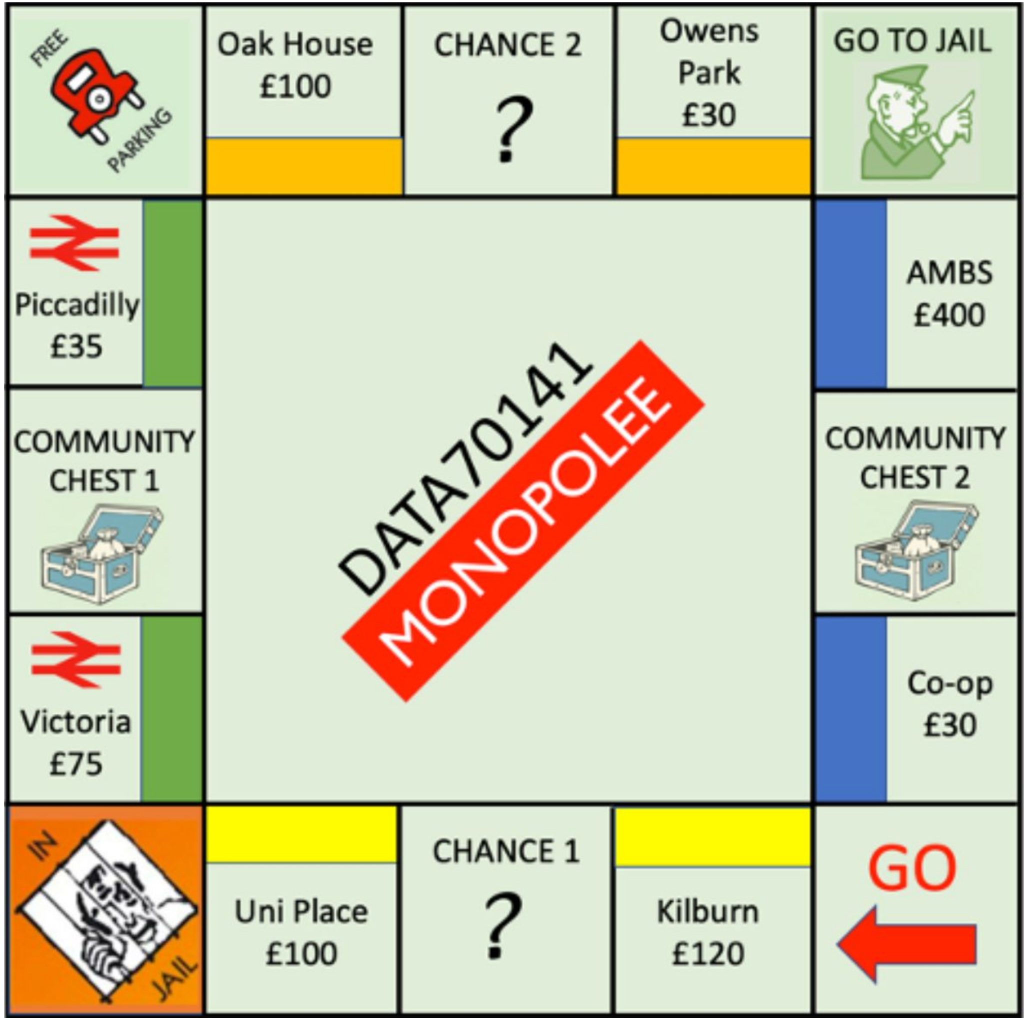
Monopolee Assignment

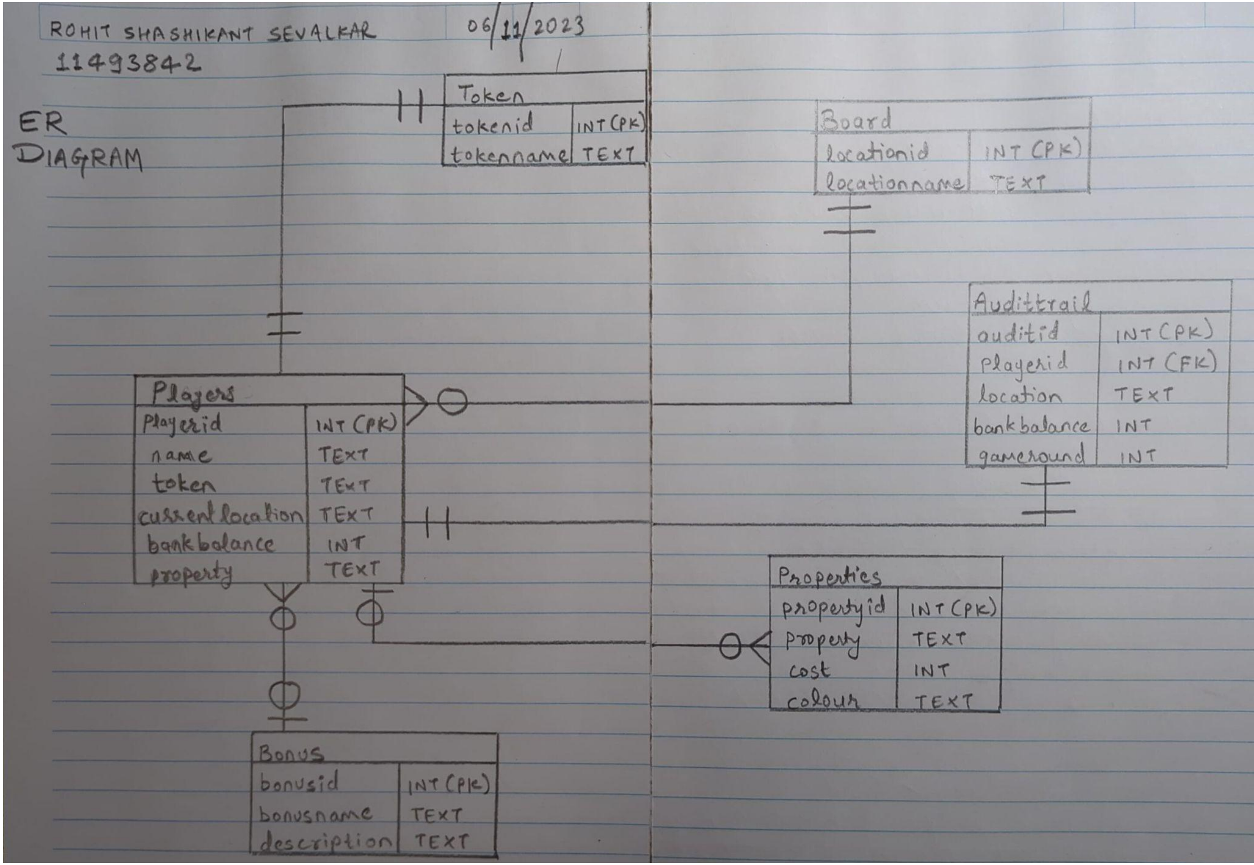
Rohit Shashikant Sevalkar
Student ID - 11493842
Date - 6th November 2023
Course_code - DATA70141

Assumptions

- In this slide I will be talking about the assumptions that I have made during this assignment.
- The first one is if a player lands in Jail, we know that after rolling a 6 on a dice they can get out of the jail.
- But in the rules it was not explicitly mentioned whether they should move those 6 places forward while they get out of jail. Hence I have assumed that when Mary rolls a 6 she reaches 'Chance 2' and then she rolls a 5, and finally reaches Co-op.
- Secondly I have assumed that in the last game, that is game 8. Bill moves from AMBS and lands at Community Chest 1.
- Now since he passes through the tile 'GO', he collects 200pounds on the way, and further collects 100pounds once he reaches community chest 1.



