IF OBJECT\_ID (N'[Ref\_Roles]', N'U') IS NOT NULL

DROP TABLE [Ref\_Roles]

CREATE TABLE [Ref\_Roles]

(

[RoleId] INT NOT NULL IDENTITY(1,1),

[Code] VARCHAR(5) NOT NULL,

[Description] VARCHAR(100) NOT NULL,

[CreatedDate] DATETIME DEFAULT GETDATE(),

[ModifiedDate] DATETIME NULL,

CONSTRAINT PK\_RefRoleId PRIMARY KEY ([RoleId]),

CONSTRAINT UC\_RefRoleCode UNIQUE ([Code])

)

INSERT [Ref\_Roles] ([Code], [Description]) VALUES ('DEV', 'Developer')

INSERT [Ref\_Roles] ([Code], [Description]) VALUES ('SA', 'Super Admin')

INSERT [Ref\_Roles] ([Code], [Description]) VALUES ('ADM', 'Admin')

INSERT [Ref\_Roles] ([Code], [Description]) VALUES ('USR', 'User')

INSERT [Ref\_Roles] ([Code], [Description]) VALUES ('BUS', 'Business')

INSERT [Ref\_Roles] ([Code], [Description]) VALUES ('COMM', 'Community')

SELECT \* FROM [Ref\_Roles]

----- END -----

IF OBJECT\_ID (N'[tbl\_Users]', N'U') IS NOT NULL

DROP TABLE [tbl\_Users]

CREATE TABLE [tbl\_Users]

(

[UserId] INT NOT NULL IDENTITY(1,1),

[PersonalKey] VARCHAR(800) NOT NULL,

[FirstName] VARCHAR(800) NOT NULL,

[LastName] VARCHAR(800) NOT NULL,

[RoleId] INT NOT NULL,

[Password] VARCHAR(800) NOT NULL,

[SaltKey] VARCHAR(800) NOT NULL,

[EmailId] VARCHAR(800) NOT NULL,

[EmailCheck] VARCHAR(800) NOT NULL,

[Mobile] VARCHAR(800) NOT NULL,

[MobileCheck] VARCHAR(800) NOT NULL,

[Photo] VARCHAR(MAX) NULL,

[ImgType] VARCHAR(800) NULL,

[Gender] VARCHAR(300) NULL,

[IsFirstTimeLogin] BIT DEFAULT 1,

[FailedLoginAttempts] INT DEFAULT 0,

[DisableAccountTime] DATETIME NULL,

[IsActive] BIT DEFAULT 1,

[CreatedBy] INT NOT NULL,

[CreatedDate] DATETIME DEFAULT GETDATE(),

[ModifiedBy] INT NULL,

[ModifiedDate] DATETIME NULL,

CONSTRAINT PK\_tblUserId PRIMARY KEY ([UserId]),

CONSTRAINT FK\_tblURole FOREIGN KEY ([CreatedBy]) REFERENCES [Ref\_Roles]([RoleId]),

CONSTRAINT FK\_tblUCreated FOREIGN KEY ([CreatedBy]) REFERENCES [tbl\_Users]([UserId]),

CONSTRAINT FK\_tblUModified FOREIGN KEY ([ModifiedBy]) REFERENCES [tbl\_Users]([UserId])

)

INSERT [tbl\_Users]

(

[PersonalKey],

[FirstName],

[LastName],

[RoleId],

[Password],

[SaltKey],

[EmailId],

[EmailCheck],

[Mobile],

[MobileCheck],

[IsFirstTimeLogin],

[IsActive],

[CreatedBy],

[CreatedDate]

)

VALUES

(

'2W6ppjh+6bi2veYXNrz3LyQVLFaj+r7/9CBL2horM8MhYF/kvm5kmSZ0t//tAStkHf0IWR8cPYE8kvPO7Qmc7w==',

'AhU+KNMJioCmoJw2oT+65g2UG+mIbJUYX2v0/BCMP6c=',

'bF3touMtIMA9H+OOVGFlNctvgi6FTNaZmF6GVQU8aw8=',

1,

'tF2Qh8Rbhhx1jRqRRtR0fOVaeYG+iVT8MeExfeKpn9w=',

'1X51Bf6FGv5ENt++VQRT/u9BTd5q9Wmvst9d/KZZ0RM=',

'wHJuzGDS7F+tj6L1bEgk9ndvWaf9T9Ub7DOTvUUWfll/4fEFoDpYyV1hzLBhM61o',

'Xk4Zw+C8PndeMJkBKVSVw5/gINK/7S+nubH4M5gQmeg=',

'h1TIGpbN6h6XheJsxVtJyw72ufPQnyxjHGc3WbgA3os=',

'wWAOX1vUTy3SOlULE+99Bw==',

1,

1,

1,

GETDATE()

)

---- Actual Values

--FirstName - Sampath

--LastName - Bandari

--Email - sampath.bandari51@gmail.com

--Mobile - 8121839354

--Password - W#9cqh;B

--Salt - Gou[{Is&

SELECT \* FROM [tbl\_Users]

----- END -----

IF OBJECT\_ID (N'[tbl\_Audits]', N'U') IS NOT NULL

DROP TABLE [tbl\_Audits]

CREATE TABLE [tbl\_Audits]

(

[UserId] INT NOT NULL,

[IPAddress] VARCHAR(100) NOT NULL,

[EntryType] VARCHAR(10) NOT NULL,

[EntryFrom] VARCHAR(10) NOT NULL,

[Browser] VARCHAR(600) NOT NULL,

[Resolution] VARCHAR(50) NOT NULL,

[Location] VARCHAR(500) NOT NULL,

[Latitude] VARCHAR(50) NOT NULL,

[Longitude] VARCHAR(50) NOT NULL,

[AccessTime] DATETIME NOT NULL,

[IsSuccess] BIT NOT NULL,

CONSTRAINT UC\_ALUserATime UNIQUE ([UserId], [AccessTime]),

CONSTRAINT FK\_ALUser FOREIGN KEY ([UserId]) REFERENCES [tbl\_Users]([UserId])

