# **Encrypt Function to encrypt password before storing it in the database**

USE [python\_project]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[ENCRYPT] Script Date: 4/22/2018 10:27:34 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER FUNCTION [dbo].[ENCRYPT]

(

@password VARCHAR(MAX)

)

RETURNS VARCHAR(MAX)

AS

BEGIN

DECLARE

@STR\_LEN NUMERIC(10),

@ENCRYPTED\_PASSWORD VARCHAR(100),

@TRIAL\_CHARACTER VARCHAR(1),

@TRIAL\_NUMBER NUMERIC(4)

SET @ENCRYPTED\_PASSWORD = NULL

SET @STR\_LEN =LEN(@password)

DECLARE

@I INT

SET @I = 1

DECLARE

@LOOP$BOUND INT

SET @LOOP$BOUND = @STR\_LEN

WHILE @I <= @LOOP$BOUND

BEGIN

SET @TRIAL\_CHARACTER = SUBSTRING(@password, @I, 1)

SET @TRIAL\_NUMBER = ASCII(@TRIAL\_CHARACTER)

IF (@TRIAL\_NUMBER % 2) = 0

SET @TRIAL\_NUMBER = @TRIAL\_NUMBER - 6

ELSE

SET @TRIAL\_NUMBER = @TRIAL\_NUMBER - 8

SET @TRIAL\_CHARACTER = CHAR(CAST(@TRIAL\_NUMBER + @I AS INT))

SET @ENCRYPTED\_PASSWORD = ISNULL(@ENCRYPTED\_PASSWORD, '') + ISNULL(@TRIAL\_CHARACTER, '')

SET @I = @I + 1

END

RETURN @ENCRYPTED\_PASSWORD

END

# **Decrypt Function to decrypt password stored in database**

USE [python\_project]

Go

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER FUNCTION [dbo].[DECRYPT]

(

@DB\_ROLE\_PASSWORD VARCHAR(MAX)

)

RETURNS VARCHAR(MAX)

AS

BEGIN

DECLARE

@STR\_LEN NUMERIC(10),

@DECRYPTED\_PASSWORD VARCHAR(100),

@TRIAL\_CHARACTER VARCHAR(1),

@TRIAL\_NUMBER NUMERIC(4),

@CHECK\_CHARACTER VARCHAR(1),

@V\_DB\_ROLE\_PASSWORD VARCHAR(100)

SET @V\_DB\_ROLE\_PASSWORD = @DB\_ROLE\_PASSWORD

SET @DECRYPTED\_PASSWORD = NULL

SET @STR\_LEN = LEN(@V\_DB\_ROLE\_PASSWORD)

DECLARE

@I INT

SET @I = 1

DECLARE

@LOOP$BOUND INT

SET @LOOP$BOUND = @STR\_LEN

WHILE @I <= @LOOP$BOUND

BEGIN

SET @TRIAL\_CHARACTER = SUBSTRING(@V\_DB\_ROLE\_PASSWORD, @I, 1)

SET @TRIAL\_NUMBER = ASCII(@TRIAL\_CHARACTER) - @I

IF (@TRIAL\_NUMBER % 2) = 0

SET @TRIAL\_NUMBER = @TRIAL\_NUMBER + 6

ELSE

SET @TRIAL\_NUMBER = @TRIAL\_NUMBER + 8

SET @DECRYPTED\_PASSWORD = ISNULL(@DECRYPTED\_PASSWORD,'') + ISNULL(CHAR(CAST(@TRIAL\_NUMBER AS INT)), '')

SET @I = @I + 1

END

RETURN @DECRYPTED\_PASSWORD

END

# **Stored procedure to validate user’s login details**

USE [python\_project]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[Login\_User] Script Date: 4/22/2018 10:29:59 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[Login\_User]

@UserName NVARCHAR(200),

@PW NVARCHAR(200),

@responseMessage NVARCHAR(250)='' OUTPUT

AS

BEGIN

SET NOCOUNT ON

DECLARE @userID INT

IF EXISTS (SELECT TOP 1 USER\_ID FROM [dbo].[Users] WHERE User\_emailid=@UserName)

BEGIN

SELECT @userID=(SELECT User\_ID FROM [dbo].[Users] WHERE User\_emailid=@UserName AND User\_Password=dbo.[ENCRYPT](@PW))

IF(@userID IS NULL)

SELECT @responseMessage='Incorrect password'

ELSE

SELECT @responseMessage='User successfully logged in'

END

else

SELECT @responseMessage='Incorrect password'

END

# **Logger stored procedure to store deleted application details along with the user details who deleted the app**

USE [python\_project]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[delete\_Application\_by\_user] Script Date: 4/22/2018 10:31:11 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER procedure [dbo].[delete\_Application\_by\_user]

@username Nvarchar(320),

@appid int

as

begin

declare @Log\_id int,

@activity varchar(25)

IF EXISTS (SELECT TOP 1 App\_id FROM [dbo].[Application] WHERE App\_id<>@appid)

begin

Select @Log\_id = (select count(\*)+1 from Application\_Audit);

SET @activity = 'DELETE';

INSERT into Application\_Audit values (@Log\_id, @appid,@activity,GETDATE(),@username);

end

end

# **Logger stored procedure to store inserted application details along with the user details who created new app**

USE [python\_project]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[insert\_Application\_by\_user] Script Date: 4/22/2018 10:32:08 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER procedure [dbo].[insert\_Application\_by\_user]

@username Nvarchar(320),

@appid int

as

begin

declare @Log\_id int,

@activity varchar(25)

IF EXISTS (SELECT App\_id FROM [dbo].[Application] WHERE App\_id=@appid)

begin

Select @Log\_id = (select count(\*)+1 from Application\_Audit);

SET @activity = 'INSERT';

INSERT into Application\_Audit values (@Log\_id, @appid,@activity,GETDATE(),@username);

End

end

# **Trigger to audit data of User table**

USE [python\_project]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[Users\_Audit] Script Date: 4/22/2018 10:33:25 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER trigger [dbo].[Users\_Audit]

on [dbo].[Users]

AFTER UPDATE, INSERT, DELETE

as

declare

@LogId int,

@User\_id int,

@message varchar(20),

@event\_timestamp datetime;

if exists(SELECT \* from inserted) and exists (SELECT \* from deleted)

begin

SELECT @LogId = count(\*)+1 from Logger\_User;

SET @message = 'UPDATE';

SELECT @User\_id = i.User\_id from inserted i;

INSERT into Logger\_User values (@LogId, @User\_id,@message,GETDATE());

end

If exists (Select \* from inserted) and not exists(Select \* from deleted)

begin

SELECT @LogId = count(\*)+1 from Logger\_User;

SET @message = 'INSERT';

SELECT @User\_id = i.User\_id from inserted i;

INSERT into Logger\_User values (@LogId, @User\_id,@message,GETDATE());

end

If exists(select \* from deleted) and not exists(Select \* from inserted)

begin

SELECT @LogId = count(\*)+1 from Logger\_User;

SET @message = 'DELETE';

SELECT @User\_id = i.User\_id from deleted i;

INSERT into Logger\_User values (@LogId, @User\_id,@message,GETDATE());

end

# **to find downloads of an application based on Country**

USE [python\_project]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[App\_count\_based\_on\_countries] Script Date: 4/22/2018 10:34:19 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER function [dbo].[App\_count\_based\_on\_countries](@Country varchar(20), @app\_id int)

returns table

return(

select i1.Country as CountryName, count(\*) as 'Total\_Count' from(

select d.download\_id, d.country as Country from download d

where d.country=@Country and App\_id=@app\_id

)i1

group by i1.Country

)