**INTRO**

Grab your professional fishing tackle and go on an extreme and thrilling fishing adventure. Enter the game, choose a representative and cast a fishing rod. It’s that simple! In each round three fishermen attempt to catch a chubby fish for dinner.

Only one of them can succeed and win a prize of 1,000 points (and a delicious fish...). Those who couldn't catch the fish will lose 100 points.

So... how skilled are you in fishing?

Game Representative

**'Fishenzon'**   
Geeky guy. Likes to calculate his ways and odds in the water.  
  
**'Gefilefish'**Mama's boy. Will do almost anything to catch a fish for his mommy to cook.  
  
**'Fish and chips'**   
Chubby and loveable. hungry all the time and will fight over his dinner with a shark if necessary.

\*\* Choose yours best Representative, grab your gear and let's go fishing \*\*

**DATABASE ARCHITECTURE**

Users Table

Main table 'Users' collect new users (insert into procedure) and declare active or inactive players.

\*\* This table is connected to 3 other tables by FK.

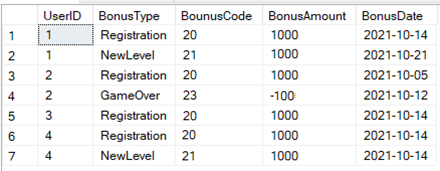


Scores table

collect all scores and bonus types by users.

The scores table are dynamic and keep track of who's allowed to play (Have enough credit) and who utilize his credit and cannot participate in the game.

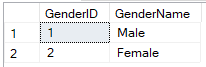
Furthermore, the credit table receiving new data in every game a player participates (insert into procedure)  
\*\* Connected to the main table 'Users' by FK 'User ID', and as a FK to Fishermen table by 'Fishmenid'. This table PK is 'Bonus code'.



Gender Table

Was created as a 'drop down' solution as part of a registration requirements  
(Multiple choice instead of a text insertion).

\*\* Connected to the main table 'Users' by PK 'Gender Name'.



Country table

Was created as a 'drop down' solution as part of a registration requirements (Multiple choice instead of a text). There are **165 different countries** added to the data base. No matter where the player placed on the planet, he will be able to go fishing with us.

\*\* Connected to the main table 'Users' by PK 'Country Name'.



