Michael G. Inso 2nd year ACCP-HCSE

Entity Framework Core can serve as an object-relational mapper, enabling .NET developers to work with a database using .NET objects, and eliminating the need for most of the data-access code they usually need to write.

Database Design and Developement

Entity Framework Core101

Table of Contents

**Table of Contents1-3**

**Table of Figures4-6**

**Acknowledgement7**

**Introduction8**

**Task 1: Introduction 9**

1. Database Structure 9
2. Entity Framework core NuGet10

**Task 2: Api11**

1. .api 11
2. launchSettings.json 12
3. ValuesController.cs 13
4. Appsettings.json 13
5. Appsettings.Development.json 13
6. Program.cs 13
7. Startup.cs 14

**Task 3: Data15**

1. .Data 15
2. EmployeeCommand.cs 15
3. CompanyConfiguration.cs 16
4. ContractorConfiguration.cs 16
5. DepartmentConfiguration.cs 17
6. DepartmentContractorConfiguration.cs 17
7. DocumentTypeConfiguration.cs 18
8. EmployeeConfiguration.cs 18
9. EmployeeDocumentConfiguration.cs 19
10. EntityConfiguration.cs 19
11. FileDescriptionConfiguration.cs 19
12. VendorConfiguration.cs 20
13. \_InitialCreate.cs 20
14. \_InitialCreate(2).cs21
15. \_InitialCreate(3).cs 22
16. \_InitialCreate(4).cs 23
17. DbContextModelSnapshot.cs 24
18. DbContextModelSnapshot(2).cs 25
19. DbContextModelSnapshot(3).cs 26
20. DbContextModelSnapshot(4).cs 27
21. DbContextModelSnapshot(5).cs 28
22. DbContextModelSnapshot(6).cs 29
23. DbContextModelSnapshot(7).cs 30
24. EmployeeProjection.cs 31
25. EmployeeQuery.cs 32
26. AssemblyInfo.cs 32
27. DataConstants.cs 33
28. DbContext.cs 33
29. DbContext(2).cs 34

**Task 4: Models35**

1. .Models 35
2. Company.cs 35
3. Contractor.cs 35
4. Department.cs 36
5. DeparmentContractor36
6. DocumentType.cs 37
7. Employee.cs 37
8. EmployeeDocument.cs 38
9. Entity.cs 38
10. FileDescription.cs 39
11. Vendor.cs 39
12. Address.cs 40
13. PersonName.cs 41
14. ValueObject.cs 42

**Task 5: Sql43**

1. EmployeeDocuments.sql 43
2. EmployeeProfileImages.sql 43
3. MigrationsHistory.sql 43
4. Company.sql 44
5. Contractor.sql 45
6. Department.sql 46
7. DepartmentContractor.sql 47
8. DocumentType.sql 48
9. Employee.sql49
10. Employee(2).sql49
11. EmployeeDocument.sql 50
12. FileDescription.sql 51
13. FileType.sql 52
14. Vendor.sql 53

**Task 6: Tests54**

1. .Tests 54
2. DataGenerator.cs 55
3. DataGenerator(2).cs 56
4. DataGenerator(3).cs 57
5. DataGenerator(4).cs 58
6. EmployeeeTests.cs 58
7. InMemoryDbTests(.cs 59
8. InMemoryDbTests(2).cs 60
9. InMemoryDbTests(3).cs 61
10. ModelFakes.cs 61
11. ValueObjectTests.cs 62
12. ValueObjectTests(2).cs 63

**Task 7: Summary64**

1. Summary64

References 64

Bibliography 64

Table of Figures

Figure 1: .api 11

Figure 2: launchSettings.json 12

Figure 3: ValuesController.cs 13

Figure 4: Appsettings.json 13

Figure 5: Appsettings.Development.json 13

Figure 6: Program.cs 13

Figure 7: Startup.cs 14

Figure 8: .Data 15

Figure 9: EmployeeCommand.cs 15

Figure 10: CompanyConfiguration.cs 16

Figure 11: ContractorConfiguration.cs 16

Figure 12: DepartmentConfiguration.cs 17

Figure 13: DepartmentContractorConfiguration.cs 17

Figure 14: DocumentTypeConfiguration.cs 18

Figure 15: EmployeeConfiguration.cs 18

Figure 16: EmployeeDocumentConfiguration.cs 19

Figure 17: EntityConfiguration.cs 19

Figure 18: FileDescriptionConfiguration.cs 19

Figure 19: VendorConfiguration.cs 20

Figure 20: \_InitialCreate.cs 20

Figure 21: \_InitialCreate(2).cs21

Figure 22: \_InitialCreate(3).cs 22

Figure 23: \_InitialCreate(4).cs 23

Figure 24: DbContextModelSnapshot.cs 24

Figure 25: DbContextModelSnapshot(2).cs 25

Figure 26: DbContextModelSnapshot(3).cs 26

Figure 27: DbContextModelSnapshot(4).cs 27

Figure 28: DbContextModelSnapshot(5).cs 28

Figure 29: DbContextModelSnapshot(6).cs 29

Figure 30: DbContextModelSnapshot(7).cs 30

Figure 31: EmployeeProjection.cs 31

Figure 32: EmployeeQuery.cs 32

Figure 33: AssemblyInfo.cs 32

Figure 34: DataConstants.cs 33

Figure 35: DbContext.cs 33

Figure 36: DbContext(2).cs 34

Figure 37:.Models 35

Figure 38: Company.cs 35

Figure 39: Contractor.cs 35

Figure 40: Department.cs 36

Figure 41: DeparmentContractor36

Figure 42: DocumentType.cs 37

Figure 43: Employee.cs 37

Figure 44: EmployeeDocument.cs 38

Figure 45: Entity.cs 38

Figure 46: FileDescription.cs 39

Figure 47: Vendor.cs 39

Figure 48: Address.cs 40

Figure 49: PersonName.cs 41

Figure 50: ValueObject.cs 42

Figure 51: EmployeeDocuments.sql 43

Figure 52: EmployeeProfileImages.sql 43

Figure 53: MigrationsHistory.sql 43

Figure 54: Company.sql 44

Figure 55: Contractor.sql 45

Figure 56: Department.sql 46

Figure 57: DepartmentContractor.sql 47

Figure 58: DocumentType.sql 48

Figure 59: Employee.sql49

Figure 60: Employee(2).sql49

Figure 61: EmployeeDocument.sql 50

Figure 62: FileDescription.sql 51

Figure 63: FileType.sql 52

Figure 64: Vendor.sql 53

Figure 65:.Tests 54

Figure 66: DataGenerator.cs 55

Figure 67: DataGenerator(2).cs 56

Figure 68: DataGenerator(3).cs 57

Figure 69: DataGenerator(4).cs 58

Figure 70: EmployeeeTests.cs 58

Figure 71: InMemoryDbTests.cs 59

Figure 72: InMemoryDbTests(2).cs 60

Figure 73: InMemoryDbTests(3).cs 61

Figure 74: ModelFakes.cs 61

Figure 75: ValueObjectTests.cs 62

Figure 76: ValueObjectTests(2).cs 63

**Acknowledgement**

This solo project would not have been possible with a lot of dedicated special individuals putting their heart, trust and faith in me helping me to persevere and be persistent.

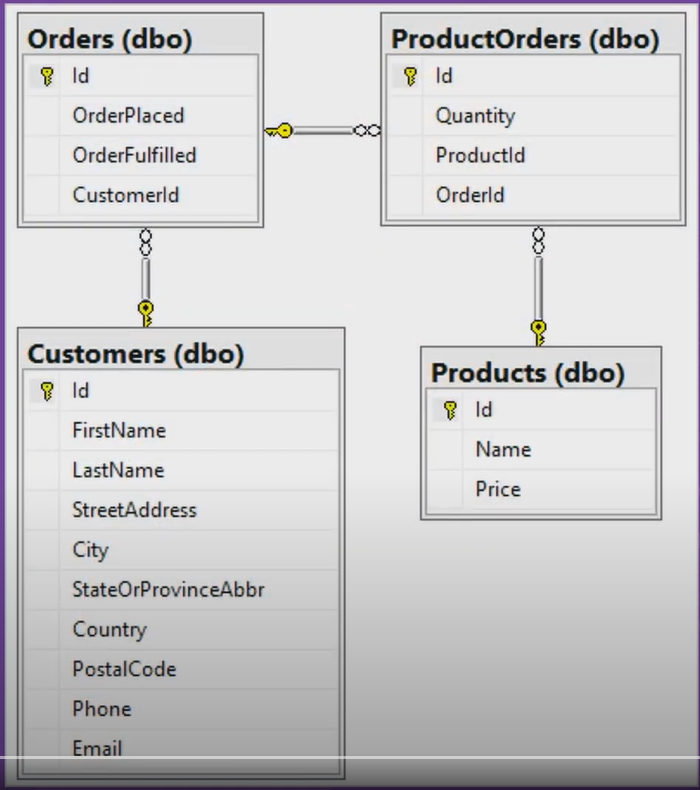
I would like to express my special thanks of gratitude to Aptech Qatar Education Centre who gave the amazing opportunity to do this strenuous project on the topic of Database Design and Development for 2nd year ACCP-HSCE, which also helped me in doing a lot of exploration and investigation and I acquired so many new knowledge and skills about Microsoft .Net core and Entity Framework core database design and development.

Secondly, I would also like to thank my family, friends and relatives who still give me assistance, sponsorship, and accommodation even I’m already a capable adult they helped me a lot in completing this project within the limited duration of time.

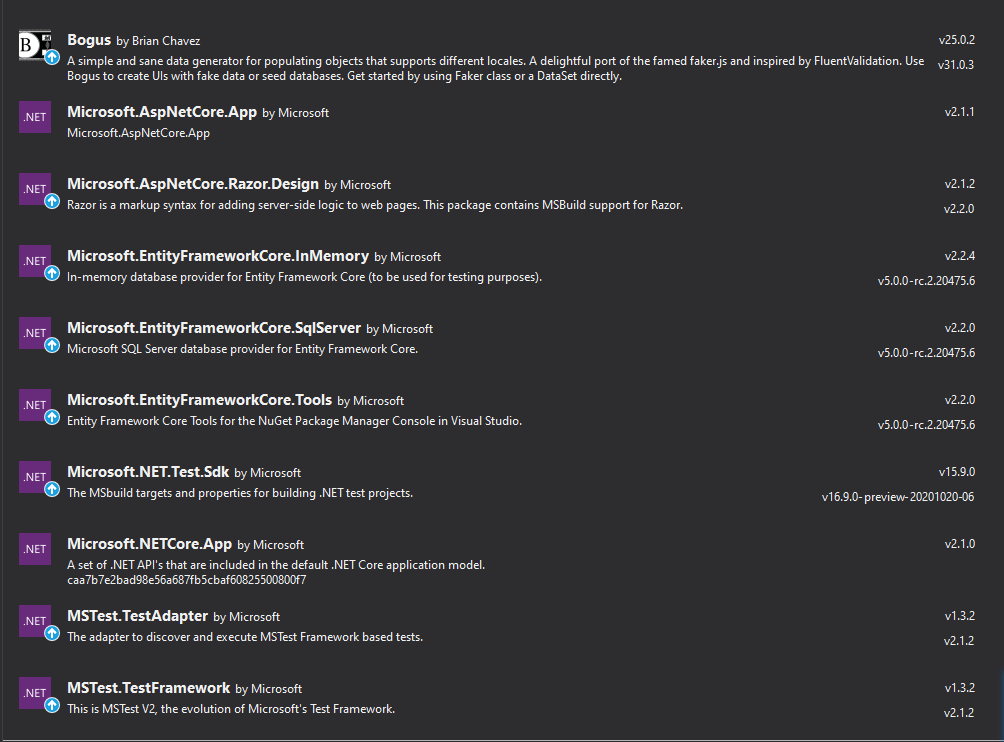
**Introduction**

I’m going to develop and design a database with the help of Entity Framework Core developed by Microsoft in 2008. It can serve as an object-relational mapper, enabling .NET developers to work with a database using .NET objects, and eliminating the need for most of the data-access code they usually need to write. This project would use the structures of a company having employees, departments, and contractors, etc. in its database. This project will include the use of SQL, C#, and JSON, etc.