)

SELECT \* FROM [tbl\_Audits]

----- END -----

IF OBJECT\_ID (N'[tbl\_UserSecurityQuestions]', N'U') IS NOT NULL

DROP TABLE [tbl\_UserSecurityQuestions]

CREATE TABLE [tbl\_UserSecurityQuestions]

(

[UserId] INT NOT NULL,

[SecurityQue] VARCHAR(200) NOT NULL,

[SecurityAns] VARCHAR(200) NULL,

CONSTRAINT UC\_USQUserSecurity UNIQUE ([UserId], [SecurityQue]),

CONSTRAINT FK\_USQUser FOREIGN KEY ([UserId]) REFERENCES [tbl\_Users]([UserId])

)

SELECT \* FROM [tbl\_UserSecurityQuestions]

----- END -----

IF OBJECT\_ID (N'[Ref\_Countries]', N'U') IS NOT NULL

DROP TABLE [Ref\_Countries]

CREATE TABLE [Ref\_Countries]

(

[CountryId] INT NOT NULL IDENTITY(1,1),

[Code] VARCHAR(5) NOT NULL,

[Description] VARCHAR(100) NOT NULL,

[CreatedDate] DATETIME DEFAULT GETDATE(),

[ModifiedDate] DATETIME NULL,

CONSTRAINT PK\_CountryId PRIMARY KEY ([CountryId]),

CONSTRAINT UC\_CountryCodeDesc UNIQUE ([Code],[Description])

)

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('AUS', 'Austria')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('BEL', 'Belgium')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('BUL', 'Bulgaria')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('CRO', 'Croatia')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('CYP', 'Cyprus')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('CZE', 'Czechia')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('DEN', 'Denmark')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('EST', 'Estonia')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('FIN', 'Finland')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('FRA', 'France')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('GER', 'Germany')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('GRE', 'Greece')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('HUN', 'Hungary')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('IRE', 'Ireland')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('ITA', 'Italy')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('LAT', 'Latvia')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('LIT', 'Lithuania')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('LUX', 'Luxembourg')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('MAL', 'Malta')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('NET', 'Netherlands')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('POL', 'Poland')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('POR', 'Portugal')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('ROM', 'Romania')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('SLO', 'Slovakia')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('SLOV', 'Slovenia')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('SPA', 'Spain')

INSERT [Ref\_Countries] ([Code], [Description]) VALUES ('SWE', 'Sweden')

SELECT \* FROM [Ref\_Countries]

----- END -----

IF OBJECT\_ID (N'[Ref\_States]', N'U') IS NOT NULL

DROP TABLE [Ref\_States]

CREATE TABLE [Ref\_States]

(

[StateId] INT NOT NULL IDENTITY(1,1),

[Code] VARCHAR(5) NOT NULL,

[Description] VARCHAR(100) NOT NULL,

[CountryId] INT NOT NULL,

[CreatedDate] DATETIME DEFAULT GETDATE(),

[ModifiedDate] DATETIME NULL,

CONSTRAINT PK\_StateId PRIMARY KEY ([StateId]),

CONSTRAINT FK\_CState FOREIGN KEY ([CountryId]) REFERENCES [Ref\_Countries]([CountryId]),

CONSTRAINT UC\_StateCodeDesc UNIQUE ([Code],[Description])

)

INSERT [Ref\_States] ([Code], [Description], [CountryId]) VALUES ('BER', 'Berlin', 11)

INSERT [Ref\_States] ([Code], [Description], [CountryId]) VALUES ('HAM', 'Hamburg', 11)

INSERT [Ref\_States] ([Code], [Description], [CountryId]) VALUES ('SAA', 'Saarland', 11)

INSERT [Ref\_States] ([Code], [Description], [CountryId]) VALUES ('WAR', 'Warsaw', 21)

INSERT [Ref\_States] ([Code], [Description], [CountryId]) VALUES ('OPO', 'Opole', 21)

INSERT [Ref\_States] ([Code], [Description], [CountryId]) VALUES ('MAS', 'Masovian', 21)

INSERT [Ref\_States] ([Code], [Description], [CountryId]) VALUES ('BUD', 'Budapest', 13)

INSERT [Ref\_States] ([Code], [Description], [CountryId]) VALUES ('FEJ', 'Fejer', 13)

INSERT [Ref\_States] ([Code], [Description], [CountryId]) VALUES ('NOG', 'Nograd', 13)

SELECT \* FROM [Ref\_States]

----- END -----

IF OBJECT\_ID (N'[tbl\_UserMapping]', N'U') IS NOT NULL

DROP TABLE [tbl\_UserMapping]

CREATE TABLE [tbl\_UserMapping]

(

[UserId] INT NOT NULL,

[IsPermanentAddress] BIT NULL,

[PermanentStreetAddress] VARCHAR(MAX) NULL,

[PermanentCity] VARCHAR(800) NULL,

[PermanentState] INT NULL,

[PermanentCountry] INT NULL,

[PermanentZipCode] BIGINT NULL,

[CurrentStreetAddress] VARCHAR(MAX) NULL,

[CurrentCity] VARCHAR(800) NULL,

[CurrentState] INT NULL,

[CurrentCountry] INT NULL,

[CurrentZipCode] BIGINT NULL,

CONSTRAINT FK\_UMUser FOREIGN KEY ([UserId]) REFERENCES [tbl\_Users]([UserId]),

CONSTRAINT FK\_UMPState FOREIGN KEY ([PermanentState]) REFERENCES [Ref\_States]([StateId]),

CONSTRAINT FK\_UMPCountry FOREIGN KEY ([PermanentCountry]) REFERENCES [Ref\_Countries]([CountryId]),

