Procedures and triggers

Create script to add points for excites Bahraini

Name: addPointInPointTable

Description: Add 5 Points if the User is Bahraini for excites users

Status: One time

```
USE [YouthProgramRegistraionV2]
GO
/****** Object:  StoredProcedure [dbo].[addPointsInPointsTable]    Script Date: 5/18/2023
7:12:35 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           <Alasmawi,,Mohammed>
-- Create date: <3/28/2023>
-- Description:      <Add 5 Points if the User is bahraini>
-- =============================================
ALTER PROCEDURE [dbo].[addPointsInPointsTable]
       -- Add the parameters for the stored procedure here
          @Points int = NULL,
       @description  nvarchar(50) = NULL,
       @Type nvarchar(50) = NULL,
       @CreatedBy nvarchar(50) = NULL,
       @ProgramID int = NULL,
       @UserID int = NULL
AS
BEGIN
       -- SET NOCOUNT ON added to prevent extra result sets from
       SET NOCOUNT ON;
       DECLARE @myV int
       DECLARE @myC CURSOR
       DECLARE @ID int
       DECLARE @NameEN varchar(50)
       DECLARE @CreatedDate datetime



       SET @myV = 1
       SET @CreatedDate = GETDATE()
       SET @Points = 5
       SET @Type = 'internal'
       SET @ProgramID = 23330
       SET @description = 'The User is Bahraini'
```

```
	SET @myC = CURSOR FOR
	SELECT ID, NameEN from [YouthProgramRegistraionV2].[dbo].[User] where [IsBahrani]
= 1
	open @myC
	fetch next from @myC into @ID, @NameEN
    -- Insert statements for procedure here
	WHILE @@FETCH_STATUS = 0
	Begin



	INSERT INTO [dbo].[Points]
	    ([Points]
	    ,[description]
	    ,[Type]
	    ,[CreatedDate]
	    ,[ProgramID]
	    ,[UserID])
     VALUES
	    (@Points,@description,@Type,@CreatedDate,@ProgramID,@ID)



	SET @myV = @myV + 1
	fetch next from @myC into @ID, @NameEN


	end
	close @myC
	DEALLOCATE @myC
END
```

Create trigger for new Bahraini users

Name: isBahraini

Description: Add 5 Points if the User is Bahraini for new users

Status: Automatic trigger in SQL

```
USE [YouthProgramRegistraionV2]
GO
/****** Object:  Trigger [dbo].[isBahraini]    Script Date: 5/18/2023 7:25:44 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER TRIGGER [dbo].[isBahraini]
ON [YouthProgramRegistraionV2].[dbo].[User]

AFTER  INSERT
AS
Begin
    DECLARE @UserID int

    select @UserID = UserList.ID from inserted UserList


    UPDATE [dbo].[User]
    SET [TotalPoints] = 5
    WHERE [IsBahrani] = 1


    INSERT INTO [dbo].[Points]
           (
            [Points]
           ,[description]
           ,[Type]
           ,[CreatedDate]
           ,[CreatedBy]
           ,[ModifiedDate]
           ,[ModifiedBy]
           ,[ProgramID]
           ,[UserID])
     VALUES
           (5,'The User is Bahraini','internal',GETDATE(),'','','',23330, @UserID)



END
```

