

**DATA70141**

# **MONOPOLEE**



I Gusti Ngurah Agung Kresna  
11164605

# **Monopolee Board**

## **Players**



## **Tokens**



## **Game Round**

Each Player : Roll Dice  
1X / Round

FREE PARKING 	Oak House £100	CHANCE 2 ?	Owens Park £30	GO TO JAIL
 Piccadilly £35	9	10	11	12
COMMUNITY CHEST 1	7			AMBS £400 13
 Victoria £75	6			COMMUNITY CHEST 2 14
JAIL	4	3	2	1 GO
Uni Place £100	?	Kilburn £120		16

DATA70141 MONOPOLEE

## **Properties** ●



Buy / Rent



## **Colour**

## **Bonuses** ●



Rewards  
Punishments  
Actions

# **Entities**

Player	
PK	Player ID
	Player
	Token
	Location
	Bank Balance (£)
	Properties Owned
	Is Playing

Property	
PK	Property ID
	Property
	Cost (£)
	Colour

Bonus	
PK	Bonus ID
	Bonus
	Description

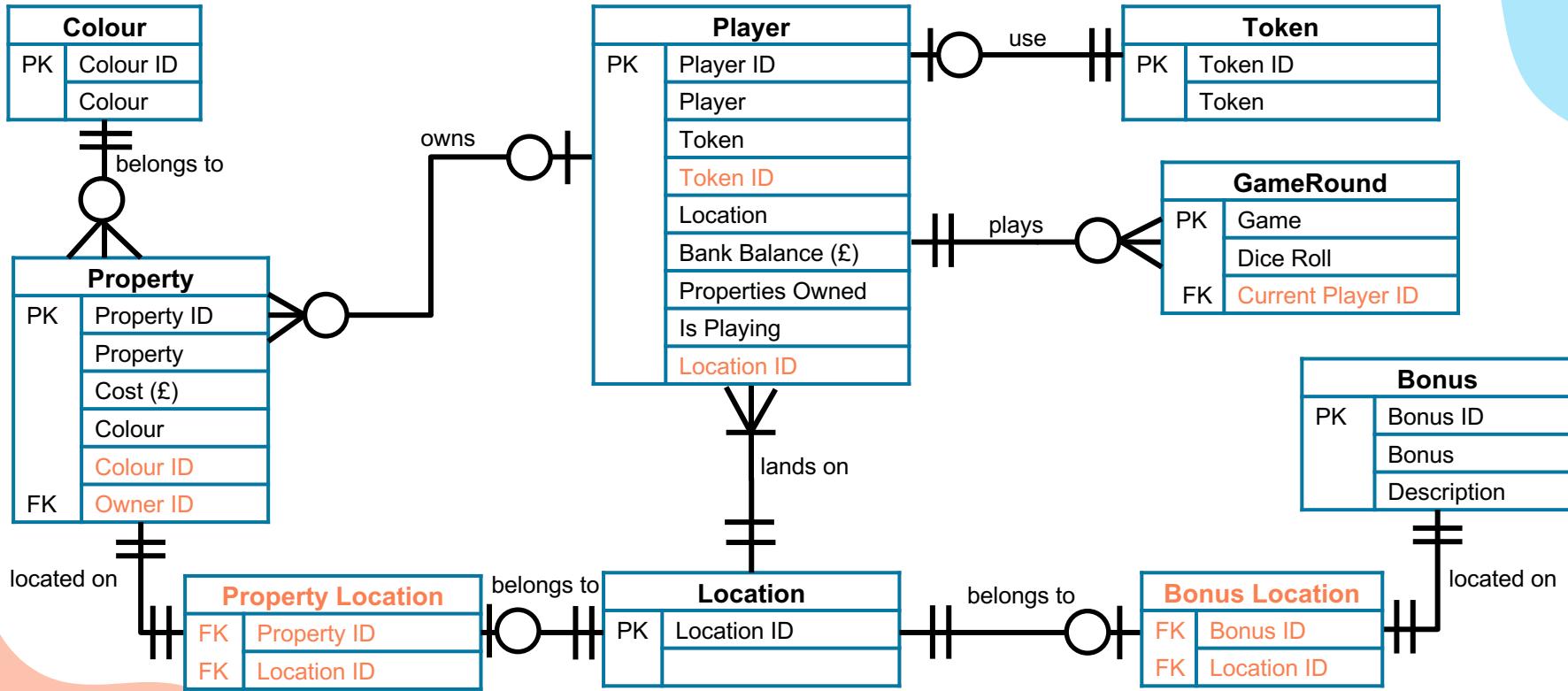
Location	
PK	Location ID

Colour	
PK	Colour ID
	Colour

GameRound	
PK	Game
	Dice Roll

Token	
PK	Token ID
	Token

# Entity Relationship Diagram



# **Normalisation – 1NF**

PlayerID	Player	Token	Token ID	Location	Bank Balance (£)	Properties Owned	Is Playing	Location ID
1	Mary	Battleship	3	Free Parking	190	Uni Place	1	8
2	Bill	Dog	1	Owens Park	500	Victoria	1	11
3	Jane	Car	2	AMBS	150	Co-Op	1	13
4	Norman	Thimble	5	Kilburn	250	Oak House, Owens Park	1	1

Violates 1NF  
(atomic)

PropertyID	Property	Cost (£)	Colour	OwnerId	LocationID
1	Oak House	100	Orange	4	9
2	Owens Park	30	Orange	4	11
3	AMBS	400	Blue	NULL	13
...	...	...	...	...	...

# **Normalisation – 2NF**

PlayerID	Player	Token	Token ID	Location	Bank Balance (£)	Is Playing	Location ID
1	Mary	Blue Chip	3	Freezing	190	1	8
2	Bill	Domino	1	Owens Park	500	1	11
3	Jane	Green	2	AMP	150	1	13
4	Norman	Thimble	5	Kilburn	250	1	1

Violates 2NF  
every non-key attribute is dependent on the PK

# **Normalisation – 2NF**

PropertyID	Property	Cost (£)	Colour	ColourID	OwnerID
1	Oak House	100	Orange	1	4
2	Owens Park	30	Orange	1	4
3	AMBS	400		2	NULL
4	Co-Op	30		2	3
5	Kilburn	120		3	NULL
6	Uni Place	100	Yellow	3	1
7	Victoria	75	Green	4	2
8	Piccadilly	35	Green	4	NULL



# ***Normalisation – 3NF***

All tables have complied to the 3NF  
(no transitive dependency)

# Summary After Normalisation

PlayerID	Player	TokenID	Bank Balance (£)	Is Playing	Location ID
1	Mary	3	190	1	8
2	Bill	1	500	1	11
3	Jane	2	150	1	13
4	Norman	5	250	1	1