CONSTRAINT FK\_UMCState FOREIGN KEY ([CurrentState]) REFERENCES [Ref\_States]([StateId]),

CONSTRAINT FK\_UMCCountry FOREIGN KEY ([CurrentCountry]) REFERENCES [Ref\_Countries]([CountryId])

)

SELECT \* FROM [tbl\_UserMapping]

----- END -----

IF OBJECT\_ID (N'[tbl\_ResetPwdLog]', N'U') IS NOT NULL

DROP TABLE [tbl\_ResetPwdLog]

CREATE TABLE [tbl\_ResetPwdLog]

(

[ResetPwdLogId] INT NOT NULL IDENTITY(1,1),

[UserId] INT NOT NULL,

[GuidVal] VARCHAR(800) NOT NULL,

[IsReset] BIT NOT NULL,

[EntryDate] DATETIME NOT NULL

CONSTRAINT PK\_ResetPwdLogId PRIMARY KEY ([ResetPwdLogId]),

CONSTRAINT FK\_RPwdUser FOREIGN KEY ([UserId]) REFERENCES [tbl\_Users]([UserId])

)

SELECT \* FROM [tbl\_ResetPwdLog]

----- END -----

IF OBJECT\_ID (N'[tbl\_Menus]', N'U') IS NOT NULL

DROP TABLE [tbl\_Menus]

CREATE TABLE [tbl\_Menus]

(

[ParentId] INT NOT NULL IDENTITY(1,1),

[Header] VARCHAR(50) NOT NULL,

[Controller] VARCHAR(50) NULL,

[Action] VARCHAR(50) NULL,

[HavingChild] BIT NOT NULL,

[Order] INT NOT NULL,

[Icon] VARCHAR(50) NOT NULL,

CONSTRAINT PK\_ParentId PRIMARY KEY ([ParentId]),

CONSTRAINT UC\_PHeaderControllerAction UNIQUE ([Header],[Controller],[Action])

)

INSERT INTO [tbl\_Menus] ([Header], [Controller], [Action], [HavingChild], [Order], [Icon]) VALUES ('Dashboard', 'CPanelDev', 'Dashboard', 0, 1, 'fa fa-bookmark-o')

INSERT INTO [tbl\_Menus] ([Header], [Controller], [Action], [HavingChild], [Order], [Icon]) VALUES ('All Users', NULL, NULL, 1, 2, 'fa fa-user-o')

INSERT INTO [tbl\_Menus] ([Header], [Controller], [Action], [HavingChild], [Order], [Icon]) VALUES ('Master Data', NULL, NULL, 1, 3, 'fa fa-star-o')

INSERT INTO [tbl\_Menus] ([Header], [Controller], [Action], [HavingChild], [Order], [Icon]) VALUES ('Error Log', 'CPanelDev', 'ViewErrorLog', 0, 4, 'fa fa-exclamation-triangle')

INSERT INTO [tbl\_Menus] ([Header], [Controller], [Action], [HavingChild], [Order], [Icon]) VALUES ('Dashboard', 'CPanelUser', 'Dashboard', 0, 1, 'fa fa-bookmark-o')

INSERT INTO [tbl\_Menus] ([Header], [Controller], [Action], [HavingChild], [Order], [Icon]) VALUES ('Accommodation', NULL, NULL, 0, 2, 'fa fa-user-o')

INSERT INTO [tbl\_Menus] ([Header], [Controller], [Action], [HavingChild], [Order], [Icon]) VALUES ('Buy / Sell', NULL, NULL, 0, 3, 'fa fa-star-o')

SELECT \* FROM [tbl\_Menus]

----- END -----

IF OBJECT\_ID (N'[tbl\_SubMenus]', N'U') IS NOT NULL

DROP TABLE [tbl\_SubMenus]

CREATE TABLE [tbl\_SubMenus]

(

[ChildId] INT NOT NULL IDENTITY(1,1),

[ParentId] INT NOT NULL,

[Header] VARCHAR(50) NOT NULL,

[Controller] VARCHAR(50) NOT NULL,

[Action] VARCHAR(50) NOT NULL,

[Order] INT NOT NULL,

CONSTRAINT PK\_ChildId PRIMARY KEY ([ChildId]),

CONSTRAINT FK\_SMenusParentId FOREIGN KEY ([ParentId]) REFERENCES [tbl\_Menus]([ParentId]),

CONSTRAINT UC\_CHeaderControllerAction UNIQUE ([Header],[Controller],[Action])

)

INSERT INTO [tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (2, 'Create', 'CPanelDev', 'CreateAllUsers', 1)

INSERT INTO [tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (2, 'Details', 'CPanelDev', 'AllUserDetails', 2)

INSERT INTO [tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (3, 'Roles', 'CPanelDev', 'CreateRole', 1)

INSERT INTO [tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (3, 'Menus', 'CPanelDev', 'CreateMenus', 2)

INSERT INTO [tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (3, 'Sub Menus', 'CPanelDev', 'CreateSubMenus', 3)

INSERT INTO [tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (3, 'Assigned Menus', 'CPanelDev', 'CreateAssignedMenus', 4)

INSERT INTO [tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (3, 'Countries', 'CPanelDev', 'CreateCountries', 5)

INSERT INTO [tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (3, 'States', 'CPanelDev', 'CreateStates', 6)

INSERT INTO [tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (6, 'Posts', 'CPanelUser', 'AccommodationPosts', 1)

INSERT INTO [tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (6, 'Requests', 'CPanelUser', 'AccommodationRequests', 2)

INSERT INTO [tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (7, 'Posts', 'CPanelUser', 'BuySellPosts', 1)

INSERT INTO [tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (7, 'Requests', 'CPanelUser', 'BuySellRequests', 2)

SELECT \* FROM [tbl\_SubMenus]