Create Script to update education field for exist users

Name: addEducationPoints

Description: update education field for existing users

Status: One Time Run

```
USE [YouthProgramRegistraionV2]
GO
/****** Object:  StoredProcedure [dbo].[addEducationPoints]    Script Date: 5/18/2023
7:27:34 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           <Alasmawi,,Mohammed>
-- Create date: <3/30/2023>
-- Description:      <update education field for exisiting users>
-- =============================================
ALTER PROCEDURE [dbo].[addEducationPoints]
       -- Add the parameters for the stored procedure here
         @Points int = NULL,
      @description nvarchar(50) = NULL,
      @Type nvarchar(50) = NULL,
      @CreatedBy nvarchar(50) = NULL,
      @ProgramID int = NULL,
      @UserID int = NULL
AS
BEGIN
       -- interfering with SELECT statements.
       SET NOCOUNT ON;

       Declare @myV int
       Declare @myC CURSOR
       Declare @ID int
       Declare @PointsID int
       Declare @Point int
       Declare @EducationalGradeID int
       Declare @CreatedDate datetime
       Declare @count int

       Set @myV = 1
       Set @CreatedDate = GETDATE()
       Set @ProgramID = 23332
       Set @description = 'This is the Education Points'
       Set @type = 'internal'
```

```
	Set @myC = CURSOR FOR
	Select ID, EducationalGradeID from [YouthProgramRegistraionV2].[dbo].[User]
	Set @count = @@ROWCOUNT
    open @myC
	fetch next from @myC into @ID, @EducationalGradeID


	while @@FETCH_STATUS = 0
	Begin
			if @EducationalGradeID = 1
			begin
				SELECT @PointsID = [ID]
					,@Point = [Points]
					FROM
[YouthProgramRegistraionV2].[dbo].[MstEducationalGrade]
					order by [ID]
					OFFSET 0 row
					fetch next 1 row only

				INSERT INTO [dbo].[Points]
				 ([Points]
				 ,[description]
				 ,[Type]
				 ,[CreatedDate]
				 ,[ProgramID]
				 ,[UserID])
				 VALUES

(@Point,@description,@type,@CreatedDate,@ProgramID,@ID);

				UPDATE [dbo].[User]
				SET [TotalPoints] = [TotalPoints] + @Point
				where [ID] = @ID


			end

			if @EducationalGradeID = 2
			begin
				SELECT @PointsID = [ID]
					,@Point = [Points]
					FROM
[YouthProgramRegistraionV2].[dbo].[MstEducationalGrade]
					order by [ID]
					OFFSET 1 row
					fetch next 1 row only

				INSERT INTO [dbo].[Points]
				 ([Points]
				 ,[description]
				 ,[Type]
				 ,[CreatedDate]
				 ,[ProgramID]
				 ,[UserID])
				 VALUES

(@Point,@description,@type,@CreatedDate,@ProgramID,@ID);
```

```
                                UPDATE [dbo].[User]
                                SET [TotalPoints] = [TotalPoints] + @Point
                                where [ID] = @ID


                        end

                        if @EducationalGradeID = 3
                        begin
                               SELECT @PointsID = [ID]
                                        ,@Point = [Points]
                                        FROM
[YouthProgramRegistraionV2].[dbo].[MstEducationalGrade]
                                        order by [ID]
                                        OFFSET 2 row
                                        fetch next 1 row only

                                INSERT INTO [dbo].[Points]
                                ([Points]
                                ,[description]
                                ,[Type]
                                ,[CreatedDate]
                                ,[ProgramID]
                                ,[UserID])
                                VALUES

(@Point,@description,@type,@CreatedDate,@ProgramID,@ID);

                                UPDATE [dbo].[User]
                                SET [TotalPoints] = [TotalPoints] + @Point
                                where [ID] = @ID


                        end

                        if @EducationalGradeID = 4
                        begin
                               SELECT @PointsID = [ID]
                                        ,@Point = [Points]
                                        FROM
[YouthProgramRegistraionV2].[dbo].[MstEducationalGrade]
                                        order by [ID]
                                        OFFSET 3 row
                                        fetch next 1 row only

                                INSERT INTO [dbo].[Points]
                                ([Points]
                                ,[description]
                                ,[Type]
                                ,[CreatedDate]
                                ,[ProgramID]
                                ,[UserID])
                                VALUES

(@Point,@description,@type,@CreatedDate,@ProgramID,@ID);

                                UPDATE [dbo].[User]
                                SET [TotalPoints] = [TotalPoints] + @Point
```

```
                                    where [ID] = @ID


                        end

                        if @EducationalGradeID = 5
                        begin
                               SELECT @PointsID = [ID]
                                        ,@Point = [Points]
                                        FROM
[YouthProgramRegistraionV2].[dbo].[MstEducationalGrade]
                                        order by [ID]
                                        OFFSET 4 row
                                        fetch next 1 row only

                                    INSERT INTO [dbo].[Points]
                                     ([Points]
                                      ,[description]
                                      ,[Type]
                                      ,[CreatedDate]
                                      ,[ProgramID]
                                      ,[UserID])
                                     VALUES

(@Point,@description,@type,@CreatedDate,@ProgramID,@ID);

                                    UPDATE [dbo].[User]
                                    SET [TotalPoints] = [TotalPoints] + @Point
                                    where [ID] = @ID


                        end
                        if @EducationalGradeID = 6
                        begin
                               SELECT @PointsID = [ID]
                                        ,@Point = [Points]
                                        FROM
[YouthProgramRegistraionV2].[dbo].[MstEducationalGrade]
                                        order by [ID]
                                        OFFSET 5 row
                                        fetch next 1 row only

                                    INSERT INTO [dbo].[Points]
                                     ([Points]
                                      ,[description]
                                      ,[Type]
                                      ,[CreatedDate]
                                      ,[ProgramID]
                                      ,[UserID])
                                     VALUES

(@Point,@description,@type,@CreatedDate,@ProgramID,@ID);

                                    UPDATE [dbo].[User]
                                    SET [TotalPoints] = [TotalPoints] + @Point
                                    where [ID] = @ID
```

```
                end
            Set @myV = @myV + 1
            fetch next from @myC into @ID, @EducationalGradeID
    End

END
```

