23111-Cwk1-S-Database Design

Goal:

This assessment is in two parts, both of which should be submitted as part of the same report. The goal for the first activity 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 (DRS), such as you have practised deriving from a data flow diagram (DFD) in Laboratory 1. The outcome of this step is a conceptual model from which a logical model of the data can then be derived.

The second task will then test you on your basic SQL skills using a simpler database. The goal of this activity is for you to demonstrate the technique of writing MySQL. You will begin by documenting the physical design, then you will create the database schema for the University Database Schema. Once you have implemented the database you will insert some test data and then run some queries on that test data. All MySQL should be executed at the MySQL command prompt. Download a suitable MySQL server for your operating system. This will make the task easier in the long run.

These techniques are often applied in real-world software development by a data analyst and is typically the first concrete step in the database design process.

Material:

Part 1

Take a specification of the Monopolee data definition and data manipulation requirements, and express that as an entity-relationship diagram using crow's foot notation (see Figure 1).

The list below gives the data definition and manipulation requirements for the game. 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.

1. There are up to six players.
2. There are six tokens the players can choose from. Each token has a unique name (dog, car, battleship, top hat, thimble, boot).
3. A player must choose one and only one token.

4. Each space on the board is called a location. A location is one of: a corner, a Chance, a Community Chest, or a property.

5. Chance, Community Chest and corners can be grouped together as “bonuses”. Every bonus has a unique id, its name (e.g., “Chance 1”) and a textual description of what it does.

6. Store data about players, properties and bonuses.

7. A property has a unique name, and a purchase cost, which is also the rent payable to the owner if another player lands on it.

8. A property has either one owner or no owner.

9. Each player has a unique id, name, their chosen token, bank balance, their current location, and any bonus that they may have at that location. All this data should have suitable default values and constraints.

10. A bonus can be used by many players. A player has at most one bonus at any time in the gameplay.

11. There must be an audit trail of the gameplay. For each turn taken by a player, the audit trail should store the player’s id, location landed on, current bank balance, and number of the game round.

Part 2

See Figure 2 below.

