--Creating Tbl\_Usermaster table –Copy this and paste it on sqlquirymanager to --create tbl\_UserMaster..

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: Table [dbo].[tbl\_UserMaster] Script Date: 11/06/2008 10:21:26 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[tbl\_UserMaster](

[UserId] [int] NOT NULL,

[FirstName] [varchar](50) NOT NULL,

[LastName] [varchar](50) NULL,

[Gender] [varchar](50) NULL,

[UserDOB] [datetime] NOT NULL,

[UserDOR] [datetime] NOT NULL,

[UserLoginId] [varchar](50) NOT NULL,

[Password] [varchar](50) NOT NULL,

[HintQuestion] [varchar](50) NULL,

[HintAnswer] [varchar](50) NULL,

[UserPhoto] [varbinary](max) NULL,

[ImageFileName] [varchar](50) NULL,

[EmailId] [varchar](50) NULL,

[AlternateEmailId] [varchar](50) NULL,

[FaxNo] [varchar](25) NULL,

[Status] [varchar](20) NULL,

CONSTRAINT [PK\_tbl\_UserMaster] PRIMARY KEY CLUSTERED

(

[UserId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

--Creating Tbl\_Loginmaster table –Copy this and paste it on sqlquirymanager ---to create tbl\_ Loginmaster..

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: Table [dbo].[tbl\_LoginMaster] Script Date: 11/06/2008 10:24:05 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[tbl\_LoginMaster](

[UserId] [int] NOT NULL,

[UserLoginDate] [datetime] NOT NULL,

[UserLoginTime] [varchar](50) NOT NULL,

[UserLogOffTime] [varchar](50) NULL,

CONSTRAINT [PK\_tbl\_LoginMaster] PRIMARY KEY CLUSTERED

(

[UserId] ASC,

[UserLoginDate] ASC,

[UserLoginTime] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

ALTER TABLE [dbo].[tbl\_LoginMaster] WITH CHECK ADD CONSTRAINT [FK\_tbl\_LoginMaster\_tbl\_UserMaster] FOREIGN KEY([UserId])

REFERENCES [dbo].[tbl\_UserMaster] ([UserId])

ON UPDATE CASCADE

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[tbl\_LoginMaster] CHECK CONSTRAINT [FK\_tbl\_LoginMaster\_tbl\_UserMaster]

--Creating Tbl\_ UserAddress table –Copy this and paste it on sqlquirymanager --to create tbl\_ UserAddress..

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: Table [dbo].[tbl\_UserAddress] Script Date: 11/06/2008 10:25:29 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[tbl\_UserAddress](

[UserId] [int] NOT NULL,

[UserAddressId] [int] NOT NULL,

[UserAddressType] [varchar](50) NOT NULL,

[HNo] [varchar](50) NULL,

[Street] [varchar](50) NULL,

[City] [varchar](50) NULL,

[State] [varchar](50) NULL,

[Country] [varchar](50) NULL,

[PinCode] [varchar](50) NULL,

CONSTRAINT [PK\_tbl\_UserAddress] PRIMARY KEY CLUSTERED

(

[UserId] ASC,

[UserAddressId] ASC,

[UserAddressType] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

ALTER TABLE [dbo].[tbl\_UserAddress] WITH CHECK ADD CONSTRAINT [FK\_tbl\_UserAddress\_tbl\_UserMaster] FOREIGN KEY([UserId])

REFERENCES [dbo].[tbl\_UserMaster] ([UserId])

ON UPDATE CASCADE

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[tbl\_UserAddress] CHECK CONSTRAINT [FK\_tbl\_UserAddress\_tbl\_UserMaster]

--Creating Tbl\_ UserPhones table –Copy this and paste it on sqlquirymanager ---to create tbl\_ UserPhones..

