**TCBID-** Technion Certified Business Intelligence Developer

**SQL server management studio Project**

Casino - Royal

**Submitted by:**

Efrat Shapira

Eran Mataraso

Liat Shadmi

**Project overview**

During our studies as an TCBID students, we required to exhibit an SQL server project that reflect the knowledge we acquire throughout the course.

The project will reflect our abilities in SQL server management studio, as we display the solutions and assumption that generate through our work.

**Project requirements-**

The requirement of this project, is to simulate an online casino website. We will start by creating the database, logic connection and constraints that will be our base for the online game. Additionally, we will build a collection of logically related scripts, that allowing the player to participate the game in a simple, fun and continuously way.

**The course of the game-**

In order to participate in the online casino, the player will be asked to register or login the game first. There is certain restriction that the player must follow in order to register or play the casino (for example: 18 years old restricted entry, using strong password and correct email address etc.).

After logging in successfully, the player will enter the CASINO ROYAL lobby. The player will be able to play the ultimate game of Slot Machine. In any given time, the player will be able to cashout his winnings, and view all statistics and game rounds reports.

.

**Data Base**

After Creating the DB\_Casino\_Royal Database, we generate the data structure, by creating the tables and the relationship within that database.

**ERD:**

**תמונה שמכילה צילום מסך

התיאור נוצר באופן אוטומטי**

**DDL**

There are seven tables overall, six of them connected by constraints:

* UTBL.UserName

A main table, that stores information about the players (‘users’) that registered the game. The user inserts the information in the register process (registration procedure).

This table has a PRIMARY KEY constraint in the User ID column, And FOREIGN KEY constraints in ‘country’ and ‘gender’ column that references to Country table and Gender table.

The table also show if the player is logged in, and count the number of times he enters wrong password. The ‘bankroll’ column calculate the amount of money the player collect in the game, after every transaction (Deposit, Cash out, Win, Lose, Bonus).

* UTBL.Country

This table contains a list of all countries worldwide. The country name column has a PRIMARY KEY constraint, that prevent from the user enter a country that not on this list, when registered.

* UTBL.Gender

The table holds the ‘male’ and ‘female’ phrase, and has a PRIMARY KEY constraint in the gender column. This constrain will prevent from the user enter a spelling mistake when he or she will be asked to enter gender, in the registration form

* UTBL.GameRound

The table provides information about the game type and number of rounds for each User ID when a game occurs. Additionally, this table hold data about the winning amount and losses for each user.

The User ID column has a FOREIGN KEY constraint references to UTBLE.UserName table.

Because there is only one game in our casino (Slot Machine), there was no need to create a Game Type table with different types of games. The column ‘Game Type’ will hold the phrase ‘Slot Machin’ in this case.

* UTBL\_Transaction\_Type

All Transactions that a player can generate in the game. This table connect to the UTBLE.BankRoll\_Trans.

* UTBL.BankRoll\_Trans

This table, provides information about every transaction that occur by each player in the game, and the amount of money in each transaction.

The ‘User ID’ column has a FOREIGN KEY constraint references to UTBLE.UserName table.

And the ‘TransactionType’ column has a FOREIGN KEY constraint references to UTBL\_Transaction\_Type in ‘TransactionType’ column.

* UTBL\_Symbols

The table show six unique symbols in the Slot Machine game. This table has no connection to the other tables in the data base, but has a logic influence in the procedure of the game.

**Registration procedure**

USP Registration PROCEDURE

When a new user enters the casino royal online game, he will be asked to register the game, and fill in the registration form.

The Registration procedure contains parameters similar to the registration form (@UserName, @Password, @FirstName, @LastName, @Address, @Country, @Email, @Gender, @BirthDate).

In this procedure we’ll be using function, if condition and view to check if the user follow a certain restriction (for example: 18 years old restricted entry, using strong password and correct email address etc.).

After the user enters a correct parameter and register successfully, all these data insert into the User Name table.

In addition, the user will get a welcome bonus of 10$. This procedure creates also a transaction row in the Bank Roll table.

* dbo.RandomNum FUNCTION

We use Label loop in this procedure, to check if a user name exists in the user name table.

if the player enters a user name that already exists, this function will provide an alternative user name with a random number. Following this massage:

'The UserName Is already exists. You can choose another UserName: '+@AlternativeUser

* COLLATE CLAUSE

The player password must be strong. We use The COLLATE clause to insure that the player enters a password that contains capital letters, small letters and digit.

**Login procedure**

USP Login PROCEDURE

After the player register the casino royal game successfully, he will be asked to log in with the correct user name and password.

After a successful login the player will get 'You Loged in secssecfuly' massage, and the amount of money in his bank roll.

If a player tries to login with a wrong password 5 times in a row, he will get 'You are blocked, please call the support' massage and will be blocked from the game. For each time, the ‘login counter’ column in the user name table will be updated.

* dbo.GeneratePassword FUNCTION

This function creates a random strong password, that contains capital letters, small letters and digits.

* USP\_UnBlock PROCEDURE

This procedure will provide a new and strong password, using the dbo.GeneratePassword FUNCTION. And sets the ‘login counter’ column in the user name table to zero.

The procedure will show this massage: 'Your new password is: @RandPassword

**Cashier procedure**

USP Cashier PROCEDURE

1. After a successful login, the player will enter the casino lobby, and will be able to play. In The lobby the player will be able to choose between the game slot machine, or the cashier.
2. when the player chooses the cashier, he can ‘deposit’ any amount of money to play, or ‘cashout’ his winnings. If the bank roll amount in the user name table, less than the cashout amount, this massage will occur:
3. 'The amount you have is less than the amount you want to Cashout. Please Try Again.'
4. for every transaction in this procedure, the data inserts in to the Bankroll\_Trans table.

**The Game procedure**

USP GameSM PROCEDURE

In order to start the slot machine game, the user required to enter a bet amount. The bet amount, must be bigger than the bank roll amount in the user name table, or the player will get this massage:

'You can only bet on amount of money you have in your bankroll. please go to the Cashier or reduce your bet amount'.

The @bet amount parameter, stored temporary amount until the game resume. If the procedure ends with 3 similar symbols, it’s a win and the player will earn the @bet amount. If the procedure not ends with 3 similar symbols, the player loses the @bet amount.

Each transaction that occur (Win or Lose), create a row in the Bank Roll trans table, and the Game Round table.

* NewID() Function

In the GameSM procedure, we use the NewID () function, to assimilate a rounds of 3 unique symbols from the symbols table.

* Bankroll Insert TRIGGER

Every time that a player makes a transaction (Deposit, Cash out, Win, Lose, Bonus), by playing slot machine, addressing the cashier or register successfully. a DDL commend will insert a row with data to the Bankroll Trans table.

The trigger invokes when a row is inserted to the Bankroll Trans table in the database.

Automatically, the trigger will update the bankroll amount in User Name table.

**Reports**

The player will be able to generate several reports, in any given time.

* Game History report

In this report we used Multi Statement TVF, in order to create a variable table that show the following data: Game type (slot machine in our case), bet amount and winnings by date and time range.

* Bankroll Transaction report

In this report we used Inline TVF. The report provides information for each player, about transactions in the game and total amount of money after every action. in addition, the player will be able to sort the report by date and time.

* Game Statistics report

In order to generate a repot that show statistics data for each user, we create a VIEW. The view contains data about the number of rounds, winnings amount and total bets amount, from the last 7 days.