**Task 1: Introduction**

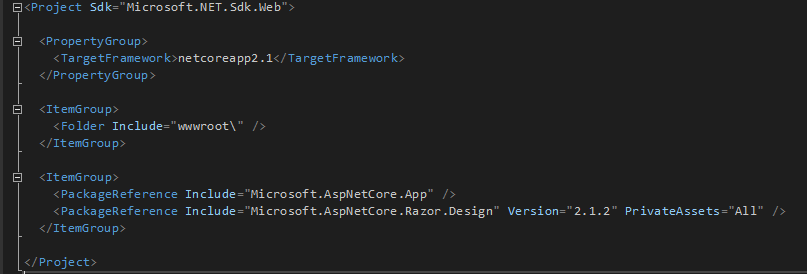
****

**Figure 1: a.) Example of Database Design**

****

**Figure 2: b.) NuGet Package**

**Task 2: Api**

****

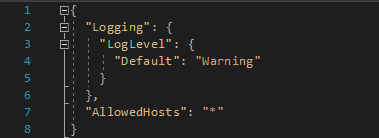
**Figure 3: a.) .api**

****

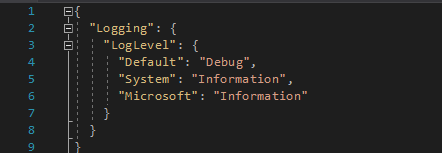
**Figure 4: b.) launchSettings.json**

****

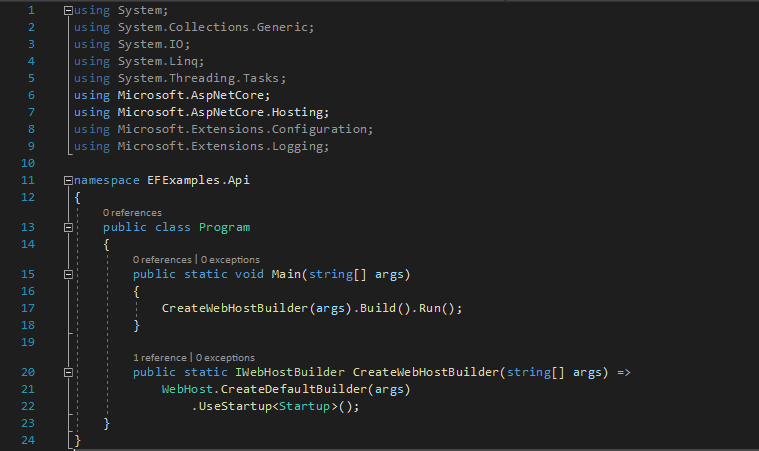
**Figure 5: c.) ValuesController.cs**

****

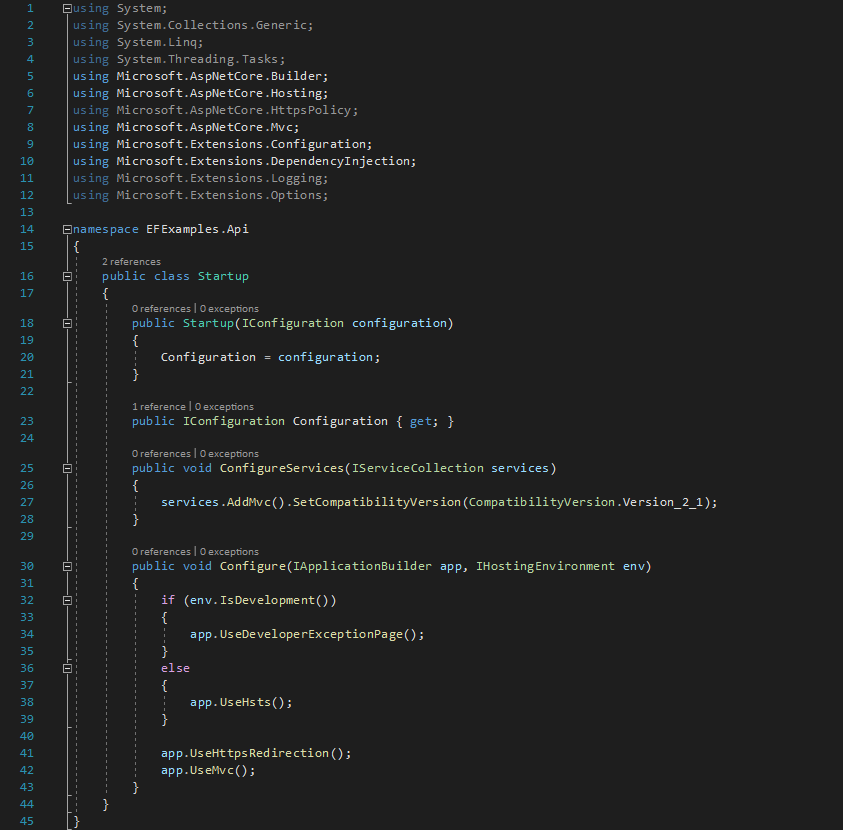
**Figure 6: d.) appsettings.json**

****

**Figure 7: e.) appsettings.Development.json**

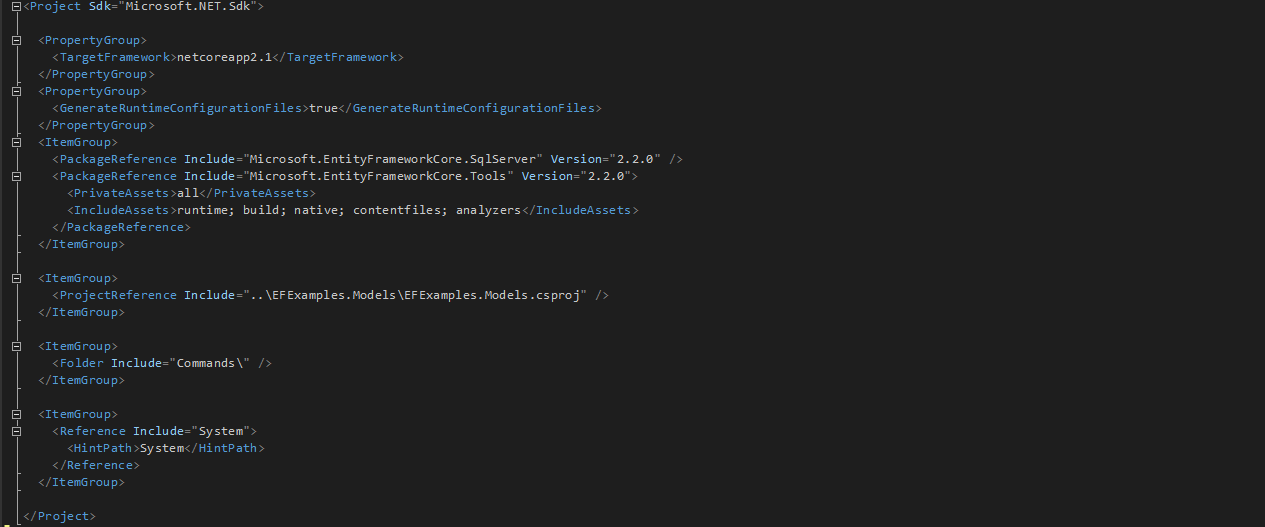
****

**Figure 8: f.) Program.cs**

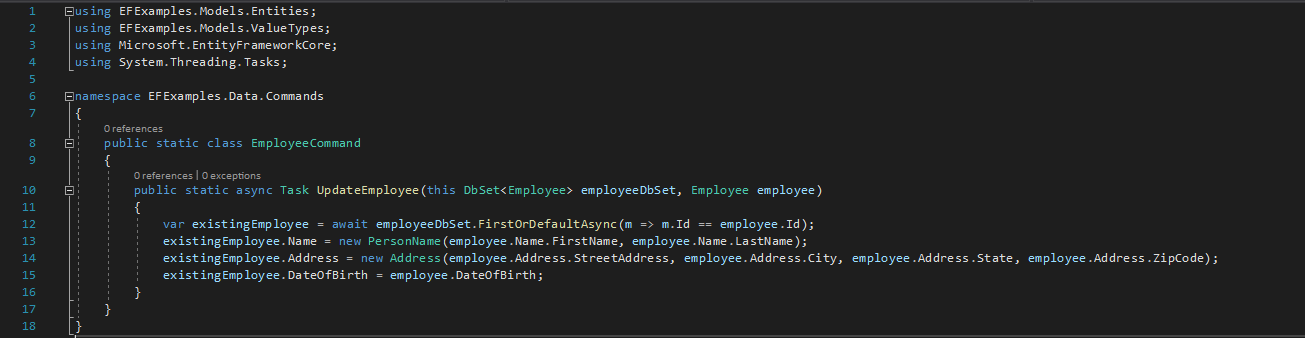
****

**Figure 9: g.) Starup.cs**

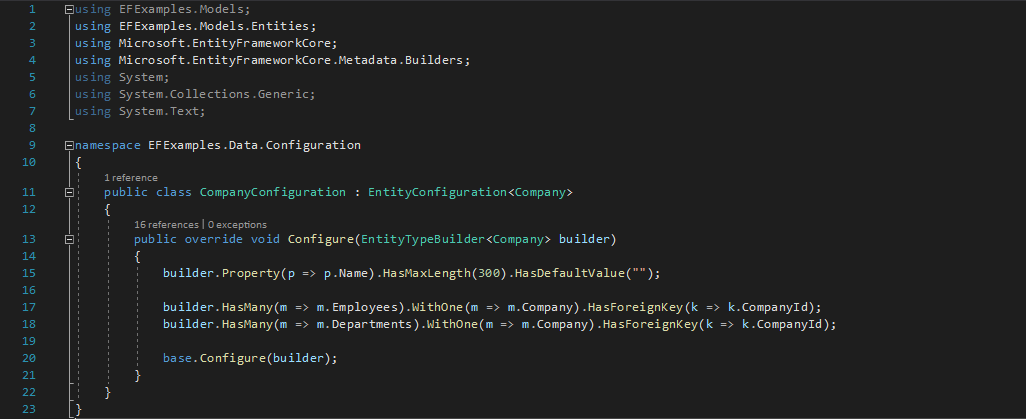
**Task 3: Data**

****

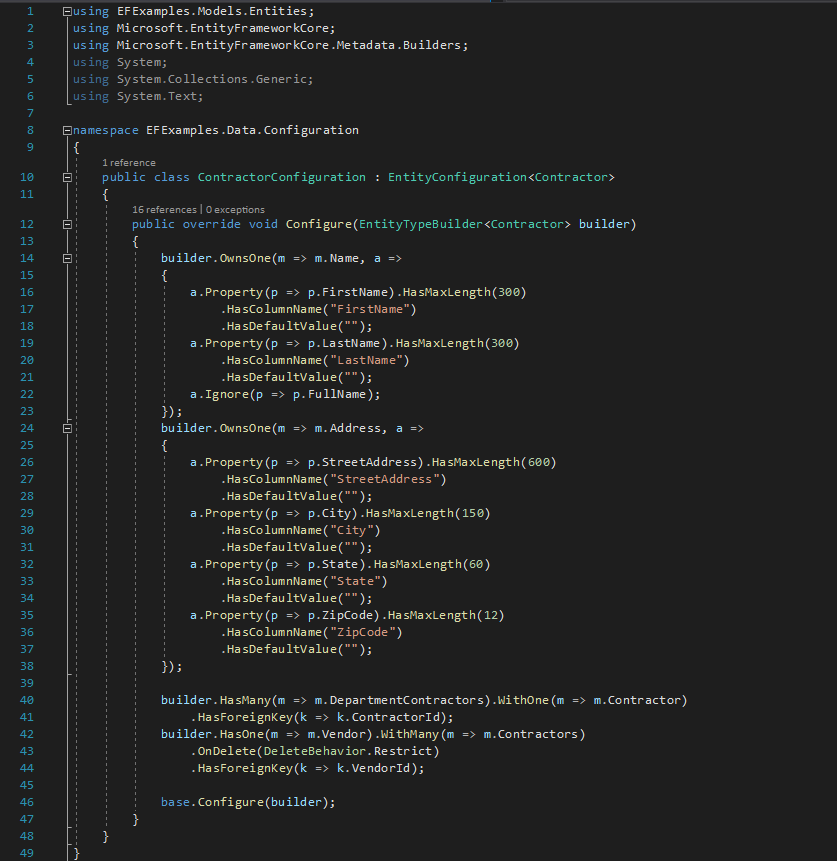
**Figure 10: a.) .Data**

****

**Figure 11: b.) EmployeeCommand.cs**

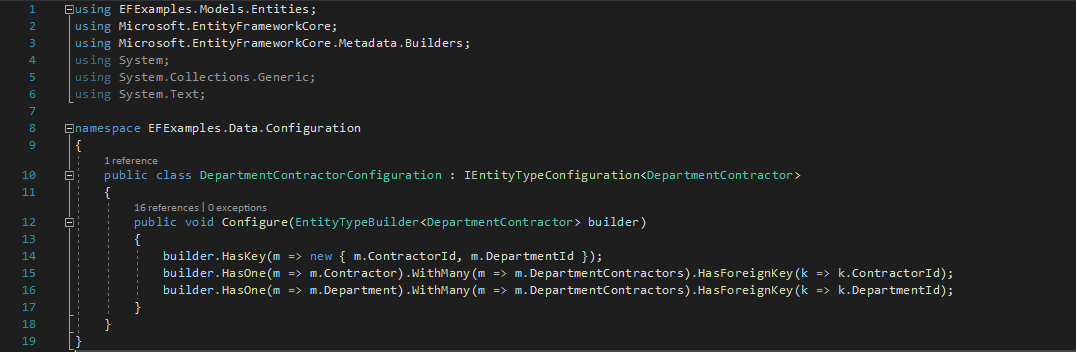
****

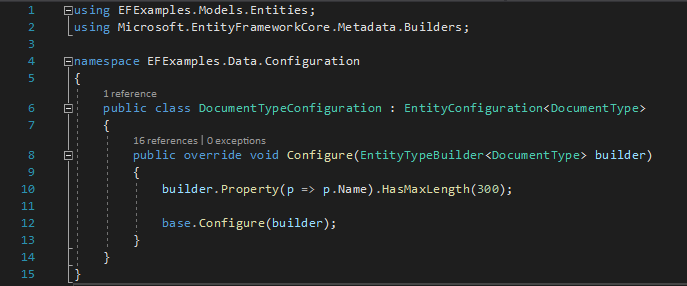
**Figure 12: c.) CompanyConfiguration.cs**



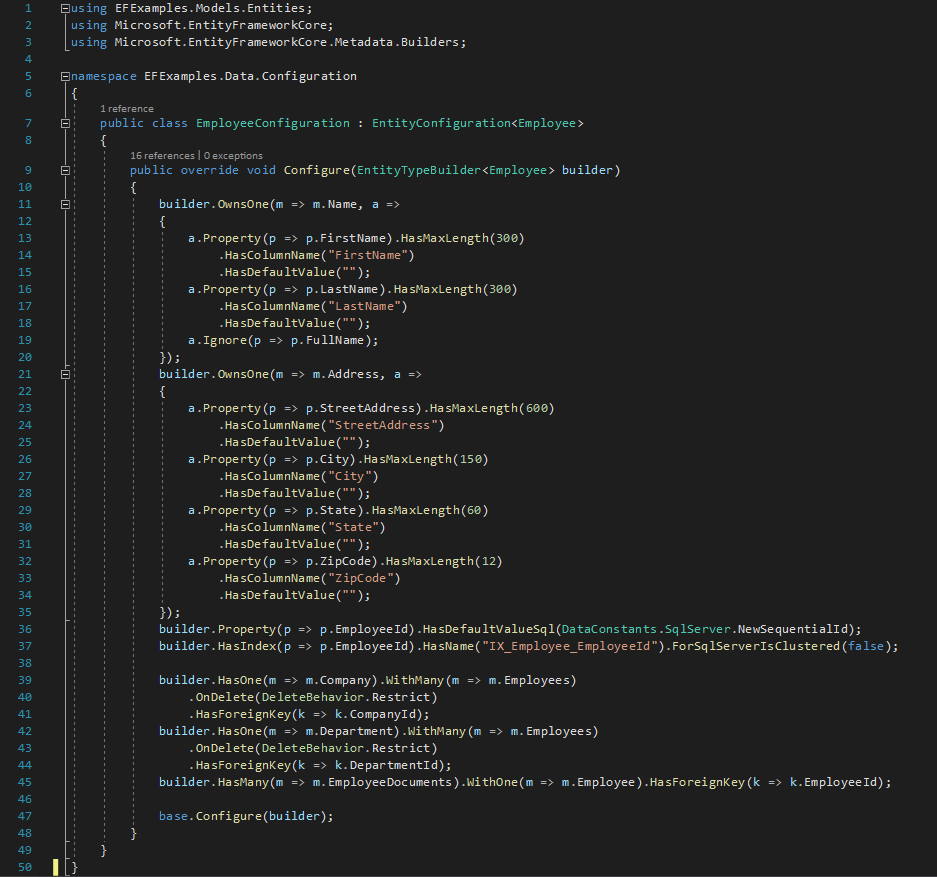
**Figure 13: d.) ContractorConfiguration.cs**



**Figure 14: e.) DepartmentConfiguration.cs**

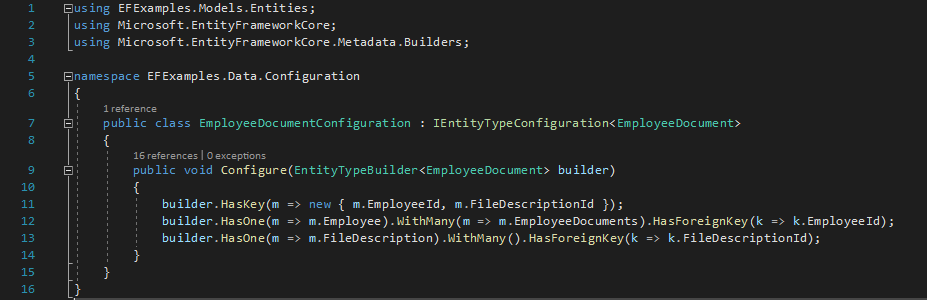
**Figure 14: f.) DepartmentContractorConfiguration.cs**

**Figure 15: g.) DocumentTypeConfiguration.cs**

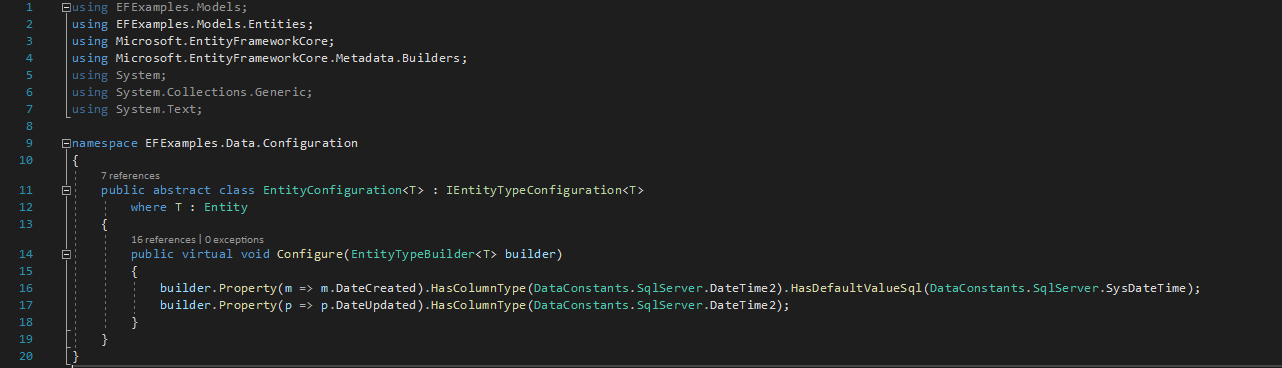
****

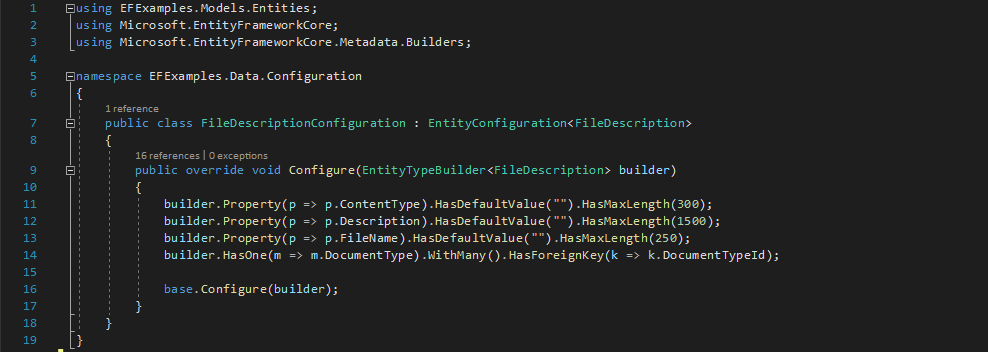
**Figure 16: i.) EmployeeConfiguration.cs**

**Figure 16: h.) EmployeeConfiguration.cs**

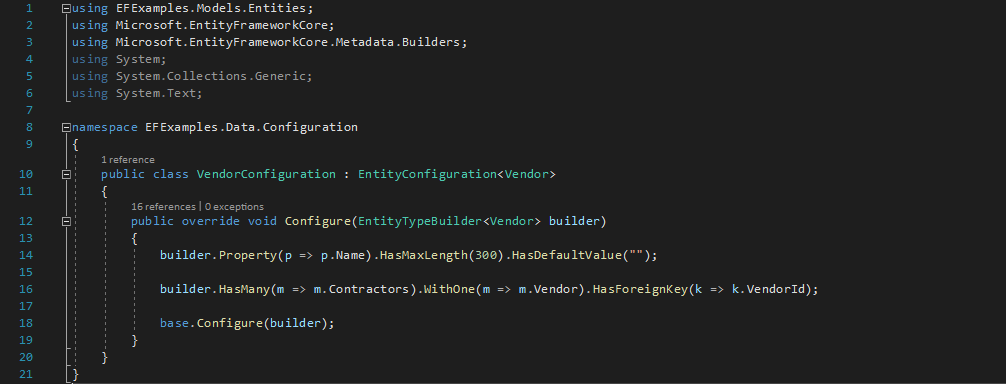


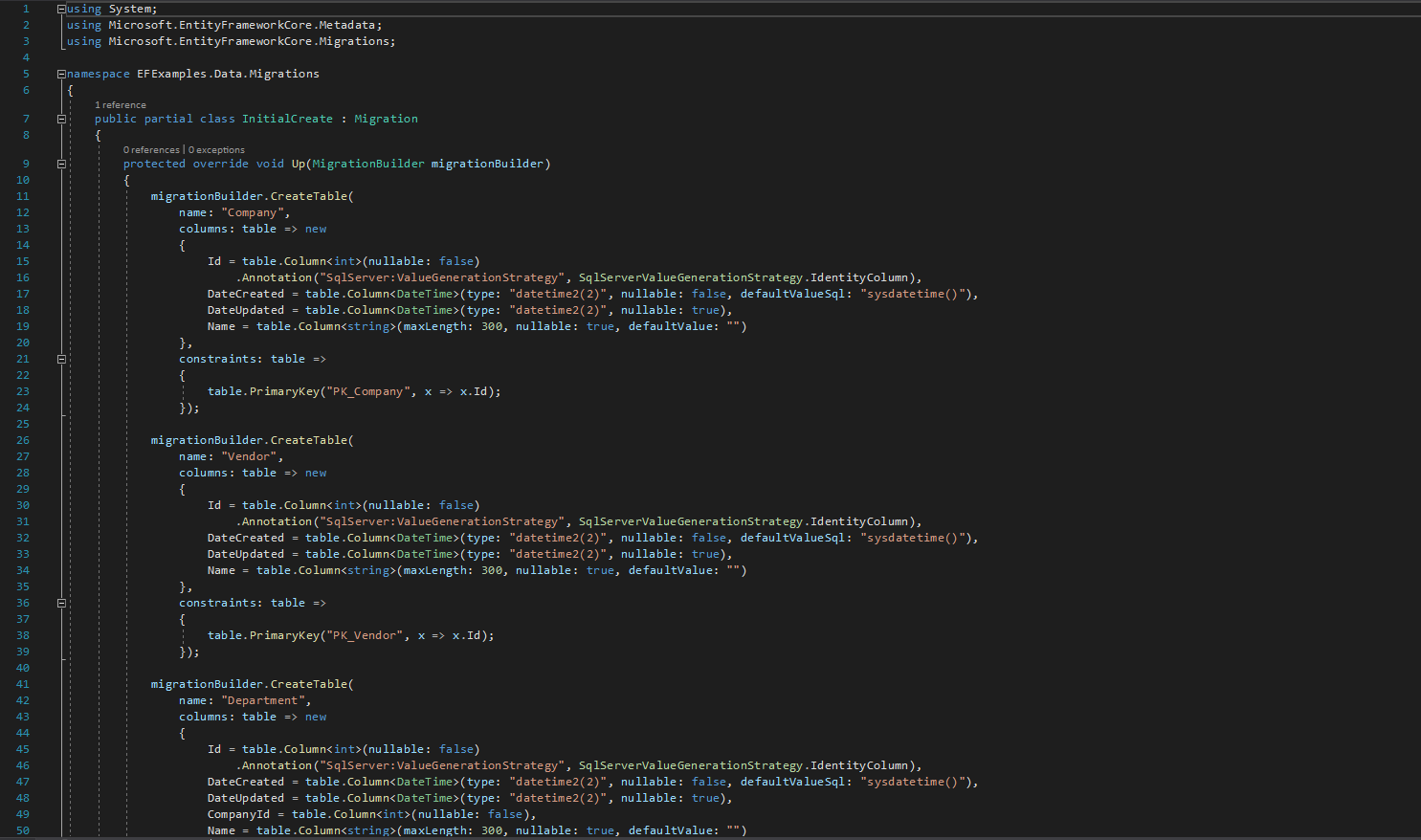
**Figure 17: i.) EmployeeDocumentConfiguration.cs**