PropertyID	Property	Cost (£)	ColourID	OwnerID
1	Oak House	100	1	4
2	Owens Park	30	1	4
3	AMBS	400	2	NULL
4	Co-Op	30	2	3
5	Kilburn	120	3	NULL
6	Uni Place	100	3	1
7	Victoria	75	4	2
8	Piccadilly	35	4	NULL

BonusID	Bonus	Description
1	Chance 1	Pay each of the other players £50
2	Chance 2	Move forward 3 spaces
3	Community Chest 1	For winning a Beauty Contest, you win £100
4	Community Chest 2	Your library books are overdue. Pay a fine of £30
5	Free Parking	No action
6	Go to Jail	Go to Jail, do not pass GO, do not collect £200
7	GO	Collect £200
8	Jail	Must roll a 6 to get out

TokenID	Token	ColourID	Colour
1	Dog	1	Orange
2	Car	2	Blue
3	Battleship	3	Yellow
4	Top Hat	4	Green
5	Thimble		
6	Boot		

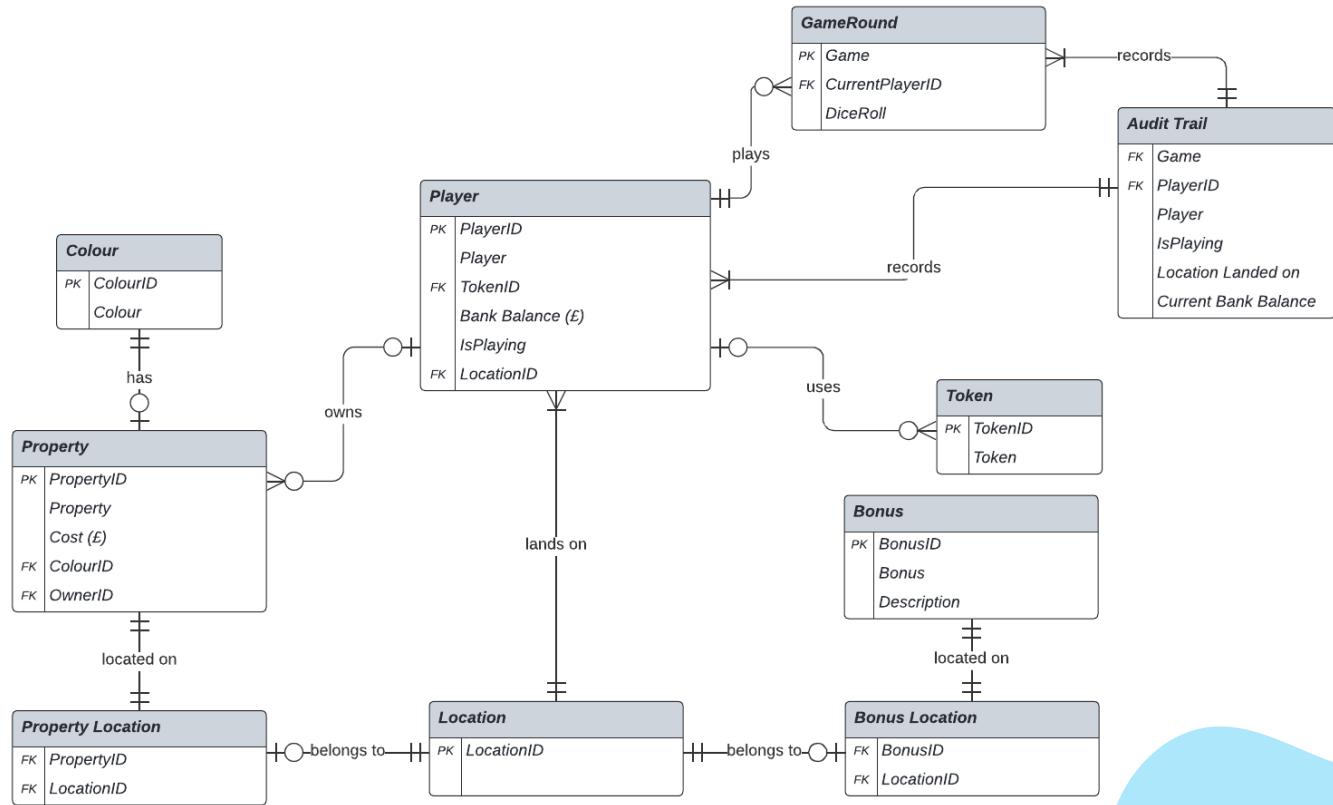
# **Summary After Normalisation**

LocationID
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

<u>PropertyID</u>	<u>LocationID</u>
1	9
2	11
3	13
4	15
5	1
6	3
7	5
8	7

<u>BonusID</u>	LocationID
1	2
2	10
3	6
4	14
5	8
6	12
7	16
8	4

# Revised ER Diagram



# Schema

## Player

<u>PlayerID</u>	Player	TokenID	Bank Balance (£)	Is Playing	Location ID
1	Mary	3	190	1	8
2	Bill	1	500	1	11
3	Jane	2	150	1	13
4	Norman	5	250	1	1

Data Type	int	VARCHAR(30)	int	int	boolean	int
Constraint	NOT NULL PlayerID ≤ 6	NOT NULL	NOT NULL Unique	NOT NULL	NOT NULL	NOT NULL
Default	Auto Increment	none	none	200	1	1
Key	PK		FK to Token(TokenID)			FK to Location(LocationID)

# Schema

## Property

<u>PropertyID</u>	Property	Cost (£)	ColourID	OwnerID
1	Oak House	100	1	4
2	Owens Park	30	1	4
3	AMBS	400	2	NULL
4	Co-Op	30	2	3
5	Kilburn	120	3	NULL
6	Uni Place	100	3	1
7	Victoria	75	4	2
8	Piccadilly	35	4	NULL

Data Type	int	VARCHAR(30)	int	int	int
Constraint	NOT NULL	NOT NULL Unique	NOT NULL	NOT NULL	none
Default	Auto Increment	none	none	none	none
Key	PK			FK to Colour(ColourID)	FK to Player(PlayerID)

# Schema

## Bonus

<u>BonusID</u>	Bonus	Description
1	Chance 1	Pay each of the other players £50
2	Chance 2	Move forward 3 spaces
3	Community Chest 1	For winning a Beauty Contest, you win £100
4	Community Chest 2	Your library books are overdue. Pay a fine of £30
5	Free Parking	No action
6	Go to Jail	Go to Jail, do not pass GO, do not collect £200
7	GO	Collect £200
8	Jail	Must roll a 6 to get out

Data Type	int	VARCHAR(30)	VARCHAR(100)
Constraint	NOT NULL	NOT NULL Unique	NOT NULL
Default	Auto Increment	none	none
Key	PK		

# Schema

**Token**

<u>TokenID</u>	Token
1	Dog
2	Car
3	Battleship
4	Top Hat
5	Thimble
6	Boot

Data Type	int	VARCHAR(30)
Constraint	NOT NULL	NOT NULL Unique
Default	Auto Increment	none
Key	PK	

**Colour**

<u>ColourID</u>	Colour
1	Orange
2	Blue
3	Yellow
4	Green

Data Type	int	VARCHAR(30)
Constraint	NOT NULL	NOT NULL Unique
Default	Auto Increment	none
Key	PK	

# **Schema**

**Location**

<u>LocationID</u>
1
2
3
4
5
6
7
...

