DATA70141 Assignment 1 - Monopolee

1 Important Information

Deadline 18:00 7th November 2022

Submission via Blackboard

Deliverable A report (as a PDF) conforming to the following:

- Section 1: Entity Relationship Diagram.
- Section 2: Short discussion (300-500 words) detailing design choices.
- Section 3: Relational Database Schema and short discussion about the mapping from ER Diagram to Schema, as well as SQL commands for initial population of your database.
- Section 4: Report containing relevant screenshots and the SQL Queries for each gameplay move.

The report must be formatted as follows:

- Margins of between 1cm and 2.5cm
- Your name and student ID is to appear in the footer
- Section headers of at least 14pt bold font
- 10pt font for regular text
- A sensible choice of serif font for regular text (such as Computer Modern, Times New Roman, etc...) - Serifs are the little ”ticks” you see at the ends of the letters in this document. A sans-serif font is one that does *not* have these, such as Arial.
- A sensible choice of monospaced font for SQL content (such as Consolas, Ubuntu Mono, etc...)
- If SQL cannot be placed into the report as typed text, then well formatted screenshots of the SQL code is acceptable
- Figures are to be numbered, and these numbers should be used for correct crossferencing
- You may include a title page if you wish (but you are not required to)

Marking and Feedback Within 3 university working weeks of submission deadline (on or before 28th November 2022)

Grade Weighting 30% of your final module grade

2 Introduction

Your task is to model the gameplay of a simplified version of Monopoly using a relational database and SQL queries.

There is no single RIGHT way to do this assignment. There is no perfect solution. If you gave this exercise to 10 database professionals you would get 10 answers that differ in their detail, and I suspect that at least 5 answers would differ in their overall approach. That’s to be expected. Requirements analysis and database design are to some extent formulaic, yet they are also arts, relying on experience, intuition and creativity. This assignment is as much about the problem-solving process itself, as it is about actually solving the problem. So the assessment will take your creativity into account. Your approach must satisfy the requirements, but exactly how you go about it, is up to you. Be creative and have fun! And remember that there is more to SQL than we have looked at in the course, so you are encouraged to research and use SQL commands that we have not covered, although this is not an assessed requirement.

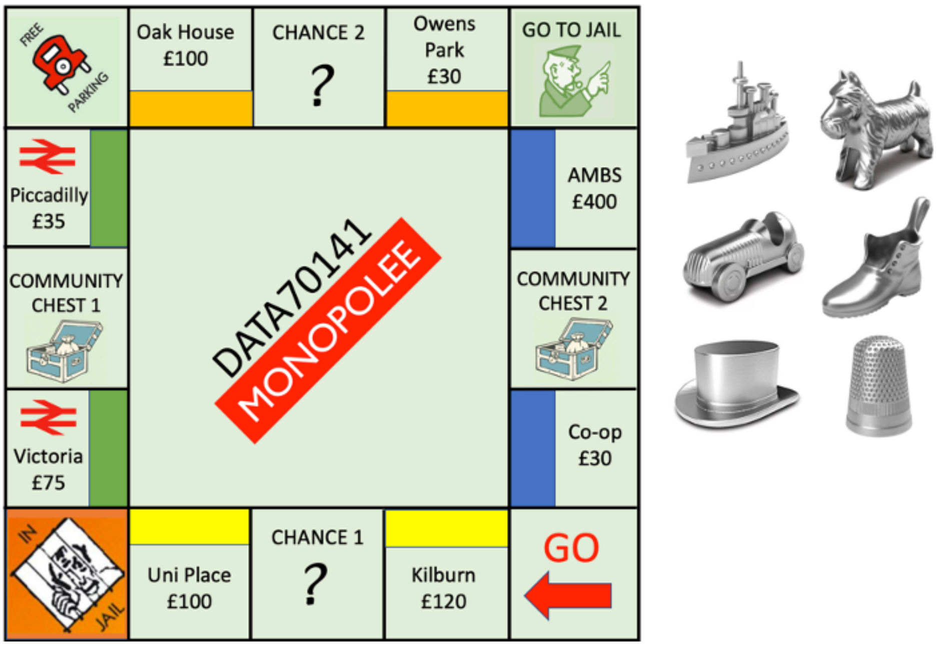

Figure 1: The Monopolee Board

3 Task 1 - ER Diagram in Crows Foot Notation

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.