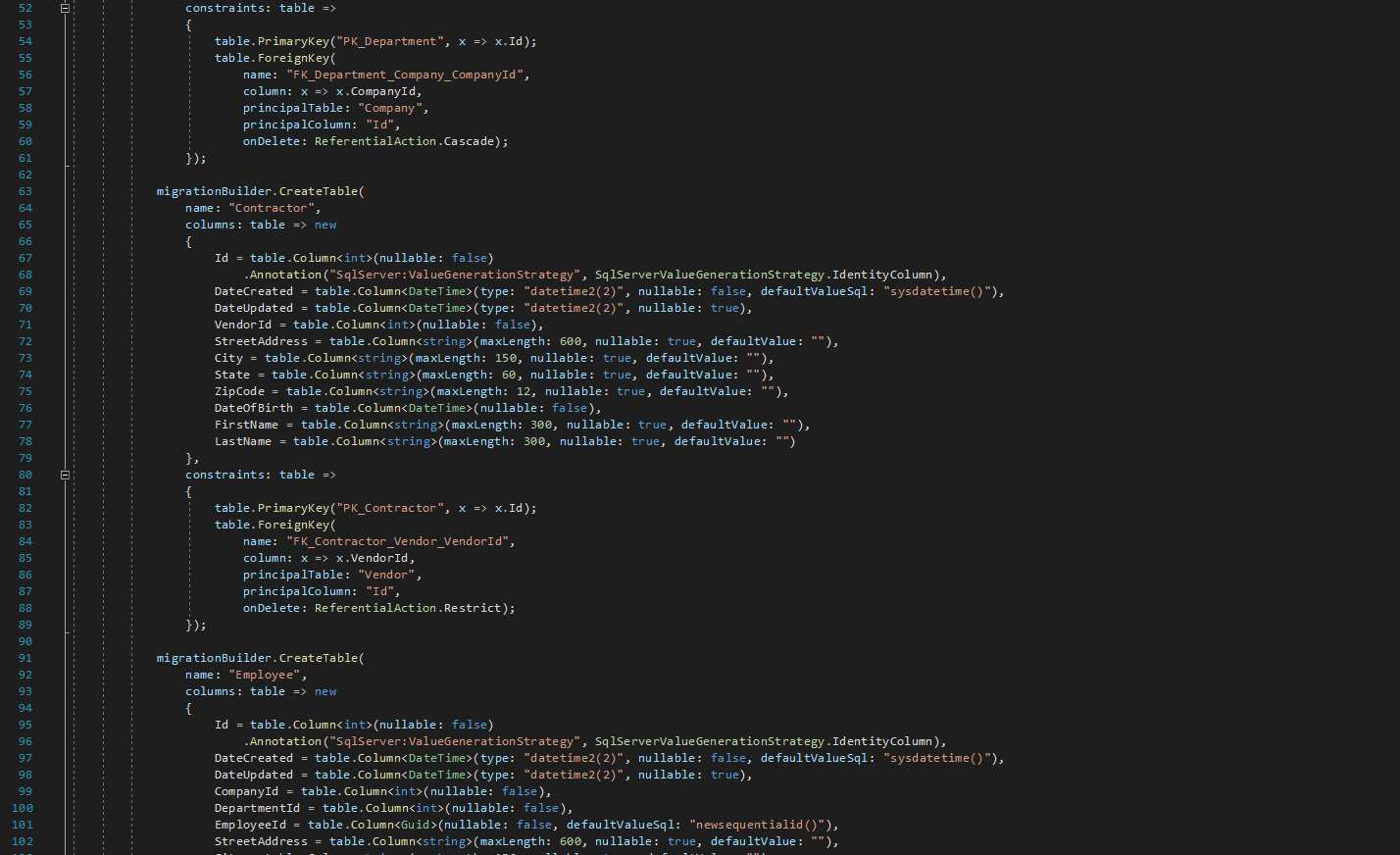
 **Figure 18: j.) EntityConfiguration.cs**

**Figure 19: k.) FileDescriptionConfiguration.cs**

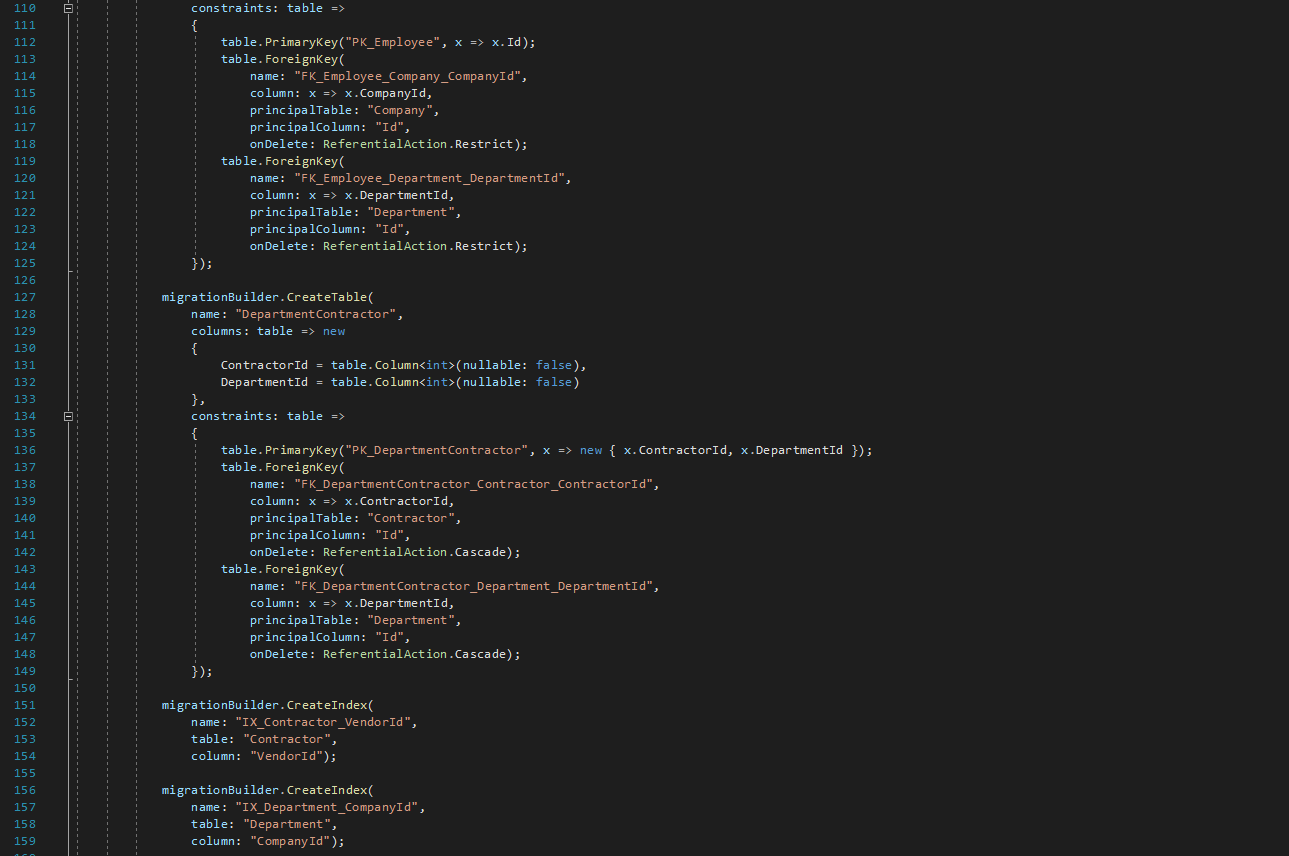


 **Figure 20: l.) VendorConfiguration.cs**

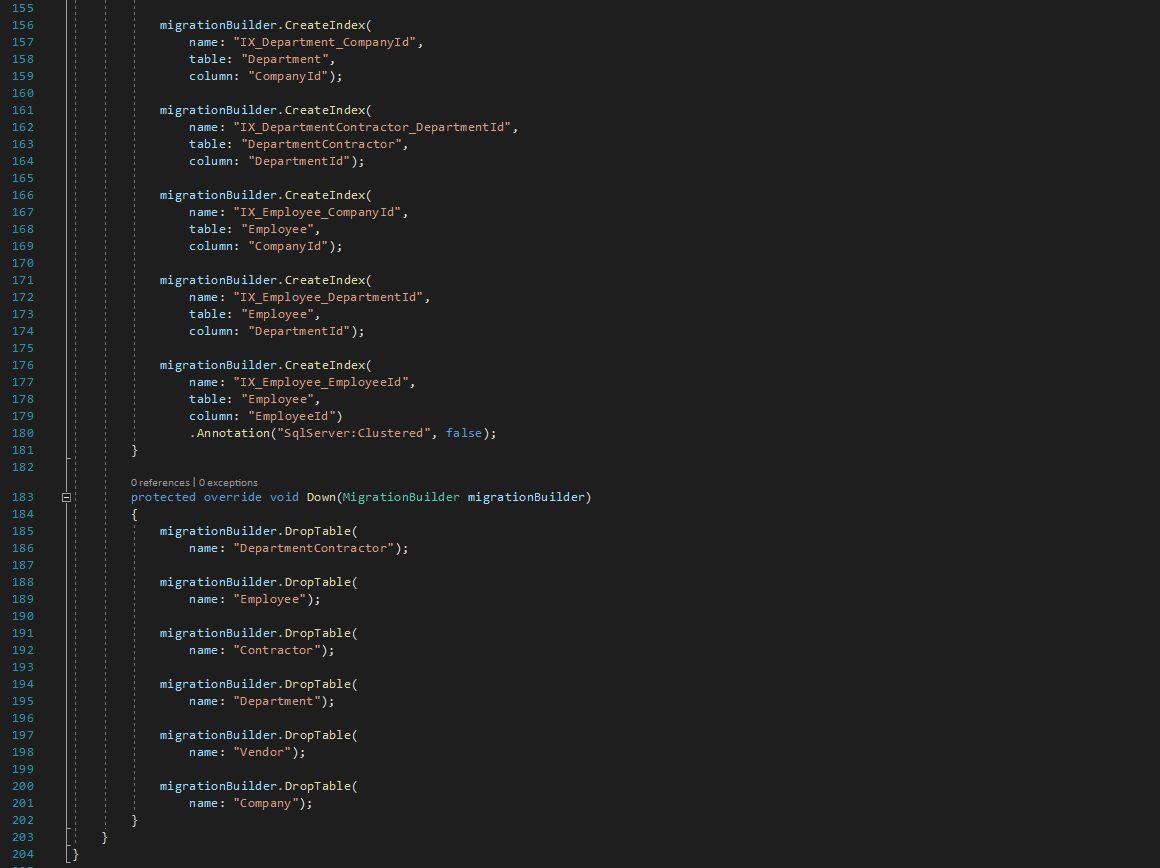
**Figure 21: m.) \_InitialCreate.cs**



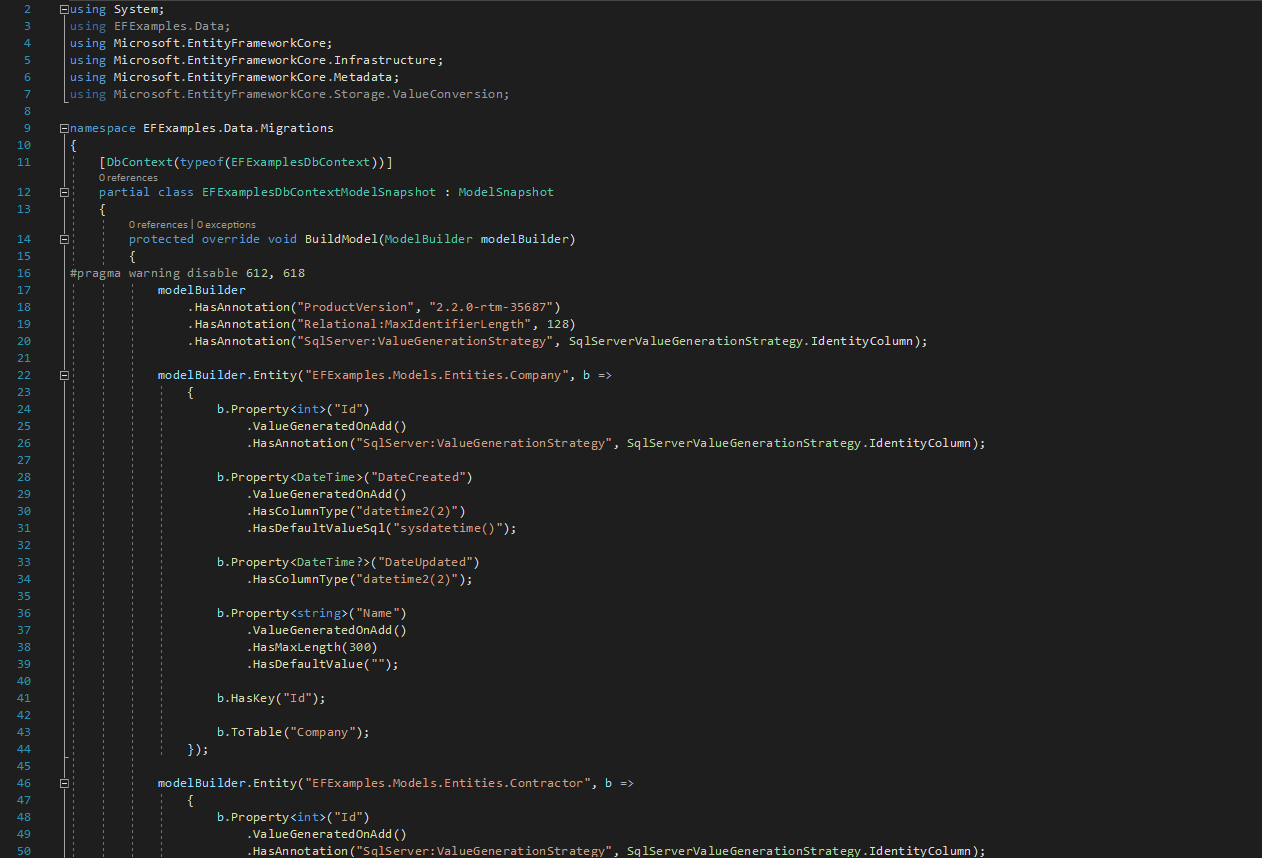
**Figure 22: n.) \_InitialCreate(2).cs**



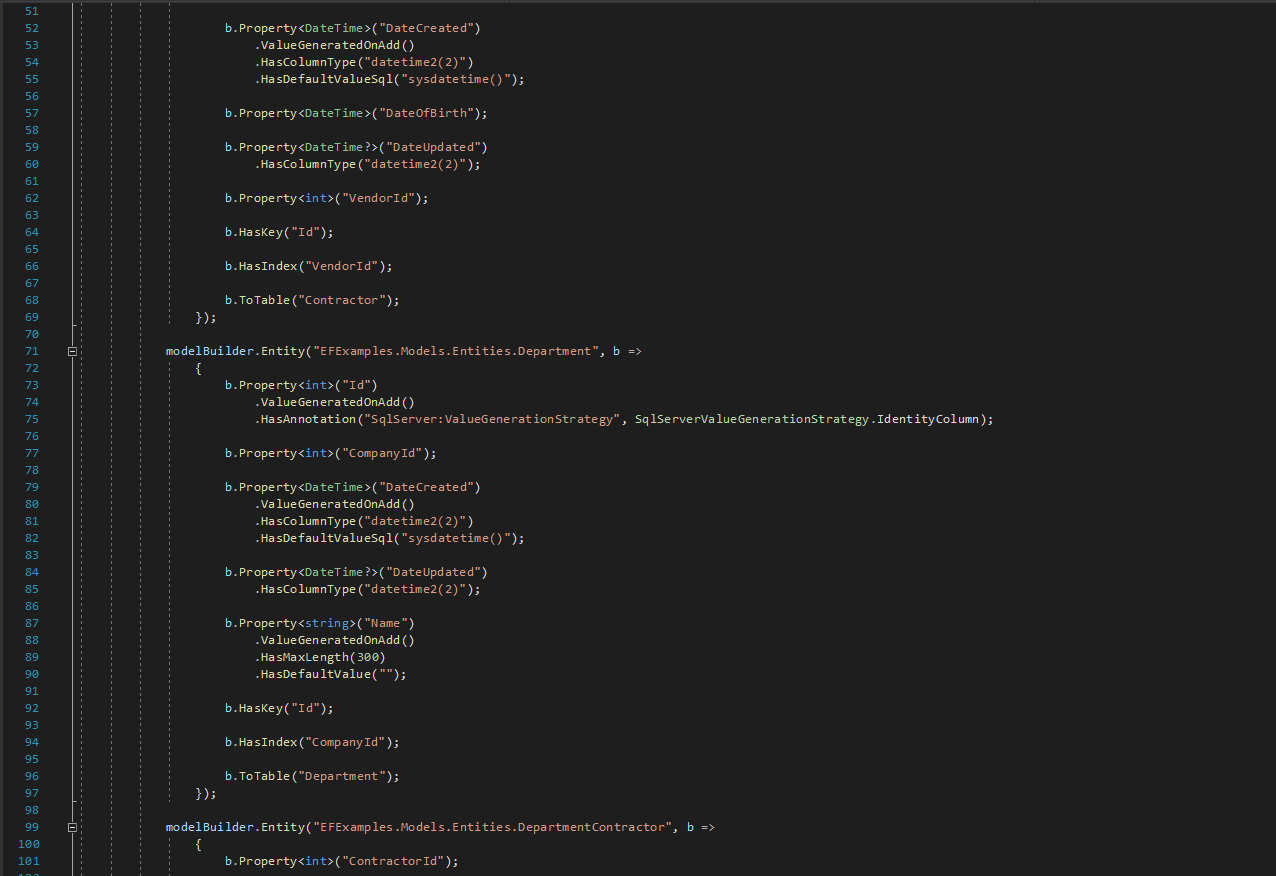
**Figure 23: o.) \_InitialCreate(3).cs**