----- END -----

IF OBJECT\_ID (N'[tbl\_AssignMenus]', N'U') IS NOT NULL

DROP TABLE [tbl\_AssignMenus]

CREATE TABLE [tbl\_AssignMenus]

(

[AssignMenuId] INT NOT NULL IDENTITY(1,1),

[ParentId] INT NOT NULL,

[RoleId] INT NOT NULL,

CONSTRAINT PK\_AssignMenuId PRIMARY KEY ([AssignMenuId]),

CONSTRAINT FK\_AMenusParentId FOREIGN KEY ([ParentId]) REFERENCES [tbl\_Menus]([ParentId]),

CONSTRAINT FK\_AMenusRoleId FOREIGN KEY ([RoleId]) REFERENCES [Ref\_Roles]([RoleId]),

CONSTRAINT UC\_AMenusParentRole UNIQUE ([ParentId],[RoleId])

)

INSERT INTO [tbl\_AssignMenus] ([ParentId], [RoleId]) VALUES (1, 1)

INSERT INTO [tbl\_AssignMenus] ([ParentId], [RoleId]) VALUES (2, 1)

INSERT INTO [tbl\_AssignMenus] ([ParentId], [RoleId]) VALUES (3, 1)

INSERT INTO [tbl\_AssignMenus] ([ParentId], [RoleId]) VALUES (4, 1)

INSERT INTO [tbl\_AssignMenus] ([ParentId], [RoleId]) VALUES (5, 4)

INSERT INTO [tbl\_AssignMenus] ([ParentId], [RoleId]) VALUES (6, 4)

INSERT INTO [tbl\_AssignMenus] ([ParentId], [RoleId]) VALUES (7, 4)

SELECT \* FROM [tbl\_AssignMenus]

----- END -----

IF OBJECT\_ID (N'[tbl\_AccommodationPosts]', N'U') IS NOT NULL

DROP TABLE [tbl\_AccommodationPosts]

CREATE TABLE [tbl\_AccommodationPosts]

(

[AccommodationPostId] INT NOT NULL IDENTITY(1,1),

[UserId] INT NOT NULL,

[Title] VARCHAR(30) NOT NULL,

[RoomType] VARCHAR(50) NOT NULL,

[Address] VARCHAR(50) NOT NULL,

[City] VARCHAR(30) NOT NULL,

[StateId] INT NOT NULL,

[CountryId] INT NOT NULL,

[ZipCode] BIGINT NOT NULL,

[Description] VARCHAR(300) NOT NULL,

[CreatedDate] DATETIME NOT NULL

CONSTRAINT PK\_AccommodationPostId PRIMARY KEY ([AccommodationPostId]),

CONSTRAINT FK\_AccPUserId FOREIGN KEY ([UserId]) REFERENCES [tbl\_Users]([UserId]),

CONSTRAINT FK\_AccPStateId FOREIGN KEY ([StateId]) REFERENCES [Ref\_States]([StateId]),

CONSTRAINT FK\_AccPCountryId FOREIGN KEY ([CountryId]) REFERENCES [Ref\_Countries]([CountryId]),

CONSTRAINT UC\_AccPTitle UNIQUE ([Title],[UserId])

)

----- END -----

IF OBJECT\_ID (N'[tbl\_PagesForComments]', N'U') IS NOT NULL

DROP TABLE [tbl\_PagesForComments]

CREATE TABLE [tbl\_PagesForComments]

(

[PageId] INT NOT NULL IDENTITY(1,1),

[Code] VARCHAR(5) NOT NULL,

[Description] VARCHAR(100) NOT NULL,

[CreatedDate] DATETIME DEFAULT GETDATE(),

[ModifiedDate] DATETIME NULL,

CONSTRAINT PK\_CPageId PRIMARY KEY ([PageId]),

CONSTRAINT UC\_CPCode UNIQUE ([Code])

)

INSERT INTO [tbl\_PagesForComments] ([Code], [Description]) VALUES ('ACPOS', 'AccPosts')

----- END -----

IF OBJECT\_ID (N'[tbl\_Comments]', N'U') IS NOT NULL

DROP TABLE [tbl\_Comments]

CREATE TABLE [tbl\_Comments]

(

[CommentId] INT NOT NULL IDENTITY(1,1),

[UserId] INT NOT NULL,

[PageId] INT NOT NULL,

[PageColumnId] INT NOT NULL,

[Comments ] VARCHAR(300) NOT NULL,

[ParentComment] INT NOT NULL,

[CommentedDate] DATETIME DEFAULT GETDATE(),

CONSTRAINT PK\_CommentId PRIMARY KEY ([CommentId]),

CONSTRAINT FK\_CommUserId FOREIGN KEY ([UserId]) REFERENCES [tbl\_Users]([UserId]),

CONSTRAINT FK\_CommPageId FOREIGN KEY ([PageId]) REFERENCES [tbl\_PagesForComments]([PageId]),

CONSTRAINT UC\_CPCode UNIQUE ([UserId],[PageId],[PageColumnId])

)

INSERT INTO [tbl\_PagesForComments] ([Code], [Description]) VALUES ('ACPOS', 'AccPosts')

----- END -----

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_Insert\_Registration] Script Date: 1/10/2019 3:00:03 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Insert\_Registration]

(

@PersonalKey VARCHAR(800),

@FirstName VARCHAR(800),

@LastName VARCHAR(800),

@Email VARCHAR(800),

@EmailCheck VARCHAR(800),

@Mobile VARCHAR(800),

@MobileCheck VARCHAR(800),

@RoleCode VARCHAR(10),

@Password VARCHAR(800),

@SaltKey VARCHAR(800)

)

AS

BEGIN

SET NOCOUNT ON;

INSERT INTO [eurodb].[tbl\_Users] (PersonalKey, FirstName, LastName, EmailId, EmailCheck, Mobile, MobileCheck, RoleId, [Password], SaltKey, CreatedBy)

