Coursework - Database Design & Implementation

Introduction

The coursework for COMP23111 is a singular piece split into two parts each with their own submission format and deadline. Please ensure you read these instructions carefully to ensure you are confident in what needs to be submitted for each milestone.

The goal for Milestone 1 is for you to demonstrate the technique of deriving a conceptual model in the form of an (**enhanced**) **entity-relationship** (EER) diagram from a data requirements specification. The outcome of this step is a conceptual model from which a logical model of the data can then be derived. Then you should normalise the data and create a relational schema ready for implementation.

Milestone 2 is all about your implementation and SQL skills. You will be creating the database based on the designs you produced in Milestone 1, inserting some data and producing queries based on the tasks outlined in Milestone 2. You will also be required to implement a simple front end which your database must connect to. The ability to connect a database to a front end is a fundamental skill needed by all database engineers. Download a suitable MySQL server for your operating system. This will make the task easier in the long run. You may use the provided version of PhPMyAdmin (web.cs.manchester.ac.uk), details of this can be found in Lab 04.

Contents

Chapter 1

Milestone 1 - Database Design

ERD

The first part of Milestone 1 is to analyse the Material section below. In there you will find all of the information about our COMP23111 company “Kilburnazon”. You need to take all the information provided and design an ERD capturing all of the information. Your ERD must be in Crow’s Foot notation, other notations will not be accepted. Examples of suitable ERD’s can be found in Lab 02’s Solutions.

You should draw your finished diagram neatly and clearly so that it looks professional and is easy to read. You can do this with pen and paper, or there are many free tools available (there’s no need to pay for a drawing tool for this). I suggest you try Lucidchart.com, draw.io or Visual Paradigm Online. Or you can of course use any other drawing tool you like. Top tip: work out your diagram with pen and paper first, and only draw it neatly once you’ve got it worked out. Otherwise you can waste a lot of time. The logic should come first; make it look nice later.

Normalisation & Relational Schema

It is important that you limit the amount of data redundancy and avoid insert, update and delete anomalies when you come to implement your database (see milestone 2 for implementation details). Therefore, you should use the normalisation techniques that you have learnt and demonstrate your understanding of 1NF, 2NF and 3NF by following the rules at each stage. Normalising all the information you have captured, and your design should divide larger tables into smaller tables and link the tables using relationships. This will strengthen your design, remove redundant data, and ensure data is stored in the most logical way possible to help avoid insert, update and delete anomalies.

When reporting your design, you should use the following notation:

```
Relation_Name (Attribute_1, Attribute_2, ... Attribute N)
    FK Attribute_Name → Relation_Name (Attribute_Name)
    ON DELETE CONSTRAINT, ON UPDATE CONSTRAINT
```

- The attribute(s) that form the primary key should be underlined (not the foreign keys)
- The relations should be named appropriately and represent the information being stored (as should the attributes)

- Any foreign keys should be identified under the relation with their associated table and attribute, and the delete and update constraints.

Report Format

For Milestone 1 you are to submit a report following the general structure outlined below. However, this is just for guidance, and you should structure your report as appropriate with headings and subheadings.

- Coverpage (your name, title of assessment, module code, date)
- Table of Contents
- ERD
  - Introduction to section
  - Your ERD
  - A short report (Max 500 words) outlining your design choices
- Normalisation
  - Introduction to section
  - Your relations in 3NF
  - A short report (Max 500 words) outlining your design choices and at which normalisation stage
- Relational Schema
  - Introduction to section
  - Your Schema
  - A short report (Max 500 words) outlining your design choices

Submission

Submit a single PDF to the appropriate place on Blackboard,
23111-Cwk1-S-Database Design.

Deadline:

18:00 on Friday the 11th November

Assessment Type:

This activity is subject to **summative** assessments regulations, therefore your submission will be marked and you will receive the associated feedback. The marks you obtain count for up to 20% of your overall mark for this unit.

Chapter 2

Milestone 2 - Implementation

Database Creation

实施你的设计，注意你的字段的数据类型和外键的任何参考行动，以更新或删除相关表中的数据。通过一系列的插入MySQL命令将一些数据插入你的数据库。我们在employees.csv中提供了随机生成的数据，你可以在你的实现中使用，你可能需要调整列名和格式以适应你的设计。对于这一部分，你将只查询整个数据库的一小部分，因此不需要填充每个单独的表，但每个表必须用适当的约束条件创建。

Implement your design being mindful of the data types for your fields and any referential actions for foreign keys to update or delete data from related tables. Insert some data into your database through a series of `INSERT` MySQL commands. We have provided randomly generated data in employees.csv which you can use in your implementation, you may need to adjust column names and formats to suit your design. For this part you will only be querying a small portion of the overall database and therefore do not need to populate each individual table but each table must be created with the appropriate constraints.