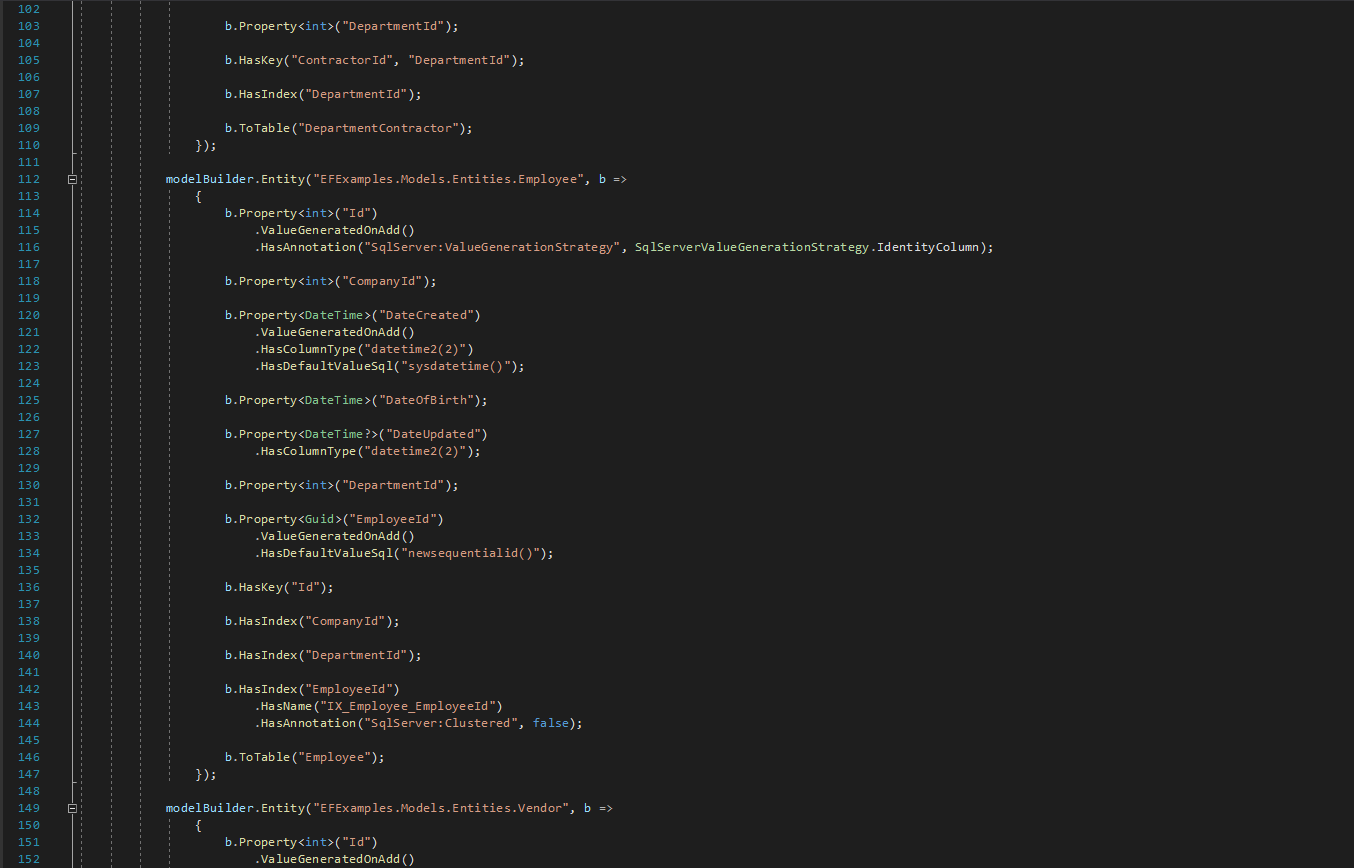
**Figure 23: p.) \_InitialCreate(4).cs**



**Figure 24: q.) DbContextModelSnapshot.cs**

****

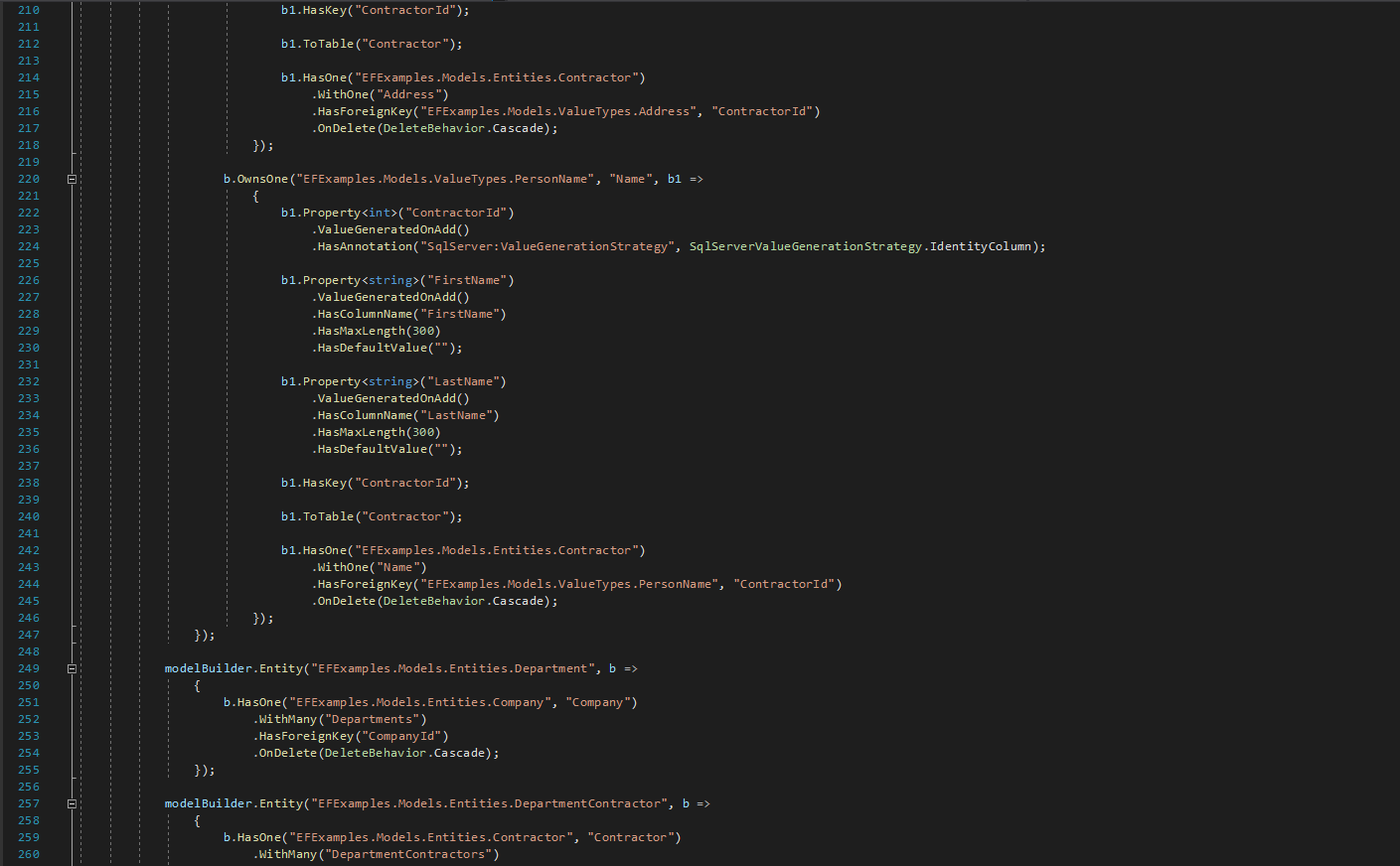
**Figure 25: r.) DbContextModelSnapshot(2).cs**



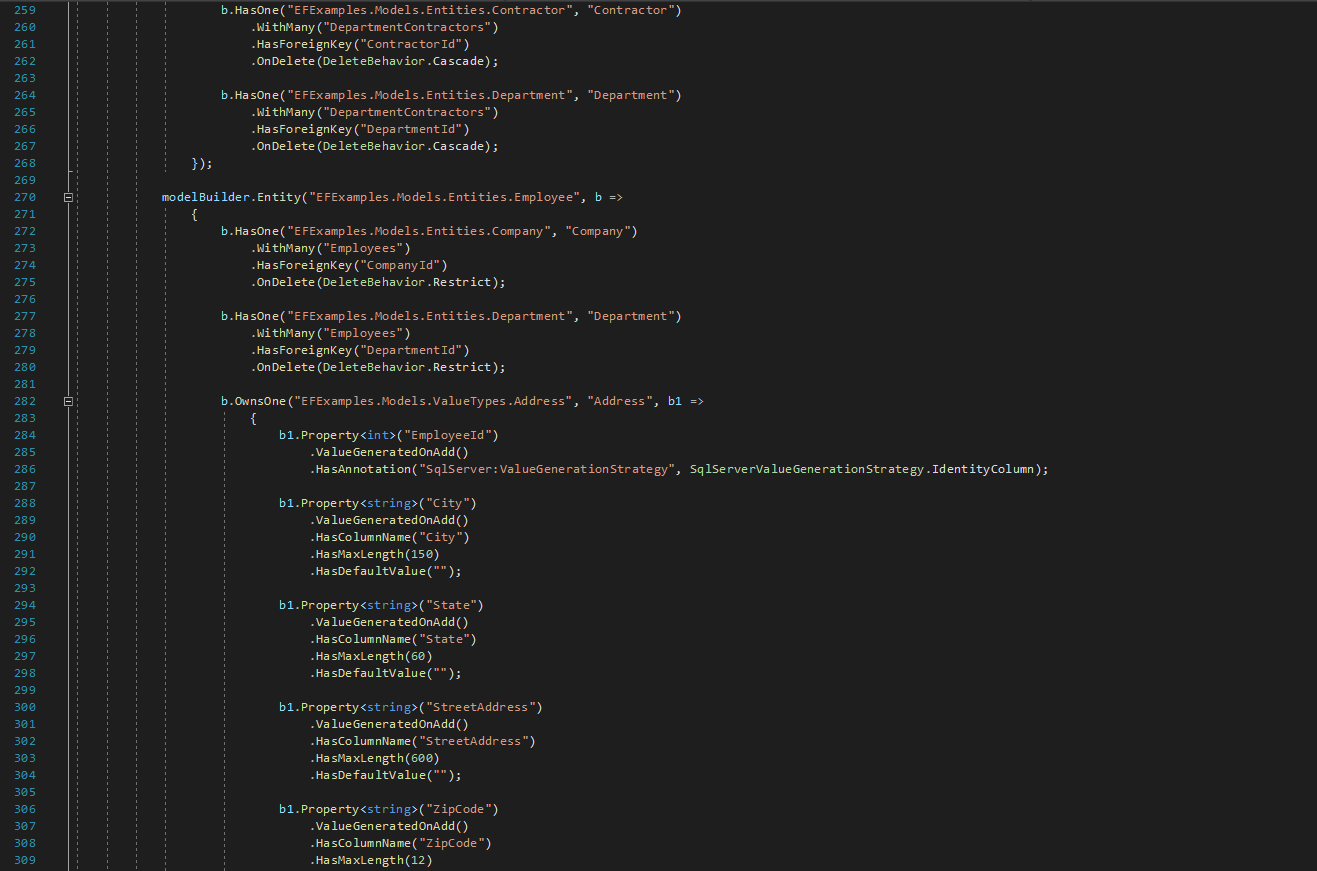
**Figure 26: s.) DbContextModelSnapshot(3).cs**



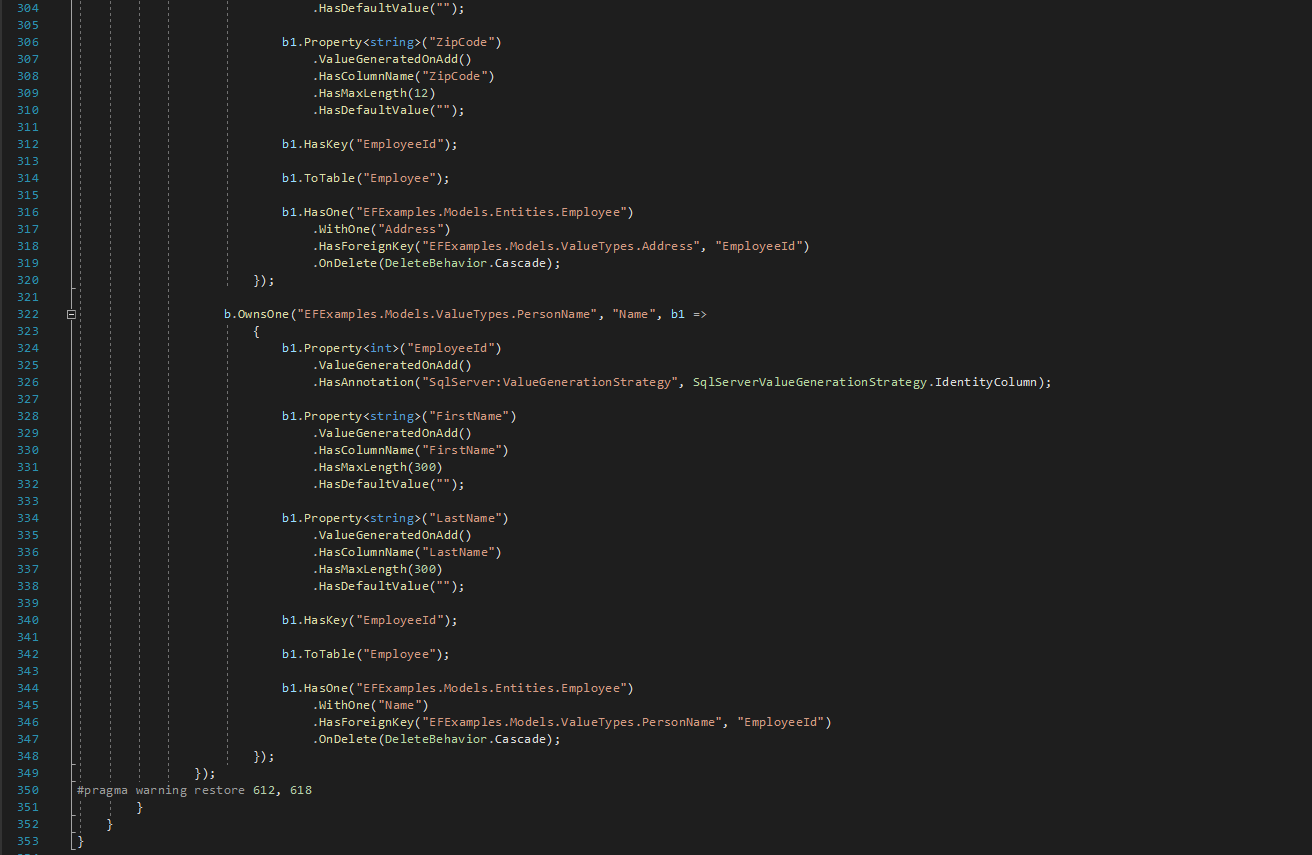
**Figure 27: t.) DbContextModelSnapshot(4).cs**

****

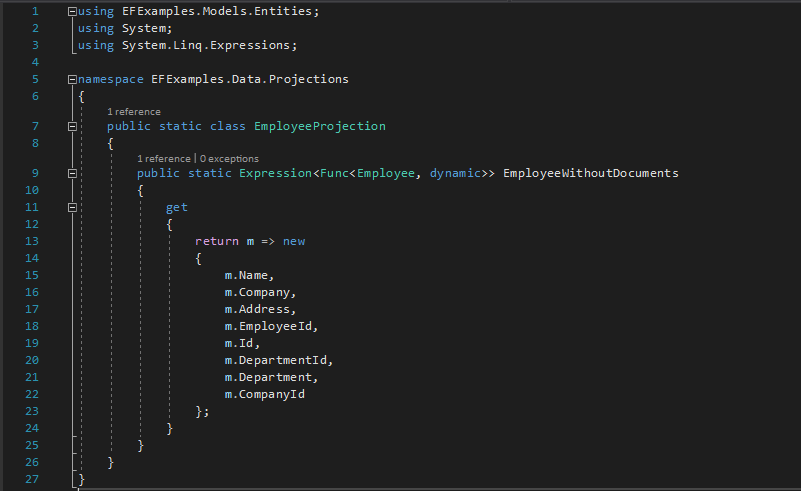
**Figure 28: u.) DbContextModelSnapshot(5).cs**



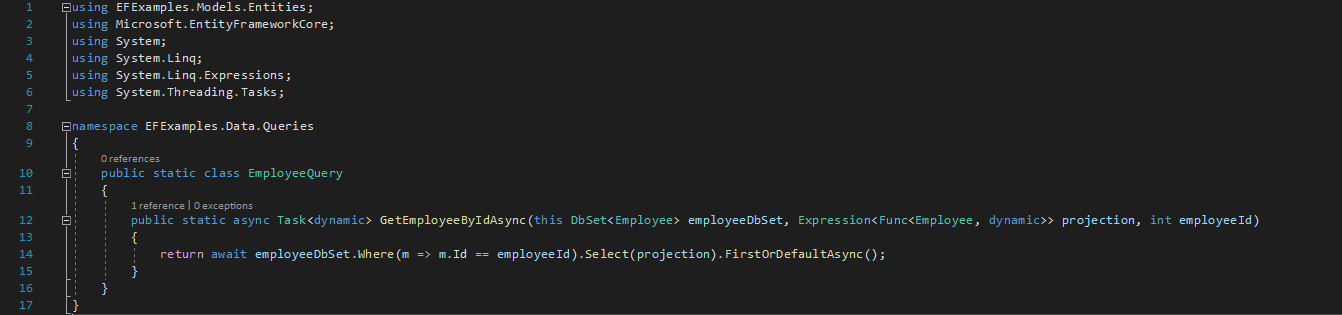
**Figure 29: v.) DbContextModelSnapshot(6).cs**



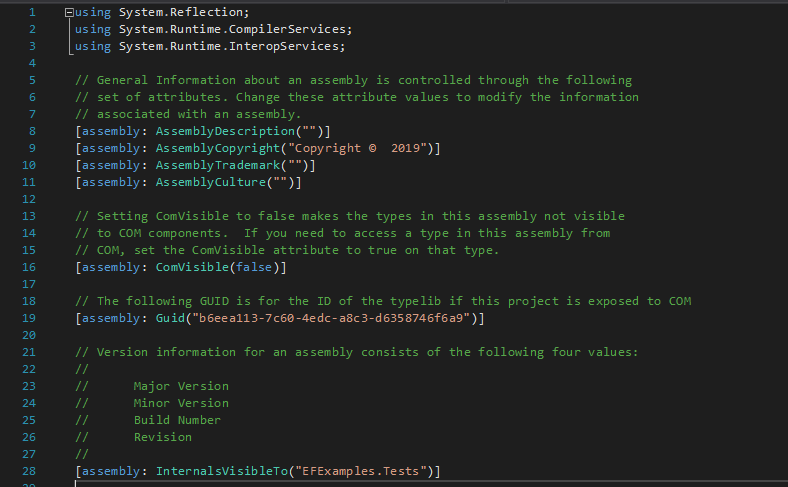
**Figure 30: w.) DbContextModelSnapshot(7).cs**



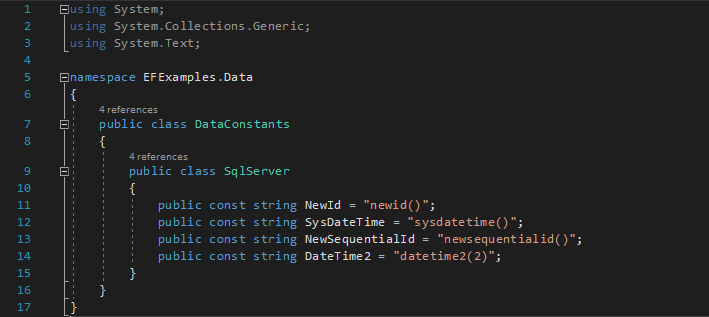
**Figure 31: x.) EmployeeProjection.cs**