**Property Location**

<u>PropertyID</u>	<u>LocationID</u>
1	9
2	11
3	13
4	15
5	1
6	3
7	5
8	7

**Bonus Location**

<u>BonusID</u>	<u>LocationID</u>
1	2
2	10
3	6
4	14
5	8
6	12
7	16
8	4

<b>Data Type</b>	int
<b>Constraint</b>	NOT NULL LocationID ≤ 16
<b>Default</b>	none
<b>Key</b>	PK

<b>Data Type</b>	int	int
<b>Constraint</b>	NOT NULL Unique	NOT NULL Unique
<b>Default</b>	Auto Increment	none
<b>Key</b>	FK to Property(PropertyID)	FK to Location(LocationID)

<b>Data Type</b>	int	int
<b>Constraint</b>	NOT NULL Unique	NOT NULL Unique
<b>Default</b>	Auto Increment	none
<b>Key</b>	FK to Bonus(BonusID)	FK to Location(LocationID)

# ***Schema***

## ***GameRound***

<u>Game</u>	CurrentPlayerID	DiceRoll

Data Type	int	int	VARCHAR(100)
Constraint	NOT NULL	NOT NULL	NOT NULL DiceRoll ≤ 12
Default	Auto Increment	none	none
Key	PK	FK to Player(PlayerID)	

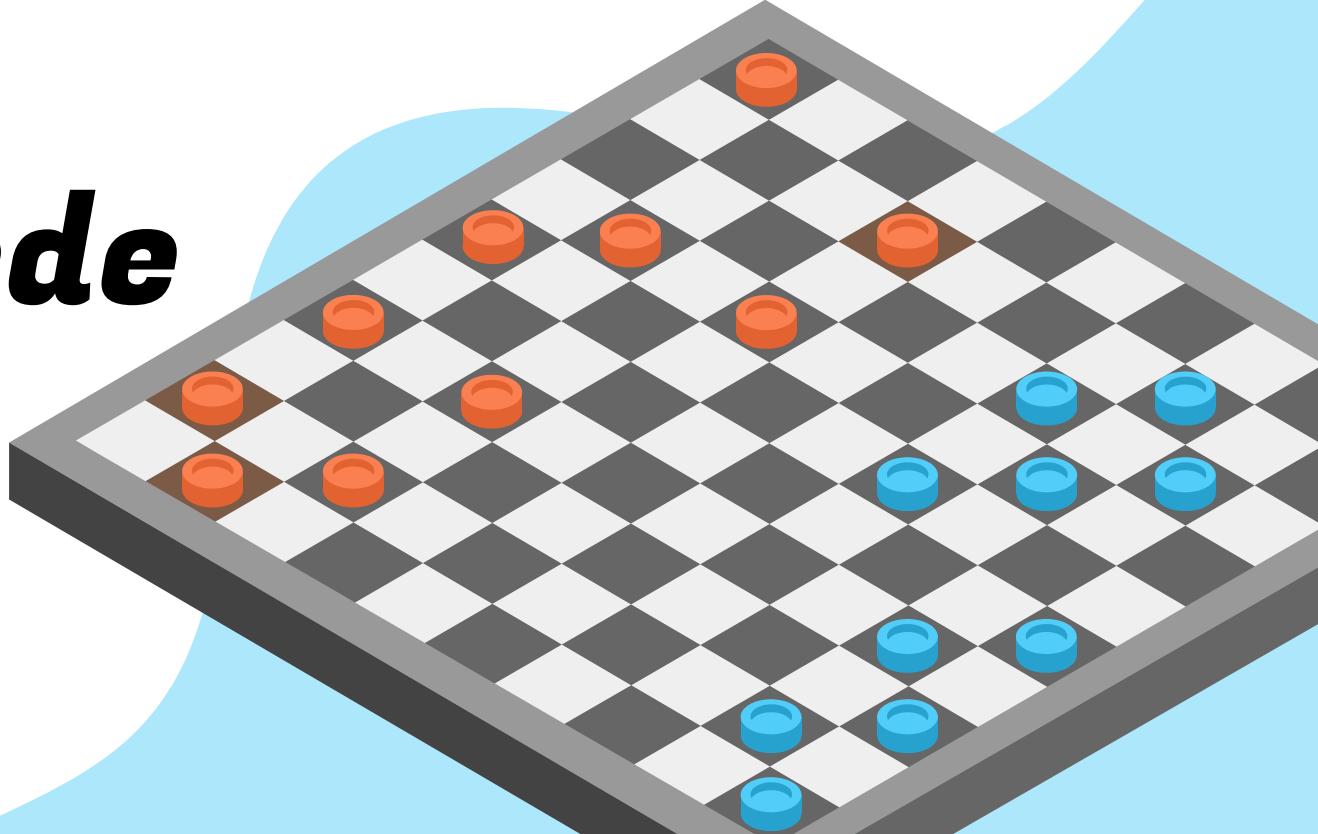
# **Schema**

## **Audit Trail**

Game	PlayerID	Player	IsPlaying	LocationLandedOn	CurrentBankBalance

Data Type	int	int	VARCHAR(30)	boolean	VARCHAR(30)	int
Constraint	NOT NULL	NOT NULL	NOT NULL	NOT NULL	NOT NULL	NOT NULL
	UNIQUE					
Default	none	none	none	none	none	none
Key	FK to GameRound(Game)	FK to Player(PlayerID)				

