

```

USE [MLM Db]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <MD.Alim Ul Karim , auk-port.webs.com>
-- Create date: <29-Jul-2012>
-- Description: <Condition A , see the Reward Class>
-- =====
ALTER PROCEDURE [dbo].[ConnectRewardMemberWithFailedUsersCode]
AS
BEGIN
    --SET NOCOUNT ON;

    DECLARE @RewardActsCount    bigint
    DECLARE @FailedActsCount    bigint
    DECLARE @CountTake          int
    DECLARE @IsTesting          bit

    EXEC @IsTesting              = dbo.SettingsValue 'IsTesting';
    EXEC @RewardActsCount        = dbo.Counter dbo,RewardPromote;
    EXEC @FailedActsCount        = dbo.Counter dbo,FailedBusiness;

    IF @FailedActsCount >= @RewardActsCount
        SET @CountTake = @RewardActsCount;
    ELSE
        SET @CountTake = @FailedActsCount;

    IF (@CountTake > 0 AND @RewardActsCount > 0 AND @FailedActsCount > 0) BEGIN
        DECLARE @FailedID          uniqueidentifier
        DECLARE @FailedMemberID    bigint
        DECLARE @RewardID          uniqueidentifier
        DECLARE @RewardMemberID    bigint
        DECLARE @ConnectionStatus  tinyint -- 1 BonusFromFailedUser
        DECLARE @AssignDate        smalldatetime
        DECLARE @it                bigint
        DECLARE @WorkingCount      int
        DECLARE @DeleteFromReward  int
        DECLARE @DeleteFromFailed  int

        SET @ConnectionStatus      = 1;-- 1 BonusFromFailedUser
        SET @AssignDate             = GETUTCDATE();
        SET @it                    = 0;
        SET @WorkingCount          = 0;
        SET @DeleteFromReward      = 0;
        SET @DeleteFromFailed      = 0;

        DECLARE c1 CURSOR local
        FOR
        SELECT FailedBusinessID,MemberID FROM FailedBusiness ORDER BY StoredDate
        OPEN c1

        FETCH NEXT FROM c1 INTO @FailedID,@FailedMemberID

        WHILE @@FETCH_STATUS = 0
        BEGIN
            SET @it = @it + 1;
            -- Getting every failed member
            -- Get RewardMemberID
            Select Top 1 @RewardID = r.RewardPromoteID, @RewardMemberID = r.MemberID from
            RewardPromote r Order By StoredDate;
            DECLARE @ExistReward bit

            --EXEC @ExistReward = RecExist dbo,RewardPromote;

            -- Create business code for this reward user under this failed user.
        
```

```
exec dbo.CreateBusinessCode @FailedMemberID,1,@RewardMemberID,@ConnectionStatus,@
AssignDate;

DELETE FROM RewardPromote WHERE RewardPromoteID = @RewardID;
SET @DeleteFromReward += @@ROWCOUNT;
DELETE FROM FailedBusiness WHERE FailedBusinessID = @FailedID;
SET @DeleteFromFailed += @@ROWCOUNT;

-- Breaking Condition and do not take more than the CountTake
if (@it >= @CountTake OR @DeleteFromReward >= @CountTake OR @DeleteFromFailed >= @
CountTake) Break;
--Cursor to next
FETCH NEXT FROM c1 INTO @FailedID,@FailedMemberID;
END
CLOSE c1
DEALLOCATE c1

if(@IsTesting = 1) Begin
    DECLARE @v Varchar(250)
    SET @v = 'Worked:' + CAST(@WorkingCount AS Varchar(100)) + ' CountTake:' + CAST(@
CountTake AS Varchar(100));
    SELECT @v as 'info';
    SELECT 'DELETE FROM Failed' = @DeleteFromFailed;
    SELECT 'DELETE FROM Reward' = @DeleteFromReward;
    SELECT 'Count Take' = @CountTake;
END
Return @DeleteFromReward;
END

END
```