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: Table [dbo].[tbl\_UserPhones] Script Date: 11/06/2008 10:26:18 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[tbl\_UserPhones](

[UserId] [int] NOT NULL,

[PhoneNumber] [varchar](50) NOT NULL,

[UserPhoneType] [varchar](50) NOT NULL,

CONSTRAINT [PK\_tbl\_UserPhones\_1] PRIMARY KEY CLUSTERED

(

[UserId] ASC,

[PhoneNumber] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

ALTER TABLE [dbo].[tbl\_UserPhones] WITH CHECK ADD CONSTRAINT [FK\_tbl\_UserPhones\_tbl\_UserMaster] FOREIGN KEY([UserId])

REFERENCES [dbo].[tbl\_UserMaster] ([UserId])

ON UPDATE CASCADE

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[tbl\_UserPhones] CHECK CONSTRAINT [FK\_tbl\_UserPhones\_tbl\_UserMaster]

--Procedure

--spInsertUserDetails

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[spInsertUserDetails] Script Date: 11/06/2008 10:28:28 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE Procedure [dbo].[spInsertUserDetails]

(

@FirstName varchar(50),

@LastName varchar(50),

@Gender varchar(50),

@UserDOB datetime,

@EmailId varchar(50),

@AlternateEmailId varchar(50),

@UserLoginId varchar(50),

@Password varchar(50),

@HintQuestion varchar(50),

@HintAnswer varchar(50),

@UserPhoto varbinary(max),

@ImageFileName varchar(50),

@FaxNo varchar(25),

@AddressCount int,

@Address1 varchar(1000),

@Address2 varchar(1000),

@Address3 varchar(1000),

@PhoneCount int,

@PhoneString1 varchar(100),

@PhoneString2 varchar(100),

@PhoneString3 varchar(100),

@Flag int output,

@PhoneFlag Varchar(50) output,

@AddressFlag varchar(50) output

)

as

begin ---------------------------------1

if not exists(Select UserLoginId from tbl\_UserMaster where UserLoginId=@UserLoginId)

begin ---------------------------------2

Declare @a int

Declare @b int

Declare @c int

set @a=0

set @b=0

set @c=0

Begin Transaction S ----------------tarnsaction 3

Declare @UserId int

exec spGeneratedUserId @UserId output

declare @UserDOR datetime

set @UserDOR=(select dateadd(day,datediff(day,0, getdate()),0))

--set @UserDOB=(Select convert(varchar(50),@UserDOB,103))

Insert into tbl\_UserMaster(UserID,FirstName, LastName,Gender,UserDOB,UserDOR,UserLoginId,

Password,HintQuestion,HintAnswer,UserPhoto,ImageFileName,EmailId,AlternateEmailId,FaxNo,Status)

Values(@UserId,@FirstName,@LastName,@Gender,@UserDOB, @UserDOR,@UserLoginId,@Password,

@HintQuestion,@HintAnswer,@UserPhoto,@ImageFileName,@EmailId,@AlternateEmailId,@FaxNo,'Active')

set @a=@@RowCount

--Starting Region for Address parameters for getting splitter out put

declare @addstart int

set @addstart=0

declare @AddType varchar(20)

declare @Hno varchar(100)

declare @Street varchar(100)

declare @City varchar(100)

declare @pincode varchar(10)

declare @State varchar(100)

declare @Country varchar(100)

declare @AId int

while (@addstart < @AddressCount)

begin

set @AId=(Select max(UserAddressId)+1 from tbl\_UserAddress)

set @addstart =@addstart+1

if(@addstart =1)

begin

exec spSplitaddressData @Address1,@AddType output,

@Hno output,@Street output,@City output,@pincode output,@State output,@Country output

end

else if(@addstart =2)

begin

exec spSplitaddressData @Address2,@AddType output,

@Hno output,@Street output,@City output,@pincode output,@State output,@Country output

end

else if( @addstart =3)

begin

exec spSplitaddressData @Address3,@AddType output,

@Hno output,@Street output,@City output,@pincode output,@State output,@Country output

end

if(@AId is null)

begin

set @AId=1001

Insert into tbl\_UserAddress(UserId,UserAddressId,UserAddressType,[HNO],Street,City,State,Country,Pincode)

values(@UserId,@AId,@AddType,@Hno,@Street,@City,@State,@Country,@pincode)

set @b= @@RowCount

end

else

begin

Insert into tbl\_UserAddress(UserId,UserAddressId,UserAddressType,[HNO],Street,City,State,Country,Pincode)

values(@UserId,@AId,@AddType,@Hno,@Street,@City,@State,@Country,@pincode)

set @b=@b+@@RowCount

if(@@Error >0)

begin

set @AddressFlag='Address Type Already assigned'--Not Inserted Primary Key violation

end

end

end

--End Region For Address Insert

--Strat Region for Insert PhoneDetails

declare @PhoneStart int

set @PhoneStart=0

declare @PhoneType varchar(20)