- Now lets begin with the explanation for the ER diagram. Here I have created the ER diagram for this dataset, which includes the 6 tables Players, token, board, Audittrail, Properties, and Bonus.
- Now the relation between the Players and token table is one-one relation, and a player can have atleast and atmost 1 token, and a token can have atleast and atmost 1 player.
- Similarly I have mentioned relations between other tables and the Players table. For example the relation between properties and Players is as follows.
- A player can have minimum 0 propertiesand maximum many properties, while a property can have atmost 1 owner or it can have no owner. This is a crow foot notation of the ER diagram.



ROHIT SHASHIKANT SEVALKAR 06/11/2023
11493842

ER DIAGRAM

Token	
tokenid	INT (PK)
tokenname	TEXT

Board	
locationid	INT (PK)
locationname	TEXT

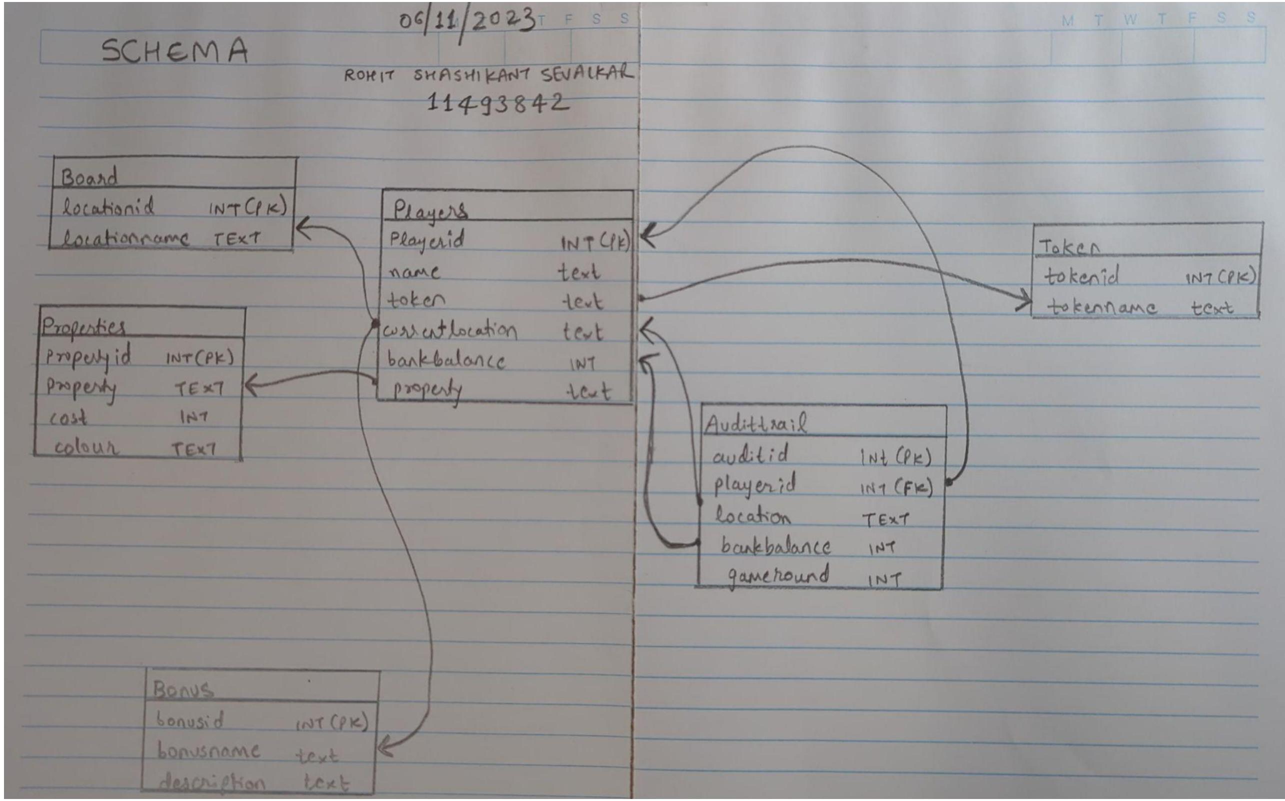
Audittrail	
auditid	INT (PK)
playerid	INT (FK)
location	TEXT
bankbalance	INT
gameround	INT

Players	
Playerid	INT (PK)
name	TEXT
token	TEXT
current location	TEXT
bankbalance	INT
property	TEXT

Properties	
propertyid	INT (PK)
property	TEXT
cost	INT
colour	TEXT

Bonus	
bonusid	INT (PK)
bonusname	TEXT
description	TEXT

- Now coming to the second slide, we have a schema for the dataset used in this assignment. Here in this diagram we can see that I have used 6 tables and I have tried to show the relationship between the tables through their attributes.
- For example, the currentlocation attribute of the Players table, is taking its information and reference from the locaitonname attribute of the Board table.
- Similarly the bankbalance attribute of the audittrail table is taking its information and reference from the 'bankbalance attribute of the Players table. In a similar fashion I have stated 7 relationships that I could find between this dataset through its tables.



SCHEMA

06/11/2023

ROHIT SHASHIKANT SEVALKAR

11493842

Board	
locationid	INT(PK)
locationname	TEXT

Players	
Playerid	INT(PK)
name	text
token	text
currentlocation	text
bankbalance	INT
property	text

Token	
tokenid	INT(PK)
tokenname	text

Properties	
propertyid	INT(PK)
property	TEXT
cost	INT
colour	TEXT

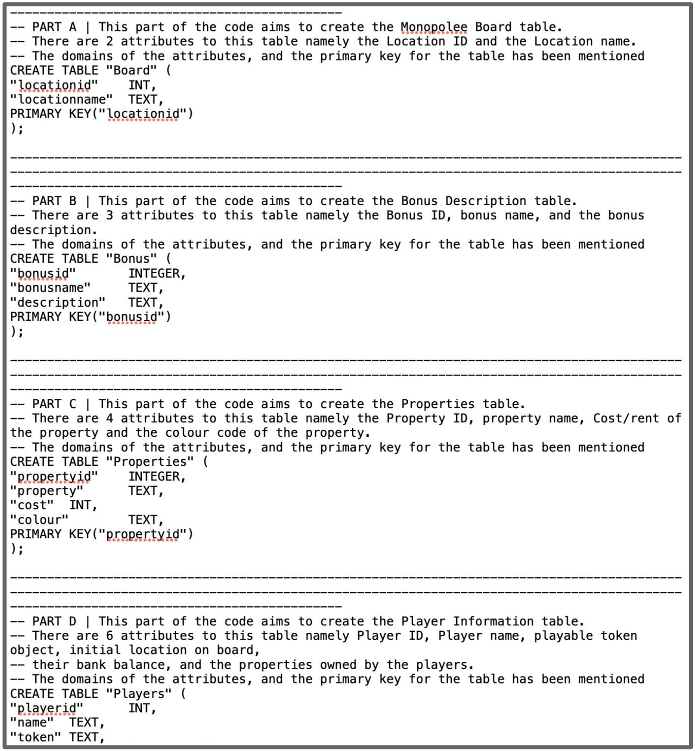
Audittrail	
auditid	INt (PK)
playerid	INT (FK)
location	TEXT
bankbalance	INT
gameround	INT

