USE [hospitalERP\_db]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[svf\_FormatAddress] Script Date: 1/31/2018 11:46:16 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE FUNCTION [dbo].[svf\_FormatAddress]

(

@address varchar(250) = NULL,

@city varchar(200) = NULL,

@state varchar(200) = NULL,

@zip varchar(50) = NULL

)

RETURNS varchar(MAX)

AS

BEGIN

DECLARE @FAddress varchar(MAX) ;

SET @Address= '';

IF NOT(@address is null OR @address = '')

BEGIN

SET @address = @address+', '+ char(10) + char(13);

END

IF NOT(@city is null OR @city = '')

BEGIN

SET @city = @city+', ';

END

IF NOT(@zip is null OR @zip = '')

BEGIN

SET @zip = ' - ' + @zip;

END

SET @FAddress = @address + @city + @state + @zip;

IF RIGHT(RTRIM(@FAddress),1) = ','

BEGIN

SET @FAddress = LEFT(@FAddress, LEN(RTRIM(@FAddress)) - 1)

END

IF LEFT(LTRIM(@FAddress),1) = '-'

BEGIN

SET @FAddress = LTRIM(SUBSTRING(@FAddress, 1, LEN(LTRIM(@FAddress)) - 1))

END

RETURN @FAddress;

END

GO

USE [hospitalERP\_db]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[svf\_FormatDrName] Script Date: 1/31/2018 11:47:05 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE FUNCTION [dbo].[svf\_FormatDrName]

(

@fname varchar(100),

@lname varchar(100)

)

RETURNS varchar(250)

AS

BEGIN

DECLARE @Name varchar(50)

SET @Name= CONCAT('Dr. ',@fname, ' ', @lname)

RETURN @Name

END

USE [hospitalERP\_db]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[svf\_FormatName] Script Date: 1/31/2018 11:47:22 AM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE FUNCTION [dbo].[svf\_FormatName]

(

@fname varchar(100),

@lname varchar(100)

)

RETURNS varchar(250)

AS

BEGIN

DECLARE @Name varchar(50)

SET @Name= CONCAT(@fname, ' ', @lname)

RETURN @Name

END

GO

CREATE FUNCTION [dbo].[svf\_GetAge]

(

@dob date

)

RETURNS int

AS

BEGIN

DECLARE @age int

SET @age= DATEDIFF(YEAR,@dob,GETDATE())

RETURN @age

END

ALTER PROCEDURE [dbo].[uspAppointments\_GetForBilling] @aid int AS

BEGIN

SELECT a.id, a.patient\_id, a.doctor\_id, a.meet\_date,p.patient\_number, dbo.svf\_FormatName(p.first\_name, p.last\_name) as patient\_name, dbo.svf\_FormatDrName(d.first\_name,d.last\_name) as doctor\_name, a.token, ISNULL(b.bill\_balance,0) as dues, s.name as status, a.notes, p.address, p.city, p.state, p.zip, p.dob, dbo.svf\_GetAge(p.dob) as age

FROM Appointments a

JOIN Patients p ON a.patient\_id=p.id

JOIN Doctors d ON d.id=a.doctor\_id

JOIN Appointment\_Status s on a.status=s.id

LEFT JOIN (SELECT patient\_id, bill\_balance FROM Billing WHERE id IN (SELECT MAX(id) FROM Appointments WHERE bill\_status=2 OR bill\_status=3 AND bill\_type=2 GROUP BY patient\_id)) as b ON b.patient\_id=a.patient\_id

WHERE a.id=@aid ORDER BY id,patient\_number;

END

ALTER PROCEDURE [dbo].[uspAppointments\_GetForDateDoc] @meet\_date datetime, @doctor\_id int, @status smallint AS

BEGIN

DECLARE @sql VARCHAR(max);