Fishermen table

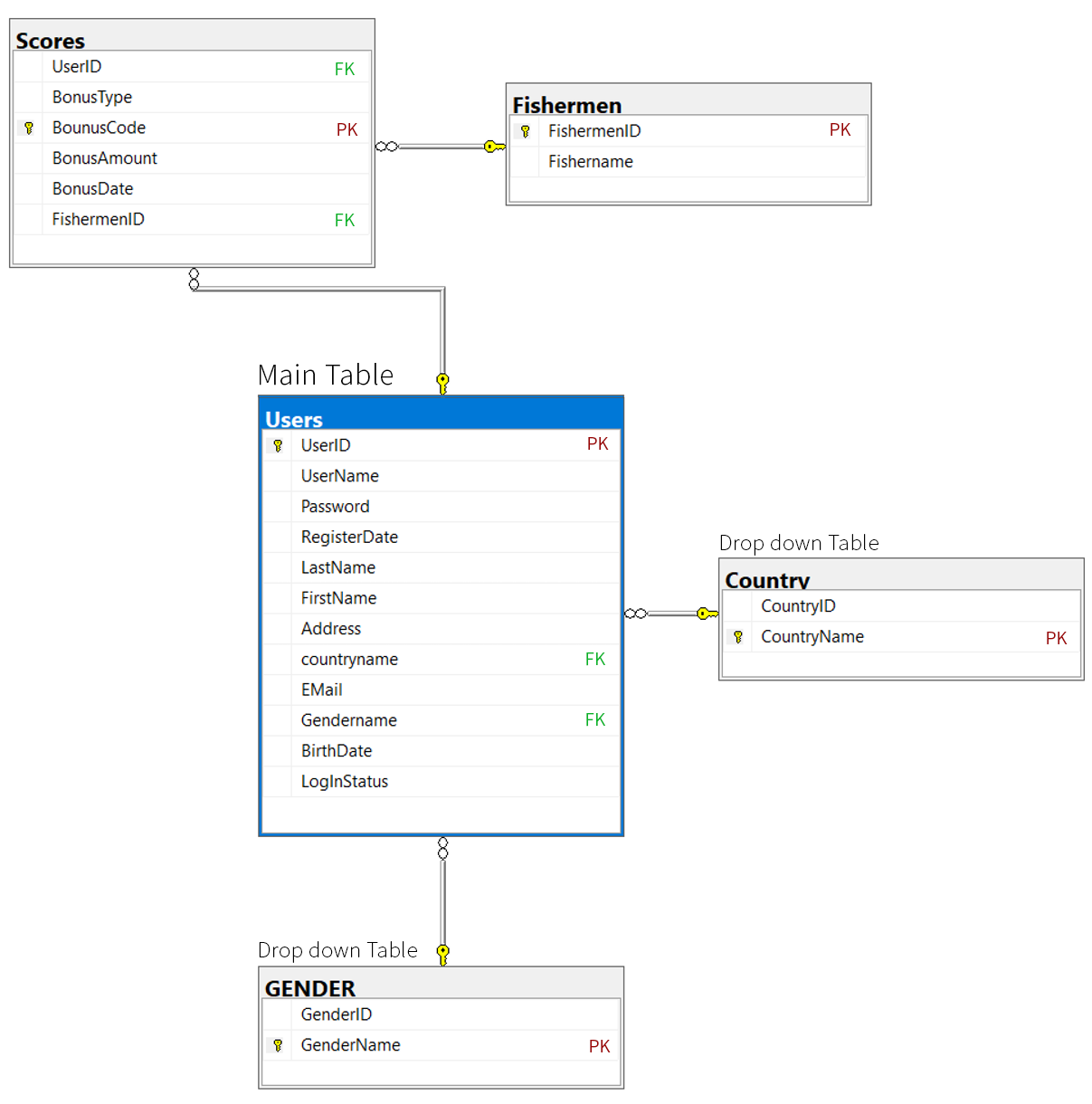
Was created as a 'drop down' solution.

The game is based on three fishermen and the users are required to choose one of them as their representative.

\*\* Connected to 'Score table' by PK 'Fishmen ID'.



**ERD**



**STORED PROCEDURES**

1. **Enter the game**

The first stored procedure was established in order to sort the prime type of the customer - does he exist in our database? or a new member of the game and needs to register first.

CREATE PROCEDURE enter\_the\_game (@first\_visit nvarchar (30))

As

IF @first\_visit = 'This is my first time'

Begin

Print ('Go to registration')

return

END

Else

Begin

Print ('Go to login')

return

END

1. **Registration Form**
   1. Age Chek - DATEDIFF function

The game has an age restriction and only 13 years old players are allowed to participate.

In order to check the player age at registration we created a procedure that calculates the date of registration (Get date()) with the Birthday information the player is required to insert in the registration form by using the function DateDiff.   
  
If the player is under the age of 13 he will not be able to register, therefore he will not be able to participate in the game until he is the right age.

PROCEDURE GET\_AGE (@DATEPICKER date)

As

Declare @number int

Set @number = (select Datediff(yyyy,@DATEPICKER,getdate()))

If @number < 13

Print

'The game permitted for 13 years old and up.   
Hope to see you again once you reach the age of 13’

Go

* 1. Mail Address check - If Exist operator

This operator was established in order to make sure the registration mail the player inserted at the registration form has not been in use and exists at our database, and to make sure the text pattern is correct for an email address.

In each case the player receives a message if any problem occurs with his email address.