Bonus	
bonusid	INT (PK)
bonusname	text
description	text

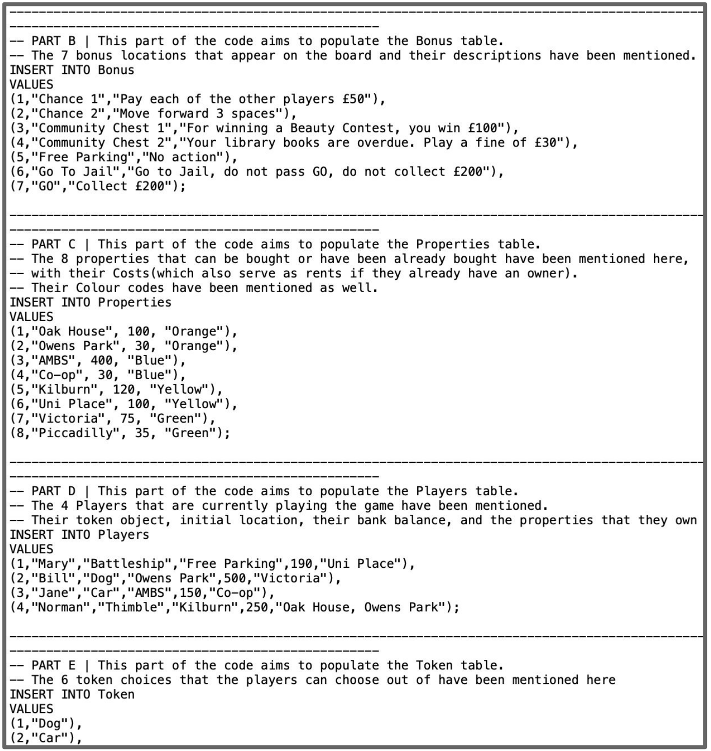
Code Explanation | create.sql and populate.sql

- Now coming to the explanations for the code files, lets begin with the create.sql file. Here in this file I have created tables for the 6 tables that are present in this dataset.
- Namely board, bonus, properties, players, audittrail and tokens. Creation of a table is pretty straightforward, we mention the names of different attributes and also mention their domains with the primary and foreign keys present in them.

  Now moving to the populate.sql table, I have populated the tables that I created in the create.sql file. While populating a table we need to make sure that the attributes of the tables need to be mentioned so that the values are inserted in the right places within the table.



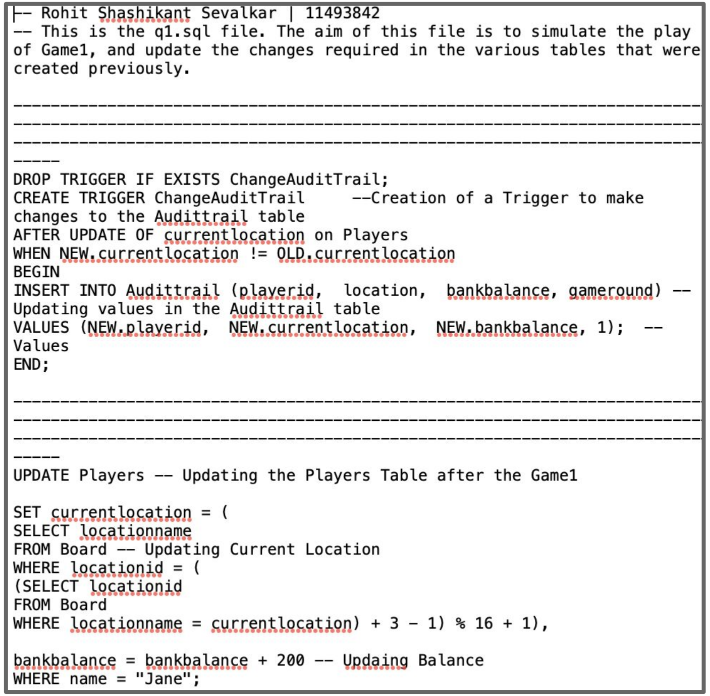
```
---------------------------------------------
-- PART A | This part of the code aims to create the Monopolee Board table.
-- There are 2 attributes to this table namely the Location ID and the Location name.
-- The domains of the attributes, and the primary key for the table has been mentioned
CREATE TABLE "Board" (
"locationid"    INT,
"locationname"  TEXT,
PRIMARY KEY("locationid")
);

------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
---------------------------------------------
-- PART B | This part of the code aims to create the Bonus Description table.
-- There are 3 attributes to this table namely the Bonus ID, bonus name, and the bonus
description.
-- The domains of the attributes, and the primary key for the table has been mentioned
CREATE TABLE "Bonus" (
"bonusid"       INTEGER,
"bonusname"     TEXT,
"description"   TEXT,
PRIMARY KEY("bonusid")
);

------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
---------------------------------------------
-- PART C | This part of the code aims to create the Properties table.
-- There are 4 attributes to this table namely the Property ID, property name, Cost/rent of
the property and the colour code of the property.
-- The domains of the attributes, and the primary key for the table has been mentioned
CREATE TABLE "Properties" (
"propertyid"    INTEGER,
"property"      TEXT,
"cost"  INT,
"colour"        TEXT,
PRIMARY KEY("propertyid")
);

------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
---------------------------------------------
-- PART D | This part of the code aims to create the Player Information table.
-- There are 6 attributes to this table namely Player ID, Player name, playable token
object, initial location on board,
-- their bank balance, and the properties owned by the players.
-- The domains of the attributes, and the primary key for the table has been mentioned
CREATE TABLE "Players" (
"playerid"      INT,
"name"  TEXT,
"token" TEXT,
```



