EF CORE:

GENCAY YILDIZ:

DERS 7: DATABASE FIRST

1. Package manager console
2. Conection strings - <https://www.connectionstrings.com/>
3. Scaffold-DbContext ‘Connection String’ Microsoft.EntityFrameworkCore.[Provider – hansina gore];
4. Standart – sql server - Server=myServerAddress;Database=myDataBase;User Id=myUsername;Password=myPassword;
5. Trusted connection – sql server - Server=myServerAddress;Database=myDataBase;Trusted\_Connection=True;
6. Scaffold-DbContext 'Server=localhost;Database=GYSALES\_RDMS\_FULL;Trusted\_Connection=True;TrustServerCertificate=True' Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
7. PMS ile isleyebilmek ucun – Microsoft.EntityFrameworkCore.Tools ve database provider yuklemek lazimdir. Mes: sql server provider - Microsoft.EntityFrameworkCore.SqlServer
8. Hedef tablo - Scaffold-DbContext ‘Connection String’ Microsoft.EntityFrameworkCore.[Provider – hansina gore] –Tables Table1, Table2 …
9. Context name vermek - Scaffold-DbContext ‘Connection String’ Microsoft.EntityFrameworkCore.[Provider] –Context ContextName
10. Path - Scaffold-DbContext ‘Connection String’ Microsoft.EntityFrameworkCore.[Provider] –ContextDir Data –OutputDir Models
11. Namespace - Scaffold-DbContext ‘Connection String’ Microsoft.EntityFrameworkCore.[Provider] –NameSpace YourNamespace -ContextNamespace YourContextNameSpace
12. Model update - Scaffold-DbContext ‘Connection String’ Microsoft.EntityFrameworkCore.[Provider] –Force
13. Modellerin ozellestirilmesi – Partial classlardan istifade et, cunki force etsen senin entitylere elave etdilerin silinecek
14. Dotnet cli
15. dotnet ef dbcontext scaffold "Server=localhost;Database=GYSALES\_RDMS\_FULL;Trusted\_Connection=True;TrustServerCertificate=True" Microsoft.EntityFrameworkCore.SqlServer -o Models -c MyDbContext -d ( powershell ya cmdden)
16. DOTNET CLI ile isleyebilmek ucun – Microsoft.EntityFrameworkCore.Design ve database provider yuklemek lazimdir. Mes: sql server provider - Microsoft.EntityFrameworkCore.SqlServer
17. Hedef tablo - dotnet ef dbcontext scaffold "Server=localhost;Database=GYSALES\_RDMS\_FULL;Trusted\_Connection=True;TrustServerCertificate=True" Microsoft.EntityFrameworkCore.SqlServer --table Table1 --table Table2 --table Table3 …
18. Context name vermek - dotnet ef dbcontext scaffold "Server=localhost;Database=GYSALES\_RDMS\_FULL;Trusted\_Connection=True;TrustServerCertificate=True" Microsoft.EntityFrameworkCore.SqlServer --context ContextName
19. dotnet ef dbcontext scaffold ‘Connection String’ Microsoft.EntityFrameworkCore.[Provider – hansina gore];
20. Path - dotnet ef dbcontext scaffold ‘Connection String’ Microsoft.EntityFrameworkCore.[Provider] --context-dir Data --output-dir Models
21. Namespace - dotnet ef dbcontext scaffold ‘Connection String’ Microsoft.EntityFrameworkCore.[Provider] –namespace YourNamespace –contextnamespace YourContextNamespace
22. Model update - dotnet ef dbcontext scaffold ‘Connection String’ Microsoft.EntityFrameworkCore.[Provider] –force
23. Modellerin ozellestirilmesi -

**Seçeneklerin Açıklamaları**

1. **-o veya --output-dir:**
   * **Açıklama:** Oluşturulan sınıfların yerleştirileceği dizini belirtir.
   * **Kullanımı:** -o Models
   * **Örnek:** -o Models komutu, oluşturulan varlık sınıflarını ve DbContext sınıfını "Models" adlı bir klasöre yerleştirir.
2. **-c veya --context:**
   * **Açıklama:** Oluşturulacak DbContext sınıfının adını belirtir.
   * **Kullanımı:** -c MyDbContext
   * **Örnek:** -c MyDbContext komutu, oluşturulan DbContext sınıfının adını MyDbContext olarak belirler
3. **-d veya --data-annotations:**

* **Açıklama:** Veri açıklamalarını (data annotations) kullanarak model oluşturur. Bu seçenek, oluşturulan model sınıflarında Fluent API yerine veri açıklamalarını kullanır.
* **Kullanımı:** -d
* **Örnek:** -d komutu, varlık sınıflarında veri açıklamalarını kullanır.

 -o Models: Oluşturulan sınıflar "Models" klasörüne yerleştirilecek.

 -c MyDbContext: Oluşturulan DbContext sınıfının adı MyDbContext olacak.

 -d: Varlık sınıflarında veri açıklamaları (data annotations) kullanılacak.