PROCEDURE GET\_mail (@mail nvarchar(50))

As

Begin

If Exist (SELECT \* FROM users WHERE @MAIL = EMail)

Print   
'This email address is taken. Please choose another one'

If (@mail not like '%@%')

Print

'Invalid email'

End

Go

* 1. User Name - RAND function

This operator was established in order to check the username the player types at the registration form. If the username he picked is already in use by another player he will receive an indicator asking him to choose a different username, plus a few system recommendations as a new user name.

The system recommendation combines the player's username and a random number between the numbers 1-1,000, after double checking that our recommendation is also not in use by another player (If Exist), meaning new user.

PROCEDURE get\_username (@username nvarchar(50))

As

Declare   
@random\_Number int

Set  
@random\_Number = FLOOR (RAND()\*(1000-1+1)+1)

If Exist (SELECT \* FROM users WHERE @username = username)

Print   
'Invalid username, you can use this: ' + @username + '' + Cast (@random\_Number as varchar(10))

* 1. Password check - If Exist operator

The password the player inserts in the registration form must answer to few constraints. With a **collate** parameter we are able to assure that each of the constraints are answered or and the password is according to list ahead:

* The password must contain at least 7 letters (LEN function)
* Contain at least 1 capital letter (Upper function)
* Contain at least 1 lowercase letter (Lower function)
* Contain numbers (Wild Card function)
* Can not use the word password in it (Wild Card function)
* Can not use the username in a new password (If Exist operator)

In each case the player receives a message if any problem occurs with his password.

else

If len(@password)<7

Print 'The Password must contain at least 7 letters'

Else

If @password not like '%[0-9]%'

Print 'The Password must contain at least 1 number'

Else

If @password Collate Latin1\_General\_BIN not LIKE '%[a-z]%'

Print 'The Password must contain at least 1 small letter'

Else

If @password Collate Latin1\_General\_BIN not like ('%[A-Z]%')

Print 'The Password must contain at least 1 capital letter'

Else

If @password like '%password%'

Print 'The Password cannot contain the word [Password] in any combination'

Else

If @Password = @Username

Print 'The Password must be unique.   
Please choose another password without using your username as part of your password’

Else

* 1. Insert Into Users Table & Score Table

If all of the constraints above are provided by the player and his registration is correct and intact, all the information is inserted into the Users database table.

All new players are being given 1,000 points as a registration bonus.

Once the registration is completed the user will have his points and the database will receive a new row at the Score table under the right UserID **(@@Identity**).

The player will receive a message informing him with the bonus amount.

Else

Begin

**Insert into Users**

Values   
(@Username,

@Password, default,

@Lastname,

@Firstname,

@Address,

@Country,

@Mail,

@Gender,

@Datepicker, default)

**Insert into Scores**

values   
(@@Identity,'Registration',1000, default, null)

print

‘Congratulations!  You have successfully registered.

We want to welcome you with 1000 points as a registration bonus.

You can log in now.

Enjoy your fishing adventure'

End

Go

* 1. Login procedure

In this procedure there are 2 ways of proceeding:

If the player inserts the correct username and password, he is allowed to enter the game. In this case our database will automatically receive an indicator of the player being active (SET LogInStatus = 1). If he is not recognized by the system, he is either not registered or already logged in.

In each case the player receives a message informing him with his login status.

PROCEDURE login\_check (@username nvarchar(50), @password nvarchar(50))

As

If Exist   
(SELECT \* FROM users WHERE username = @username and [password] = @password)

Begin

Update users

Set LogInStatus = 1

where UserName = @username and [password] = @password

Print   
'Log in complete'

End

Else

Begin

Print   
'you don't have a user yet, please go to registration.  
If you have registered before you are probably already logged in'

End

Go

* 1. 'Main Menu Options' Procedure

Once the player can make a login, he receives 2 options:  
A. Start the game

B. Score chart

Declare @Catch NVARCHAR

Exec @Catch = Selection\_menu 'score','dani1 (User Name)'

Option A  
Redirecting to game screen for an active play.

