DATA70141 Assignment 1 - Monopolee

1 Important Information

Assigmment Released 16th October 2023

Deadline 18:00 3rd November 2023

Submission via Blackboard Assessment Page

Deliverable 1 SQL code (Compatible with SQLite)

Deliverable 2 A video

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. Your database and the queries must be compatible with SQLite.

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.

Please note: SQLite does not support procedures, and this needs to be taken into account when creating your database.

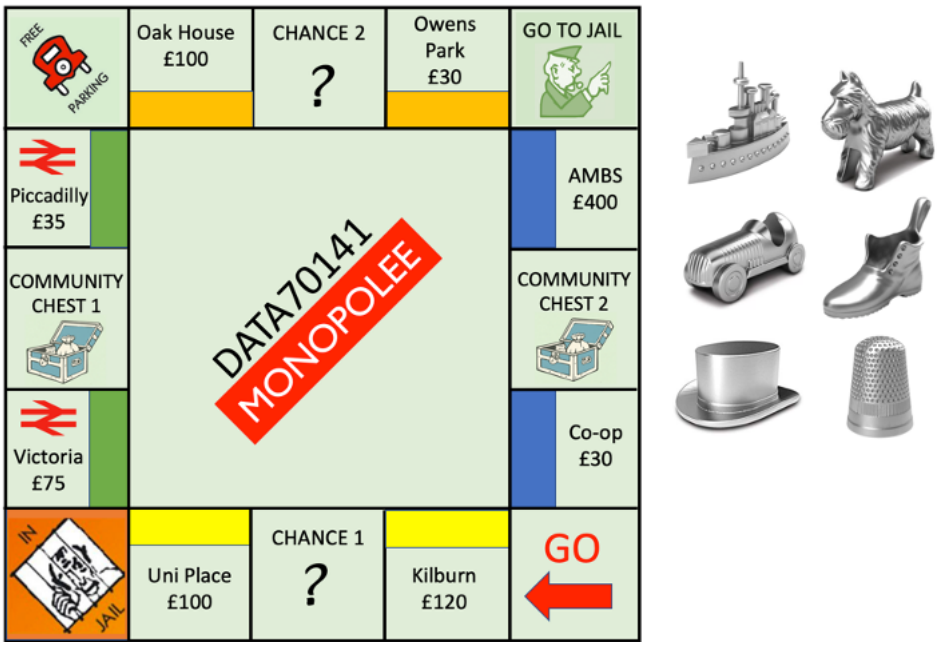

Figure 1: The Monopolee Board

3 Scenario

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 here, make your own decision about what to do, and mention this decision in your video.

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.

3.1 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.

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".

The game is taking place at the 2024 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.

Gameplay Round 1:

G1 Jane rolls a 3

G2 Norman rolls a 1

G3 Mary rolls a 4

G4 Bill rolls a 2

Gameplay Round 2:

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

4 Task 1 - ER Diagram in Crows Foot Notation

Firstly, read the requirements as detailed above, and from this create an ER diagram in Crows Foot notation. 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). 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 is to be used in your video to explain the database design.

5 Task 2 - Schema

You should now 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.

Your schema will be used when you create your database, and also will be used in your video for explaining the design.

6 Task 3 - Implementation

You are required to create several SQL files, which will create your database, populate it accordingly, and execute the queries that simulate the gameplay. You must use the file names as specified here, in order for the automated testing to work. If you do not use these names, then the automated tests might not work and you may not be awareded all marks you would otherwise be due:

create.sql this must contain all queries required to successfully create your database, with the correct constraints.

populate.sql this must contain all queries required to successfully populate your database to match the initial state, as detailed above.

q1.sql This should simulate the play of G1, as detailed above.

q2.sql This should simulate the play of G2, as detailed above.

q3.sql This should simulate the play of G3, as detailed above.

q4.sql This should simulate the play of G4, as detailed above.

q5.sql This should simulate the play of G5, as detailed above.

q6.sql This should simulate the play of G6, as detailed above.

q7.sql This should simulate the play of G7, as detailed above.

q8.sql This should simulate the play of G8, as detailed above.

view.sql This should contain an SQL View that displays a leaderboard of the gameplay, whereby it could be called after each turn (ie: in the middle of a round). As a minimum, this should contain the player name, their poisition on the board, their bank balance, and the properties owned.

These files are to be uploaded, as per the instructions on the submission page on Blackboard.

7 Task 4 - Video

You are required to create a video that contains the following:

- An explanation of your Database design, with clear display of your ER diagram and your Schema (please display each clearly for at least 5 seconds). You should ensure that any normalisation applied to the schema is explained.
- An explanation of your Database SQL code. You do not need to explain every line of code, or even every query, but you should explain key aspects of the code, as well as any features that you have used to help simulate the gameplay.
- You should show the view you created, displaying the state of play after round 1 and round 2.
- The video should last no longer than **7 minutes**.

Please upload your video as per instructions on the submission page on Blackboard.

[END]