CET1031 - Databases and Data Modelling - Practicum 3

Topics Covered: E-R Diagrams and Transactions

Learning Objectives:

- Familiarizing with drawing an E-R Diagram for a database system.
- Implementing the database system from the E-R Diagram.
- Familiarizing and applying transaction theory

Deliverables

- A zip file of consisting of a `.sql` file, a `.png` file & a `.pdf` file using the given filenames as defined in this document, the naming format of the zip file is `CET1031_P03_<Your_Name>.zip`. e.g. `CET1031_P03_John_Doe.zip`.
- Note that a non-loadable submission will result in a zero

Important Information
There are 2 parts to this practicum: E-R Diagram and Transactions

For the **E-R Diagram** part, you will be designing and implementing an E-R diagram. The E-R diagram is to be created using `draw.io` or any graphical program of your choosing, resulting in a `.png` **file** and the implementation is to be done in a `.sql` **file**.

For the **Transactions** part, create **a typed report (** `.pdf` **file)** with the answers to the questions listed in this document. Any created diagrams can be created using `draw.io` or any graphical program of your choosing. The diagrams are to be **embedded to your report**.

E-R Diagram Part

Your task is to design and implement an database system for a fictional Vehicle Repair Company.

The company's database needs to record the following:

- The various repair centers that the company has. Each repair center as a center id, an address and an hourly rate. Each repair center also hires their own employees, processes service tickets and has repair bays.
- Each repair bay has a slot id, records the work done, date of repairs, the mechanic that does the repairs and the resulting invoice number.
- Employee's data consisting of employee's id, name, salary. There are also 2 types of employees, namely mechanics and administrative staff. The mechanics have different ranks and the administrative staff have different job titles.
- Each problem has to be recorded with the description, hours to fix and a id.
- Vehicles that comes in, will be given a service ticket. Data required from the vehicles (which the service ticket can indirectly access) are the year, model, make and a unique vehicle id. Additional data accessible from the service ticket are the unique vehicle id and the repair bay's id.
- Each service ticket has a ticket id and the date of which the ticket was created.

The relationships between the various parts of the company:

- The repair center has more than 1 employees
- The repair center has more than 1 repair bay
- Each mechanic can solve more than 1 problem
- Each mechanic can work in more than 1 repair bay
- Each repair bay can service more than 1 service ticket
- Each service ticket addresses 1 problem
- Each vehicle comes in with at least 1 service ticket

Task 1 - Design the E-R diagram

Design the E-R diagram (using either Chen's or Crow's Feet notation) based on the company's requirements and their stated relationships. **Name this file `er_diagram.png`.**

Task 2 - Create the tables for this Database

Create the tables for this database in a `.sql` file with all required constraints. Include any other constraints that you think the tables requires that is not already defined in the description. **Name this file `vehicle_repair_db.sql`.** Note that this file **has to be repeatedly loadable** via MariaDB's `mysql` client terminal.

basically you need to create the DDL and make sure your integrity constraints are applied.

Transactions Part

Question 1

Consider the following schedule S_2 :

$$S_2 = r_1(P), r_2(R), r_3(P), r_1(R), r_2(Q), r_3(Q), w_1(P), w_2(R), w_3(Q), w_2(Q)$$

a) Draw the precedence graph of schedule S_2.
b) Is S_2 conflict serializable? Explain.
c) If S_2 is conflict serializable, write down all equivalent serial schedules.
d) If S_2 is not conflict serializable, make the schedule conflict serializable. Show all working required to make this schedule conflict serializable, include the new precedence graph and changes made to the schedule.

Question 2

Imagine that you are writing some procedures for an e-commerce website that allows new users to register with the site. The following pseudocode is given:

1: $\ldots\ get\ login\ from\ the\ user\ inputs,\ in\ variable\ newLogin\ \ldots$
2: $exists = SELECT\ COUNT(*)\ FROM\ users\ WHERE\ login = newLogin;$
3: $if\ (exists == True)$
4: $\quad \ldots\ give\ error\ msg\ and\ start\ over$
5: $else$
6: $\quad \ldots\ get\ firstName,\ lastName,\ password\ from\ user\ inputs\ \ldots$
7: $INSERT\ INTO\ users\ VALUES\ (newLogin,\ firstName,\ lastName,\ password,\ now());$

The `user` table in the database is created with the following DDL:

```
CREATE TABLE Users (
    login VARCHAR(20) PRIMARY KEY,
    firstName VARCHAR(30), lastName VARCHAR(30),
    password CHAR(20) CHECK (CHAR_LENGTH(password) >= 8),
    dateJoined DATE,
    description VARCHAR(250) );
```

a) For the program specified above, what atomicity problems could arise if it was not run as a transaction?
b) For the program specified above, what isolation problems could arise if it was not run as a serializable transaction?
c) Compare what would happen if the program above was run as a transaction with isolation level **SERIALIZABLE** compared to if it was run with isolation level **READ COMMITTED**. Point out benefits and drawbacks of the two choices and give a recommendation for the best isolation level for this particular problem.

For the maximum allocation of marks, refer to the table below.

Description	Marks (%)
Properly formated E-R Diagram of task 1	20
Repeatedly loadable `.sql` file with appropriate constraints for task 2	30
Transactions - Complete question 1	20
Transactions - Complete question 2	30