Option B

View the score chart to be informed with the bonus amount accumulated to this identity user. The score is drawn from the score table by userID checks.

The player can view the following data - FishName, sum amount of points.

In blue marked the alternative score chart the user see.

PROCEDURE Selection\_menu (@menu nvarchar(10), @username nvarchar(50))

As

If @menu = 'score'

select u.UserName, BonusAmount, FisherNAME

From Scores s full join users u

On s.UserID = u.UserID full join Fishermen f

On f.FishermenID = s.FishermenID

Where u.username = @username

Else

If @menu = 'game'

Begin

Print   
'Welcome. Enjoy your fishing adventure'

End

Go

* 1. Fishermen Representative Procedure

As elaborated in the intro of this document, the player must choose a representative from a fishermen list (one of 3 options).

Declare @Catch Nvarcahar

Exec @CATCH =   
fishermen representative 'GefilteFish'(Fisherman), 'dani1' (Username)

Once the player enters the game, and chooses a fisherman on his behalf, the choice he makes will automatically be inserted into the score table.

In this case we will be able to show the player his score (paragraph 4 option B),

and as a data analyst we will be able to keep track of the player choices of a representative, odds of winning and scores per representative.

Procedure  fishermen representative (@fishername nvarchar(30), @username nvarchar(50))

As

If Exist (SELECT \* FROM Fishermen WHERE @fishername = fishername)

Begin

If @fishername = 'Fishenzon' and @username =

(select username from users where UserName = @username)

Begin

Update scores

Set **FishermenID = 1**

where userID =

(select userid from Users where UserName = @username)

End

Else

If @fishername = 'GefilteFish' and @username =

(select username from users where UserName = @username)

Begin

Update scores

Set **FishermenID = 2**

where userID =

(select userid from Users where UserName = @username)

End

Else

If @fishername = 'FishAndChips' and  @username =

(select username from users where UserName = @username)

Begin

Update scores

Set **FishermenID = 3**

where userID =

(select userid from Users where UserName = @username)

End

Go

1. **Catch The Fish – The Game**

As elaborated in the intro of this document, there are 3 representatives as fishermen. The player must choose one of them as an avatar:

Declare   
@Fishermen\_1 nvarchar (20)

\*\* changes to Fishermen \_2 OR Fishermen \_3 \*\*

The script has 3 Different sets - Fishername no1, no2 and no3.

Declare @f1 int (Fishmen ID)

Each fisherman represents themself during fishing - so the player visual will be a fish in the water.

Set @Fishermen\_1 = '><((0>'

Each fish is swimming in the water in a different paist (like in a race).

The movement is monitored by a @ variable.

In order to keep the race under control by a different levels of advancement,

Each level (1- 9) is monitored by how many “steps” the fish is allowed to proceed

Declare @upper\_f1 int

Declare @lower\_f1 int

The fish are proceeding in random steps so one game will not be the same as the one that took place before. In every level (1-10) the variable upper\lower receives a different number of paces in a loop (Using **Counter Loop**)

Also, all three fish are monitored to ensure they can only proceed from their last place. Meaning - they can move forward but never backwards.

The final level - Fish catching and declaring a winner

The game ends with the 10th level, in which one fish is caught by the winning Fishmen. Therefor, two other fishmen are losing.

A disclaimer, with a print exec, appear to the players – represent on the winner by User name, Fishmen representative, Bonus amount for this round (1,000 points).

At the same time the Score table is automatically updated with the new data –

1. The winner gains 1,000 points (by userId)
2. The loser reduced with -100 points (by userid)

Hopefully you enjoyed reading our paper and imagined, as us, the thrilling fishing adventure of our game.

Thank you for your time and patience,

Tal Cohen

Or Shukrun

Daniel Pinchasov