The Front End

使用PHP和MySQL为你的数据库创建一个应用程序和前端。

Using PHP and MySQL create an application and front-end for your database. The application should:

1. Allow a user to add a new employee to the system, the details for the employee are:

   emp_id: 55-3623151

   允许用户在系统中增加一个新的雇员

   name: Malissia Osgardby

   address: 29416 Grover Alley

   salary: £17424.03

   dob: 26/12/1989

   nin: it152291r

   department: Driver

   emergency_name: Marcie Prattington

   emergency_relationship: Mother

   emergency_phone: 07297 230 400

   一名员工已经晋升，因此有了新的工资，他们的紧急联系人的电话号码也发生了变化。请使用您的系统更新雇员编号的详细信息。07-4517183，使他们的新工资为33,027.23英镑，他们的紧急联系人的新电话号码为07231 462 728。

2. An employee has been promoted and therefore has a new salary and their emergency contact's phone number has also changed. Please use your system to update the details of Employee Id: 07-4517183 so that their new salary is £33,027.23 and their emergency contacts new phone number is 07231 462 728.

3. Employee 71-7374760 has left the company so please delete their record and any associated data.

4. Display all employees who are a Driver and their emergency contact is their Father. The output should display the employees name, their department, the emergency_relationship and the name of their manager.
5. The CEO want to send out Birthday Cards to each employee. Create a stored procedure that displays all employees whose birthday is in the current calendar month.
6. The CEO thinks it would be wise to keep a log of all contract terminations. Therefore, create a trigger so that whenever an employee is deleted, a record of the employee id that left the company, the current date, the current time and employee id of the person who deleted the record are logged in an auditing table.

Screencast Format

For Milestone 2 you are to submit a screencast following the structure outlined below. We have provided duration timestamps for each of the objectives that requires demonstrating, however, these are indicative and demonstrating some objectives may take longer than others. It is important that you follow the structure and that your screencast does not exceed 10 minutes in duration – any deviation from the structure and time limit will result in lost marks.

00:00 - 02:00 Discuss your **CREATE** and **INSERT** statements, highlighting any significant complexities in your system (i.e., tell us how you implemented the backend).

02:00 - 04:00 Provide an overview of your front end system and what the features are and how they work.

04:00 - 05:00 Front End 1. Demonstrate adding the new employee to the system. If you have used any client-side and/or server-side validation techniques explain these. You should also show your updated table(s) in the database.

05:00 - 06:00 Front End 2. Demonstrate updating employee 07-4517183's record to include the new data. You should also show your updated table(s) in the database.

06:00 - 07:00 Front End 3. Demonstrate deleting employee 71-7374760 from the system. You should also show your updated table(s) in the database.

07:00 - 08:00 Front End 4. Demonstrate your systems ability to display all the data required for Front End task 4

08:00 - 09:00 Front End 5. Demonstrate the functionality of your stored procedure running in your system. You should also show the back end functionality of this.

09:00 - 10:00 Front End 6. Demonstrate the functionality of your trigger running in your system. You should also show the back end functionality of this.

Submission

Submit a .zip folder containing the below to the appropriate place on Blackboard, **23111-Cwk2-S-Implementation**.

- A backup of your database (its structure, procedures, triggers and test data)

- Your html & php (and other) files required to build your front end
- your screencast which should be an .mp4

Deadline:

18:00 on Friday the 2nd December

Assessment Type:

This activity is subject to **summative** assessments regulations, therefore your submission will be marked and you will receive the associated feedback. The marks you obtain count for up to 30% of your overall mark for this unit.

Chapter 3

Material

The information below gives the data definition and manipulation requirements for the system. This is deliberately not written with super-precise wording. When working on real projects, initial drafts of requirements are rarely complete and unambiguous. If you find any ambiguities, omissions, or imprecision's here, make a note of the issue, make your own decision about what to do, and justify this decision in your report.

A new delivery company, "Kilburnazon" has just been established and you are the lead database engineer tasked with designing and implementing the database.