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 your own decision about what to do, and make a note of the issue, and your 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.

4 Task 2 - Relational Database Schema, Implementation, and Gameplay

Your first step is to derive from your ER Diagram your Relational Database Schema diagram. Your schema should include for each attribute the domain and any default values and constraints. You should ensure that the diagram is represented clearly and neatly in the report - get the design right first and then draw it nicely. Your report should include a discussion about normalisation applied to your schema (if any).

4.1 Initial State of the Game

The definitions of property and bonuses, and the state of the game after the players have been playing for some time, are shown below. Write SQL commands to populate your database tables accordingly; you can do this with explicit `INSERT INTO` commands. Give these commands in your report.

Property	Cost (£)	Colour
Oak House	100	Orange
Owens Park	30	Orange
AMBS	400	Blue
Co-Op	30	Blue
Kilburn	120	Yellow
Uni Place	100	Yellow
Victoria	75	Green
Piccadilly	35	Green

Bonus	Description
Chance 1	Pay each of the other players £50
Chance 2	Move forward 3 spaces
Community Chest 1	For winning a Beauty Contest, you win £100
Community Chest 2	Your library books are overdue. Play a fine of £30
Free Parking	No action
Go to Jail	Go to Jail, do not pass GO, do not collect £200
GO	Collect £200

Player	Token	Location	Bank Balance (£)	Properties Owned
Mary	Battleship	Free Parking	190	Uni Place
Bill	Dog	Owens Park	500	Victoria
Jane	Car	AMBS	150	Co-Op
Norman	Thimble	Kilburn	250	Oak House, Owens Park

The game will now continue from the above state, applying the following rules R1-7. The game is played in a clockwise direction.

R1 If a player lands on a property without an owner, they must buy it.

R2 If player P lands on a property owned by player Q, then P pays Q a rent equal to the cost of the property. If Q owns all the properties of a particular colour, P pays double rent.

R3 If a player is in jail, they must roll a 6 to get out. They immediately roll again.

R4 If a player lands on or passes GO they receive £200.

R5 If a player rolls a 6, they move 6 squares; whatever location they land on has no effect. They then get another roll immediately.

R6 If a player lands on "Go to Jail", they move to Jail, without passing GO.

R7 If a player lands on a Chance or Community Chest location, the action described by the bonus happens.

A *round* of the game is defined as each of the players taking their next turn. For each of the following gameplay steps G1 to G8, which represent two rounds of the game:

1. Apply the rules of the game to work out the changes needed to reflect the state of the game in the database.
2. Write, and run, the SQL commands needed to update the database. You might like to research "SQL UPDATE arithmetic operations".
3. In your report list your SQL commands, and give a screenshot from phpMyAdmin of the state of the tables affected by the commands.

The game is taking place at the 2021 Monopolee World Championships in Havana, and you are required to create a view, "gameView", of the database which will be displayed on a giant screen and will give the current status of each player, the round number, the player's name, their current balance, their current board location etc.

Design and implement a suitable view and in your report give the SQL code to create it in your report, and include a screenshot of it in your report.

Gameplay Round 1, Write the SQL Commands to Model the Following:

G1 Jane rolls a 3

G2 Norman rolls a 1

G3 Mary rolls a 4

G4 Bill rolls a 2

Now give an updated screen shot of gameView after Round 1.

Gameplay Round 2, Write the SQL Commands to Model the Following:

G5 Jane rolls a 5

G6 Norman rolls a 4

G7 Mary rolls a 6, and then a 5

G8 Bill rolls a 6, and then a 3

Now give an updated screen shot of gameView after Round 2.

[END]