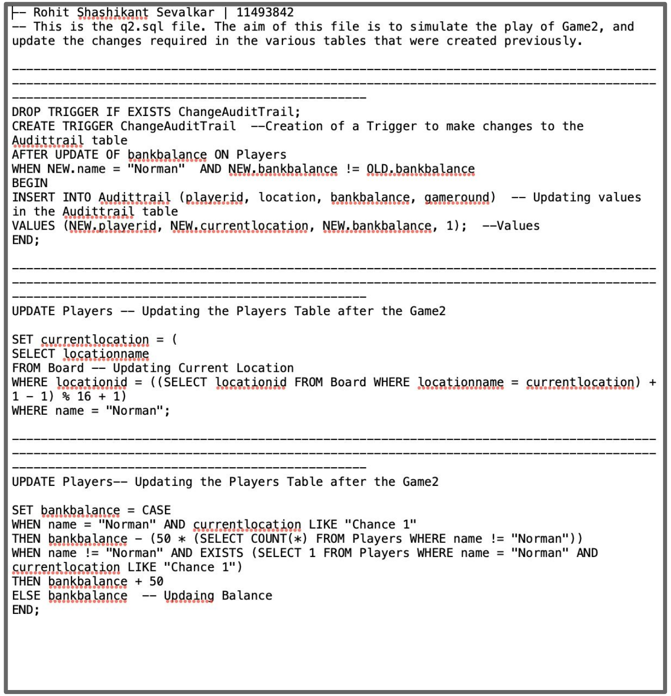
```
-------------------------------------------------------------------------------------------
---------------------------------------------
-- PART B | This part of the code aims to populate the Bonus table.
-- The 7 bonus locations that appear on the board and their descriptions have been mentioned.
INSERT INTO Bonus
VALUES
(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. Play 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");

-------------------------------------------------------------------------------------------
---------------------------------------------
-- PART C | This part of the code aims to populate the Properties table.
-- The 8 properties that can be bought or have been already bought have been mentioned here,
-- with their Costs(which also serve as rents if they already have an owner).
-- Their Colour codes have been mentioned as well.
INSERT INTO Properties
VALUES
(1,"Oak House", 100, "Orange"),
(2,"Owens Park", 30, "Orange"),
(3,"AMBS", 400, "Blue"),
(4,"Co-op", 30, "Blue"),
(5,"Kilburn", 120, "Yellow"),
(6,"Uni Place", 100, "Yellow"),
(7,"Victoria", 75, "Green"),
(8,"Piccadilly", 35, "Green");

-------------------------------------------------------------------------------------------
---------------------------------------------
-- PART D | This part of the code aims to populate the Players table.
-- The 4 Players that are currently playing the game have been mentioned.
-- Their token object, initial location, their bank balance, and the properties that they own
INSERT INTO Players
VALUES
(1,"Mary","Battleship","Free Parking",190,"Uni Place"),
(2,"Bill","Dog","Owens Park",500,"Victoria"),
(3,"Jane","Car","AMBS",150,"Co-op"),
(4,"Norman","Thimble","Kilburn",250,"Oak House, Owens Park");

-------------------------------------------------------------------------------------------
---------------------------------------------
-- PART E | This part of the code aims to populate the Token table.
-- The 6 token choices that the players can choose out of have been mentioned here
INSERT INTO Token
VALUES
(1,"Dog"),
(2,"Car"),
```

Code Explanation | q1.sql and q2.sql

Now coming to the first game file that in q1.sql. Here I have demonstrated the working of the game 1. I have used a trigger to automate the population of the ‘Audittrail table’ This is done in the following way. Initially i created a trigger called ChangeAudittrail. Now this trigger is activated only when the values of the Player Information table is updated. As we can see, I have set the trigger to activate when the players information table is updated with “new location”.



```
-- Rohit Shashikant Sevalkar | 11493842
-- This is the q1.sql file. The aim of this file is to simulate the play
of Game1, and update the changes required in the various tables that were
created previously.

--------------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
-----
DROP TRIGGER IF EXISTS ChangeAuditTrail;
CREATE TRIGGER ChangeAuditTrail     --Creation of a Trigger to make
changes to the Audittrail table
AFTER UPDATE OF currentlocation on Players
WHEN NEW.currentlocation != OLD.currentlocation
BEGIN
INSERT INTO Audittrail (playerid,  location,  bankbalance, gameround) --
Updating values in the Audittrail table
VALUES (NEW.playerid,  NEW.currentlocation,  NEW.bankbalance, 1);  --
Values
END;

--------------------------------------------------------------------------
--------------------------------------------------------------------------
--------------------------------------------------------------------------
-----
UPDATE Players -- Updating the Players Table after the Game1

SET currentlocation = (
SELECT locationname
FROM Board -- Updating Current Location
WHERE locationid = (
(SELECT locationid
FROM Board
WHERE locationname = currentlocation) + 3 - 1) % 16 + 1),

bankbalance = bankbalance + 200 -- Updaing Balance
WHERE name = "Jane";
```

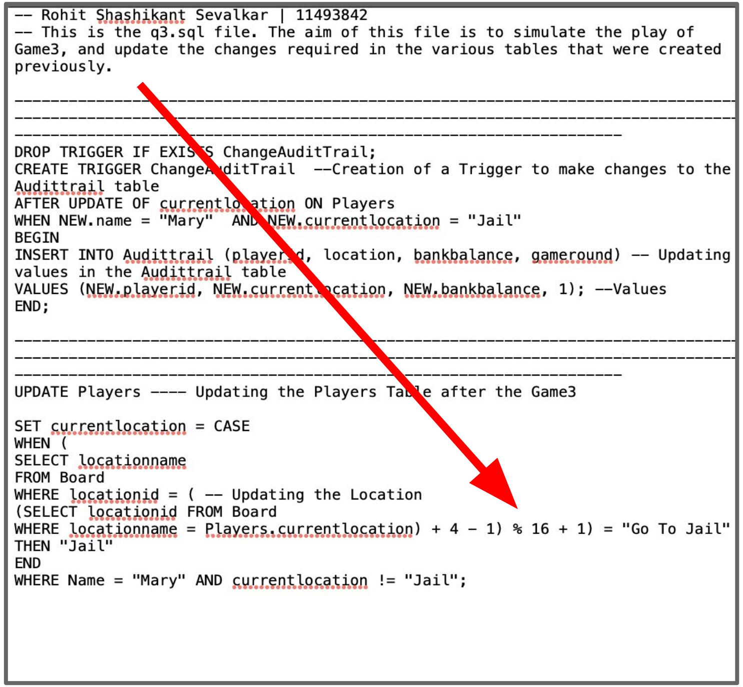


```
-- Rohit Shashikant Sevalkar | 11493842
-- This is the q2.sql file. The aim of this file is to simulate the play of Game2, and
update the changes required in the various tables that were created previously.

-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
-------------------------------------------------
DROP TRIGGER IF EXISTS ChangeAuditTrail;
CREATE TRIGGER ChangeAuditTrail  --Creation of a Trigger to make changes to the
Audittrail table
AFTER UPDATE OF bankbalance ON Players
WHEN NEW.name = "Norman"  AND NEW.bankbalance != OLD.bankbalance
BEGIN
INSERT INTO Audittrail (playerid, location, bankbalance, gameround)  -- Updating values
in the Audittrail table
VALUES (NEW.playerid, NEW.currentlocation, NEW.bankbalance, 1);  --Values
END;

-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
-------------------------------------------------
UPDATE Players -- Updating the Players Table after the Game2

SET currentlocation = (
SELECT locationname
FROM Board -- Updating Current Location
WHERE locationid = ((SELECT locationid FROM Board WHERE locationname = currentlocation) +
1 - 1) % 16 + 1)
WHERE name = "Norman";

-------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------
-------------------------------------------------
UPDATE Players-- Updating the Players Table after the Game2

SET bankbalance = CASE
WHEN name = "Norman" AND currentlocation LIKE "Chance 1"
THEN bankbalance - (50 * (SELECT COUNT(*) FROM Players WHERE name != "Norman"))
WHEN name != "Norman" AND EXISTS (SELECT 1 FROM Players WHERE name = "Norman" AND
currentlocation LIKE "Chance 1")
THEN bankbalance + 50
ELSE bankbalance  -- Updaing Balance
END;
```