declare @PhoneNumber varchar(100)

while (@PhoneStart < @PhoneCount)

begin

set @PhoneStart =@PhoneStart+1

if(@PhoneStart =1)

begin

exec spSplitPhoneDetails @PhoneString1,@PhoneType output, @PhoneNumber output

end

else if(@PhoneStart =2)

begin

exec spSplitPhoneDetails @PhoneString2,@PhoneType output, @PhoneNumber output

end

else if( @PhoneStart =3)

begin

exec spSplitPhoneDetails @PhoneString3,@PhoneType output, @PhoneNumber output

end

Insert into tbl\_UserPhones(UserId,PhoneNumber,UserPhoneType)Values

(@UserId,@PhoneNumber,@PhoneType)

set @c= @c+@@RowCount

if(@@Error >0)

begin

set @PhoneFlag='PhoneNo already Used by other User'--Not Inserted Primary Key violation

end

end

---End Region for Phone Insert Details

if(@AddressCount=@b)

set @b=1

if(@PhoneCount=@c)

set @c=1

if(@a=1 and @b=1 and @c=1)

begin

commit Transaction S

set @Flag=1

end

else

begin

rollback Transaction S

set @Flag=0

end

end

end

--spGeneratedUserId

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[spGeneratedUserId] Script Date: 11/06/2008 10:34:06 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE Procedure [dbo].[spGeneratedUserId]

(@UserId int Output)

as

begin

set @UserId=(Select max(UserId)from tbl\_userMaster)

if(@UserId is Null)

begin

set @UserId=500601

return

end

else

begin

set @UserId=(Select max(UserId)+1 from tbl\_userMaster)

return

end

end

--spSplitaddressData

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[spSplitAddressData] Script Date: 11/06/2008 10:36:58 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

Create proc [dbo].[spSplitAddressData](

@indata varchar(1000),

@AddType varchar(50) output,

@Hno varchar(100) output,

@Street varchar(100) output,

@City varchar(50) output,

@pincode varchar(10) output,

@State varchar(25) output,

@Country varchar(20) output

)

as

begin

declare @value nvarchar(50)

declare @splitter nvarchar(1000)

declare @delimit nvarchar(1)

declare @str1 varchar(100)

declare @cou int

declare @Len int

declare @DeliCou int

set @DeliCou=0

select @delimit = '^'

select @splitter = @indata

set @len=Len(@splitter)

set @cou=0

while(@cou <@len)

begin

set @cou=@cou+1

set @str1=(select substring(@splitter,@cou,1))

if(@str1 = '^')

begin

set @DeliCou=@DeliCou+1

end

end

--we got the seperator value

declare @delCount int

set @delCount=0

declare @temp varchar(100)

declare @temp1 varchar(100)

while(@delCount<@DeliCou)

begin

set @delCount=@delCount+1

set @temp=(select substring(@splitter,1,charindex(@delimit,@splitter)))

set @splitter=replace(@splitter,@temp,'')

set @temp1=replace(@temp,'^','')

if(@delCount=1)

set @AddType=@temp1

else if(@delCount=2)

set @Hno=@temp1

else if(@delCount=3)

set @Street=@temp1

else if(@delCount=4)

set @City=@temp1

else if(@delCount=5)

set @pincode=@temp1

else if(@delCount=6)

begin

set @State=@temp1

set @Country=@splitter

end

end

end

--spSplitPhoneDetails

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[spSplitPhoneDetails] Script Date: 11/06/2008 10:38:19 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

Create proc [dbo].[spSplitPhoneDetails](

@indata varchar(1000),

@PhoneType varchar(50) output,

@PhoneNumber varchar(100) output

)

as

begin

declare @value nvarchar(50)

declare @splitter nvarchar(1000)

declare @delimit nvarchar(1)

declare @str1 varchar(100)

declare @cou int

declare @Len int

declare @DeliCou int

set @DeliCou=0

select @delimit = '^'

select @splitter = @indata

set @len=Len(@splitter)

set @cou=0

while(@cou <@len)

begin

set @cou=@cou+1

set @str1=(select substring(@splitter,@cou,1))

if(@str1 = '^')

begin

set @DeliCou=@DeliCou+1

end

end

--we got the seperator value

declare @delCount int

set @delCount=0

declare @temp varchar(100)

declare @temp1 varchar(100)

while(@delCount<@DeliCou)

begin

set @delCount=@delCount+1

set @temp=(select substring(@splitter,1,charindex(@delimit,@splitter)))

set @splitter=replace(@splitter,@temp,'')

set @temp1=replace(@temp,'^','')

if(@delCount=1)

begin

set @PhoneType=@temp1

set @PhoneNumber=@splitter

end

end

end

-- spUpdateUserPersonalDetails

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[spUpdateUserPersonalDetails] Script Date: 11/06/2008 10:42:32 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create Procedure [dbo].[spUpdateUserPersonalDetails]

