**Connection String:**

1. Scaffold-DbContext "Server=your\_server; Database=your\_database; User=your\_username; Password=your\_password; " Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
2. Scaffold-DbContext "Server=DESKTOP-8SC7SR9; Database=qv\_elite; Trusted\_Connection=True; encrypt=false" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models

**Data Annotation:**

1. Key / Composite keys:

public partial class Student{

[Key]

public int Id { get; set; }

[Key]

[Column("Name")]

public string SName { get; set; } = null!;

[Column(Order = 2)]

public int? Roll { get; set; }

}

1. [ForeignKey("Passport")]
2. [Display(Name = "Name")]
3. [Column("Name")]
4. [MaxLength(10),MinLength(5)]
5. [MaxLength(10, ErrorMessage="must be 10"),MinLength(5)]
6. [NotMapped]
7. [ComplexType]
8. [Timestamp]
9. [Table("InternalBlogs")]
10. [Column("BlogDescription", TypeName="ntext")]
11. [Index("PostRatingIndex")]
12. [StringLength(200)]
13. [InverseProperty("CreatedBy")]

**Database builder:**

builder.Services.AddDbContext<QvEliteContext>(options => options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));

**Encryption and Decryption:**

1. Symmetric or secret key encryption algorithms(**AES algorithm**)

System.Security.Cryptography

public async Task<byte[]> EncryptAsync(string clearText, string passphrase){

using Aes aes = Aes.Create();

aes.Key = DeriveKeyFromPassword(passphrase);

aes.IV = IV;

using MemoryStream output = new();

using CryptoStream cryptoStream = new(output, aes.CreateEncryptor(), CryptoStreamMode.Write);

await cryptoStream.WriteAsync(Encoding.Unicode.GetBytes(clearText));

await cryptoStream.FlushFinalBlockAsync();

return output.ToArray();

}

public async Task<string> DecryptAsync(byte[] encrypted, string passphrase){

using Aes aes = Aes.Create();

aes.Key = DeriveKeyFromPassword(passphrase);

aes.IV = IV;

using MemoryStream input = new(encrypted);

using CryptoStream cryptoStream = new(input, aes.CreateDecryptor(), CryptoStreamMode.Read);

using MemoryStream output = new();

await cryptoStream.CopyToAsync(output);

return Encoding.Unicode.GetString(output.ToArray());

}

1. Asymmetric or public key encryption is based on public/private key pairs.
2. Hashing algorithms provide one-way encryption