# **SQL Code**



# Create.sql (tables)

```
1 ----- CREATING TABLES -----
2 
3 CREATE TABLE Bonus (
4     BonusID INTEGER PRIMARY KEY AUTOINCREMENT,
5     Bonus VARCHAR(30) NOT NULL UNIQUE,
6     Description VARCHAR(100) NOT NULL
7 );
8 
9 CREATE TABLE Token(
10    TokenID INTEGER PRIMARY KEY AUTOINCREMENT,
11    Token VARCHAR(30) NOT NULL UNIQUE
12 );
13 
14 CREATE TABLE Colour(
15     ColourID INTEGER PRIMARY KEY AUTOINCREMENT,
16     Colour VARCHAR(30) NOT NULL UNIQUE
17 );
18 
19 CREATE TABLE Location(
20     LocationID INTEGER PRIMARY KEY AUTOINCREMENT CHECK(LocationID <= 16) -- 16 spaces on board
21 );
22 
23 CREATE TABLE BonusLocation(
24     BonusID INTEGER NOT NULL UNIQUE,
25     LocationID INTEGER NOT NULL UNIQUE,
26     FOREIGN KEY(BonusID) REFERENCES Bonus(BonusID),
27     FOREIGN KEY(LocationID) REFERENCES Location(LocationID)
28 );
29 
30 CREATE TABLE Player (
31     PlayerID INTEGER PRIMARY KEY AUTOINCREMENT CHECK(PlayerID <= 6), -- max 6 players
32     Player VARCHAR(30) NOT NULL,
33     TokenID INTEGER NOT NULL UNIQUE, -- using only provided token
34     BankBalance INTEGER NOT NULL DEFAULT 200, -- default bank balance is 200 if not provided
35     IsPlaying BOOLEAN NOT NULL DEFAULT 1, -- yes 1, no 0
36     LocationID INTEGER NOT NULL DEFAULT 1, -- location 1 is the start point if no information provided
37     FOREIGN KEY(TokenID) REFERENCES Token(TokenID)
38     FOREIGN KEY(LocationID) REFERENCES Location(LocationID)
39 );
```

```
40 
41 CREATE TABLE Property (
42     PropertyID INTEGER PRIMARY KEY AUTOINCREMENT,
43     Property VARCHAR(30) UNIQUE NOT NULL,
44     Cost INTEGER NOT NULL,
45     ColourID INTEGER NOT NULL,
46     OwnerID INT,
47     FOREIGN KEY(ColourID) REFERENCES Colour(ColourID),
48     FOREIGN KEY(OwnerID) REFERENCES Player(PlayerID)
49 );
50 
51 CREATE TABLE PropertyLocation(
52     PropertyID INTEGER NOT NULL UNIQUE,
53     LocationID INTEGER NOT NULL UNIQUE,
54     FOREIGN KEY(PropertyID) REFERENCES Property(PropertyID),
55     FOREIGN KEY(LocationID) REFERENCES Location(LocationID)
56 );
57 
58 CREATE TABLE GameRound(
59     Game INTEGER PRIMARY KEY AUTOINCREMENT ,
60     CurrentPlayerID INTEGER NOT NULL,
61     -- max dice roll is 12 since rolling 6 will get another roll
62     DiceRoll INTEGER NOT NULL CHECK (DiceRoll <= 12),
63     FOREIGN KEY (CurrentPlayerID) REFERENCES Player(PlayerID)
64 );
65 
66 CREATE TABLE AuditTrail(
67     Game INTEGER NOT NULL,
68     PlayerID INTEGER NOT NULL,
69     Player VARCHAR(30) NOT NULL,
70     IsPlaying BOOLEAN NOT NULL,
71     LocationLandedOn VARCHAR(30) NOT NULL,
72     CurrentBankBalance INTEGER NOT NULL,
73     UNIQUE(Game, PlayerID),
74     FOREIGN KEY (Game) REFERENCES GameRound(Game),
75     FOREIGN KEY (PlayerID) REFERENCES Player(PlayerID)
76 );
```

# Create.sql (triggers)

## 1. Update Location

```
2      ----- CREATING TRIGGERS -----
3
4  -- updating player's location for each game play
5  CREATE TRIGGER UpdateLocation
6  AFTER INSERT ON GameRound
7  FOR EACH ROW
8  BEGIN
9    UPDATE Player
10   SET LocationID =
11     CASE
12       -- maximum space is 16
13       WHEN (LocationID + NEW.DiceRoll > 16)
14         THEN (LocationID + NEW.DiceRoll) - 16
15
16       -- when player lands on "Chance 2"
17       WHEN (LocationID + NEW.DiceRoll = 10)
18         THEN
19           CASE
20             WHEN (LocationID + NEW.DiceRoll + 3 > 16)
21               THEN (LocationID + NEW.DiceRoll + 3 - 16)
22             ELSE (LocationID + NEW.DiceRoll + 3)
23           END
24
25
26   -- when player lands on "Go to Jail"
27   WHEN (LocationID + NEW.DiceRoll = 12)
28     THEN 4
29
30   -- when player in Jail they need to roll 6 to get out
31   WHEN LocationID = 4
32     THEN
33       CASE
34         WHEN NEW.DiceRoll < 6
35           THEN LocationID
36         -- when player get 6, the movement is the second roll
37         ELSE (LocationID + NEW.DiceRoll) - 6
38       END
39
40   ELSE (LocationID + NEW.DiceRoll)
41
42   END
43   WHERE PlayerID = NEW.CurrentPlayerID;
44   END;
```

# **Create.sql (triggers)**

## **2. Lands on Property**

```
46 -- player lands on property
47 CREATE TRIGGER LandsOnProperty
48 AFTER UPDATE OF LocationID ON Player
49 FOR EACH ROW
50 BEGIN
51     -- 1. Update owner's bank balance from rent fee only if property has owner
52     UPDATE Player
53     SET BankBalance =
54         -- Check if location is a property
55     CASE
56         WHEN(SELECT EXISTS (
57             SELECT * FROM PropertyLocation Pl
58             WHERE Pl.LocationID = NEW.LocationID))
59     THEN
60         CASE
61             -- property has owner
62             WHEN (SELECT Pr.OwnerID FROM Property Pr
63                 LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID
64                 WHERE Pl.LocationID = NEW.LocationID) IS NOT NULL
65         THEN
66             CASE
67                 -- owner owns all properties of certain colour (double rent fees)
68                 WHEN (SELECT EXISTS (
69                     SELECT COUNT(ColourID) FROM Property
70                     WHERE OwnerID = (SELECT Pr.OwnerID FROM Property Pr
71                         INNER JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID
72                         WHERE Pl.LocationID = NEW.LocationID)
73                         GROUP BY ColourID
74                         HAVING COUNT(ColourID) = 2))
75             THEN
76                 UPDATE Player
77                 SET BankBalance = BankBalance + 2 * (SELECT COUNT(*) FROM Property
78                     WHERE OwnerID = NEW.OwnerID AND ColourID = NEW.ColourID)
79             ELSE
80                 UPDATE Player
81                 SET BankBalance = BankBalance + (SELECT COUNT(*) FROM Property
82                     WHERE OwnerID = NEW.OwnerID AND ColourID = NEW.ColourID)
83             END IF
84         ELSE
85             UPDATE Player
86             SET BankBalance = BankBalance + (SELECT COUNT(*) FROM Property
87                 WHERE OwnerID = NEW.OwnerID AND ColourID = NEW.ColourID)
88         END IF
89     END CASE
90 END
```