1. **--context veya -c:**
   * Uzun Yol: --context ContextName
   * Kısa Yol: -c ContextName
2. **--context-dir veya --contextdir:**
   * Uzun Yol: --context-dir Data
   * Kısa Yol: --contextdir Data – buna error verir
3. **--output-dir veya -o:**
   * Uzun Yol: --output-dir Models
   * Kısa Yol: -o Models

dotnet ef dbcontext scaffold komutunun çeşitli seçenekleri vardır:

* **-c veya --context**: Oluşturulacak DbContext sınıfının adını belirtir.
* **-t veya --table**: Belirtilen tabloyu ya da tabloları dahil eder.
* **-s veya --schema**: Belirtilen şemayı ya da şemaları dahil eder.
* **-f veya --force**: Var olan dosyaların üzerine yazmayı sağlar.
* **-d veya --data-annotations**: Veri açıklamalarını kullanarak model oluşturur.
* **--context-dir**: DbContext sınıfının oluşturulacağı dizini belirtir.

dotnet ef dbcontext scaffold "Server=localhost;Database=GYSALES\_RDMS\_FULL;Trusted\_Connection=True;TrustServerCertificate=True" Microsoft.EntityFrameworkCore.SqlServer -c MyContext –context-dir Data -o Models

DERS 8: CODE FIRST

1. DbContext ucun – Microsoft.EntityFrameworkCore yuklenmelidir.
2. OnConfiguring
   * 1. PMS
        + Migration olusturmak:
        + add-migration [Migration Name]
        + Migrate etmek: update
        + update-database
        + Migration down etmek:
        + update-database [Migration Name]
        + Migration path:
        + add-migration [Migration Name] –OutputDir [Path]
        + Remove:
        + remove-migration
        + Migration listelemek:
        + get-migration
     2. Dotnet CLI
        + Migration olusturmak:
        + dotnet ef migrations add [Migration Name]
        + Migrate etmek: update
        + dotnet ef database update
        + Migration down etmek:
        + dotnet ef database update [Migration Name]
        + Migration path:
        + dotnet ef migrations add [Migration Name] –output-dir [Path]
        + Remove:
        + dotnet ef migrations remove
        + Migration listelemek:
        + dotnet ef migrations list

Kod ile migration etmek: ProductDbContext context = new();await context.Database.MigrateAsync();

DERS 9:

EN TEMEL ENTITY KURALI: Primary Key

Public int ID

Public int Id

Public int UserID

Public int UserId

DERS 10: Veri kaliciligi

1. context nesnesine ihtiyacimiz var
2. context.AddAsync();
3. context.DbSet.AddAsync();
4. await context.SaveChangesAsync();
5. Console.WriteLine(context.Entry(product4).State);//detached//added//unchanged//(=EntityState=)
6. Birden cox data elave etmek.

DERS 11: Data update

// deyisdirmek istediyimiz datani elde edek

Product product\_9 = await context.Products.FirstOrDefaultAsync(p => p.Id == 9);

product\_9!.Name = "deyisdirilmis Name9";

product\_9.Category = "deyisdirilmis kategoriya";

await context.SaveChangesAsync();

1. changetracker nedir?

//(context uzerinden gelen) obyektlerin takibinden sorumlu mexanizmdir

//delete ya update sorgularinin yaradilacagi anlasilir.

1. Takip edilmeyen nesnelerin guncellenmesi

Product product5 = new()

{

Id=5,//id elave etdim

Name = "name5 deyisdirildi takibsiz",

Category = "category5",

Description = "description5"

};

context.Products.Update(product5);

await context.SaveChangesAsync();

1. entitystate nedir?

DERS 12: Data remove

DERS 13: Veri kaliciligi

DERS 14: Veri kaliciligi

DERS 15: Veri kaliciligi

dotnet ef dbcontext scaffold "Server=localhost;Database=Northwind;Trusted\_Connection=True;TrustServerCertificate=True" Microsoft.EntityFrameworkCore.SqlServer -o ModelsNorthwind -c MyNorthWindContext -d

dotnet ef dbcontext scaffold "Server=localhost;Database=TaskDotnet;Trusted\_Connection=True;TrustServerCertificate=True" Microsoft.EntityFrameworkCore.SqlServer -o Models -c ExamDbContext -d

MyNorthWindContext context2 = new();

#region singleasync

//edilen sorguda sadece ve sadece tek bir data getirmek isteyirikse single ve ya singleordefault istifade edirik

//birden cox data gelirse ve ya hec data gelmirse (exception) verir

Urunler c\_product = await context2.Urunlers.SingleAsync(p => p.BirimFiyati == 20);

Console.WriteLine(c\_product.BirimdekiMiktar);

//tek data gelende

//hec data gelmeyende

//cox data gelende

#endregion

#region singleordefaultasync

//birden cox data gelirse (exception) atir, hec data gelmirse (null) verir

var c\_productDef = await context2.Urunlers.SingleOrDefaultAsync(p => p.BirimFiyati == 300);

Console.WriteLine($"null: {c\_productDef}");

#endregion

#region firstasync

#endregion

#region firstordefaultasync

#endregion

#region qarsilasdirma

#endregion

#region findasync

#endregion

#region findasync ile digerleri qarsilasdirilmasi

#endregion

#region lastasync

#endregion

#region lastordefaultasync

#endregion