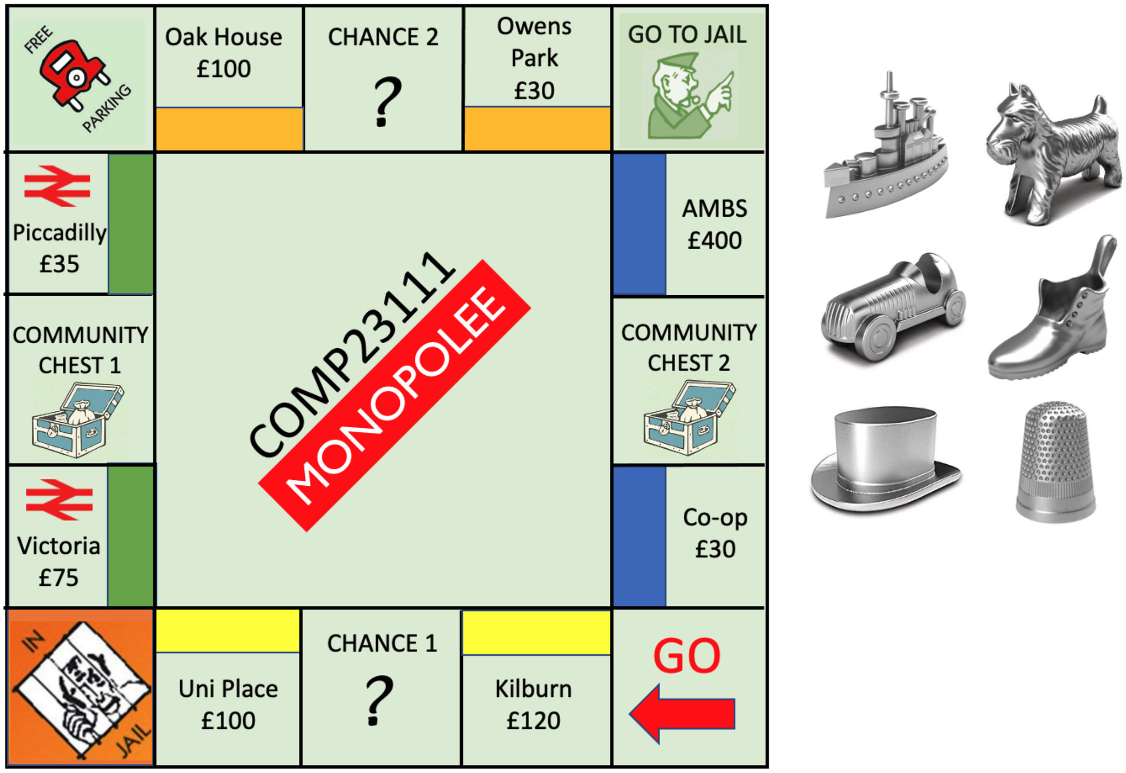

Figure 1: The Game of Monopolee and the 6 Tokens

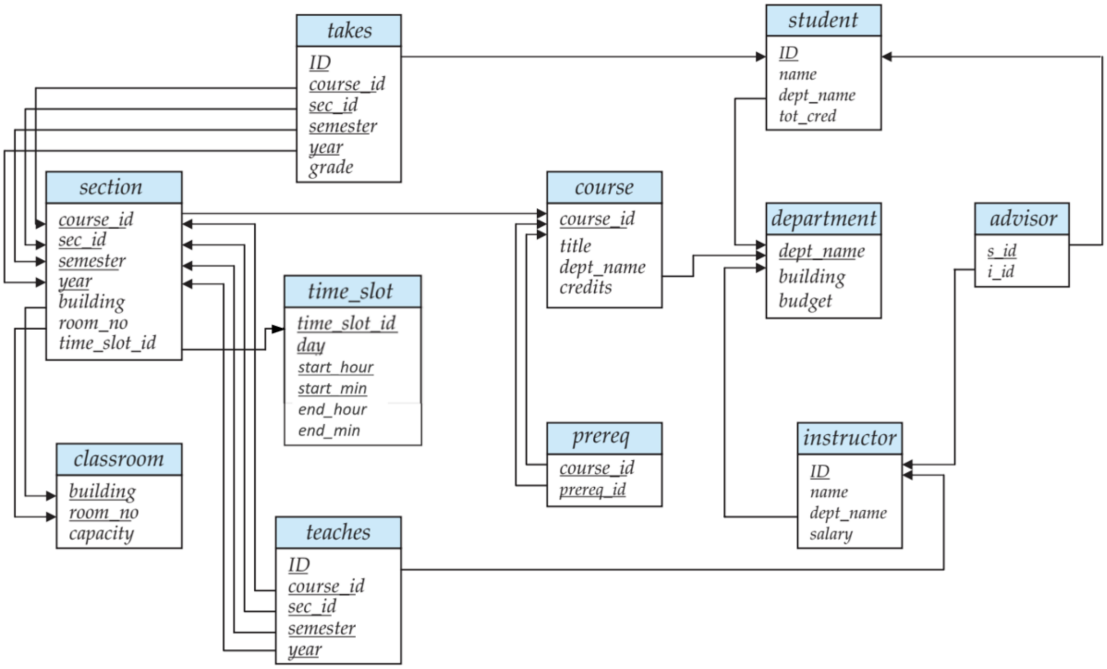

Figure 2: The University Database

Tasks:

Part 1:

Your task is to take a specification of data definition and data manipulation requirements, and express that as an entity-relationship diagram using crow's foot notation. The "problem domain" or "business logic" that you are going to analyse and model is a simplified version of Monopoly.

First, read the requirements below. Then design and create an ER diagram. 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. Your diagram should be accompanied by a short report (300-500 words) describing your design choices.

Part 2:

a. Create the documentation for the physical design of the University Database (see Figure 2). This document should contain the names of the fields, data types and constraints. Your datatypes and constraints must be sensible. Example:

Student

Field	**Data Type**	**Constraints**
id	Integer	Unsigned, 0-4294967295, primary key
name	VarChar	255 characters, not null
dept_name	VarChar	100 characters, foreign key
tot_cred	SmallInt	Unsigned, 0-65535

b. Open a text editor. All you MySQL statements will be written in the text editor, then copied and pasted into the MySQL terminal window. We do this just in case something goes wrong, at least you will have the original MySQL statements in the text file.

c. Implement the University Database using MySQL statements, be mindful of primary and foreign keys. Note: Once you are satisfied the MySQL statement works it should be copied into your main coursework document and formatted in accordance with the assessment instructions. Test your statements again by dropping the tables created and ensure the MySQL will recreate them.

d. Add some test records for each of the tables. Sample data can be downloaded from Blackboard (`SampleData.sql`).

e. You should create queries to match the following requirements:

   i. Find the names of all students who have taken at least one computer science course, making sure there are no duplicate names in the result.

   ii. Find the IDs and names of all students with a fail grade.

iii. For each department, find the maximum salary of instructors in that department. Every department should have at least one instructor.

iv. Find the names of all course and the students enrolled on them which take place on a Friday afternoon and have at least 2 students enrolled on it.

v. Find the names of all instructors and their course unit who teach on a course which uses a classroom with a capacity greater than 50.

Submission Procedure:

Submit a .zip file containing all of your work to the appropriate place on Blackboard, **23111-Cwk1-S-Database Design**.

Part 1: A singular PDF which contains your ER Diagram and any Description of design choices you made.

Part 2: A separate sql file for the following:

- All of the queries used to create the database.
- The 5 queries used for Part 2, e.

Deadline:

18:00 on Friday the 22nd October

Assessment Type:

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

Marks:

Marks are awarded for:

- Overall, a well formatted and structured document with a logical order.
- Correct, complete entity types (plus attributes) and relationship types (plus attributes), where specialisations/generalisations count as the latter, of course.
- Roughly (i.e. not strictly), each correct, complete entity type (plus attributes) is worth one mark, and each correct, complete relationship type (plus attributes) is worth two marks.
- Each correct, complete MySQL expression/statement.

- Each correct explanation in the form of required comments for the SQL statement, here also justify your datatypes and constraints.
- Clear use of conventions and indentation which makes the script easier to read by the marker.
- If the marker finds that any of your submitted files are unreadable, you lose all the marks, so, test it beforehand.

The marker will run your MySQL code, so ensure that the script works as intended. An early mistake in your script may cause problems with other parts following it.

Please note: Your work will not be re-marked because you disagree with the mark you were awarded. The markers all follow the same rubric which has been designed to be as fair and comprehensive as possible. Any queries with your marking can be raised with either Gareth Henshall or Stewart Blakeway within one week of the marks being released, any queries made after this time will not be considered.

End of Assessment