O campo de email será criptografado, pois é uma informação que não deve estar visível para todos, mas pode ser necessário em alguns casos, como no envio de um e-mail para recuperação de senha. A password não pode ser recuperada por ninguém além do seu dono, por esse motivo, será utilizado um hashing.

--criação da master key e simmectric key

Create MASTER KEY ENCRYPTION BY PASSWORD = 'Mk1729';

Create CERTIFICATE EncryptCert With subject='Protect data';

Create SYMMETRIC KEY EmailAddressKey

With Algorithm = AES\_256

ENCRYPTION BY CERTIFICATE EncryptCert;

--atualização da procedure de login para funcionar com o hashing e crypting

CREATE OR ALTER PROCEDURE spLogin

(@EmailAddress NVARCHAR(255),

@Password NVARCHAR(255),

@NewPassword NVARCHAR(255) = NULL)

AS

BEGIN

DECLARE CursorEmployee CURSOR FOR select top 1 EmployeeNumber, ResetPassword

from Employee

where Password = CONVERT(VARCHAR(32), HashBytes('MD5', @Password), 2)

AND convert(Nvarchar(256), DECRYPTBYKEY(EmailAddress)) = @EmailAddress

DECLARE @EmployeeNumber VARCHAR(50)

DECLARE @ResetPassword CHAR

OPEN CursorEmployee

FETCH NEXT FROM CursorEmployee INTO @EmployeeNumber, @ResetPassword

IF (@EmployeeNumber IS NULL)

EXEC sp\_PreencherErrorLog 9100, '''', @EmailAddress;

ELSE IF (@ResetPassword = 1)

IF (@NewPassword IS NULL)

EXEC sp\_PreencherErrorLog 9250, '''', @EmailAddress;

ELSE

BEGIN

UPDATE Employee SET [Password] = HashBytes('MD5', @NewPassword), ResetPassword = 0 WHERE EmployeeNumber = @EmployeeNumber

PRINT 'Password reset. Your EmployeeNumber: ' + @EmployeeNumber

END

ELSE

PRINT 'Login successfull. Your EmployeeNumber: ' + @EmployeeNumber

CLOSE CursorEmployee

DEALLOCATE CursorEmployee

END

GO

--test

OPEN SYMMETRIC KEY EmailAddressKey DECRYPTION BY CERTIFICATE EncryptCert;

exec spLogin @EmailAddress='user1', @Password='user1'

Close SYMMETRIC KEY EmailAddressKey;

CREATE OR ALTER PROCEDURE spInsertEmployee @EmployeeNumber nvarchar (30), @FirstName nvarchar (30), @MiddleName nvarchar (30) = NULL, @LastName nvarchar (30), @UserName nvarchar (30), @EmailAddress nvarchar (30), @Password nvarchar (50) = NULL, @ResetPassword nvarchar (50) = NULL AS

DECLARE @temp VARCHAR(250)

IF EXISTS (SELECT \* FROM Employee WHERE EmployeeNumber = @EmployeeNumber)

BEGIN

SET @temp = CAST(('EmployeeNumber: '+ @EmployeeNumber) AS varchar);

EXEC sp\_PreencherErrorLog 9220, '', @temp;

RETURN

END

IF EXISTS (SELECT \* FROM Employee WHERE EmailAddress = @EmailAddress)

BEGIN

SET @temp = CAST(('EmailAddress: '+ @EmailAddress) AS varchar);

EXEC sp\_PreencherErrorLog 9220, '', @temp;

RETURN

END

INSERT INTO Employee (EmployeeNumber, FirstName, MiddleName, LastName, UserName, EmailAddress, Password, ResetPassword) VALUES ( @EmployeeNumber, @FirstName, @MiddleName, @LastName, @UserName, ENCRYPTBYKEY(KEY\_GUID('EmailAddressKey'), @EmailAddress), CONVERT(VARCHAR(32), HashBytes('MD5', @Password), 2), @ResetPassword)

--test

OPEN SYMMETRIC KEY EmailAddressKey DECRYPTION BY CERTIFICATE EncryptCert;

exec spInsertEmployee @EmployeeNumber=20, @FirstName='test', @LastName='name', @UserName='user10', @EmailAddress='p@b.com', @Password='p@b.com', @ResetPassword=0

Close SYMMETRIC KEY EmailAddressKey;