(

@UserId int,

@EmailId varchar(50),

@AlternateEmailId varchar(50),

@UserPhoto varbinary (max),

@ImageFileName varchar(150),

@faxNo varchar(20)

)

as

begin

Update tbl\_UserMaster set EmailId=@EmailId, AlternateEmailId=@AlternateEmailId,

UserPhoto=@UserPhoto,ImageFileName=@ImageFileName,faxNo=@faxNo Where UserId=@UserId

Return @@RowCount

end

-- spUpdatePhoneDetails

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[spUpdatePhoneDetails] Script Date: 11/06/2008 10:44:35 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

Create PROCEDURE [dbo].[spUpdatePhoneDetails]

(

@UserId int,

@PhoneType varchar(50),

@PhoneNumber varchar(20),

@Msg varchar(50) output

)

AS

begin

if not exists(select \* from tbl\_UserPhones where

userId=@UserId and PhoneNumber=@PhoneNumber)

begin

update tbl\_UserPhones set PhoneNumber=@PhoneNumber where

userId=@UserId and UserPhoneType=@PhoneType

set @Msg='Phone No Added'

end

else

begin

set @Msg='Already this number given,choose another.'

end

end

-- spUpdateAddressDetails

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[spUpdateAddressDetails] Script Date: 11/06/2008 10:45:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

Create PROCEDURE [dbo].[spUpdateAddressDetails]

(

@UserId int,

@UserAddressType varchar(20),

@Hno varchar(50),

@Street varchar(100),

@City varchar(50),

@State varchar(50),

@Country varchar(50),

@pincode varchar(10),

@Msg varchar(150) output

)

AS

begin

update tbl\_UserAddress set Hno=@Hno,Street=@Street,City=@City,

State=@State,Country=@Country,PinCode=@PinCode where UserId=@UserId and

UserAddressType=@UserAddressType

if(@@rowcount =1)

set @Msg='Update sucessfully Completed.'

end

RETURN

-- spInsertOneAddressData

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[spInsertOneAddressData] Script Date: 11/06/2008 10:46:13 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

Create PROCEDURE [dbo].[spInsertOneAddressData]

(

@UserId int,

@UserAddressId int,

@UserAddressType varchar(50),

@Hno varchar(250),

@Street varchar(100),

@City varchar(50),

@State varchar(50),

@Country varchar(50),

@PinCode varchar(10),

@Msg varchar(150) output

)

AS

begin

if not exists(select \* from tbl\_UserAddress where UserId=@UserId and UserAddressType=@UserAddressType)

begin

insert into tbl\_UserAddress(UserId,UserAddressId,UserAddressType,Hno,Street,City,State,Country,PinCode) values(@UserId,@UserAddressId,@UserAddressType,

@Hno,@Street,@City,@State,@Country,@PinCode)

set @Msg=@UserAddressType +'AddressType Data Inserted'

end

else

begin

set @Msg='This type of address already given by U.'

end

end

-- spInsertOnePhoneDetails

USE [GiveTheDatabaseName]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[spInsertOnePhoneDetails] Script Date: 11/06/2008 10:46:54 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

Create PROCEDURE [dbo].[spInsertOnePhoneDetails]

(

@UserId int,

@PhoneType varchar(50),

@PhoneNumber varchar(50),

@Msg varchar(100) output

)

AS

begin

if not exists(select \* from tbl\_UserPhones where UserId=@UserId and

UserPhoneType=@PhoneType and PhoneNumber=@PhoneNumber)

begin

insert into tbl\_UserPhones values(@UserId,@PhoneNumber,@PhoneType)

set @Msg='Your '+@PhoneType+' Type Phone No Added.'

end

else

begin

set @Msg='Given data already exists.'

end

end