# Create.sql (triggers)

## 2. Lands on Property

```
107      END  
108      -- if property has no owner, owner's bank balance is not updated  
109      ELSE BankBalance  
110      END  
111      ELSE BankBalance  
112      END  
113      -- playerID refers to the property owner on which player is landing  
114      WHERE PlayerID = (SELECT Pr.OwnerID FROM Property Pr  
115          INNER JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID  
116          WHERE Pl.LocationID = NEW.LocationID);  
117  
118      -- 2. Update current player's bank balance for renting or buying property  
119      UPDATE Player  
120      SET BankBalance =  
121      -- Check if location is a property  
122      CASE  
123      WHEN (SELECT EXISTS (  
124          SELECT * FROM PropertyLocation Pl  
125          WHERE Pl.LocationID = NEW.LocationID))  
126      THEN  
127          CASE  
128              -- property has no owner (player should buy the property)  
129              WHEN (SELECT Pr.OwnerID FROM Property Pr  
130                  LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID  
131                  WHERE Pl.LocationID = NEW.LocationID) IS NULL  
132              THEN  
133                  CASE  
134                      -- checking current player's bank balance  
135                      WHEN NEW.BankBalance >= (SELECT Pr.Cost FROM Property Pr  
136                          LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID  
137                          WHERE Pl.LocationID = NEW.LocationID)
```

```
138      -- current player's bank balance is deducted to buy property  
139      THEN BankBalance - (SELECT Pr.Cost FROM Property Pr  
140          LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID  
141          WHERE Pl.LocationID = NEW.LocationID)  
142  
143      -- if it is insufficient  
144      -- current player's bank balance will be deducted to 0  
145      ELSE 0  
146      END  
147  
148      -- property has owner (player should pay rent)  
149      WHEN (SELECT Pr.OwnerID FROM Property Pr  
150          LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID  
151          WHERE Pl.LocationID = NEW.LocationID) IS NOT NULL  
152      THEN  
153          CASE  
154              -- owner owns all properties of certain colour (double rent fees)  
155              WHEN (SELECT EXISTS (  
156                  SELECT COUNT(ColourID) FROM Property  
157                  WHERE OwnerID = (SELECT Pr.OwnerID FROM Property Pr  
158                      INNER JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID  
159                      WHERE Pl.LocationID = NEW.LocationID)  
160                      GROUP BY ColourID  
161                      HAVING COUNT(ColourID) = 2))  
162              THEN  
163                  CASE  
164                      -- checking current player's bank balance  
165                      WHEN NEW.BankBalance >=  
166                          (SELECT Pr.Cost FROM Property Pr  
167                          LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID  
168                          WHERE Pl.LocationID = NEW.LocationID) * 2
```

# Create.sql (triggers)

## 2. Lands on Property

```
169      -- reducing current player's bank balance to pay double rent
170      THEN BankBalance - (SELECT Pr.Cost FROM Property Pr
171          LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID
172          WHERE Pl.LocationID = NEW.LocationID) * 2
173
174      -- if the bank balance is insufficient, player gives up all their balance
175      ELSE 0
176      END
177
178      -- owner only owns partial property of certain colour (standard rent fees)
179      ELSE
180          CASE
181              -- checking current player's bank balance
182              WHEN NEW.BankBalance >=
183                  (SELECT Pr.Cost FROM Property Pr
184                      LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID
185                      WHERE Pl.LocationID = NEW.LocationID)
186
187              -- reducing current player's bank balance to pay standard rent
188              THEN BankBalance - (SELECT Pr.Cost FROM Property Pr
189                  LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID
190                  WHERE Pl.LocationID = NEW.LocationID)
191
192              -- if the bank balance is insufficient, player gives up all their balance
193              ELSE 0
194          END
195      ELSE BankBalance
196      END
197      -- refers to current playerID
198      WHERE PlayerID = NEW.PlayerID;
```

```
200      -- 3. Update ownerID if player successfully buy property only if property has no owner
201      UPDATE Property
202          SET OwnerID =
203          CASE
204              -- checking if property has no owner
205              WHEN (SELECT Pr.OwnerID FROM Property Pr
206                  LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID
207                  WHERE Pl.LocationID = NEW.LocationID) IS NULL
208
209              THEN
210                  CASE
211                      -- checking current player's bank balance
212                      WHEN NEW.BankBalance >=
213                          (SELECT Pr.Cost FROM Property Pr
214                              LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID
215                              WHERE Pl.LocationID = NEW.LocationID)
216
217                      -- set ownerID to current PlayerID
218                      THEN NEW.PlayerID
219
220                      -- keep NULL if player has insufficient bank balance
221                      ELSE NULL
222                  END
223
224          -- if property has owner, keep the OwnerID
225          ELSE (SELECT Pr.OwnerID FROM Property Pr
226              LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID
227              WHERE Pl.LocationID = NEW.LocationID)
228
229      END
230      WHERE PropertyID = (SELECT Pr.PropertyID FROM Property Pr
231          LEFT JOIN PropertyLocation Pl ON Pr.PropertyID = Pl.PropertyID
232          WHERE Pl.LocationID = NEW.LocationID);
233      END;
```

# Create.sql (triggers)

## 3. Get Bonus

```
234      -- player lands on bonus
235      CREATE TRIGGER GetBonus
236          AFTER UPDATE OF LocationID ON Player
237          FOR EACH ROW
238          BEGIN
239              -- update current player's bank balance associated with bonus
240              UPDATE Player
241                  SET BankBalance =
242                  CASE
243                      -- checking if the location is bonus
244                      WHEN (SELECT EXISTS (
245                          SELECT * FROM BonusLocation Bl
246                          WHERE Bl.LocationID = NEW.LocationID))
247                      THEN
248                          CASE
249                              -- when player lands on "Chance 1"
250                              WHEN NEW.LocationID = 2
251                                  THEN
252                                      CASE
253                                          WHEN NEW.BankBalance >= 50*3
254                                              THEN BankBalance - 50*3
255                                          ELSE 0
256                                      END
257                                      -- when player lands on "Community Chest 1"
258                                      WHEN NEW.LocationID = 6
259                                          THEN BankBalance + 100
260
261                                      -- when player lands on "Community Chest 2"
262                                      WHEN NEW.LocationID = 14
263                                          THEN
264                                              CASE
265                                              WHEN NEW.BankBalance >= 30
266                                              THEN BankBalance - 30
267                                              ELSE 0
268          END
269          -- else, keep current bank balance
270          ELSE BankBalance
271          END
272          ELSE BankBalance
273          END
274          WHERE PlayerID = NEW.PlayerID;
275
276          -- update when player lands or passes GO (except for going to jail)
277          UPDATE Player
278              SET BankBalance =
279              CASE
280                  WHEN (SELECT EXISTS (
281                      SELECT * FROM BonusLocation Bl
282                      WHERE Bl.LocationID = NEW.LocationID))
283                  THEN
284                      CASE
285                          WHEN (NEW.LocationID = 16 OR NEW.LocationID < OLD.LocationID) AND (NEW.LocationID != 4)
286                              THEN BankBalance + 200
287                          ELSE BankBalance
288                      END
289
290
291          WHERE PlayerID = NEW.PlayerID;
```

