0 as part of a web application.

Note: If you plan to work on your own, please check with me.

3.0 Deliverables

Deliverable 1 (25 pts)

i) A detailed, conceptual design using the E-R model should be included, showing the entities, relationships, cardinalities and integrity constraints. Any diagramming tool may be used, but a detailed legend to identify the notation used, should be included (you may be given recommendations for an automated tool).

Deliverable 2

ii) The process of translating the E-R model into a relational model should be clearly shown with the resulting tables. Show any normalization process that is applied to each of the tables and identify if the tables are in 3NF and BCNF. Clearly identify the primary keys and foreign keys**. (25 pts)**

iii) The script files to create the tables using SQL and load the tables with data of your choice. The script files and program files to implement the queries that are necessary to offer the functionality required by the dealership. **(50 pts)**

iv) A demo of your implementation (in week 10) (100 pts)