SET @sql = 'SELECT a.id, a.patient\_id, a.doctor\_id, p.patient\_number, dbo.svf\_FormatName(p.first\_name, p.last\_name) as patient\_name, dbo.svf\_FormatDrName(d.first\_name,d.last\_name) as doctor\_name, a.token, am.prev\_date,ISNULL(b.bill\_balance,0) as dues, s.name as status

FROM Appointments a

JOIN Patients p ON a.patient\_id=p.id

JOIN Doctors d ON d.id=a.doctor\_id

JOIN Appointment\_Status s on a.status=s.id

LEFT JOIN (SELECT patient\_id, MAX(meet\_date) as prev\_date FROM Appointments WHERE id NOT IN (SELECT MAX(id)FROM Appointments GROUP BY patient\_id) GROUP BY patient\_id) as am ON am.patient\_id=a.patient\_id

LEFT JOIN (SELECT patient\_id, bill\_balance FROM Billing WHERE id IN (SELECT MAX(id) FROM Appointments WHERE bill\_status=2 OR bill\_status=3 AND bill\_type=2 GROUP BY patient\_id)) as b ON b.patient\_id=a.patient\_id

WHERE a.meet\_date='''+ CAST(@meet\_date AS varchar(20))+'''';

IF(@doctor\_id >0)

BEGIN

SET @sql = @sql + ' AND doctor\_id =''' + CAST(@doctor\_id AS varchar(20)) + '''';

END

IF(@status >0)

BEGIN

SET @sql = @sql + ' AND status =''' + CAST(@status AS varchar(20)) + ''' ';

END

SET @sql = @sql + ' ORDER BY id,patient\_number';

EXEC(@sql);

END

ALTER PROCEDURE [dbo].[uspConsultationDet\_Get\_History] @id int , @patient\_id int

AS

BEGIN

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

-- interfering with SELECT statements.

SET NOCOUNT ON;

SELECT a.id,a.meet\_date, dbo.svf\_FormatDrName(d.first\_name,d.last\_name) as doctor\_name,a.notes , aps.name as status\_name

FROM Appointments a JOIN Doctors d ON d.id=a.doctor\_id

LEFT JOIN Appointment\_Status aps ON a.status=aps.id WHERE /\*a.id!=@id AND \*/a.patient\_id=@patient\_id ORDER BY meet\_date DESC;

END

ALTER PROCEDURE [dbo].[uspDoctors\_Combo] @id int

AS

BEGIN

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

-- interfering with SELECT statements.

SET NOCOUNT ON;

DECLARE @sql varchar(max);

SET @sql = 'SELECT id , dbo.svf\_FormatDrName(first\_name,last\_name) as name from Doctors WHERE Active = 1';

IF(@id > 0)

BEGIN

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

END

SET @sql = @sql + ' ORDER BY active, first\_name, last\_name';

EXEC(@sql);

END

ALTER procedure [dbo].[uspDoctors\_Get] @SearchBy varchar(50), @SearchValue varchar(100) as

begin

declare @sql nvarchar(max);

set @sql = 'select D.id as id, U.emp\_id,dbo.svf\_FormatDrName(D.first\_name,D.last\_name) as name, De.name as department,

D.designation, D.phone, D.gender, D.legal\_id as pathaka, D.consultation\_fee, D.active from Doctors D JOIN Users U ON U.id=D.user\_id

JOIN Departments De ON De.id=D.department\_id';

if(@SearchBy <> 'All')

begin

set @sql = @sql + ' where ' + @SearchBy + ' like ''' + @SearchValue + '%''';

end

set @sql = @sql + ' order by D.first\_name';

exec(@sql);

end

ALTER PROCEDURE [dbo].[uspUsers\_LoggedUser] @empid varchar(20)

AS

BEGIN

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

-- interfering with SELECT statements.