# Create.sql (triggers)

## 3. Get Bonus

```
292  
293    -- update other's player balance (for Chance 1)  
294    UPDATE Player  
295    SET BankBalance =  
296    CASE  
297        -- Chance 1  
298        WHEN NEW.LocationID = 2  
299            THEN  
300                CASE  
301                    WHEN NEW.BankBalance >= 50*3  
302                        THEN BankBalance + 50  
303                        -- if current player's bank balance is insufficient,  
304                        -- all their remaining balance will be distributed evenly to the other players  
305                        ELSE BankBalance + NEW.BankBalance / 3  
306                END  
307                ELSE BankBalance  
308            END  
309            -- all but the current player  
310            WHERE PlayerID != NEW.PlayerID;  
311  
312        END;
```

# **Create.sql (triggers)**

## **4. Updating Current Player Status (IsPlaying)**

```
315 | -- updating current player's playing status (IsPlaying)
316 | CREATE TRIGGER UpdatePlayerPlayingStatus
317 | AFTER UPDATE OF BankBalance ON Player
318 | FOR EACH ROW
319 | BEGIN
320 |   -- when bank balance become 0, Player is bankrupt hence 0
321 |   UPDATE Player
322 |   SET IsPlaying=
323 |   CASE
324 |     WHEN BankBalance = 0
325 |     THEN 0
326 |     ELSE 1
327 |   END
328 |   WHERE PlayerID = NEW.PlayerID;
329 | END;
--
```

# Create.sql (triggers)

## 5. Audit Trail

```
331 -- audit trail
332 CREATE TRIGGER Audit
333 AFTER UPDATE OF IsPlaying ON Player
334 FOR EACH ROW
335 BEGIN
336     INSERT OR REPLACE INTO AuditTrail(Game, PlayerID, Player, IsPlaying, LocationLandedOn, CurrentBankBalance)
337     SELECT
338         (SELECT Game FROM GameRound ORDER BY Game DESC LIMIT 1),
339         P.PlayerID, P.Player, P.IsPlaying,
340         COALESCE(Pr.Property, B.Bonus),
341         P.BankBalance
342     FROM Player P
343     LEFT JOIN GameRound G ON P.PlayerID = G.CurrentPlayerID
344     LEFT JOIN PropertyLocation Pl ON P.LocationID = Pl.LocationID
345     LEFT JOIN BonusLocation Bl on P.LocationID = Bl.LocationID
346     LEFT JOIN Property Pr ON Pl.PropertyID = Pr.PropertyID
347     LEFT JOIN Bonus B ON Bl.BonusID = B.BonusID;
348 END;
349
```

# *populate.sql*

```
4   INSERT INTO Bonus(Bonus, Description)
5     VALUES
6     ('Chance 1', 'Pay each of the other players £50'),
7     ('Chance 2', 'Move forward 3 spaces'),
8     ('Community Chest 1', 'For winning a Beauty Contest, you win £100'),
9     ('Community Chest 2', 'Your library books are overdue. Pay a fine of £30'),
10    ('Free Parking', 'No action'),
11    ('Go to Jail', 'Go to Jail, do not pass GO, do not collect £200'),
12    ('GO', 'Collect £200'),
13    ('Jail', 'Must roll a 6 to get out');
14
15  INSERT INTO Token(Token)
16    VALUES
17    ('Dog'),
18    ('Car'),
19    ('Battleship'),
20    ('Top Hat'),
21    ('Thimble'),
22    ('Boot');
23
24  INSERT INTO Colour(Colour)
25    VALUES
26    ('Orange'),
27    ('Blue'),
28    ('Yellow'),
29    ('Green');

32  INSERT INTO Location (LocationID)
33    VALUES
34    (1),
35    (2),
36    (3),
37    (4),
38    (5),
39    (6),
40    (7),
41    (8),
42    (9),
43    (10),
44    (11),
45    (12),
46    (13),
47    (14),
48    (15),
49    (16);
50
51  INSERT INTO BonusLocation(BonusID, LocationID)
52    VALUES
53    (1, 2),
54    (2, 10),
55    (3, 6),
56    (4, 14),
57    (5, 8),
58    (6, 12),
59    (7, 16),
60    (8, 4);
```

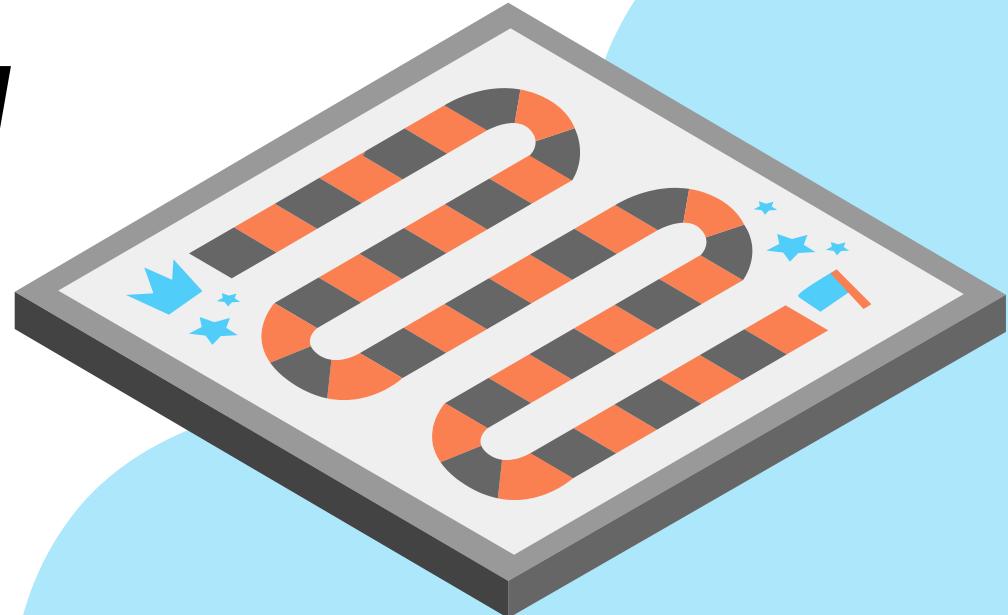
# ***populate.sql***

```
62 INSERT INTO Player (Player, TokenID, BankBalance, LocationID)
63 VALUES
64 ('Mary', 3, 190, 8),
65 ('Bill', 1, 500, 11),
66 ('Jane', 2, 150, 13),
67 ('Norman', 5, 250, 1);
68
69 INSERT INTO Property (Property, Cost, ColourID, OwnerID)
70 VALUES
71 ('Oak House', 100, 1,4),
72 ('Owens Park', 30, 1,4),
73 ('AMBS', 400, 2, NULL),
74 ('Co-Op', 30, 2,3),
75 ('Killburn', 120, 3, NULL),
76 ('Uni Place', 100, 3,1),
77 ('Victoria', 75, 4,2),
78 ('Piccadilly', 35, 4, NULL);
79
80 INSERT INTO PropertyLocation(PropertyID, LocationID)
81 VALUES
82 (1, 9),
83 (2, 11),
84 (3, 13),
85 (4, 15),
86 (5, 1),
87 (6, 3),
88 (7, 5),
89 (8, 7);
90
```