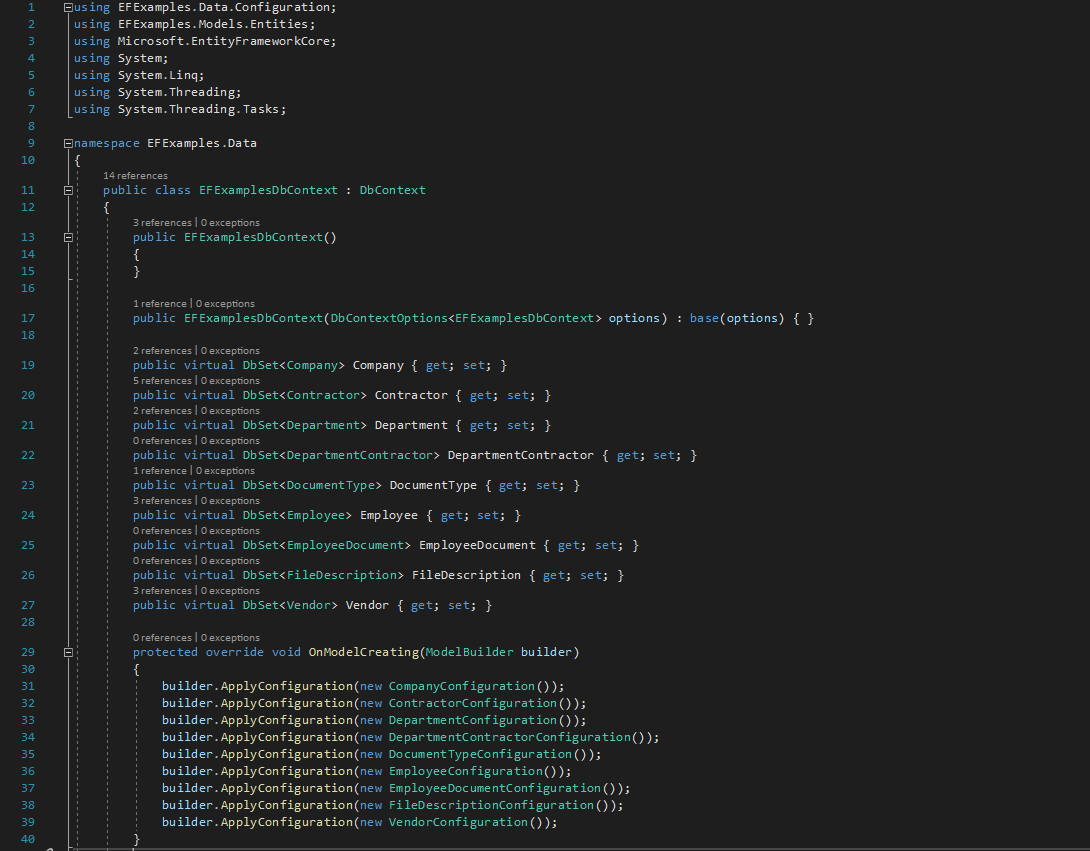


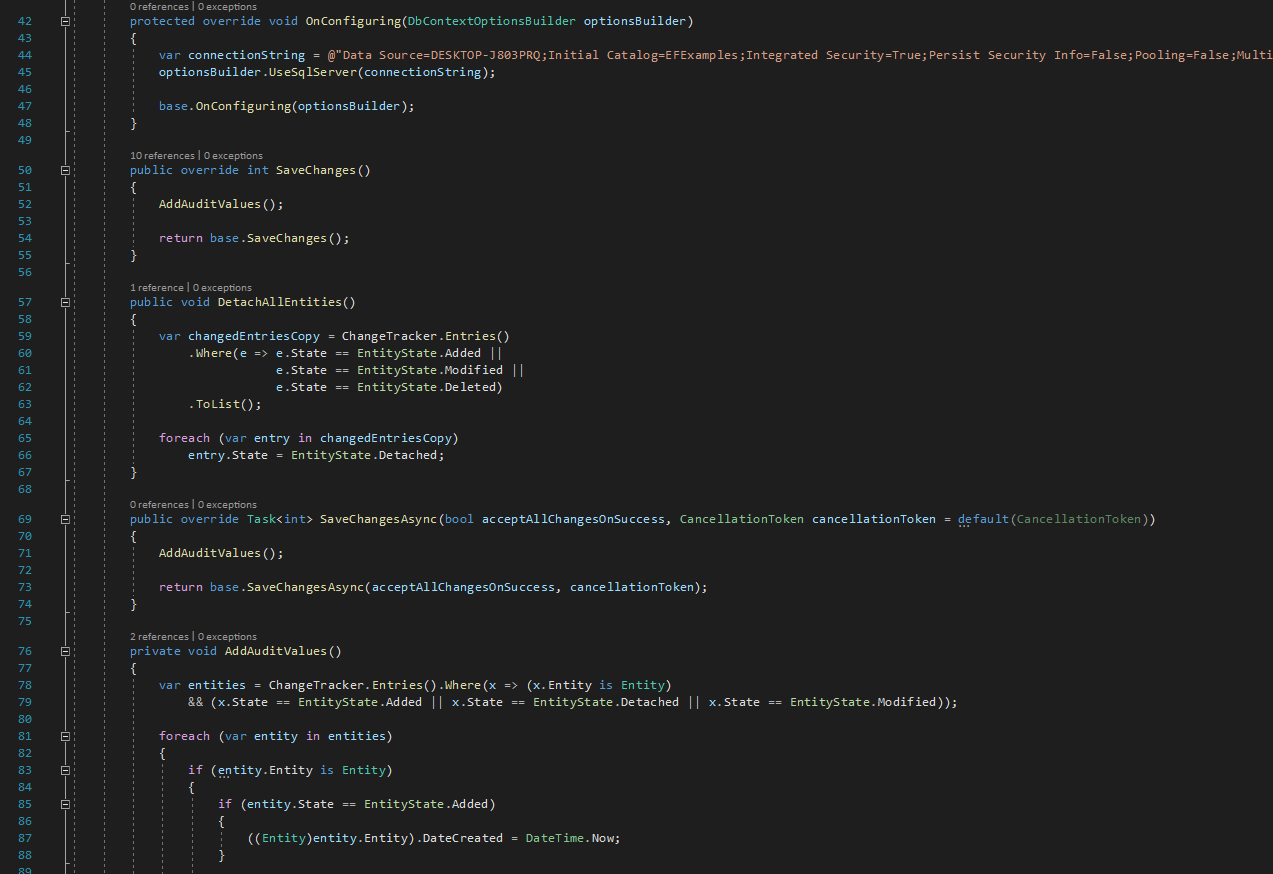
**Figure 32: u.) EmployeeQuery.cs**



**Figure 33: y.) AssemblyInfo.cs**

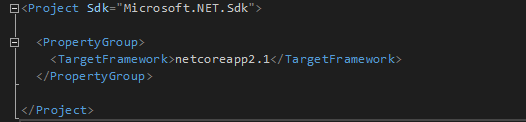


**Figure 34: z.) DataConstants.cs Figure 35: aa.) DbContext.cs**

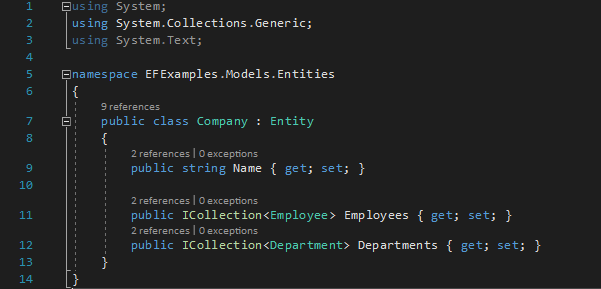


**Figure 36: ab.) DbContext(2).cs**

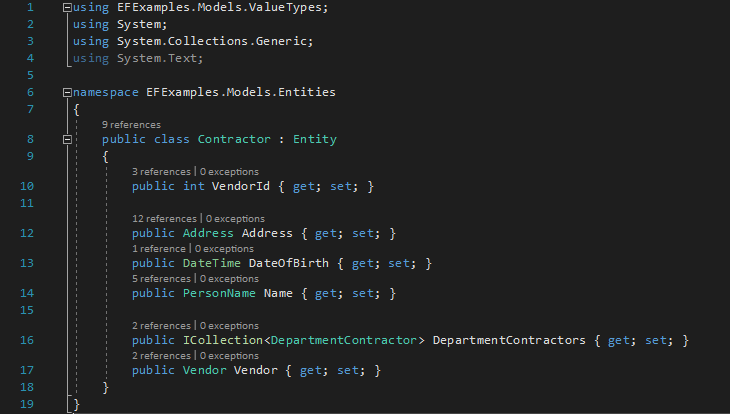
**Task 4: Models**

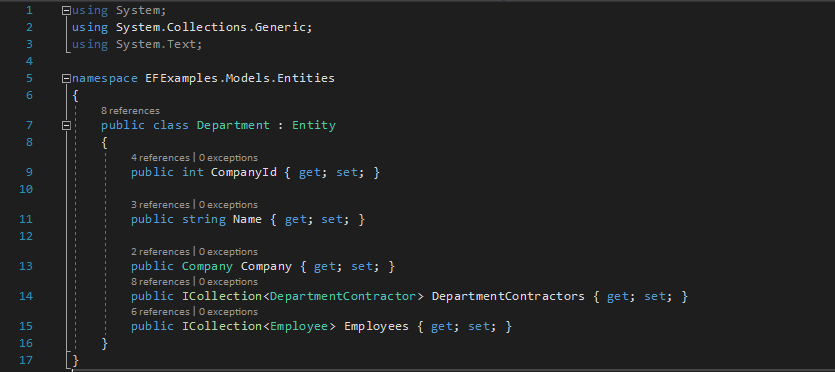
****

**Figure 37: a.) .Models**

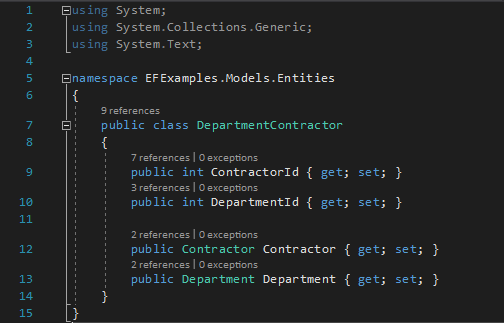
****

**Figure 38: b.) Company.cs Figure 39: c.) Contractor.cs**

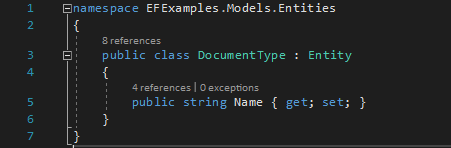
****



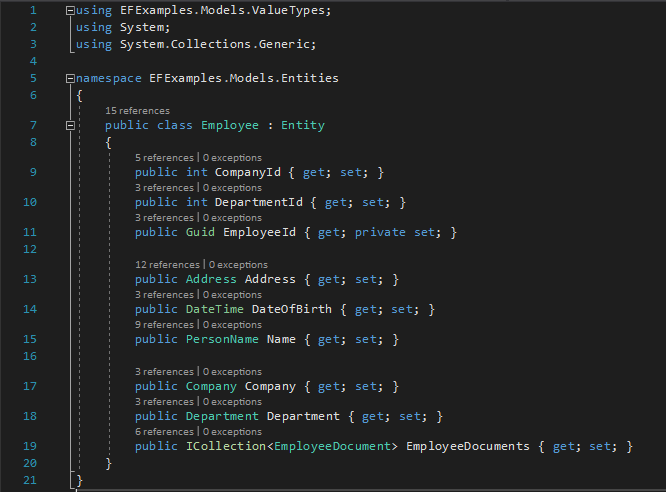
**Figure 40: d.) Department.cs**

****

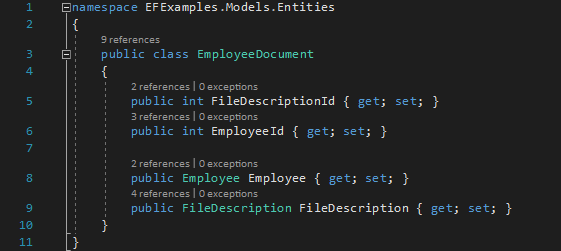
**Figure 41: e.) DepartmentContractor.cs**