SET NOCOUNT ON;

SELECT u.\*, t.type\_name,s.\* FROM dbo.Users u JOIN dbo.User\_Types t ON u.user\_type\_id=t.id

LEFT JOIN

(SELECT d.id as staff\_id, d.user\_id as staff\_user\_id, dbo.svf\_FormatDrName(d.first\_name, d.last\_name) as staff\_name, d.phone as staff\_phone, 'Doctors' as staff\_tbl, 'Doctor' as staff\_type,d.nationality, d.gender, d.dob, dbo.svf\_GetAge(d.dob) as age, d.designation, dp.name as department FROM dbo.Users u JOIN dbo.Doctors d ON u.id=d.user\_id JOIN dbo.Departments dp ON dp.id=d.department\_id WHERE u.emp\_id=@empid

UNION

SELECT d.id as staff\_id, d.user\_id as staff\_user\_id, dbo.svf\_FormatName(d.first\_name, d.last\_name) as staff\_name, d.phone as staff\_phone, 'Staffs' as staff\_tbl,st.type\_title as staff\_type,d.nationality, d.gender, d.dob, dbo.svf\_GetAge(d.dob) as age, d.designation, dp.name as department FROM dbo.Users u JOIN dbo.Staffs d ON u.id=d.user\_id JOIN dbo.Departments dp ON dp.id=d.department\_id JOIN dbo.Staff\_Types st ON d.staff\_type\_id=st.id WHERE u.emp\_id=@empid)

s ON u.id=s.staff\_user\_id

WHERE u.emp\_id=@empid;

END

ALTER procedure [dbo].[uspStaffs\_Get] @SearchBy varchar(50), @SearchValue varchar(100) as

begin

declare @sql nvarchar(max);

set @sql = 'select D.id as id, U.emp\_id,dbo.svf\_FormatName( D.first\_name,D.last\_name) as name, De.name as department,

D.designation, ST.type\_title, UT.type\_name as user\_role,D.gender, D.phone, D.legal\_id as pathaka, D.active from Staffs D JOIN Users U ON U.id=D.user\_id

JOIN Departments De ON De.id=D.department\_id JOIN Staff\_Types ST ON ST.id=D.staff\_type\_id JOIN User\_Types UT ON U.user\_type\_id=UT.id ';

if(@SearchBy <> 'All')

begin

set @sql = @sql + ' where ' + @SearchBy + ' like ''' + @SearchValue + '%''';

end

set @sql = @sql + ' order by D.first\_name';

print @sql;

exec(@sql);

end

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

AS

BEGIN

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

-- interfering with SELECT statements.

SET NOCOUNT ON;

DECLARE @sql varchar(max);

SET @sql = 'SELECT id, type\_name from User\_Types WHERE (Active = 1 AND id!=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

ALTER PROCEDURE [dbo].[uspStaffs\_Add] @first\_name varchar(50), @last\_name varchar(50) = NULL, @department\_id int,

@designation varchar(50), @qualification varchar(max) = NULL, @staff\_type\_id int, @address varchar(max) = NULL,

@city varchar(max)=NULL, @state varchar(max) = NULL, @zip varchar(50)=NULL, @phone varchar(50), @email varchar(50)=NULL, @legal\_id varchar(50),

@legal\_id\_expiry date=NULL, @nationality varchar(50),@gender char(1), @dob date,

@active bit, @added\_id int, @user\_type\_id int As

SET NOCOUNT ON;

Declare @user\_id as int;

Declare @Ret int;

Set @Ret = -1;

Declare @emp\_id int;

select @emp\_id = (MAX(id) + 1) From Users;

if(@emp\_id IS NULL)

set @emp\_id = 1

Declare @prefix varchar(10);

Select @prefix = op\_value FROM Options WHERE op\_name = 'EMPID\_PREFIX';

Declare @start int;

Select @start = TRY\_CAST (op\_value AS INT) FROM Options WHERE op\_name= 'EMPID\_START';

Set @emp\_id = @emp\_id+@start;

Set @prefix = @prefix + cast(@emp\_id as varchar)

DECLARE @password VARBINARY(MAX);

SELECT @password = CAST(op\_value AS VARBINARY(MAX)) FROM Options WHERE op\_name = 'DEFAULT\_PWD';