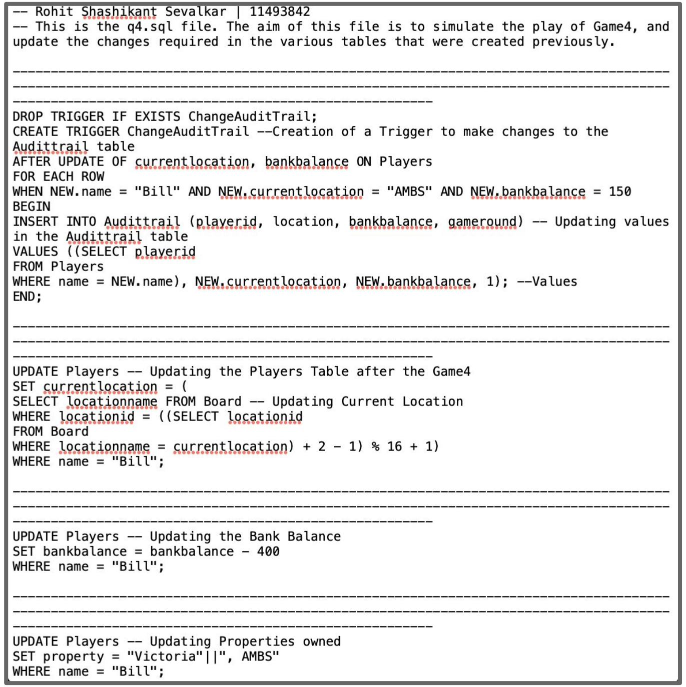
Now as soon as this trigger is activated, 1 row denoting the 1st gameplay is appended in the Audittrail table. Moreover in the second part of the code, I have updated the locations and the bankbalances of Jane semiautomatically, keeping in mind the rules of the game as well as the costs of properties. As I have mentioned earlier, this updation would lead to the activation of the trigger in the first part of the code. Similarly in the gameplay 2, I have used the same format. Step 1 is creation of trigger, and step 2 is updating the values which in turn activates the trigger.

Code Explanation | q3.sql and q4.sql

- Here are the codes for the gameplays 3 and 4. These have a similar format to the ones explained before.
- Here in the left image we can see some numbers. The number 16 denoted the number of tiles on the monopoly board, while %16 denotes that the board is in a circular manner.
- Moreover i have subtracted and added 1 to the numbers as the counting starts from 0 in the syntax.



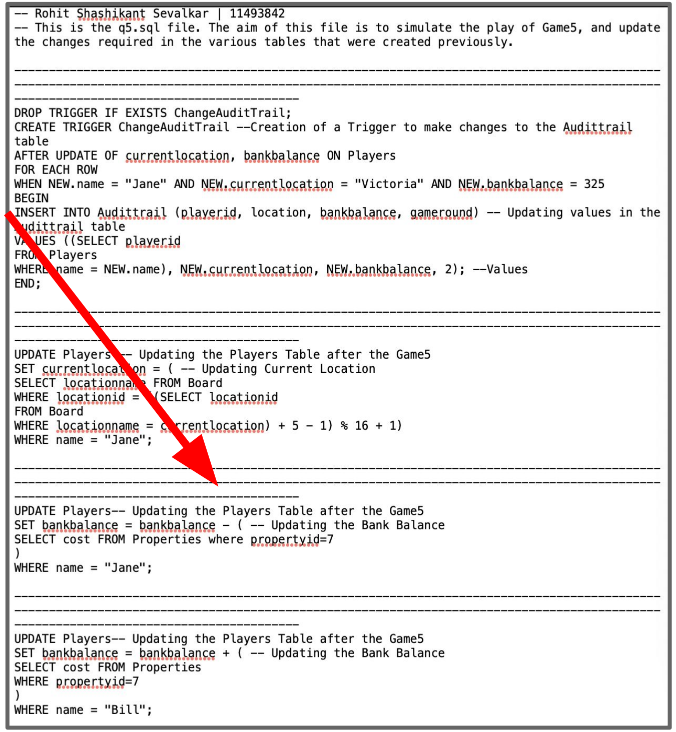
```
-- Rohit Shashikant Sevalkar | 11493842
-- This is the q3.sql file. The aim of this file is to simulate the play of
Game3, and update the changes required in the various tables that were created
previously.

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-------------------------------------------------------------------
DROP TRIGGER IF EXISTS ChangeAuditTrail;
CREATE TRIGGER ChangeAuditTrail  --Creation of a Trigger to make changes to the
Audittrail table
AFTER UPDATE OF currentlocation ON Players
WHEN NEW.name = "Mary"  AND NEW.currentlocation = "Jail"
BEGIN
INSERT INTO Audittrail (playerid, location, bankbalance, gameround) -- Updating
values in the Audittrail table
VALUES (NEW.playerid, NEW.currentlocation, NEW.bankbalance, 1); --Values
END;

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
-------------------------------------------------------------------
UPDATE Players ---- Updating the Players Table after the Game3

SET currentlocation = CASE
WHEN (
SELECT locationname
FROM Board
WHERE locationid = ( -- Updating the Location
(SELECT locationid FROM Board
WHERE locationname = Players.currentlocation) + 4 - 1) % 16 + 1) = "Go To Jail"
THEN "Jail"
END
WHERE Name = "Mary" AND currentlocation != "Jail";
```



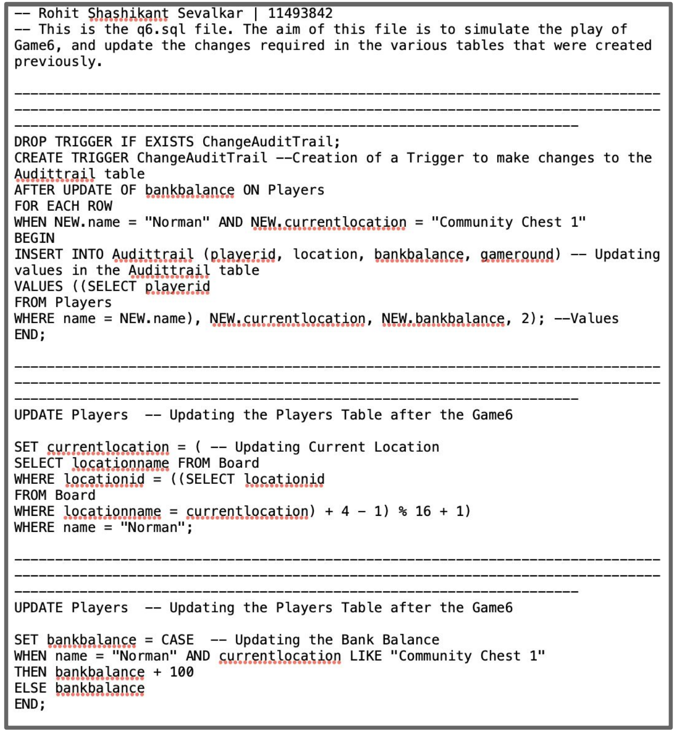
```
-- Rohit Shashikant Sevalkar | 11493842
-- This is the q4.sql file. The aim of this file is to simulate the play of Game4, and
update the changes required in the various tables that were created previously.

--------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
-------------------------------------------------------
DROP TRIGGER IF EXISTS ChangeAuditTrail;
CREATE TRIGGER ChangeAuditTrail --Creation of a Trigger to make changes to the
Audittrail table
AFTER UPDATE OF currentlocation, bankbalance ON Players
FOR EACH ROW
WHEN NEW.name = "Bill" AND NEW.currentlocation = "AMBS" AND NEW.bankbalance = 150
BEGIN
INSERT INTO Audittrail (playerid, location, bankbalance, gameround) -- Updating values
in the Audittrail table
VALUES ((SELECT playerid
FROM Players
WHERE name = NEW.name), NEW.currentlocation, NEW.bankbalance, 1); --Values
END;

--------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
-------------------------------------------------------
UPDATE Players -- Updating the Players Table after the Game4
SET currentlocation = (
SELECT locationname FROM Board -- Updating Current Location
WHERE locationid = ((SELECT locationid
FROM Board
WHERE locationname = currentlocation) + 2 - 1) % 16 + 1)
WHERE name = "Bill";

--------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
-------------------------------------------------------
UPDATE Players -- Updating the Bank Balance
SET bankbalance = bankbalance - 400
WHERE name = "Bill";

--------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------
-------------------------------------------------------
UPDATE Players -- Updating Properties owned
SET property = "Victoria"||", AMBS"
WHERE name = "Bill";
```