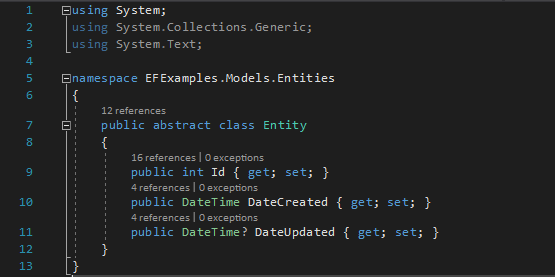
**Figure 42: f.) DocumentType.cs**

****

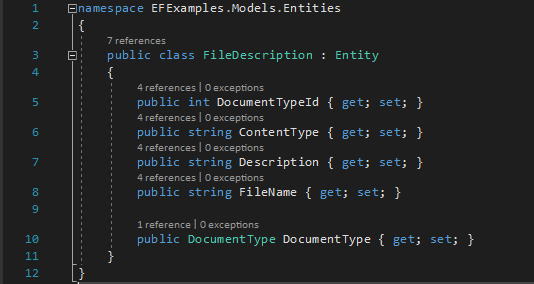
**Figure 43: f.) Employee.cs**



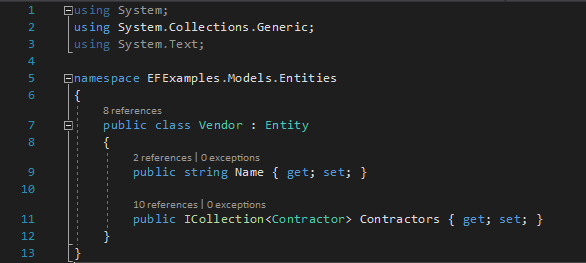
**Figure 44: g.) EmployeeDocument.cs**

****

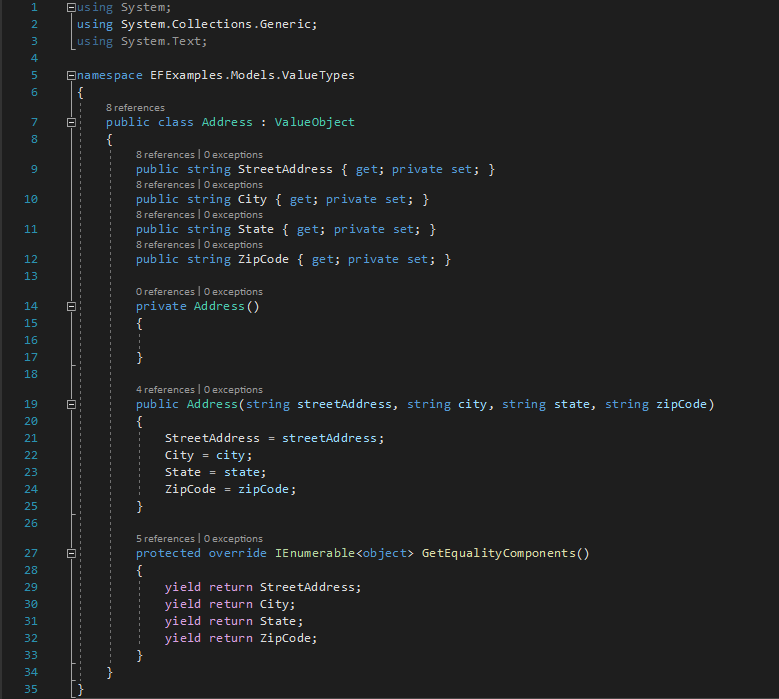
**Figure 45: h.) Entity.cs**



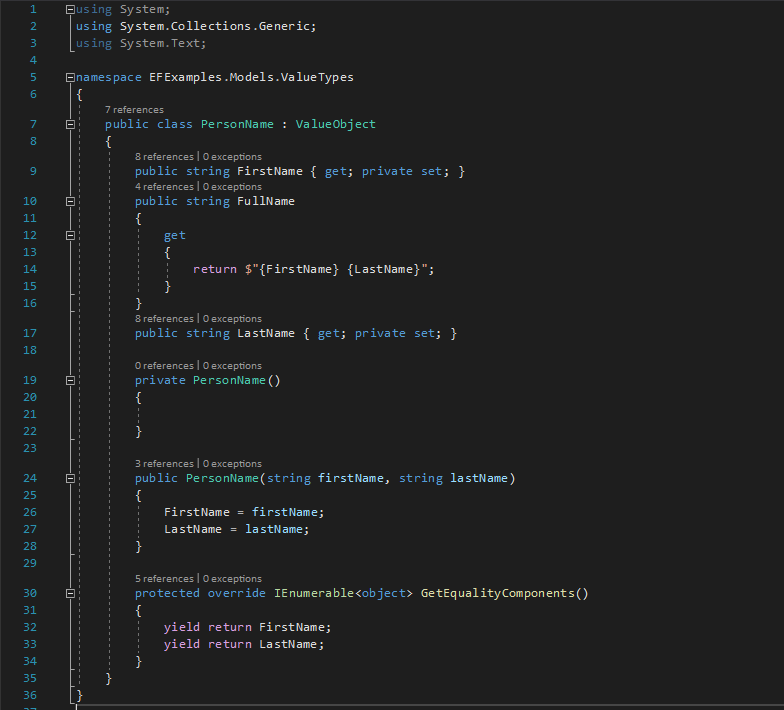
**Figure 46: i.) FileDescripiton.cs**

****

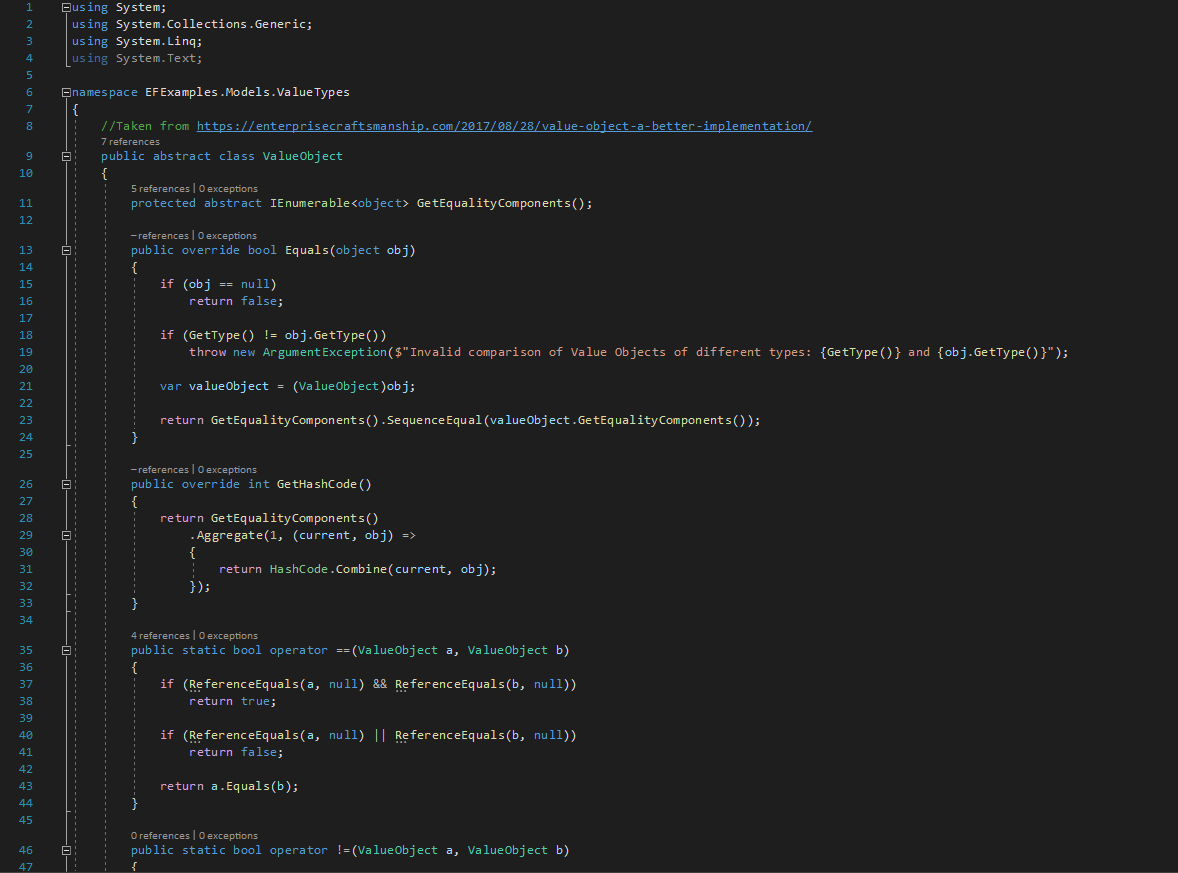
**Figure 47: j.) Vendor.cs**

****

**Figure 48: k.) Address.cs**



**Figure 49: l.) PersonName.cs**

****

**Figure 50: m.) ValueObject.cs**

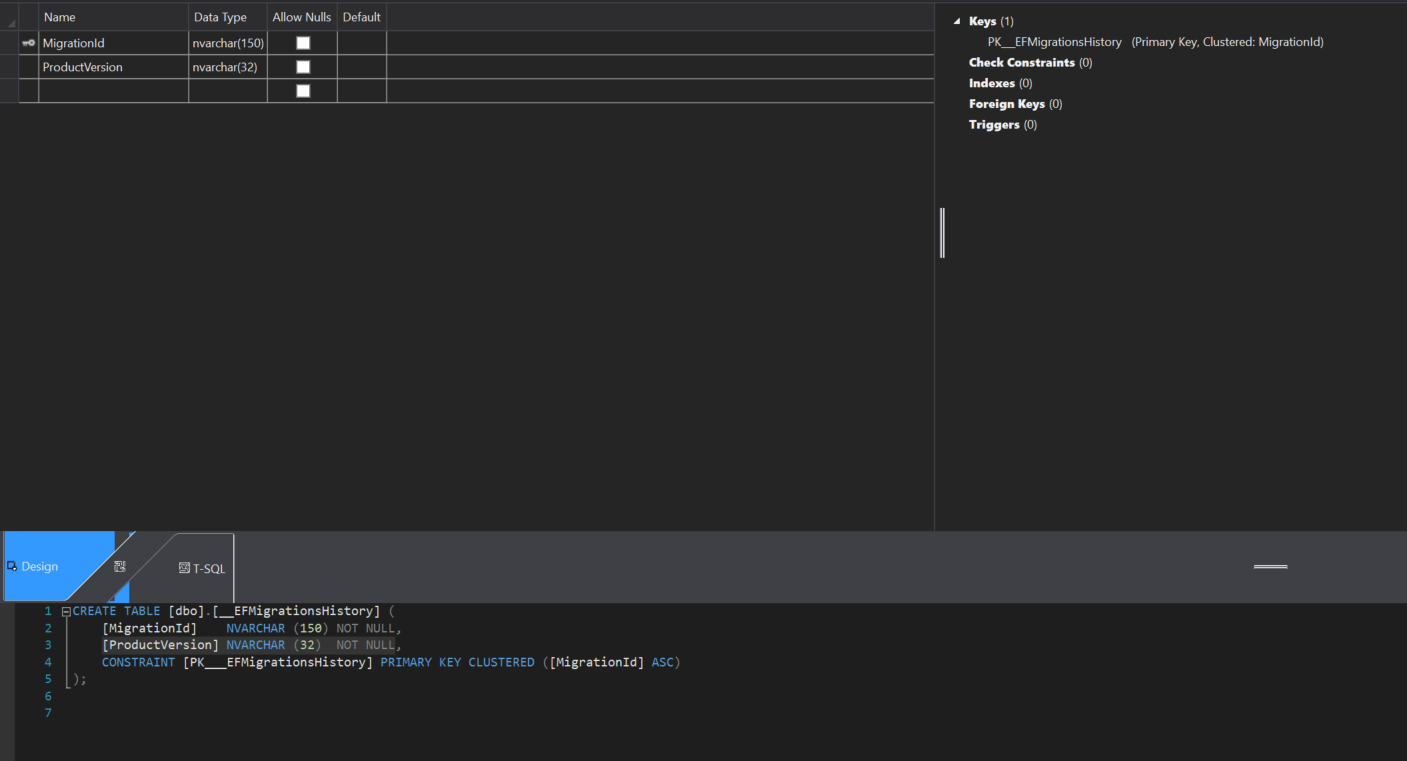
**Task 5: Sql**

****

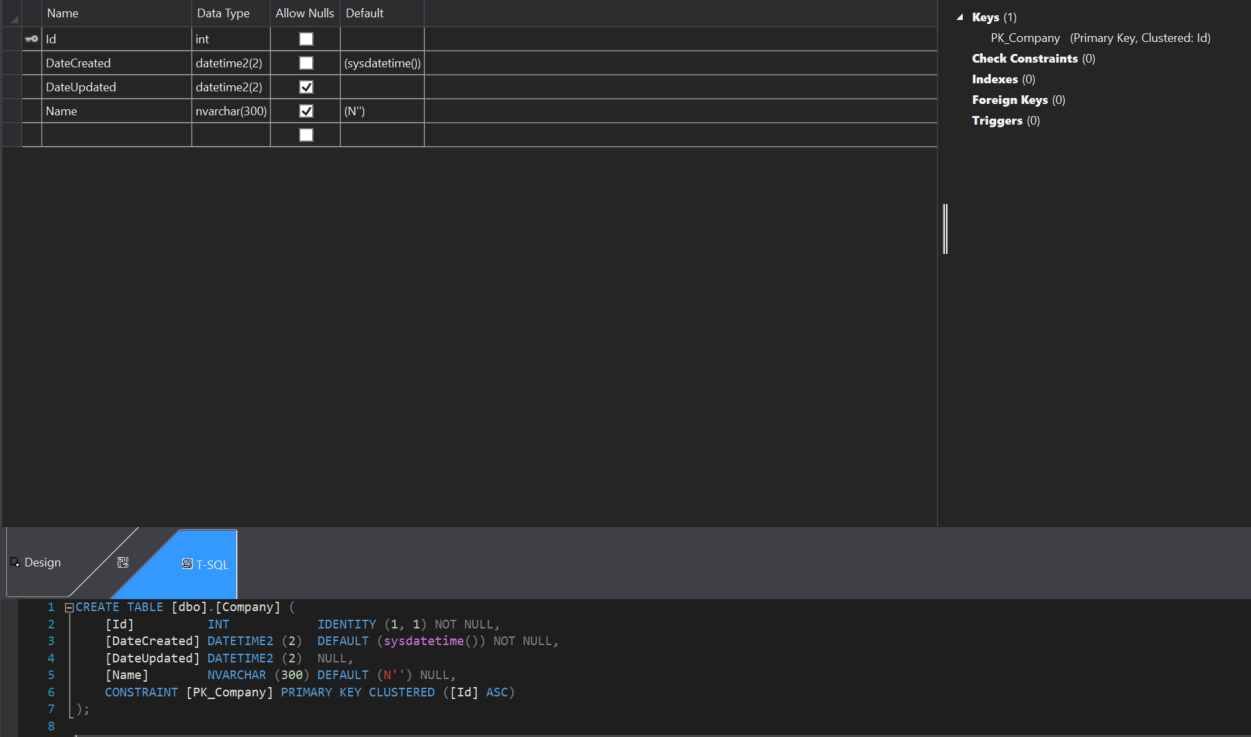
**Figure 51: a.) EmployeeDocuments.sql**

****

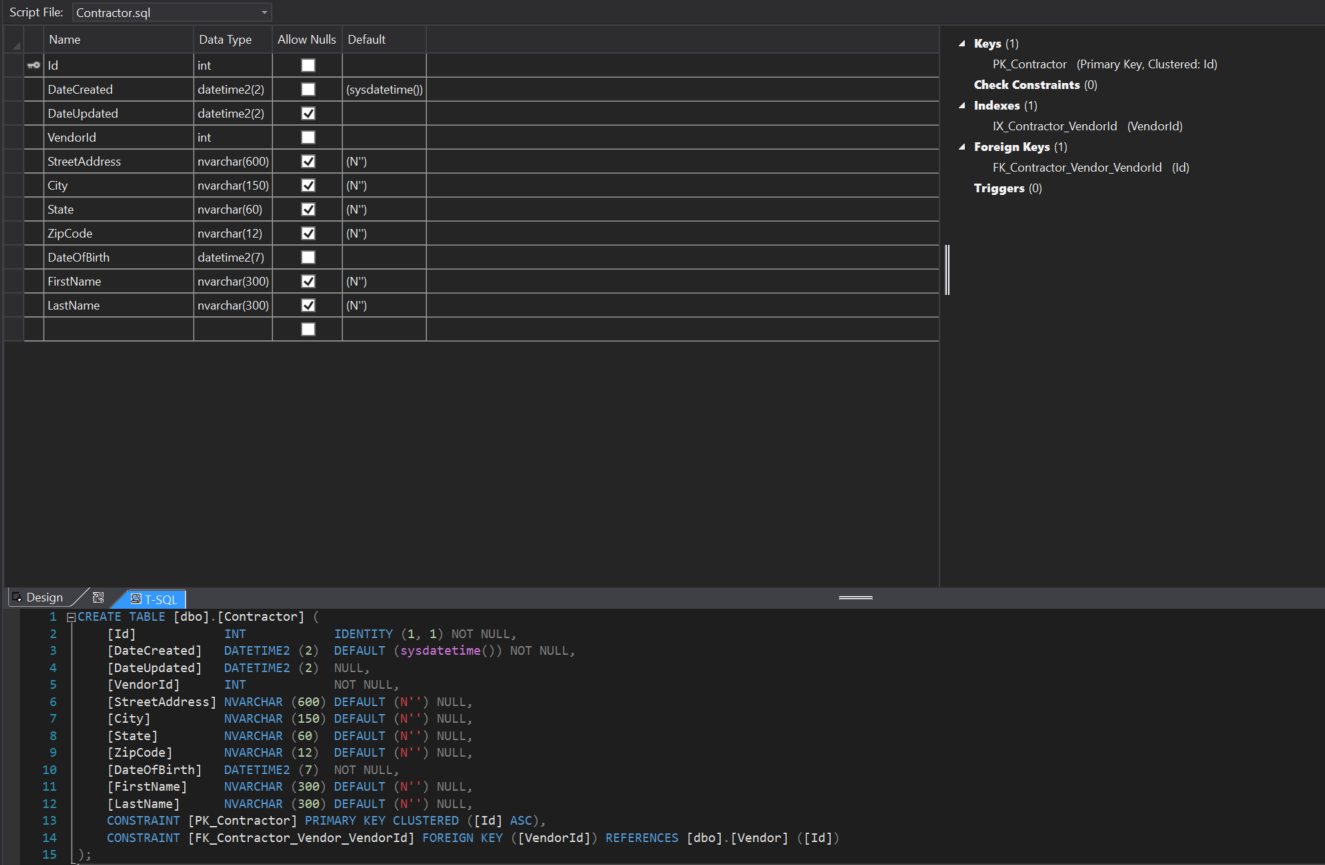
**Figure 52: b.) EmployeeProfileImages.sql**