# *view.sql*

```
4  CREATE VIEW gameView AS
5  SELECT
6  (SELECT G.Game FROM GameRound G
7  WHERE P.PlayerID = G.CurrentPlayerID
8  ORDER BY G.Game DESC LIMIT 1
9  ) AS GameRound,
10 P.PlayerID, P.Player, P.BankBalance, P.isPlaying,
11 COALESCE(Pr.Property, B.Bonus) AS CurrentLocation,
12 (SELECT GROUP_CONCAT(Pr.Property, ',')
13 FROM Property Pr
14 WHERE P.PlayerID = Pr.OwnerID
15 ) AS OwnedProperty
16 FROM Player P
17 LEFT JOIN PropertyLocation Pl ON P.LocationID = Pl.LocationID
18 LEFT JOIN BonusLocation Bl on P.LocationID = Bl.LocationID
19 LEFT JOIN Property Pr ON Pl.PropertyID = Pr.PropertyID
20 LEFT JOIN Bonus B ON Bl.BonusID = B.BonusID
21 ;
```

# **GAME SIMULATION**



# ***Initial Condition***

```
1 SELECT * FROM gameView;
```

	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	NULL	1	Mary	190	1	Free Parking	Uni Place
2	NULL	2	Bill	500	1	Owens Park	Victoria
3	NULL	3	Jane	150	1	AMBS	Co-Op
4	NULL	4	Norman	250	1	Killburn	Oak House, Owens Park

# ***Game 1 – Round 1***

	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	NULL	1	Mary	190	1	Free Parking	Uni Place
2	NULL	2	Bill	500	1	Owens Park	Victoria
3	NULL	3	Jane	150	1	AMBS	Co-Op
4	NULL	4	Norman	250	1	Killburn	Oak House, Owens Park

```
5 -- Game 1, Player 3 (Jane), Rolls a 3  
6 INSERT INTO GameRound  
7 VALUES (1, 3, 3);
```



	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	NULL	1	Mary	190	1	Free Parking	Uni Place
2	NULL	2	Bill	500	1	Owens Park	Victoria
3	1	3	Jane	350	1	GO	Co-Op
4	NULL	4	Norman	250	1	Killburn	Oak House, Owens Park

# ***Game 2 – Round 1***

	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	NULL	1	Mary	190	1	Free Parking	Uni Place
2	NULL	2	Bill	500	1	Owens Park	Victoria
3	1	3	Jane	350	1	GO	Co-Op
4	NULL	4	Norman	250	1	Killburn	Oak House, Owens Park

```
1 -- Game 2, Player 4 (Norman), Rolls a 1  
2 INSERT INTO GameRound  
3 VALUES (2, 4, 1);
```



	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	NULL	1	Mary	240	1	Free Parking	Uni Place
2	NULL	2	Bill	550	1	Owens Park	Victoria
3	1	3	Jane	400	1	GO	Co-Op
4	2	4	Norman	100	1	Chance 1	Oak House, Owens Park

# **Game 3 – Round 1**

	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	NULL	1	Mary	240	1	Free Parking	Uni Place
2	NULL	2	Bill	550	1	Owens Park	Victoria
3	1	3	Jane	400	1	GO	Co-Op
4	2	4	Norman	100	1	Chance 1	Oak House, Owens Park

```
1 -- Game 3, Player 1 (Mary), Rolls a 4  
2 INSERT INTO GameRound  
3 VALUES (3, 1, 4);
```



	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	3	1	Mary	240	1	Jail	Uni Place
2	NULL	2	Bill	550	1	Owens Park	Victoria
3	1	3	Jane	400	1	GO	Co-Op
4	2	4	Norman	100	1	Chance 1	Oak House, Owens Park

# **Game 4 – Round 1**

	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	3	1	Mary	240	1	Jail	Uni Place
2	NULL	2	Bill	550	1	Owens Park	Victoria
3	1	3	Jane	400	1	GO	Co-Op
4	2	4	Norman	100	1	Chance 1	Oak House, Owens Park

```
1 -- Game 4, Player 2 (Bill), Rolls a 2  
2 INSERT INTO GameRound  
3 VALUES (4, 2, 2);
```



	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	3	1	Mary	240	1	Jail	Uni Place
2	4	2	Bill	150	1	AMBS	AMBS, Victoria
3	1	3	Jane	400	1	GO	Co-Op
4	2	4	Norman	100	1	Chance 1	Oak House, Owens Park

# **Game 5 – Round 2**

	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	3	1	Mary	240	1	Jail	Uni Place
2	4	2	Bill	150	1	AMBS	AMBS, Victoria
3	1	3	Jane	400	1	GO	Co-Op
4	2	4	Norman	100	1	Chance 1	Oak House, Owens Park

1 -- Game 5, Player 3 (Jane), Rolls a 5

2 **INSERT INTO GameRound**

3 **VALUES (5, 3, 5);**



	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	3	1	Mary	240	1	Jail	Uni Place
2	4	2	Bill	225	1	AMBS	AMBS, Victoria
3	5	3	Jane	325	1	Victoria	Co-Op
4	2	4	Norman	100	1	Chance 1	Oak House, Owens Park

# **Game 6 – Round 2**

	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	3	1	Mary	240	1	Jail	Uni Place
2	4	2	Bill	225	1	AMBS	AMBS, Victoria
3	5	3	Jane	325	1	Victoria	Co-Op
4	2	4	Norman	100	1	Chance 1	Oak House, Owens Park

1 -- Game 6, Player 4 (Norman), Rolls a 4  
2 **INSERT INTO GameRound**  
3 **VALUES (6, 4, 4);**