Code Explanation | q5.sql and q6.sql

- Here in the code for gameplay5 we can see that the bankbalance for the players is updated in such a way that it retrieves the cost for the properties from another table and used that number to update the bankbalance.



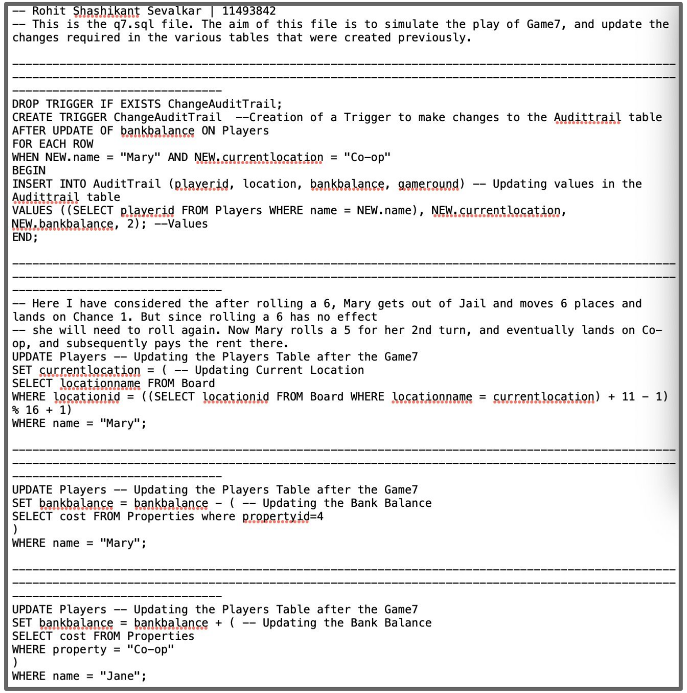
```
-- Rohit Shashikant Sevalkar | 11493842
-- This is the q5.sql file. The aim of this file is to simulate the play of Game5, and update
the changes required in the various tables that were created previously.

-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------
DROP TRIGGER IF EXISTS ChangeAuditTrail;
CREATE TRIGGER ChangeAuditTrail --Creation of a Trigger to make changes to the Audittrail
table
AFTER UPDATE OF currentlocation, bankbalance ON Players
FOR EACH ROW
WHEN NEW.name = "Jane" AND NEW.currentlocation = "Victoria" AND NEW.bankbalance = 325
BEGIN
INSERT INTO Audittrail (playerid, location, bankbalance, gameround) -- Updating values in the
Audittrail table
VALUES ((SELECT playerid
FROM Players
WHERE name = NEW.name), NEW.currentlocation, NEW.bankbalance, 2); --Values
END;

-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------
UPDATE Players -- Updating the Players Table after the Game5
SET currentlocation = ( -- Updating Current Location
SELECT locationname FROM Board
WHERE locationid = ((SELECT locationid
FROM Board
WHERE locationname = currentlocation) + 5 - 1) % 16 + 1)
WHERE name = "Jane";

-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------
UPDATE Players-- Updating the Players Table after the Game5
SET bankbalance = bankbalance - ( -- Updating the Bank Balance
SELECT cost FROM Properties where propertyid=7
)
WHERE name = "Jane";

-----------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------
-----------------------------------------
UPDATE Players-- Updating the Players Table after the Game5
SET bankbalance = bankbalance + ( -- Updating the Bank Balance
SELECT cost FROM Properties
WHERE propertyid=7
)
WHERE name = "Bill";
```



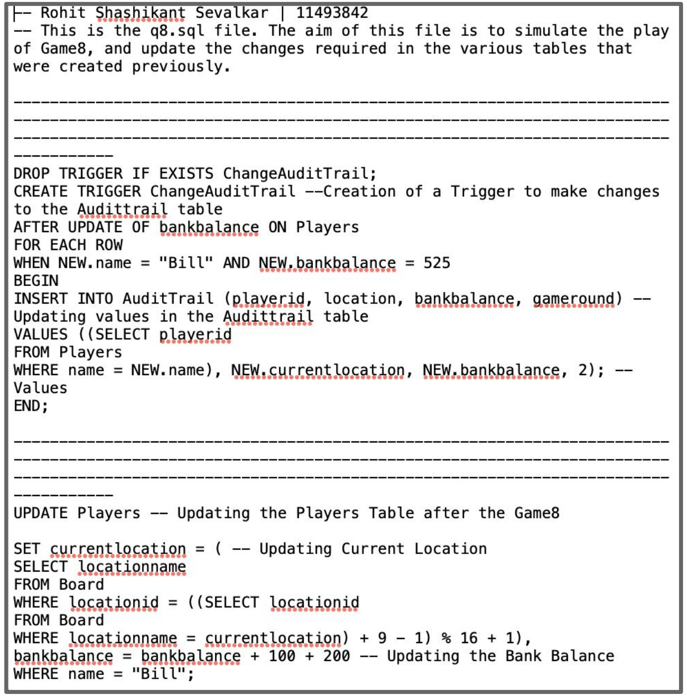
```
-- Rohit Shashikant Sevalkar | 11493842
-- This is the q6.sql file. The aim of this file is to simulate the play of
Game6, and update the changes required in the various tables that were created
previously.

-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-----------------------------------------------------------------
DROP TRIGGER IF EXISTS ChangeAuditTrail;
CREATE TRIGGER ChangeAuditTrail --Creation of a Trigger to make changes to the
Audittrail table
AFTER UPDATE OF bankbalance ON Players
FOR EACH ROW
WHEN NEW.name = "Norman" AND NEW.currentlocation = "Community Chest 1"
BEGIN
INSERT INTO Audittrail (playerid, location, bankbalance, gameround) -- Updating
values in the Audittrail table
VALUES ((SELECT playerid
FROM Players
WHERE name = NEW.name), NEW.currentlocation, NEW.bankbalance, 2); --Values
END;

-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-----------------------------------------------------------------
UPDATE Players  -- Updating the Players Table after the Game6

SET currentlocation = ( -- Updating Current Location
SELECT locationname FROM Board
WHERE locationid = ((SELECT locationid
FROM Board
WHERE locationname = currentlocation) + 4 - 1) % 16 + 1)
WHERE name = "Norman";

-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-----------------------------------------------------------------
UPDATE Players  -- Updating the Players Table after the Game6

SET bankbalance = CASE  -- Updating the Bank Balance
WHEN name = "Norman" AND currentlocation LIKE "Community Chest 1"
THEN bankbalance + 100
ELSE bankbalance
END;
```