****

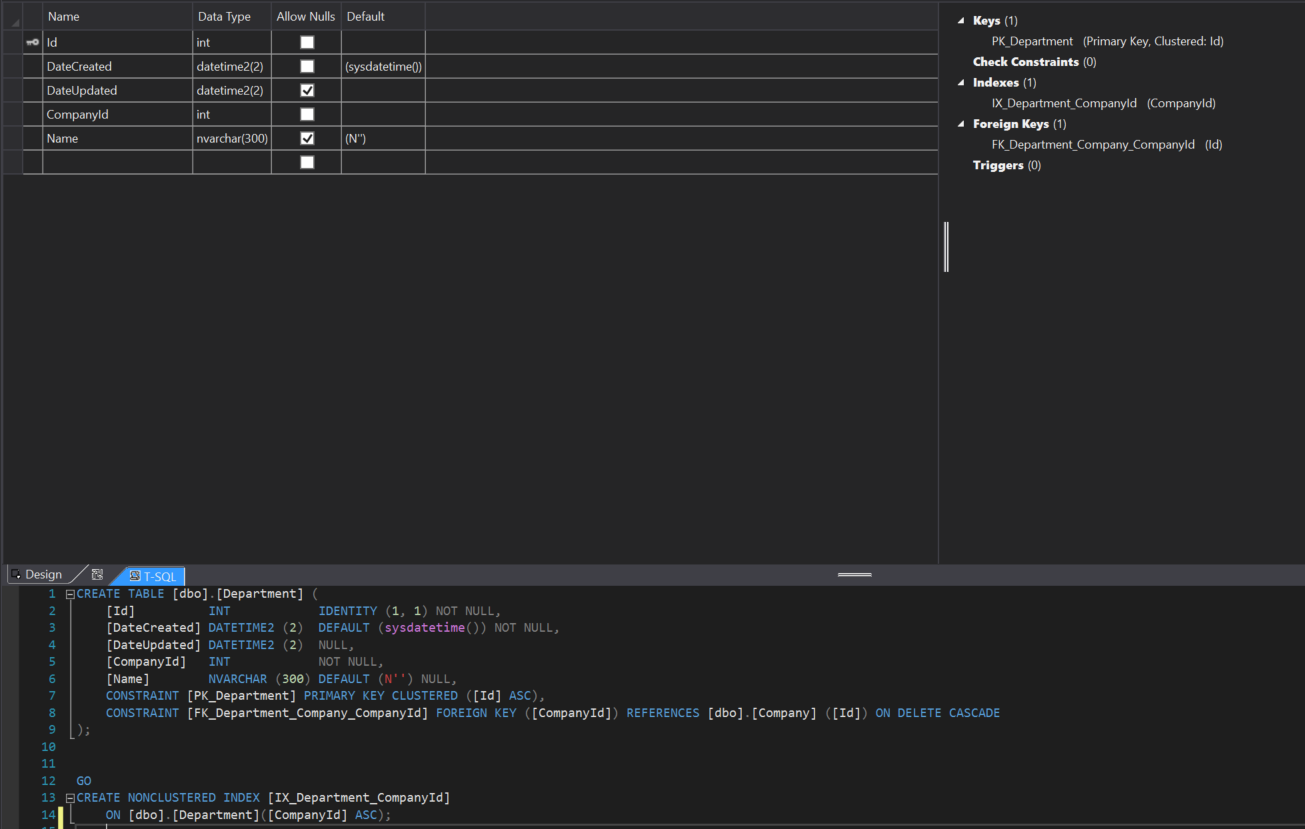
**Figure 53: c.) MigrationHistory.sql**



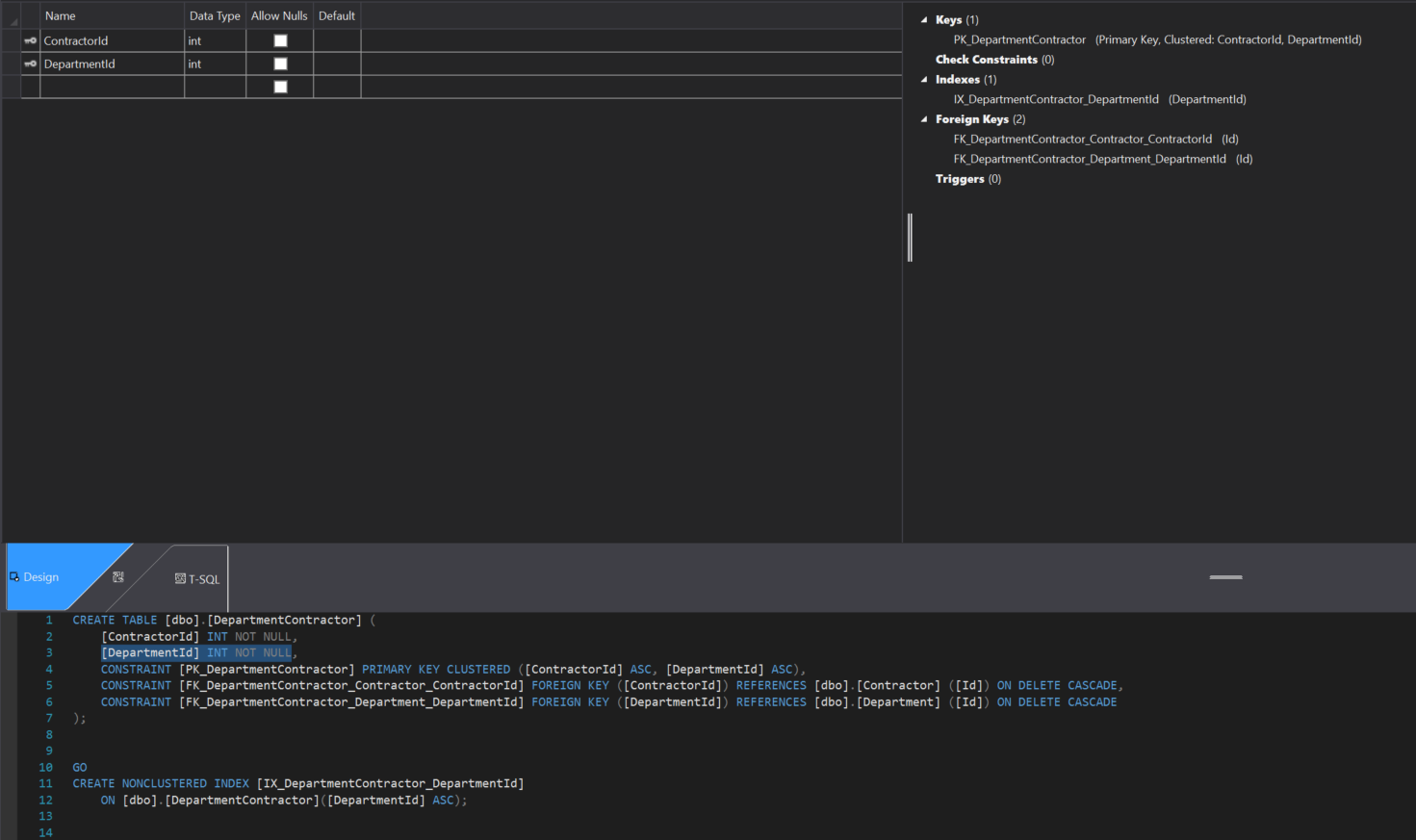
**Figure 54: d.) Company.sql**

****

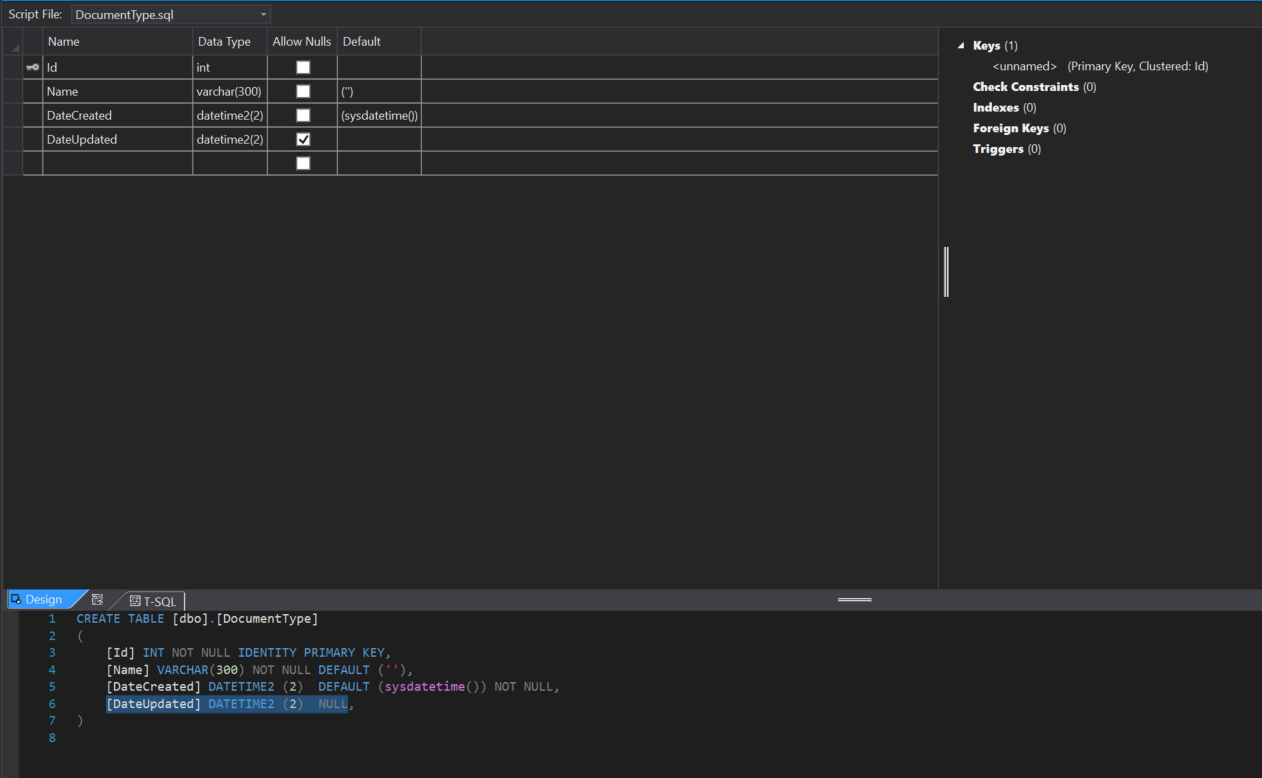
**Figure 55: e.) Contractor.sql**

****

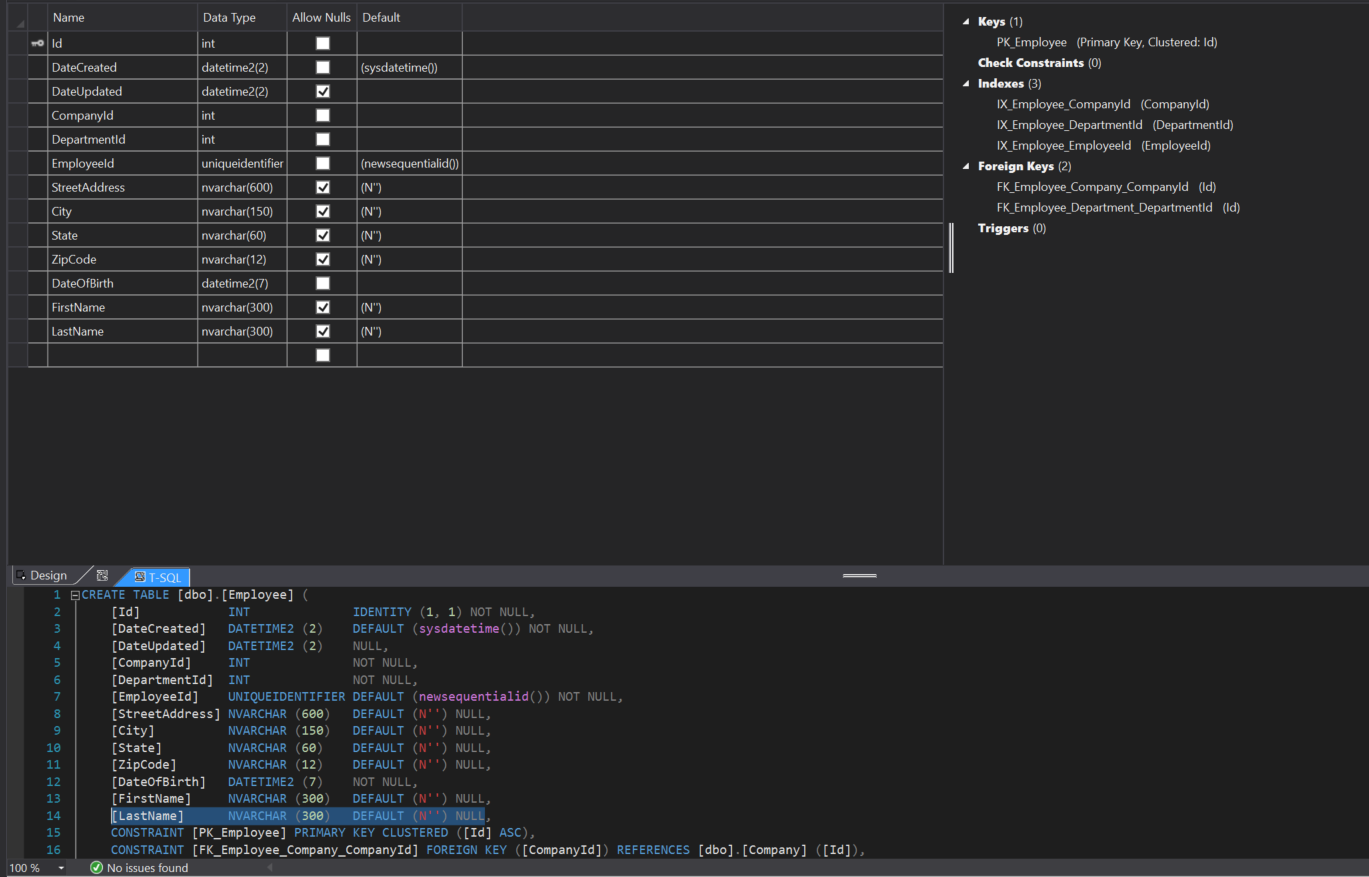
**Figure 56: f.) Department.sql**



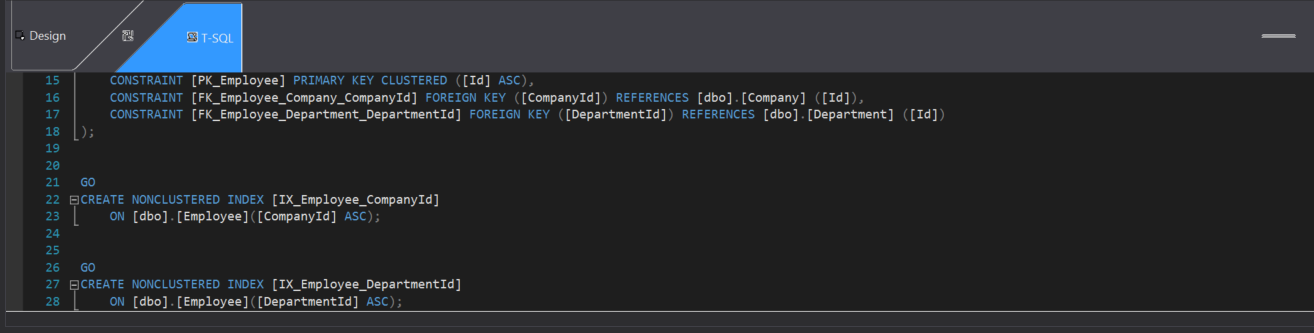
**Figure 57: g.) DepartmentContractor.sql**



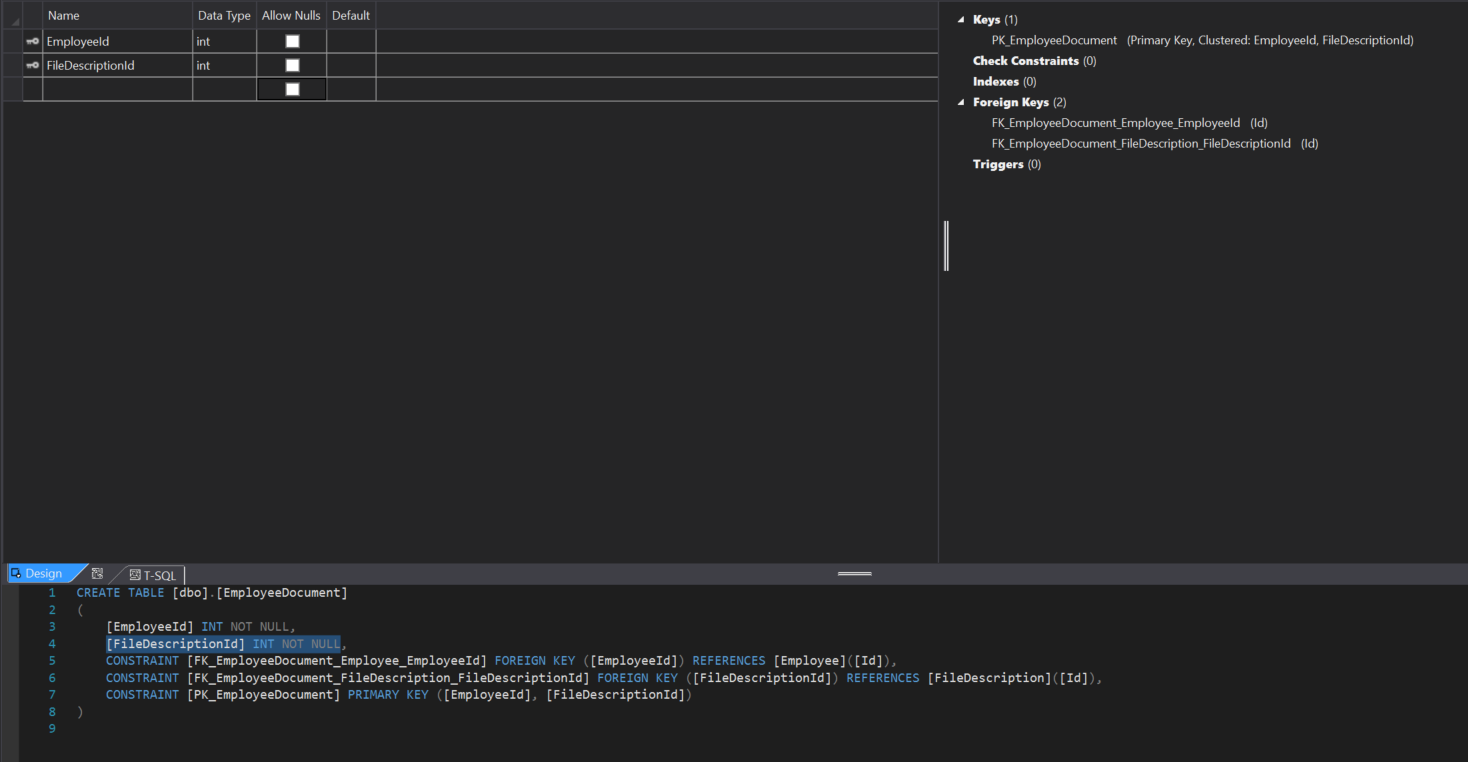
**Figure 58: h.) DocumentType.sql**



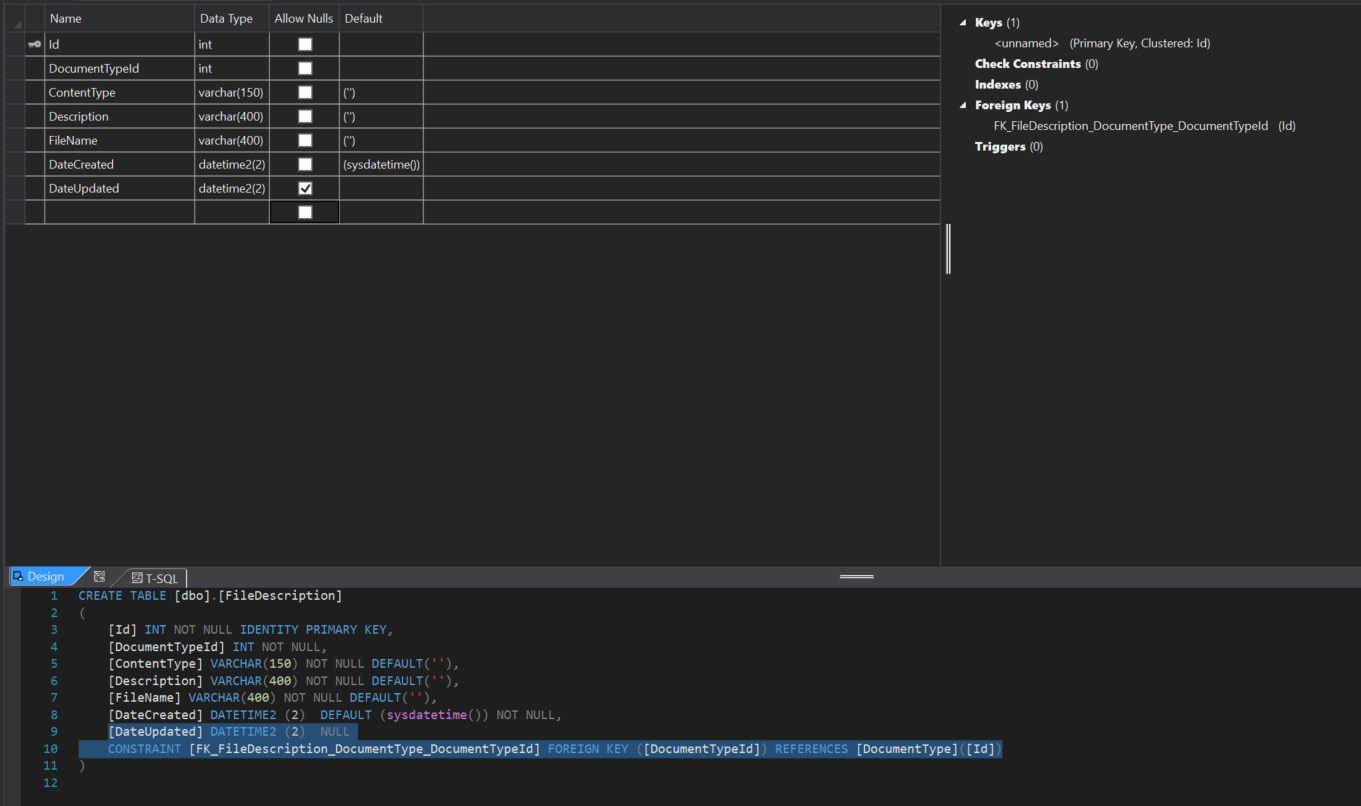
**Figure 59: i.) Employee.sql**

****

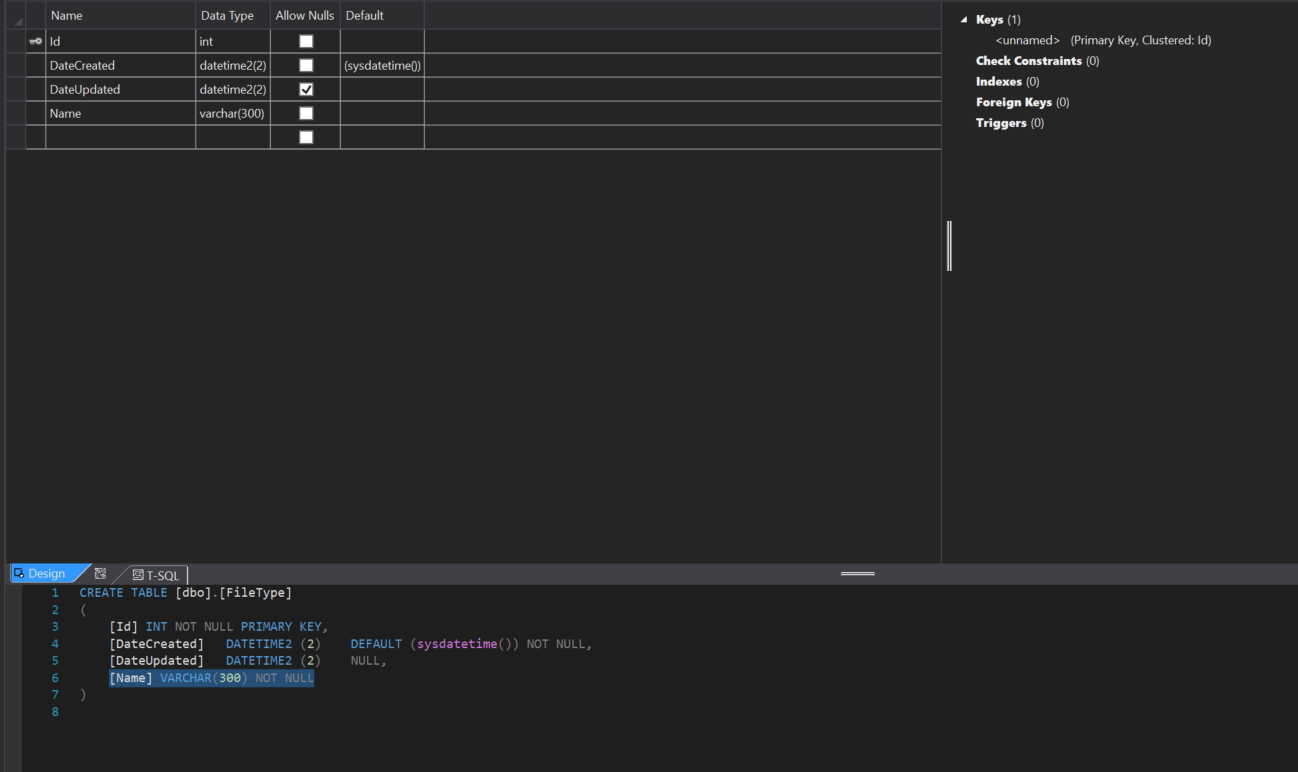
**Figure 60: j.) Employee(2).sql**



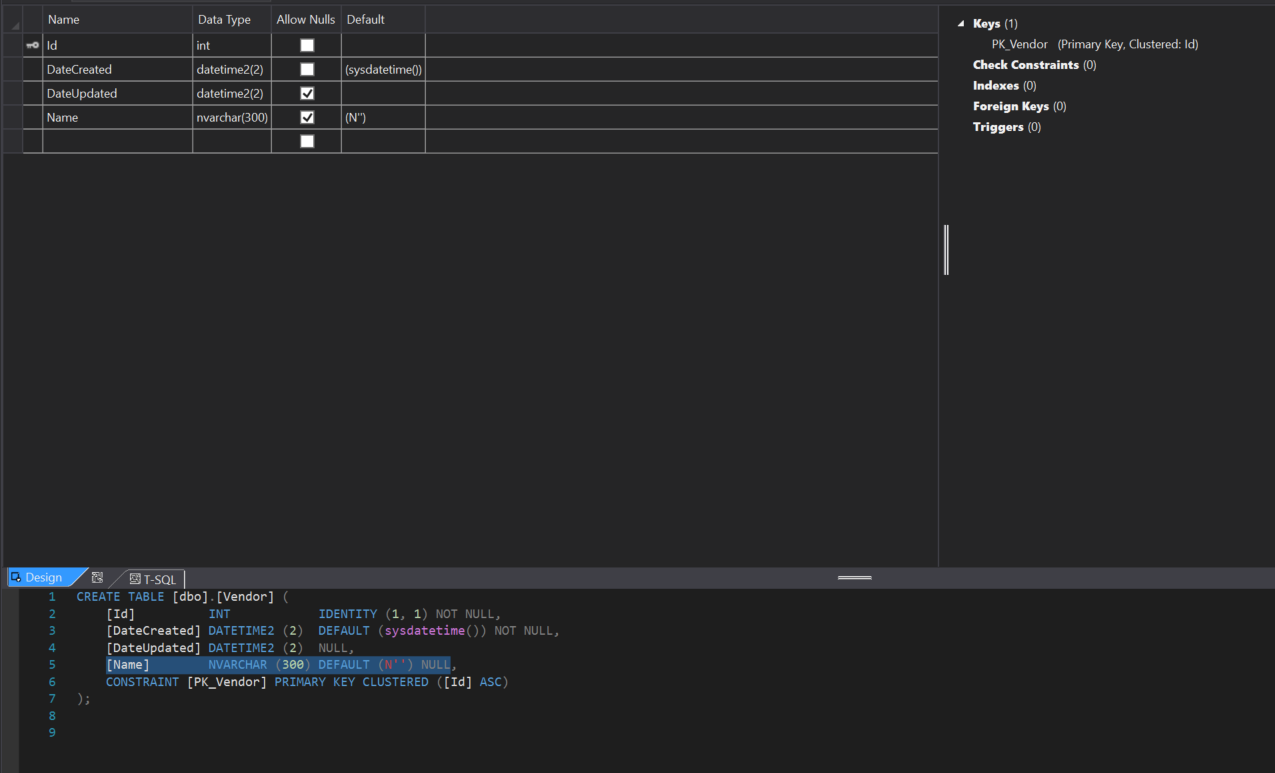
**Figure 61: k.) EmployeeDocuments.sql**