VALUES (@PersonalKey, @FirstName, @LastName, @Email, @EmailCheck, @Mobile, @MobileCheck, (SELECT RoleId FROM [eurodb].[Ref\_Roles] WHERE Code = @RoleCode), @Password, @SaltKey, 1)

END

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Get\_AuthenticatedUser]

(

@EmailCheck VARCHAR(800),

@Password VARCHAR(800)

)

AS

BEGIN

SET NOCOUNT ON;

DECLARE @Status VARCHAR(5),

@UserId INT = NULL,

@FirstName VARCHAR(800) = NULL,

@LastName VARCHAR(800) = NULL,

@FullName VARCHAR(800) = NULL,

@Email VARCHAR(800) = NULL,

@Mobile VARCHAR(800) = NULL,

@RoleID INT = NULL,

@RoleCode VARCHAR(50) = NULL,

@IsFirstTimeLogin BIT = NULL,

@ProfilePic VARCHAR(MAX) = NULL,

@FailedLoginAttempts INT = NULL,

@DisableAccountTime DATETIME = NULL,

@TimeDiff INT = NULL

IF (SELECT Count(\*) FROM [eurodb].[tbl\_Users] WHERE EmailCheck = @EmailCheck AND @Password = @Password) = 0

BEGIN

SET @Status = 0

SET @DisableAccountTime = (SELECT DisableAccountTime FROM [eurodb].[tbl\_Users] WHERE EmailCheck = @EmailCheck)

SET @FailedLoginAttempts = (SELECT FailedLoginAttempts FROM [eurodb].[tbl\_Users] WHERE EmailCheck = @EmailCheck)

END

ELSE

BEGIN

SELECT @Status = 1,

@UserId = U.UserId,

@FirstName = U.FirstName,

@LastName = U.LastName,

@FullName = (U.FirstName + ' ' + U.LastName),

@Email = U.EmailCheck,

@Mobile = U.MobileCheck,

@RoleID = U.RoleId,

@RoleCode = R.Code,

@ProfilePic = U.Photo,

@IsFirstTimeLogin = U.IsFirstTimeLogin,

@FailedLoginAttempts = U.FailedLoginAttempts,

@DisableAccountTime = U.DisableAccountTime

FROM

[eurodb].[tbl\_Users] U

INNER JOIN [eurodb].[Ref\_Roles] R ON U.RoleId = R.RoleId

WHERE U.EmailCheck = @EmailCheck AND U.[Password] = @Password

IF (SELECT Count(\*) FROM [eurodb].[tbl\_UserMapping] WHERE UserId = @UserId) = 0

BEGIN

INSERT INTO [eurodb].[tbl\_UserMapping] (UserId) VALUES (@UserId)

END

DECLARE @DisableTime DATETIME

SET @DisableTime = (SELECT DisableAccountTime FROM [eurodb].[tbl\_Users] WHERE UserId = (SELECT UserId FROM [eurodb].[tbl\_Users] WHERE EmailCheck = @EmailCheck))

IF(@DisableTime) IS NULL

BEGIN

UPDATE [eurodb].[tbl\_Users] SET FailedLoginAttempts = NULL, DisableAccountTime = NULL WHERE UserId= (SELECT UserId FROM [eurodb].[tbl\_Users] WHERE EmailCheck = @EmailCheck)

END

ELSE

SET @TimeDiff = (SELECT DATEDIFF (mi, @DisableTime, GETDATE()))

IF((SELECT DATEDIFF (mi, @DisableTime, GETDATE())) >= 5)

BEGIN

UPDATE [eurodb].[tbl\_Users] SET FailedLoginAttempts = NULL, DisableAccountTime = NULL WHERE UserId= (SELECT UserId FROM [eurodb].[tbl\_Users] WHERE EmailCheck = @EmailCheck)

END

END

SELECT @Status AS [Status],

@UserId AS UserId,

@FirstName AS FirstName,

@LastName AS LastName,

@FullName AS FullName,

@Email AS EmailCheck,

@Mobile AS MobileCheck,

@RoleID AS RoleID,

@RoleCode AS RoleCode,

@ProfilePic AS Photo,

@IsFirstTimeLogin AS IsFirstTimeLogin,

@FailedLoginAttempts AS FailedLoginAttempts,

@DisableAccountTime AS DisableAccountTime,

@TimeDiff AS TimeDiff

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_Update\_FailedLoginAttempts] Script Date: 1/10/2019 3:00:04 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Insert\_Audit]

(

@UserId INT,

@IPAddr VARCHAR(100),

@EntryType VARCHAR(10),

@EntryFrom VARCHAR(10),

@Browser VARCHAR(600),

@Resolution VARCHAR(50),

@Location VARCHAR(500),

@Latitude VARCHAR(50),

@Longitude VARCHAR(50),

@IsSuccess BIT

)

AS

BEGIN

SET NOCOUNT ON;

INSERT INTO [eurodb].[tbl\_Audits] (UserId, IPAddress, EntryType, EntryFrom, Browser, Resolution, [Location], Latitude, Longitude, AccessTime, IsSuccess)

VALUES (@UserId, @IPAddr, @EntryType, @EntryFrom, @Browser, @Resolution, @Location, @Latitude, @Longitude, GETDATE(), @IsSuccess)

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_Update\_FailedLoginAttempts] Script Date: 1/10/2019 3:00:04 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Update\_FailedLoginAttempts]

(

@UserId INT

)

AS

BEGIN

SET NOCOUNT ON;

IF (SELECT FailedLoginAttempts FROM [eurodb].[tbl\_Users] WHERE UserId = @UserId) IS NOT NULL

BEGIN

DECLARE @CountNum INT

SET @CountNum = (SELECT FailedLoginAttempts FROM [eurodb].[tbl\_Users] WHERE UserId = @UserId) + 1

IF(@CountNum > 4)