Code Explanation | q7.sql and q8.sql

- The format of the codes for the gameplays q7 and q8 is no different than what I have mentioned in the earlier codes.
- The format as well as the syntax is similar to the previous gameplays. With game8 the 2 rounds for the games end, and the game is complete.

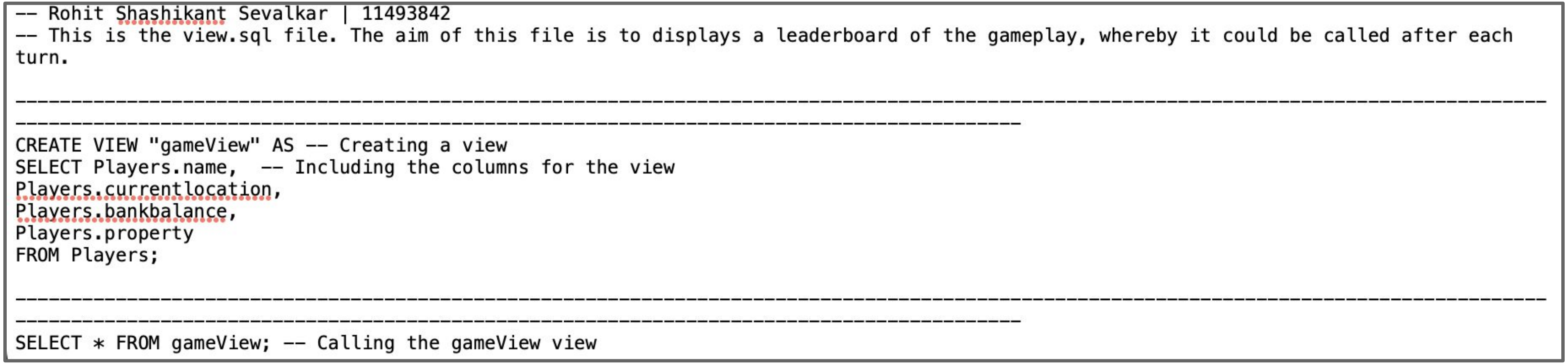


```sql
-- Rohit Shashikant Sevalkar | 11493842
-- This is the q7.sql file. The aim of this file is to simulate the play of Game7, and update the
changes required in the various tables that were created previously.

-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------
------------------------------
DROP TRIGGER IF EXISTS ChangeAuditTrail;
CREATE TRIGGER ChangeAuditTrail  --Creation of a Trigger to make changes to the Audittrail table
AFTER UPDATE OF bankbalance ON Players
FOR EACH ROW
WHEN NEW.name = "Mary" AND NEW.currentlocation = "Co-op"
BEGIN
INSERT INTO AuditTrail (playerid, location, bankbalance, gameround) -- Updating values in the
Audittrail table
VALUES ((SELECT playerid FROM Players WHERE name = NEW.name), NEW.currentlocation,
NEW.bankbalance, 2); --Values
END;

-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------
------------------------------
-- Here I have considered the after rolling a 6, Mary gets out of Jail and moves 6 places and
lands on Chance 1. But since rolling a 6 has no effect
-- she will need to roll again. Now Mary rolls a 5 for her 2nd turn, and eventually lands on Co-
op, and subsequently pays the rent there.
UPDATE Players -- Updating the Players Table after the Game7
SET currentlocation = ( -- Updating Current Location
SELECT locationname FROM Board
WHERE locationid = ((SELECT locationid FROM Board WHERE locationname = currentlocation) + 11 - 1)
% 16 + 1)
WHERE name = "Mary";

-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------
------------------------------
UPDATE Players -- Updating the Players Table after the Game7
SET bankbalance = bankbalance - ( -- Updating the Bank Balance
SELECT cost FROM Properties where propertyid=4
)
WHERE name = "Mary";

-------------------------------------------------------------------------------------------------
-------------------------------------------------------------------------------------------------
------------------------------
UPDATE Players -- Updating the Players Table after the Game7
SET bankbalance = bankbalance + ( -- Updating the Bank Balance
SELECT cost FROM Properties
WHERE property = "Co-op"
)
WHERE name = "Jane";
```

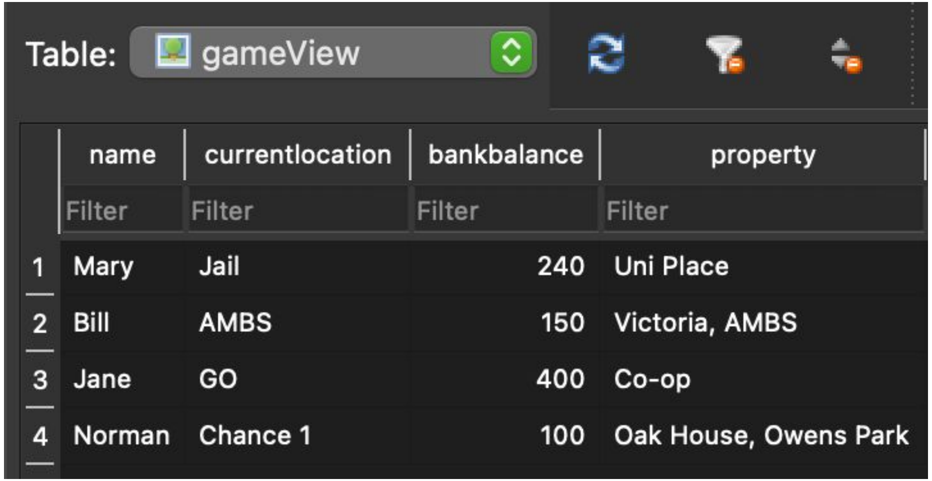


```sql
-- Rohit Shashikant Sevalkar | 11493842
-- This is the q8.sql file. The aim of this file is to simulate the play
of Game8, and update the changes required in the various tables that
were created previously.

---------------------------------------------------------------------
---------------------------------------------------------------------
---------------------------------------------------------------------
-----------
DROP TRIGGER IF EXISTS ChangeAuditTrail;
CREATE TRIGGER ChangeAuditTrail --Creation of a Trigger to make changes
to the Audittrail table
AFTER UPDATE OF bankbalance ON Players
FOR EACH ROW
WHEN NEW.name = "Bill" AND NEW.bankbalance = 525
BEGIN
INSERT INTO AuditTrail (playerid, location, bankbalance, gameround) --
Updating values in the Audittrail table
VALUES ((SELECT playerid
FROM Players
WHERE name = NEW.name), NEW.currentlocation, NEW.bankbalance, 2); --
Values
END;

---------------------------------------------------------------------
---------------------------------------------------------------------
---------------------------------------------------------------------
-----------
UPDATE Players -- Updating the Players Table after the Game8

SET currentlocation = ( -- Updating Current Location
SELECT locationname
FROM Board
WHERE locationid = ((SELECT locationid
FROM Board
WHERE locationname = currentlocation) + 9 - 1) % 16 + 1),
bankbalance = bankbalance + 100 + 200 -- Updating the Bank Balance
WHERE name = "Bill";
```

