-- ================================================

-- Template generated from Template Explorer using:

-- Create Procedure (New Menu).SQL

--

-- Use the Specify Values for Template Parameters

-- command (Ctrl-Shift-M) to fill in the parameter

-- values below.

--

-- This block of comments will not be included in

-- the definition of the procedure.

-- ================================================

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Insync Tech Solutions

-- Create date: 2017-12-01

-- Description: Create new procedures/tests

-- =============================================

CREATE PROCEDURE [dbo].[uspProcedures\_Add] @name Varchar(max), @desc text, @type int, @fee decimal(10,2), @active bit

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

-- Insert statements for procedure here

Declare @Ret int;

Set @Ret = -1;

If Exists (Select 1 From dbo.Procedures Where name = @name)

Begin

Set @Ret = 0;

return;

End

Else

Begin

Insert Into dbo.Procedures(name,description,type,fee, active)

Values (@name,@desc,@type,@fee, @active);

set @Ret = 1;

End

select @Ret

END

GO

-- ================================================

-- Template generated from Template Explorer using:

-- Create Procedure (New Menu).SQL

--

-- Use the Specify Values for Template Parameters

-- command (Ctrl-Shift-M) to fill in the parameter

-- values below.

--

-- This block of comments will not be included in

-- the definition of the procedure.

-- ================================================

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Insync Tech Solutions

-- Create date: 2017-12-01

-- Description: Create new procedures/tests

-- =============================================

CREATE PROCEDURE [dbo].[uspProcedures\_Edit] @id int, @name Varchar(max), @desc text, @type int, @fee decimal(10,2), @active bit

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

DECLARE @Ret int;

SET @Ret = -1;

IF Exists (SELECT 1 FROM dbo.[Procedures] WHERE [name] = @name and id <> @id)

BEGIN

SET @Ret = 0;

END

ELSE

BEGIN

UPDATE dbo.[Procedures] SET [name] = @name, [description]=@desc, type=@type, fee=@fee, active=@active WHERE [id]=@id

SET @Ret = 1;

END

SELECT @Ret

END

GO

-- ================================================

-- Template generated from Template Explorer using:

-- Create Procedure (New Menu).SQL

--

-- Use the Specify Values for Template Parameters

-- command (Ctrl-Shift-M) to fill in the parameter

-- values below.

--

-- This block of comments will not be included in

-- the definition of the procedure.

-- ================================================

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Insync Tech Solutions

-- Create date: 2017-12-01

-- Description: Create new procedures/tests

-- =============================================

CREATE PROCEDURE [dbo].[uspProcedures\_Get] @SearchBy varchar(50), @SearchValue varchar(100)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

DECLARE @sql nvarchar(max);

SET @sql = 'SELECT p.id, p.name, p.description, p.type, t.type\_name, p.fee, p.active from Procedures p JOIN Procedure\_Types t ON t.id=p.type ';

IF(@SearchBy <> 'All')

BEGIN

SET @sql = @sql + ' WHERE ' + @SearchBy + ' like ''' + @SearchValue + '%''';

END

EXEC(@sql);

END

GO

-- ================================================

-- Template generated from Template Explorer using:

-- Create Procedure (New Menu).SQL

--

-- Use the Specify Values for Template Parameters

-- command (Ctrl-Shift-M) to fill in the parameter

-- values below.

--

-- This block of comments will not be included in

-- the definition of the procedure.

-- ================================================

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Insync Tech Solutions

-- Create date: 2017-12-01

-- Description: Datatable for Procedure Types Combobox

-- =============================================

CREATE PROCEDURE [dbo].[uspProcedureTypes\_Combo] @id int

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

DECLARE @sql nvarchar(max);

SET @sql = 'SELECT id, type\_name from Procedure\_Types WHERE Active = 1';

IF(@id > 0)

BEGIN

SET @sql = @sql + ' OR id=''' + CAST(@id AS varchar(30)) + '''';

END

SET @sql = @sql + ' ORDER BY Active, type\_name';

EXEC(@sql);

END

GO

USE [hospitalERP\_db]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[uspProcedures\_Single] Script Date: 12/1/2017 9:37:41 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Insync Tech Solutions

-- Create date: 2017-12-01

-- Description: Datatable for Procedure Types Combobox

-- =============================================

CREATE PROCEDURE [dbo].[uspProcedures\_Single] @id int

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

SELECT p.\*, t.type\_name FROM dbo.[PROCEDURES] p JOIN dbo.Procedure\_Types t ON t.id=p.type WHERE p.id=@id

END

USE [hospitalERP\_db]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[uspOptions\_Get] Script Date: 12/2/2017 11:30:52 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE procedure [dbo].[uspOptions\_Get] @SearchBy varchar(50), @SearchValue varchar(100) AS

BEGIN

DECLARE @sql nvarchar(max);

SET @sql = 'SELECT id, op\_name, op\_description, op\_value FROM Options';

IF(@SearchBy <> 'All')

BEGIN

SET @sql = @sql + ' WHERE ' + @SearchBy + ' LIKE ''' + @SearchValue + '%''';

END

EXEC(@sql);

END

USE [hospitalERP\_db]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[uspProcedureTypes\_Add] Script Date: 12/2/2017 11:17:36 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[uspOptions\_Add] @name Varchar(max), @description text, @val Varchar(max) AS

DECLARE @Ret int;

SET @Ret = -1;

IF Exists (SELECT 1 From dbo.[Options] WHERE op\_name = @name)

BEGIN

SET @Ret = 0;

return;

END

ELSE

BEGIN

INSERT INTO dbo.[Options] (op\_name, op\_description, op\_value)

Values (@name, @description, @val);

SET @Ret = 1;

End

SELECT @Ret;

USE [hospitalERP\_db]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[uspProcedures\_Edit] Script Date: 12/2/2017 11:26:50 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: Insync Tech Solutions

-- Create date: 2017-12-01

-- Description: Modify options

-- =============================================

CREATE PROCEDURE [dbo].[uspOptions\_Edit] @id int, @name Varchar(max), @desc text, @val Varchar(max)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

SET NOCOUNT ON;

DECLARE @Ret int;

SET @Ret = -1;

IF Exists (SELECT 1 FROM dbo.[Options] WHERE [op\_name] = @name and id <> @id)

BEGIN

SET @Ret = 0;

END

ELSE

BEGIN

UPDATE dbo.[Options] SET op\_name = @name, op\_description=@desc, op\_value=@val WHERE [id]=@id

SET @Ret = 1;

END

SELECT @Ret

END

USE [hospitalERP\_db]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[uspOptions\_Single]

Script Date: 12/2/2017 11:33:02 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[uspOptions\_Single] @name varchar(100) AS

BEGIN

SELECT \* FROM dbo.[Options] WHERE op\_name=@name

END

USE [hospitalERP\_db]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[uspOptions\_Single]

Script Date: 12/2/2017 11:33:02 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[uspOptions\_Many] @name varchar(100) AS

BEGIN

SELECT op\_name, op\_value FROM dbo.[Options] WHERE op\_name IN (@name)

END