DECLARE @salt VARBINARY(4) = CRYPT\_GEN\_RANDOM(4);

DECLARE @hash VARBINARY(MAX);

SET @hash = 0x0200 + @salt + HASHBYTES('SHA2\_512',@password + @salt);

Insert Into Users (emp\_id,password,[salt],user\_type\_id,active,log\_date)

Values (@prefix,@hash,@salt,@user\_type\_id,@active,GETDATE());

select @user\_id = SCOPE\_IDENTITY();

insert into Staffs (first\_name, last\_name, department\_id,designation, qualification, staff\_type\_id, address,

city, state, zip, phone, email, legal\_id,legal\_id\_expiry, nationality,gender,dob, user\_id,

active, added\_date,modified\_date,added\_id,modified\_id) values (@first\_name, @last\_name, @department\_id,

@designation, @qualification, @staff\_type\_id, @address,

@city, @state, @zip, @phone, @email , @legal\_id, @legal\_id\_expiry, @nationality,@gender, @dob, @user\_id,

@active, GETDATE(), getdate(),@added\_id ,@added\_id);

Set @Ret =1;

SELECT @Ret;

ALTER PROCEDURE [dbo].[uspStaffs\_Edit] @id int, @user\_id int, @first\_name varchar(50), @last\_name varchar(50) = NULL, @department\_id int,

@designation varchar(50), @qualification varchar(max) = NULL, @staff\_type\_id int, @address varchar(max) = NULL,

@city varchar(max)=NULL, @state varchar(max) = NULL, @zip varchar(50)=NULL, @phone varchar(50), @email varchar(50)=NULL, @legal\_id varchar(50),

@legal\_id\_expiry date=NULL, @nationality varchar(50),@gender char(1), @dob date,

@active bit, @modified\_id int, @user\_type\_id int

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;

BEGIN

UPDATE dbo.[Users] SET active=@active, user\_type\_id=@user\_type\_id WHERE [ID]=@user\_id

UPDATE dbo.[Staffs] SET [first\_name] = @first\_name, [last\_name]=@last\_name, department\_id=@department\_id, designation=@designation, qualification=@qualification,

staff\_type\_id=@staff\_type\_id, address=@address, city=@city,state=@state, zip=@zip, phone=@phone,email=@email,

legal\_id=@legal\_id,legal\_id\_expiry=@legal\_id\_expiry,nationality=@nationality, gender=@gender, dob=@dob,modified\_id=@modified\_id, modified\_date=getdate(), active=@active WHERE [id]=@id

SET @Ret = 1;

END

SELECT @Ret

END

ALTER PROCEDURE [dbo].[uspPatients\_Get] @SearchBy varchar(50), @SearchValue varchar(100) AS

BEGIN

DECLARE @sql nvarchar(max);

SET @sql = 'SELECT id, patient\_number, dbo.svf\_FormatName(first\_name,last\_name) as name,

gender, dbo.svf\_GetAge(dob) AS age , dbo.svf\_FormatAddress(address,city,state,zip) as address1, phone FROM Patients';

IF(@SearchBy <> 'All')

BEGIN

SET @sql = @sql + ' WHERE ' + @SearchBy + ' LIKE ''' + @SearchValue + '%'' ORDER BY patient\_number';

END

EXEC(@sql);

END

ALTER PROCEDURE [dbo].[uspStaffs\_GetSingle] @id int

AS

BEGIN

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

-- interfering with SELECT statements.

SET NOCOUNT ON;

SELECT d.\*, dt.name,U.emp\_id ,st.type\_title,ut.id as user\_type\_id, ut.type\_name as user\_type\_name FROM dbo.[Staffs] d JOIN dbo.Departments dt ON dt.id=d.department\_id

JOIN USERS U ON U.id=d.user\_id JOIN Staff\_Types st ON st.id=d.staff\_type\_id JOIN User\_Types ut ON ut.id=u.user\_type\_id WHERE d.id=@id

END