BEGIN

UPDATE [eurodb].[tbl\_Users] SET FailedLoginAttempts = @CountNum, DisableAccountTime = GETDATE() WHERE UserId = @UserId

END

ELSE

BEGIN

UPDATE [eurodb].[tbl\_Users] SET FailedLoginAttempts = @CountNum WHERE UserId = @UserId

END

END

ELSE

BEGIN

UPDATE [eurodb].[tbl\_Users] SET FailedLoginAttempts = 1 WHERE UserId = @UserId

END

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_Insert\_Registration] Script Date: 1/10/2019 3:00:03 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Insert\_ResetPwdLog]

(

@UserId INT,

@GuidVal VARCHAR(800)

)

AS

BEGIN

SET NOCOUNT ON;

INSERT INTO [eurodb].[tbl\_ResetPwdLog] (UserId, GuidVal, IsReset, EntryDate)

VALUES (@UserId, @GuidVal, 0, GETDATE())

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_Update\_FailedLoginAttempts] Script Date: 1/10/2019 3:00:04 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Update\_ResetPwd]

(

@UserId INT,

@Password VARCHAR(800),

@SaltKey VARCHAR(800),

@GuidVal VARCHAR(800),

@ExpiryTime INT,

@OutputResult INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

DECLARE @CurrentTime DATETIME = GETDATE()

DECLARE @DiffTime INT = (SELECT DATEDIFF(MINUTE,(SELECT EntryDate FROM [eurodb].[tbl\_ResetPwdLog] WHERE GuidVal = @GuidVal), @CurrentTime))

IF(@DiffTime > @ExpiryTime)

BEGIN

SET @OutputResult = 99 --Expierd

END

ELSE

BEGIN

IF((SELECT IsReset FROM [eurodb].[tbl\_ResetPwdLog] WHERE GuidVal = @GuidVal) > 0)

BEGIN

SET @OutputResult = 100 --Already Update

END

ELSE

BEGIN

BEGIN TRANSACTION;

SAVE TRANSACTION MySavePoint;

BEGIN TRY

UPDATE [eurodb].[tbl\_ResetPwdLog] SET IsReset = 1 WHERE GuidVal = @GuidVal

UPDATE [eurodb].[tbl\_Users] SET [Password] = @Password , [SaltKey] = @SaltKey WHERE UserId = @UserId

SET @OutputResult = 1

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

BEGIN

SET @OutputResult = 0

ROLLBACK TRANSACTION MySavePoint;

END

END CATCH

COMMIT TRANSACTION

END

END

END

-- EXEC [USP\_Update\_ResetPassword] 2,'asd','dsa','Ojypmq83TwS8IJXRzJ/nXC9V2gdu7HFcwYVdpwIpQ+tgkvIcNXeBwGjay1wgDsp1f1/jd6LI/AFKjWSCB70jNw==',900, 0

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_Get\_MenuListByRole] Script Date: 1/17/2019 3:28:52 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Get\_MenuListByRole]

(

@RoleCode VARCHAR(10)

)

AS

BEGIN

SET NOCOUNT ON;

DECLARE @RoleId INT

SET @RoleId= (SELECT RoleId FROM [eurodb].[Ref\_Roles] WHERE Code = @RoleCode)

SELECT

AM.ParentId,

M.Header,

M.Controller,

M.[Action],

M.HavingChild,

M.[Order],

M.Icon

FROM [eurodb].[tbl\_AssignMenus] AS AM

INNER JOIN [eurodb].[tbl\_Menus] M ON M.ParentId= AM.ParentId

WHERE AM.RoleId = @RoleId

SELECT

AM.ParentId,

SM.Header,

SM.Controller,

SM.[Action],

SM.[Order]

FROM [eurodb].[tbl\_AssignMenus] AS AM

INNER JOIN [eurodb].[tbl\_SubMenus] SM ON SM.ParentId= AM.ParentId

WHERE AM.RoleId = @RoleId

END

-- EXEC USP\_Get\_MenuListByRole 132,4

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_Update\_OneTimeLogin] Script Date: 1/17/2019 3:28:52 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Update\_OneTimeLogin]

(

@UserId INT ,

@OldPwd VARCHAR(800),

@OldKey VARCHAR(800),

@NewPwd VARCHAR(800),

@NewKey VARCHAR(800),

@OutPut INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

IF((SELECT COUNT(\*) FROM [eurodb].[tbl\_Users] WHERE UserId = @UserId) > 0)

BEGIN

IF((SELECT COUNT(\*) FROM [eurodb].[tbl\_Users] WHERE [Password] = @OldPwd AND SaltKey = @OldKey) > 0)

BEGIN

UPDATE [eurodb].[tbl\_Users] SET [Password] = @NewPwd , SaltKey = @NewKey, IsFirstTimeLogin = 0 WHERE UserId = @UserId

SET @OutPut = 1

END

ELSE

SET @OutPut = 99 -- Old Password Incorrect

END

ELSE

SET @OutPut = 100 -- No User

END

-- EXEC [USP\_Update\_OneTimeLogin] 132,4

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_InsertUpdate\_Roles] Script Date: 1/17/2019 3:28:52 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_InsertUpdate\_Roles]

(

@Code VARCHAR(5),

@Desc VARCHAR(100),

@btnType VARCHAR(15),

@OutPut INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

IF(@btnType = 'Save')

BEGIN

IF((SELECT COUNT(\*) FROM [eurodb].[Ref\_Roles] WHERE Code= @Code) = 0)

BEGIN

INSERT INTO [eurodb].[Ref\_Roles] (Code, [Description], CreatedDate) VALUES (@Code, @Desc, GETDATE())

SET @OutPut = 1

END

ELSE

SET @OutPut = 99 -- Duplicate Code

END

ELSE IF(@btnType = 'Update')