Create Script to add Point in education field for new user

Name: addNewEducationPoint1

Description: update education field for new users

Status: added to the code to run each time for new user

```
USE [YouthProgramRegistraionV2]
GO
/****** Object:  StoredProcedure [dbo].[addNewEducationPoints1]    Script Date: 5/18/2023
7:28:39 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           <Alasmawi,,Mohammed>
-- Create date: <29/3/2023,,>
-- Description:      <add Education Points to point table and update the total points
field in user>
-- =============================================
ALTER PROCEDURE [dbo].[addNewEducationPoints1] (
       -- Add the parameters for the stored procedure here

      @CreatedBy nvarchar(50) = NULL,
         @EducationalGradeID int = Null,
      @UserID int = NULL
         )
AS
BEGIN
       -- SET NOCOUNT ON added to prevent extra result sets from
       -- interfering with SELECT statements.
       SET NOCOUNT ON;

       Declare @description nvarchar(50)
       Declare @PointsID int
       Declare @Points int
       Declare @Type nvarchar(50)
       Declare @CreatedDate datetime
       Declare @programID int

       Set @CreatedDate = GETDATE()
       Set @Type = 'internal'
       Set @programID = 23332
       Set @description = 'This is User Education'

    -- Insert statements for procedure here

       Select @Points = Points from
[YouthProgramRegistraionV2].[dbo].[MstEducationalGrade] where @EducationalGradeID = [ID]
```

```
  INSERT INTO [dbo].[Points]
      ([Points]
      ,[description]
      ,[Type]
      ,[CreatedDate]
      ,[CreatedBy]
      ,[ProgramID]
      ,[UserID])
VALUES
      (@Points,@description,@Type,@CreatedDate,@CreatedBy,@programID,@UserID)



  UPDATE [dbo].[User]
  SET [TotalPoints] += @Points
         WHERE @UserID = [ID] AND @EducationalGradeID = [EducationalGradeID]
  End
```

Create Script to grant users points for each program

Name: addPointToQualifyUser

Description: calculate the points for the existing qualify users from programRegistration table and add them to point table

Status: One Time Run

```
USE [YouthProgramRegistraionV2]
GO
/****** Object:  StoredProcedure [dbo].[addPointsToQualifyUsers]    Script Date:
5/18/2023 7:30:11 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           <Author,,Name>
-- Create date: <Create Date,,>
-- Description:      <calculate the points for the exisiting qualify users from
programRegistration table and add them to point table>
-- =============================================
ALTER PROCEDURE [dbo].[addPointsToQualifyUsers]
       -- Add the parameters for the stored procedure here
AS
BEGIN

       SET NOCOUNT ON;
    -- Insert statements for procedure here
       insert into Points([Points],
                                          [description],
                                          [Type],
                                          [CreatedDate],
                                          [ProgramID],
                                          [UserID])
       select m.Points, s.DescriptionEn, c.CategoryEn,GETDATE(), r.ProgramID,r.UserID
       from Program s
       join MstSubCategory m
       on s.SubCategoryID = m.ID
       join ProgramRegistration r
       on r.ProgramID = s.ID
       join MstCategory c
       on c.ID = m.CategoryID

       End
```

Create Script to Update the total points for the user

Name: addPointToUser

Description: Update the qualify points from the points table to total points in the user table