	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	3	1	Mary	240	1	Jail	Uni Place
2	4	2	Bill	225	1	AMBS	AMBS, Victoria
3	5	3	Jane	325	1	Victoria	Co-Op
4	6	4	Norman	200	1	Community Chest 1	Oak House, Owens Park

# **Game 7 – Round 2**

	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	3	1	Mary	240	1	Jail	Uni Place
2	4	2	Bill	225	1	AMBS	AMBS, Victoria
3	5	3	Jane	325	1	Victoria	Co-Op
4	6	4	Norman	200	1	Community Chest 1	Oak House, Owens Park

1 -- Game 7, Player 1 (Mary), Rolls a 6, and a 5 (11)

2 **INSERT INTO GameRound**  
3 **VALUES (7, 1, 11);**



	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	7	1	Mary	40	1	Oak House	Uni Place
2	4	2	Bill	225	1	AMBS	AMBS, Victoria
3	5	3	Jane	325	1	Victoria	Co-Op
4	6	4	Norman	400	1	Community Chest 1	Oak House, Owens Park

# **Game 8 – Round 2**

	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	7	1	Mary	40	1	Oak House	Uni Place
2	4	2	Bill	225	1	AMBS	AMBS, Victoria
3	5	3	Jane	325	1	Victoria	Co-Op
4	6	4	Norman	400	1	Community Chest 1	Oak House, Owens Park

```
1 -- Game 8, Player 2 (Bill), Rolls a 6, and a 3 (9)
2 INSERT INTO GameRound
3 VALUES (8, 2, 9);
```



	GameRound	PlayerID	Player	BankBalance	IsPlaying	CurrentLocation	OwnedProperty
1	7	1	Mary	40	1	Oak House	Uni Place
2	8	2	Bill	525	1	Community Chest 1	AMBS, Victoria
3	5	3	Jane	325	1	Victoria	Co-Op
4	6	4	Norman	400	1	Community Chest 1	Oak House, Owens Park

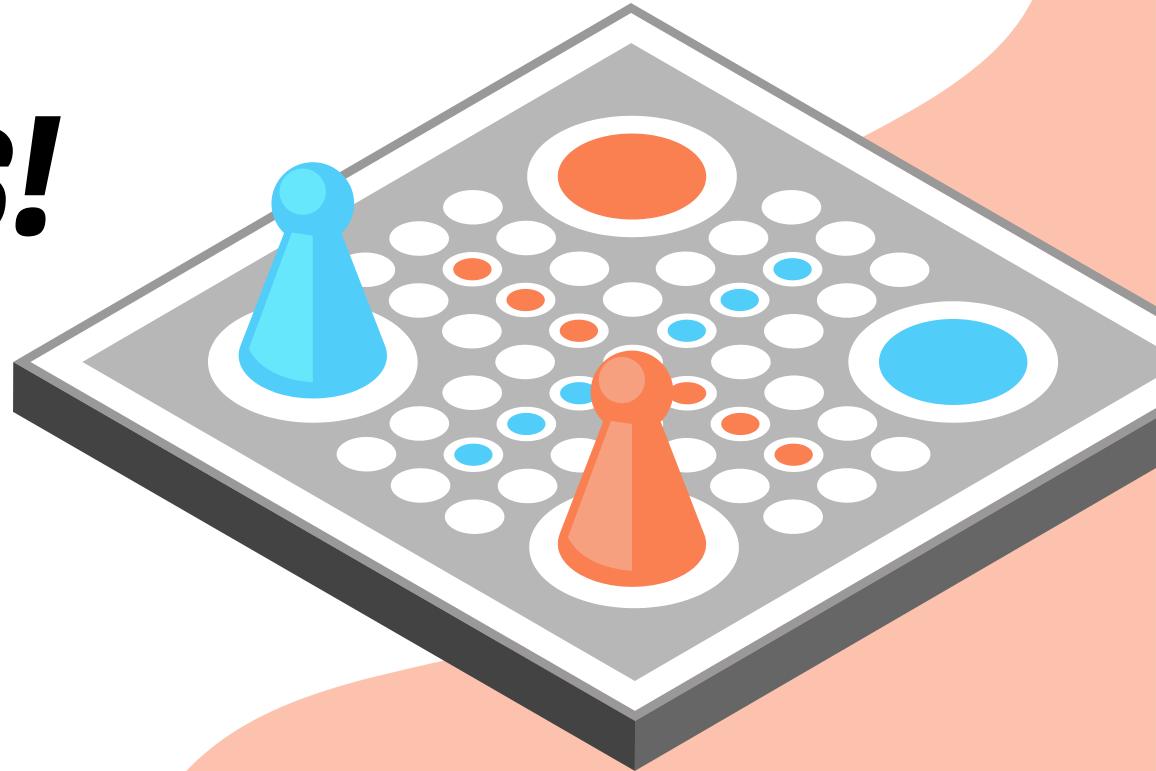
# Audit Trail

1    **SELECT \* FROM AuditTrail;**

	Game	PlayerID	Player	IsPlaying	LocationLandedOn	CurrentBankBalance
1	1	1	Mary	1	Free Parking	190
2	1	2	Bill	1	Owens Park	500
3	1	3	Jane	1	GO	350
4	1	4	Norman	1	Killburn	250
5	2	1	Mary	1	Free Parking	240
6	2	2	Bill	1	Owens Park	550
7	2	3	Jane	1	GO	400
8	2	4	Norman	1	Chance 1	100
9	3	1	Mary	1	Jail	240
10	3	2	Bill	1	Owens Park	550
11	3	3	Jane	1	GO	400
12	3	4	Norman	1	Chance 1	100
13	4	1	Mary	1	Jail	240
14	4	2	Bill	1	AMBS	150
15	4	3	Jane	1	GO	400
16	4	4	Norman	1	Chance 1	100

17	5	1	Mary	1	Jail	240
18	5	2	Bill	1	AMBS	225
19	5	3	Jane	1	Victoria	325
20	5	4	Norman	1	Chance 1	100
21	6	1	Mary	1	Jail	240
22	6	2	Bill	1	AMBS	225
23	6	3	Jane	1	Victoria	325
24	6	4	Norman	1	Community Chest 1	200
25	7	1	Mary	1	Oak House	40
26	7	2	Bill	1	AMBS	225
27	7	3	Jane	1	Victoria	325
28	7	4	Norman	1	Community Chest 1	400
29	8	1	Mary	1	Oak House	40
30	8	2	Bill	1	Community Chest 1	525
31	8	3	Jane	1	Victoria	325
32	8	4	Norman	1	Community Chest 1	400

# THANKS!



This presentation template was created by  
**Slidesgo**, including icons by **Flaticon**, and  
infographics & images by **Freepik**.