BEGIN

UPDATE [eurodb].[Ref\_Roles] SET [Description] = @Desc, ModifiedDate = GETDATE() WHERE Code = @Code

SET @OutPut = 1

END

ELSE

SET @OutPut = 100 -- Button Text

END

-- EXEC [USP\_InsertUpdate\_Roles] 132,4

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_Delete\_Roles] Script Date: 1/17/2019 3:28:52 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Delete\_Roles]

(

@Code VARCHAR(5),

@OutPut INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

IF((SELECT COUNT(\*) FROM [eurodb].[Ref\_Roles] WHERE Code= @Code) > 0)

BEGIN

DELETE FROM [eurodb].[Ref\_Roles] WHERE Code = @Code

SET @OutPut = 1

END

ELSE

SET @OutPut = 99 -- Duplicate Code

END

-- EXEC [USP\_Delete\_Roles] 132,4

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_InsertUpdate\_Roles] Script Date: 1/17/2019 3:28:52 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_InsertUpdate\_MyProfile]

(

@ActualEmail VARCHAR(800),

@FirstName VARCHAR(800),

@LastName VARCHAR(800),

@Gender VARCHAR(500) = NULL,

@PAddress VARCHAR(800) = NULL,

@PCity VARCHAR(800) = NULL,

@PStateId INT = NULL,

@PCountryId INT = NULL,

@PZipCode BIGINT = NULL,

@isCurrentPermanent BIT = NULL,

@CAddress VARCHAR(800) = NULL,

@CCity VARCHAR(800) = NULL,

@CStateId INT = NULL,

@CCountryId INT = NULL,

@CZipCode BIGINT = NULL,

@ProfilePic VARCHAR(MAX) = NULL,

@ImgType VARCHAR(800) = NULL,

@OutputResult INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRANSACTION;

SAVE TRANSACTION MySavePoint;

BEGIN TRY

DECLARE @UserId INT

SET @UserId= (SELECT UserId FROM [eurodb].[tbl\_Users] WHERE EmailCheck = @ActualEmail)

IF(@isCurrentPermanent = 1)

BEGIN

UPDATE [eurodb].[tbl\_Users] SET [FirstName] = @FirstName, [LastName] = @LastName, [Gender] = @Gender, [Photo] = @ProfilePic, ImgType = @ImgType, [ModifiedDate] = GETDATE() WHERE UserId = @UserId

UPDATE [eurodb].[tbl\_UserMapping]

SET [IsPermanentAddress] = @isCurrentPermanent,

[PermanentStreetAddress] = @PAddress,

[PermanentCity] = @PCity,

[PermanentState] = @PStateId,

[PermanentCountry] = @PCountryId,

[PermanentZipCode] = @PZipCode,

[CurrentStreetAddress] = NULL,

[CurrentCity] = NULL,

[CurrentState] = NULL,

[CurrentCountry] = NULL,

[CurrentZipCode] = NULL

WHERE UserId = @UserId

SET @OutputResult = 1

END

ELSE

BEGIN

UPDATE [eurodb].[tbl\_Users] SET [FirstName] = @FirstName, [LastName] = @LastName, [Gender] = @Gender, [Photo] = @ProfilePic, ImgType = @ImgType, [ModifiedDate] = GETDATE() WHERE UserId = @UserId

UPDATE [eurodb].[tbl\_UserMapping]

SET [IsPermanentAddress] = @isCurrentPermanent,

[PermanentStreetAddress] = @PAddress,

[PermanentCity] = @PCity,

[PermanentState] = @PStateId,

[PermanentCountry] = @PCountryId,

[PermanentZipCode] = @PZipCode,

[CurrentStreetAddress] = @CAddress,

[CurrentCity] = @CCity,

[CurrentState] = @CStateId,

[CurrentCountry] = @CCountryId,

[CurrentZipCode] = @CZipCode

WHERE UserId = @UserId

SET @OutputResult = 1

END

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0

BEGIN

SET @OutputResult = 0

ROLLBACK TRANSACTION MySavePoint;

END

END CATCH

COMMIT TRANSACTION

END

--EXEC USP\_InsertUpdate\_MyProfile

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_InsertUpdate\_Roles] Script Date: 1/17/2019 3:28:52 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_InsertUpdate\_Menus]

(

@Type VARCHAR(15),

@ParentId INT = NULL,

@Header VARCHAR(50),

@Controller VARCHAR(50) = NULL,

@Action VARCHAR(50) = NULL,

@HavingChild BIT,

@Order INT,

@Icon VARCHAR(50),

@OutPut INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

IF(@Type = 'Insert')

BEGIN

INSERT INTO [eurodb].[tbl\_Menus] ([Header], [Controller], [Action], [HavingChild], [Order], [Icon]) VALUES (@Header, @Controller, @Action, @HavingChild, @Order, @Icon)

SET @OutPut = 1

END

ELSE IF(@Type = 'Update')

BEGIN

UPDATE [eurodb].[tbl\_Menus] SET [Header] = @Header, [Controller] = @Controller, [Action] = @Action, [HavingChild] = @HavingChild, [Order] = @Order, [Icon] = @Icon WHERE ParentId = @ParentId

SET @OutPut = 1

END

ELSE

SET @OutPut = 0

END

-- EXEC [USP\_InsertUpdate\_Menus] 132,4

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[USP\_InsertUpdate\_Roles] Script Date: 1/17/2019 3:28:52 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Delete\_Menus]

(

@ParentId INT,

@OutPut INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

IF((SELECT COUNT(\*) FROM [eurodb].[tbl\_Menus] WHERE ParentId= @ParentId) > 0)

BEGIN

DELETE FROM [eurodb].[tbl\_Menus] WHERE ParentId = @ParentId

SET @OutPut = 1

END

ELSE

SET @OutPut = 99 -- Not Id

END

-- EXEC [USP\_Delete\_Menus] 132,4

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_InsertUpdate\_SubMenus]