Status: One Time Run

```
USE [YouthProgramRegistraionV2]
GO
/****** Object:  StoredProcedure [dbo].[addPointsToUser]    Script Date: 5/18/2023
7:31:13 AM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           <Alasmawi,,Mohammed>
-- Create date: <3/30/2023>
-- Description:      <Update the qualify points from the points table to total points in
the user table>
-- =============================================
ALTER PROCEDURE [dbo].[addPointsToUser]

AS
BEGIN
Declare @UserID int
Declare @myV int
Declare @myC CURSOR
Declare @points int

       SET @myV = 0
       Set @myC = CURSOR FOR
       select UserID, SUM(Points) from Points GROUP BY UserID
       open @myC
       fetch next from @myC into @UserID ,@points


       WHILE @@FETCH_STATUS = 0
       Begin


       UPDATE [YouthProgramRegistraionV2].[dbo].[User]
       SET [TotalPoints] = @points
       WHERE [ID] =@UserID;
```

```
        SET @myV = @myV + 1
        fetch next from @myC into @UserID ,@points
        end

END
```

Create Script to Users Points once qualify

Name: isQualify

Description: Add Points for the Qualified users after program ends

Status: added to the code to run each time for new user

```
USE [YouthProgramRegistraionV2]
GO
/****** Object:  StoredProcedure [dbo].[IsQualified]    Script Date: 5/18/2023 7:32:07 AM
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           <Alasmawi,,Mohammed>
-- Create date: <3/31/2023>
-- Description:      <Add Points for the Qualified users after program ends>
-- =============================================
ALTER PROCEDURE [dbo].[IsQualified](
       -- Add the parameters for the stored procedure here

      @CreatedBy nvarchar(50) = NULL,
      @ProgramID int = NULL,
      @UserID int = NULL
       )
AS
BEGIN

       Declare @Points int
       Declare @CreatedDate datetime
       Declare @description nvarchar(MAX)
       Declare @Type nvarchar(MAX)
       Declare @CategoryID int
       Declare @SubCategoryID int


       SET NOCOUNT ON;
       Set @CreatedDate = GETDATE()
    -- Select statements for procedure here
       Select @description = [DescriptionEn],@CategoryID = [CategoryID], @SubCategoryID =
[SubCategoryID] from [Program] where [ID] = @ProgramID
       Select @Type = [CategoryEn] from [MstCategory] where [ID] = @CategoryID
       Select @Points = [Points] from [MstSubCategory] where [ID] = @SubCategoryID

       --Insert
       INSERT INTO [dbo].[Points]
           ([Points]
```

```
           ,[description]
           ,[Type]
           ,[CreatedDate]
           ,[CreatedBy]
           ,[ProgramID]
           ,[UserID])
     VALUES
           (@Points,@description,@Type,@CreatedDate,@CreatedBy,@ProgramID,@UserID)

       --Update Total Points in User table

       UPDATE [dbo].[User]
       SET [TotalPoints] += @Points
       WHERE [ID] = @UserID


END
```

Create Procedure to Update achievement approval

Name: UpdateApplicationDetails

Description: Approve the attachments and give points

Status: added to the code to run each time for new achievement approval

```
USE [YouthProgramRegistraionV2]
GO
/****** Object:  StoredProcedure [dbo].[UpdateApplicationDetails]    Script Date:
5/25/2023 12:44:16 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:           <Mohammed Alasmawi>
-- Create date: <25/5/2023>
-- Description:      <Approve the attachments and give points>
-- =============================================
ALTER PROCEDURE [dbo].[UpdateApplicationDetails] (
       -- Add the parameters for the stored procedure here
       @UserID int = NULL,
       @attachmentID int = NULL,
       @approvedBy nvarchar(50) = NULL,
       @comment nvarchar(50) = NULL,
       @ProgramID int = NULL
       )
AS
BEGIN
       -- SET NOCOUNT ON added to prevent extra result sets from
       -- interfering with SELECT statements.
       SET NOCOUNT ON;

       Declare @status nvarchar(50)
       Declare @approvalDate datetime
       Declare @participationType int
       Declare @Points int



       Declare @Type nvarchar(50)

       Set @approvalDate = GETDATE()
       Set @status = 'Approved'
       Set @type = 'Achievement'
```

```
      Select @Points = Points from [YouthProgramRegistraionV2].[dbo].[MstSubCategory]
where @participationType = [ID]



    -- Insert statements for procedure here
       INSERT INTO [dbo].[Points]
           ([Points]
           ,[description]
           ,[Type]
           ,[CreatedDate]
           ,[CreatedBy]
           ,[ProgramID]
           ,[UserID])
     VALUES
           (@Points,@comment,@Type,@approvalDate,@approvedBy,@programID,@UserID)


                UPDATE [dbo].[User]
                           SET [TotalPoints] += @Points
                           WHERE @UserID = [ID] AND @participationType = [ID]


END
```