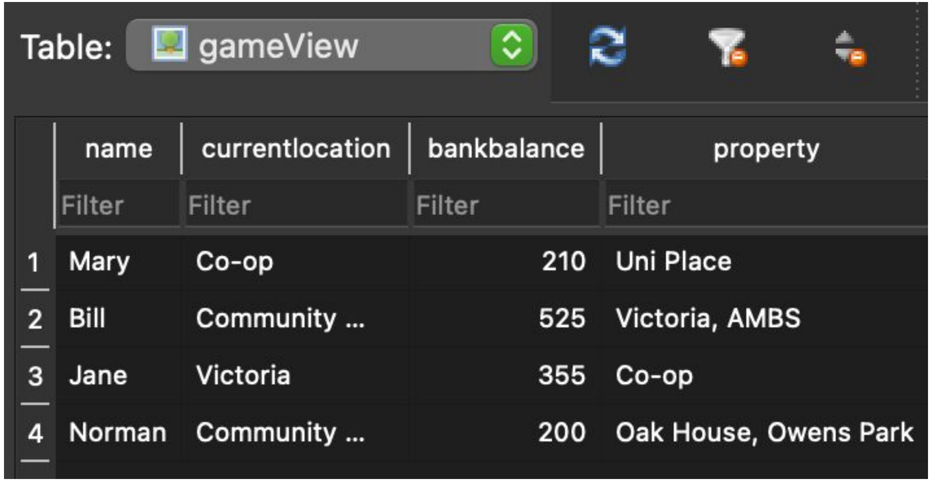
View.sql Code Explanation | View for Gameround 1 & 2



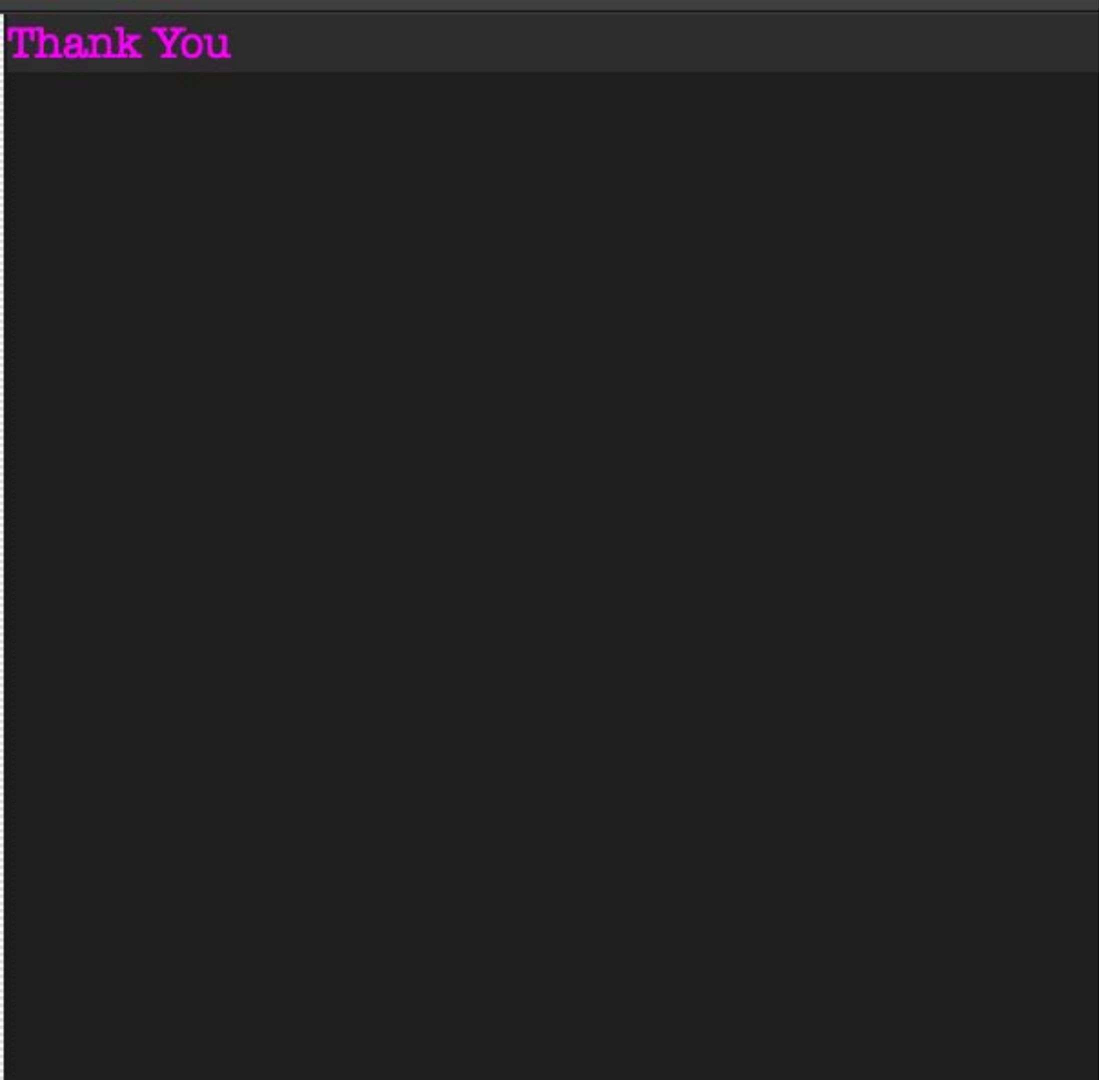
```
-- Rohit Shashikant Sevalkar | 11493842
-- This is the view.sql file. The aim of this file is to displays a leaderboard of the gameplay, whereby it could be called after each
turn.

-----------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------

CREATE VIEW "gameView" AS -- Creating a view
SELECT Players.name,  -- Including the columns for the view
Players.currentlocation,
Players.bankbalance,
Players.property
FROM Players;

-----------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------
SELECT * FROM gameView; -- Calling the gameView view
```



	name	currentlocation	bankbalance	property
1	Mary	Jail	240	Uni Place
2	Bill	AMBS	150	Victoria, AMBS
3	Jane	GO	400	Co-op
4	Norman	Chance 1	100	Oak House, Owens Park



	name	currentlocation	bankbalance	property
1	Mary	Co-op	210	Uni Place
2	Bill	Community ...	525	Victoria, AMBS
3	Jane	Victoria	355	Co-op
4	Norman	Community ...	200	Oak House, Owens Park



Thank You