(

@Type VARCHAR(15),

@ChildId INT = NULL,

@ParentId INT,

@Header VARCHAR(50),

@Controller VARCHAR(50) = NULL,

@Action VARCHAR(50) = NULL,

@Order INT,

@OutPut INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

IF(@Type = 'Insert')

BEGIN

INSERT INTO [eurodb].[tbl\_SubMenus] ([ParentId], [Header], [Controller], [Action], [Order]) VALUES (@ParentId, @Header, @Controller, @Action, @Order)

SET @OutPut = 1

END

ELSE IF(@Type = 'Update')

BEGIN

UPDATE [eurodb].[tbl\_SubMenus] SET [ParentId] = @ParentId, [Header] = @Header, [Controller] = @Controller, [Action] = @Action, [Order] = @Order WHERE ChildId = @ChildId

SET @OutPut = 1

END

ELSE

SET @OutPut = 0

END

-- EXEC [[USP\_InsertUpdate\_SubMenus] 132,4

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Delete\_SubMenus]

(

@ChildId INT,

@OutPut INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

IF((SELECT COUNT(\*) FROM [eurodb].[tbl\_SubMenus] WHERE ChildId= @ChildId) > 0)

BEGIN

DELETE FROM [eurodb].[tbl\_SubMenus] WHERE ChildId = @ChildId

SET @OutPut = 1

END

ELSE

SET @OutPut = 99 -- Not Id

END

-- EXEC [USP\_Delete\_SubMenus] 132,4

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_InsertUpdate\_AssignedMenus]

(

@Type VARCHAR(15),

@AssignedMenuId INT = NULL,

@ParentId INT,

@RoleId INT,

@OutPut INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

IF(@Type = 'Insert')

BEGIN

INSERT INTO [eurodb].[tbl\_AssignMenus] ([ParentId], [RoleId]) VALUES (@ParentId, @RoleId)

SET @OutPut = 1

END

ELSE IF(@Type = 'Update')

BEGIN

UPDATE [eurodb].[tbl\_AssignMenus] SET [ParentId] = @ParentId, [RoleId] = @RoleId WHERE AssignMenuId= @AssignedMenuId

SET @OutPut = 1

END

ELSE

SET @OutPut = 0

END

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Delete\_AssignedMenus]

(

@AssignMenuId INT,

@OutPut INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

IF((SELECT COUNT(\*) FROM [eurodb].[tbl\_AssignMenus] WHERE AssignMenuId = @AssignMenuId) > 0)

BEGIN

DELETE FROM [eurodb].[tbl\_AssignMenus] WHERE AssignMenuId = @AssignMenuId

SET @OutPut = 1

END

ELSE

SET @OutPut = 99 -- Not Id

END

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Get\_UserMappingDetails]

(

@EmailCheck VARCHAR(800)

)

AS

BEGIN

SET NOCOUNT ON;

DECLARE @UserId INT

IF (SELECT Count(\*) FROM [eurodb].[tbl\_Users] WHERE EmailCheck = @EmailCheck) > 0

BEGIN

SET @UserId= (SELECT UserId FROM [eurodb].[tbl\_Users] WHERE EmailCheck = @EmailCheck)

SELECT U.UserId,

U.Gender,

U.Photo,

U.ImgType,

UM.IsPermanentAddress,

UM.PermanentStreetAddress [PAddress],

UM.PermanentCity [PCity],

UM.PermanentState [PStateId],

UM.PermanentCountry [PCountryId],

UM.PermanentZipCode [PZipCode],

UM.CurrentStreetAddress [CAddress],

UM.CurrentCity [CCity],

UM.CurrentState [CStateId],

UM.CurrentCountry [CCountryId],

UM.CurrentZipCode [CZipCode]

FROM

[eurodb].[tbl\_Users] U

INNER JOIN [eurodb].[tbl\_UserMapping] UM ON U.UserId = UM.UserId

WHERE U.UserId = @UserId

END

END

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_InsertUpdate\_Countries]

(

@Type VARCHAR(15),

@Id INT = NULL,

@Code VARCHAR(50) = NULL,

@Desc VARCHAR(50) = NULL,

@OutPut INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

IF(@Type = 'Insert')

BEGIN

INSERT INTO [eurodb].[Ref\_Countries] ([Code], [Description]) VALUES (@Code, @Desc)

SET @OutPut = 1

END

ELSE IF(@Type = 'Update')

BEGIN

UPDATE [eurodb].[Ref\_Countries] SET [Description] = @Desc WHERE CountryId = @Id

SET @OutPut = 1

END

ELSE

SET @OutPut = 0

END

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Delete\_Countries]

(

@Id INT,

@OutPut INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

IF((SELECT COUNT(\*) FROM [eurodb].[Ref\_Countries] WHERE CountryId= @Id) > 0)

BEGIN

DELETE FROM [eurodb].[Ref\_Countries] WHERE CountryId = @Id

SET @OutPut = 1

END

ELSE

SET @OutPut = 99 -- Not Id

END

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[USP\_Insert\_AccPosts]

(

@UserId INT,

@Title VARCHAR(30),

@RoomType VARCHAR(50),

@Address VARCHAR(50),

@City VARCHAR(30),

@StateId INT,

@CountryId INT,

@Description VARCHAR(300),

@ZipCode BIGINT,

@OutPut INT = 0 OUT

)

AS

BEGIN

SET NOCOUNT ON;

INSERT INTO [eurodb].[tbl\_AccommodationPosts] ([UserId], [Title], [RoomType], [Address], [City], [StateId], [CountryId], [ZipCode], [Description], [CreatedDate]) VALUES (@UserId, @Title, @RoomType, @Address, @City, @StateId, @CountryId, @ZipCode, @Description, GETDATE())

SET @OutPut = 1

END

-- EXEC [USP\_Insert\_AccPosts] 132,4