Your task is to design the database and implement the database which models the structure of the company. The following specification has been acquired through a requirements interview with the companies CEO, Elon Bazos. Since the language is not precise, you may need to make some assumptions and decisions as you go.

Kilburnazon Design Specification:

Kilburnazon's staff are organised into departments including Management, HR, Drivers and Packagers. Each of these departments has a name, number and an employee who manages it. We should also note the numbers of employees in each department and the department's head office location.

Each employee in Kilburnazon has an employee number which stays with them for the duration of their employment. We must also know their name, home address, salary, date of birth and national insurance number (NIN). Every employee is assigned to a single department and has a manager they report to, the manager will be from the management department. Each employee will also have an emergency contact on record, we need to know their name, relationship to the employee and phone number.

Every company building is situated in one of the areas of the UK, each area has a unique name. All drivers and packagers report to the warehouses in their area whereas managers and HR staff work in a single office. Each warehouse is contained within a specific area, has a unique id, location, size and purpose. The company has two main offices, one in Manchester and the other in London, these are named after their location.

A manager may oversee the work of multiple employees from multiple departments. For drivers we must also keep track of the number of hours a week they are working. The company owns lots of vehicles each one has a unique ID, name and area it operates in. Routes for vehicles always contain a starting and ending location with the possibility for multiple stops in the middle. Each route has a unique name given to them (e.g. red-002). The time a vehicle arrives at each location must also be. A driver will be assigned to a vehicle from their area each day and then allocated a delivery route.

Every time a customer places an order we capture their full name, email address, postal location and the products that have been ordered. We then receive this information in their order forms. Each order can contain multiple products with varying quantities. Each order also has an id and purchase date.

We must keep track of our full product list at all times. Each product has a unique id, name, description, price and quantity left in the warehouses.

Finally, HR deal with any complaints received, these must all be logged with their unique complaint number, date of the complaint, passenger complaints name and the reason for the complaint. This complaint will then be allocated to a member of the HR team to deal with.

Chapter 4

FAQs

Could I develop the applications with modern technologies like NodeJS, Flask, Java Spring, GoLang, etc.?

- No frameworks are to be used. The assessment scripts are designed to test you on certain aspects on the course and any restrictions built into them is intentional. You can, however, use Bootstrap framework for the front-end if you wish.

Are any marks awarded for validation of data?

- It is imperative that data is entered correctly into a database. Validation should be used to reduce the number of errors during the process of data input.

What types of validation should I use?

- I would suggest you look at the different validation techniques, any data captured by the user should be validated, as a guide we would expect:
  - Unique Values, a question last year was asked "if a user tries to register with the same username as an existing user, should this be accounted for?".

    The reply: We would expect some validation here that will inform the user that the username is already taken and give the user the option to try another username.
  - Type validation, is the data entered the correct type? For example, if the input is a surname, you would not expect numeric digits.
  - Non-Null Values – if there are constraints of NOT NULL for any fields in the database, then validation should ensure the user enters a value for any fields that have this constraint.
  - Range Check is often used for numeric fields.
  - Restricted Choice is used when the user can only enter a choice from a valid list of choices, an obvious example might be asking the user which quiz they wish to compete. They user only be able to enter a quiz that exists.
  - Format is often the most useful validation check and has many applications. You will have seen examples such as checking for valid postcodes, email addresses, course codes etc. A format check is essentially checking a string matches a specific pattern and often is implemented using regular expression.

Should I use client-side or server-side validation?

- HTML 5 offers nice features for client-side validation (such as regular expression), anything that HTML 5 doesn't provide can normally be provided by JavaScript. However, you should not assume the user is using HTML5 or JavaScript, so server-side validation can also be used for these situations.

Milestone 2 asks for a backup of the database. Is this backup the .sql file that we get when we export the database on phpMyAdmin?

- Yes, you need to include the database file. This can be achieved in phpMyAdmin using the export feature, or using the mysql command mysqldump.

Wanted to ask if the 3NF form and relational schema should be the same as the one which we will use to implement the database in the application. By this I mean, if we want to change/add something into the database for the app, that we did not deduce from the information source, should we also change the 3NF and Relational Schema.

- When you implement the database, it will be different from the relational scheme since the information source does not capture everything.

Could you please clarify if we will be marked on the appearance of the website, or you will mark the functionality of the website?

- The user interface does not have to look amazing. We are interested in the interaction with the database; however, why not apply a little CSS? There is no credit given for an attempt to make the interface look better – focus your efforts on the functionality, make it look nicer if you have time.

End of Assessment