**Figure 62: l.) FileDescription.sql**



**Figure 63: m.) FileType.sql**

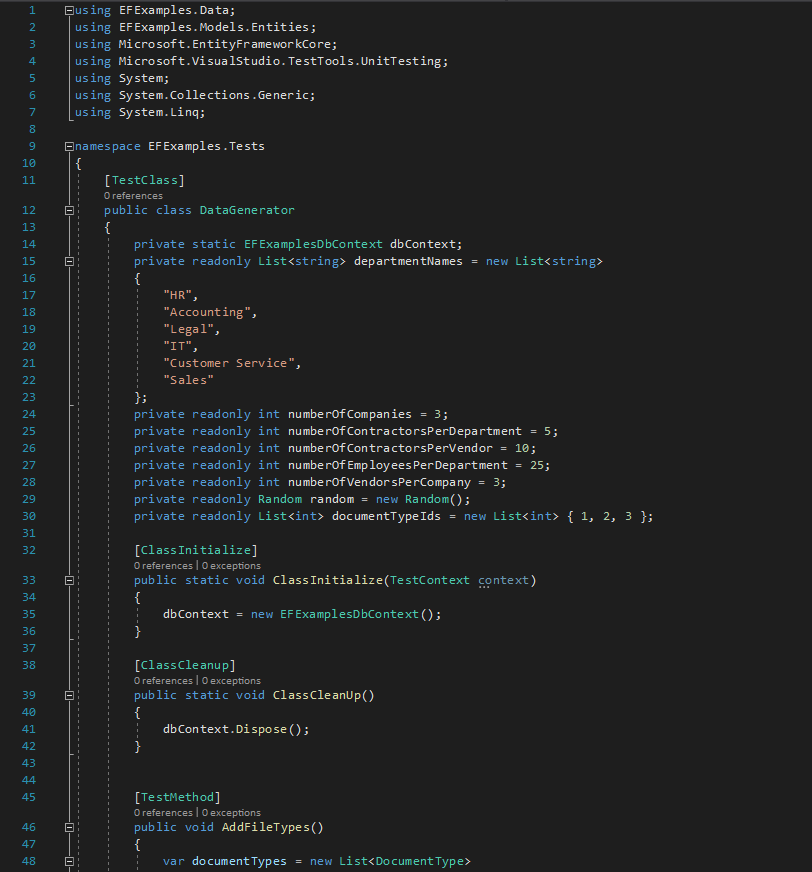
****

**Figure 64: n.) Vendor.sql**

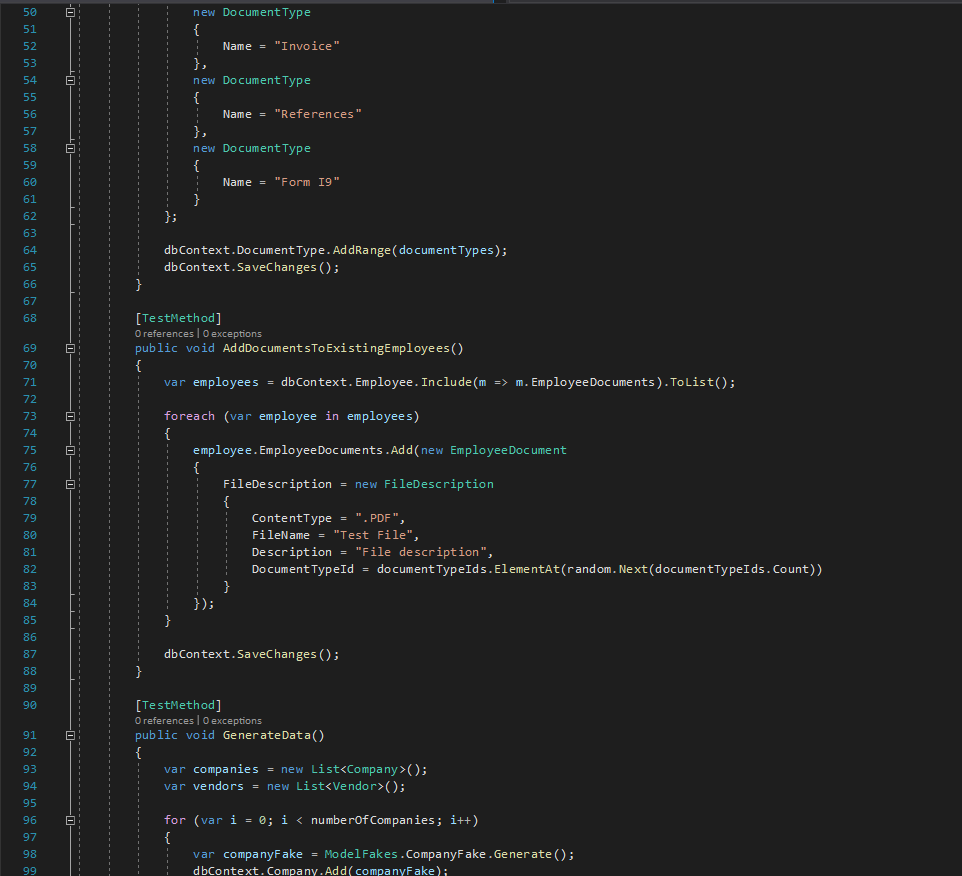
**Task 6: Tests**



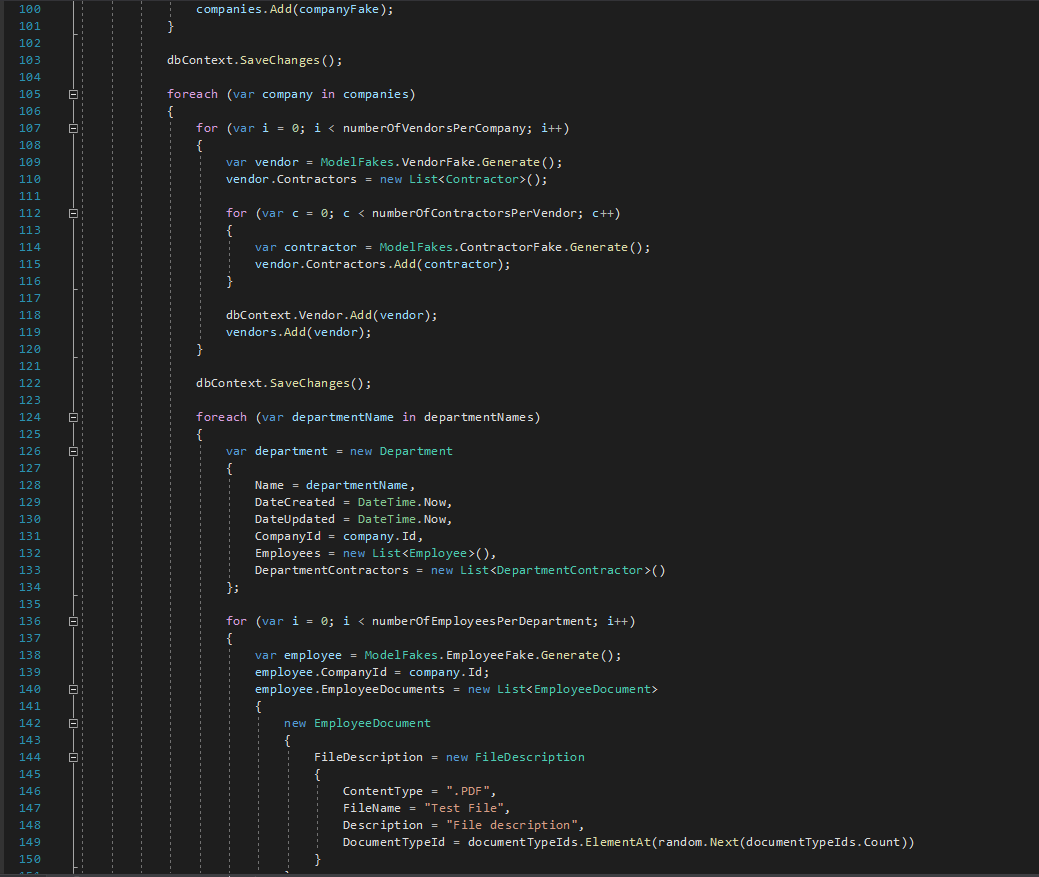
**Figure 65: a.) .Tests**

****

**Figure 66: b.) DataGenerator.cs**



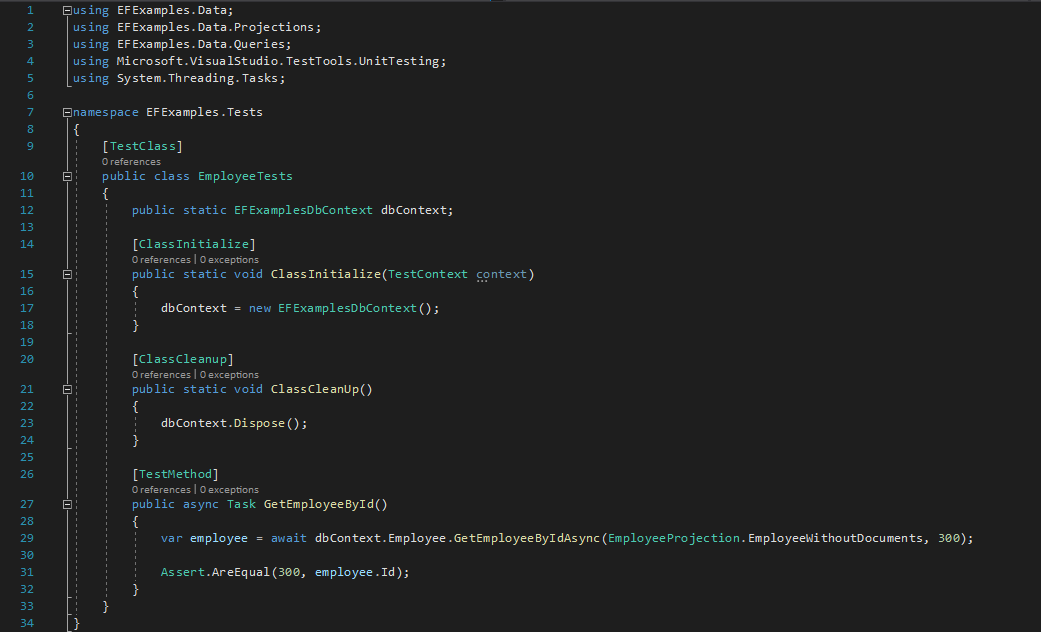
**Figure 67: c.) DataGenerator(2).cs**

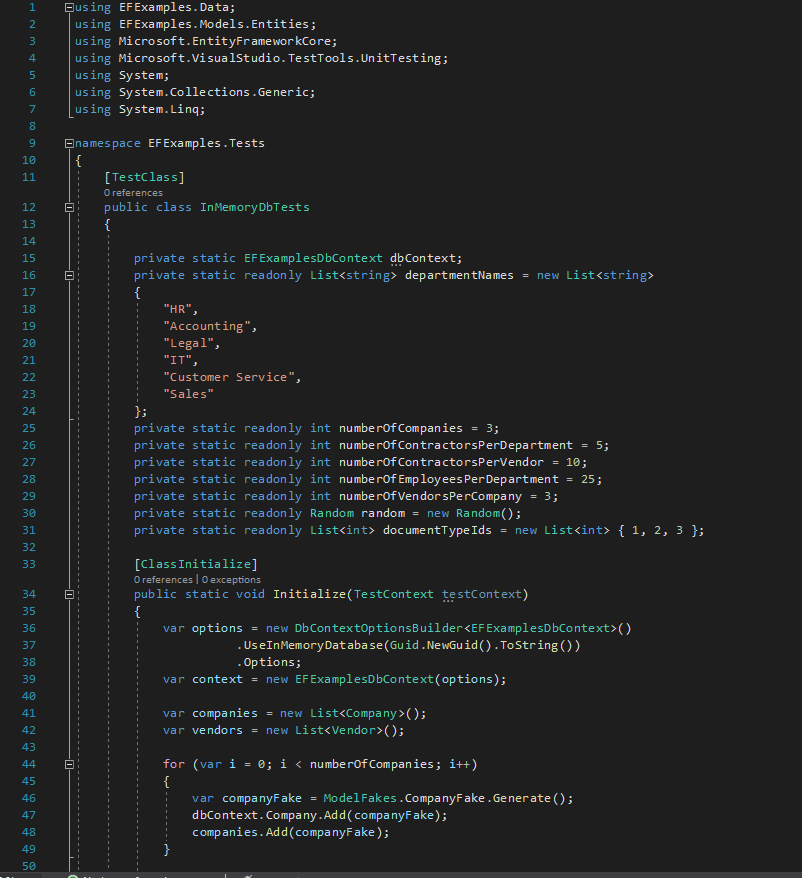


**Figure 68: d.) DataGenerator(3).cs**

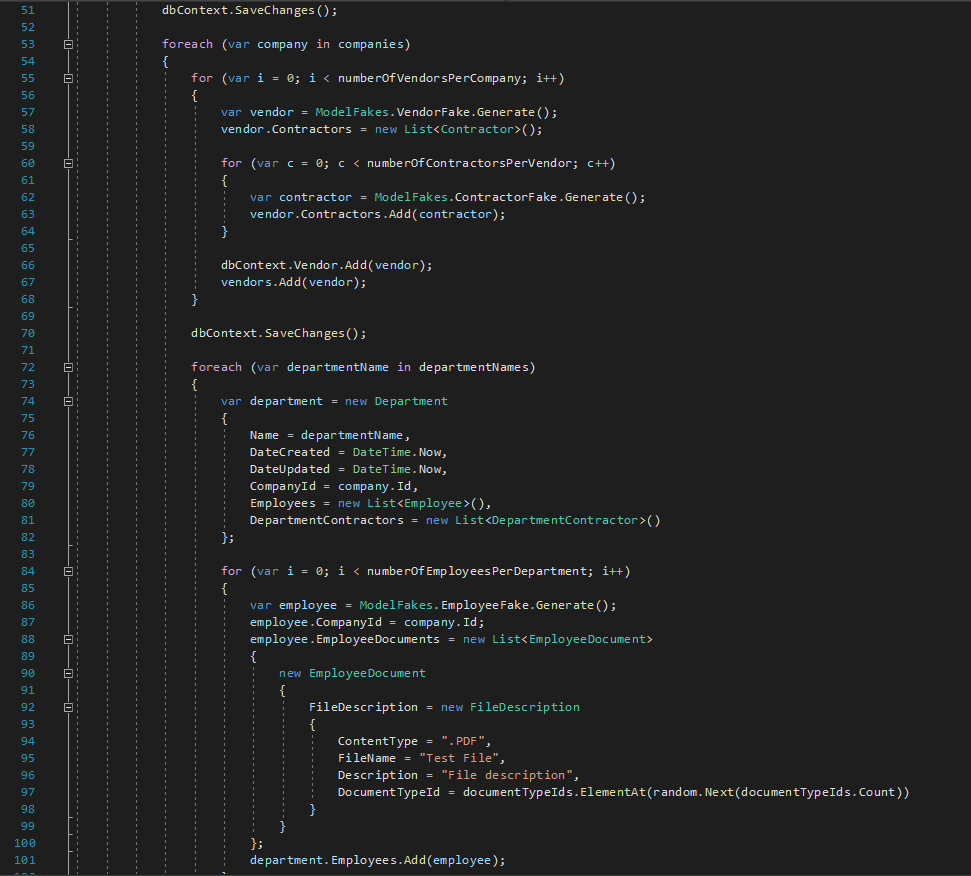


**Figure 69: e.) DataGenerator(4).cs Figure 70: f.) EmployeeTests.cs**

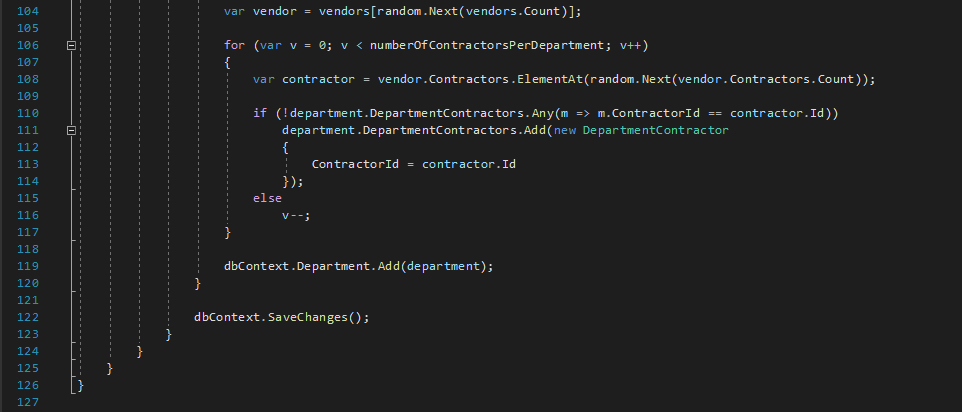
****



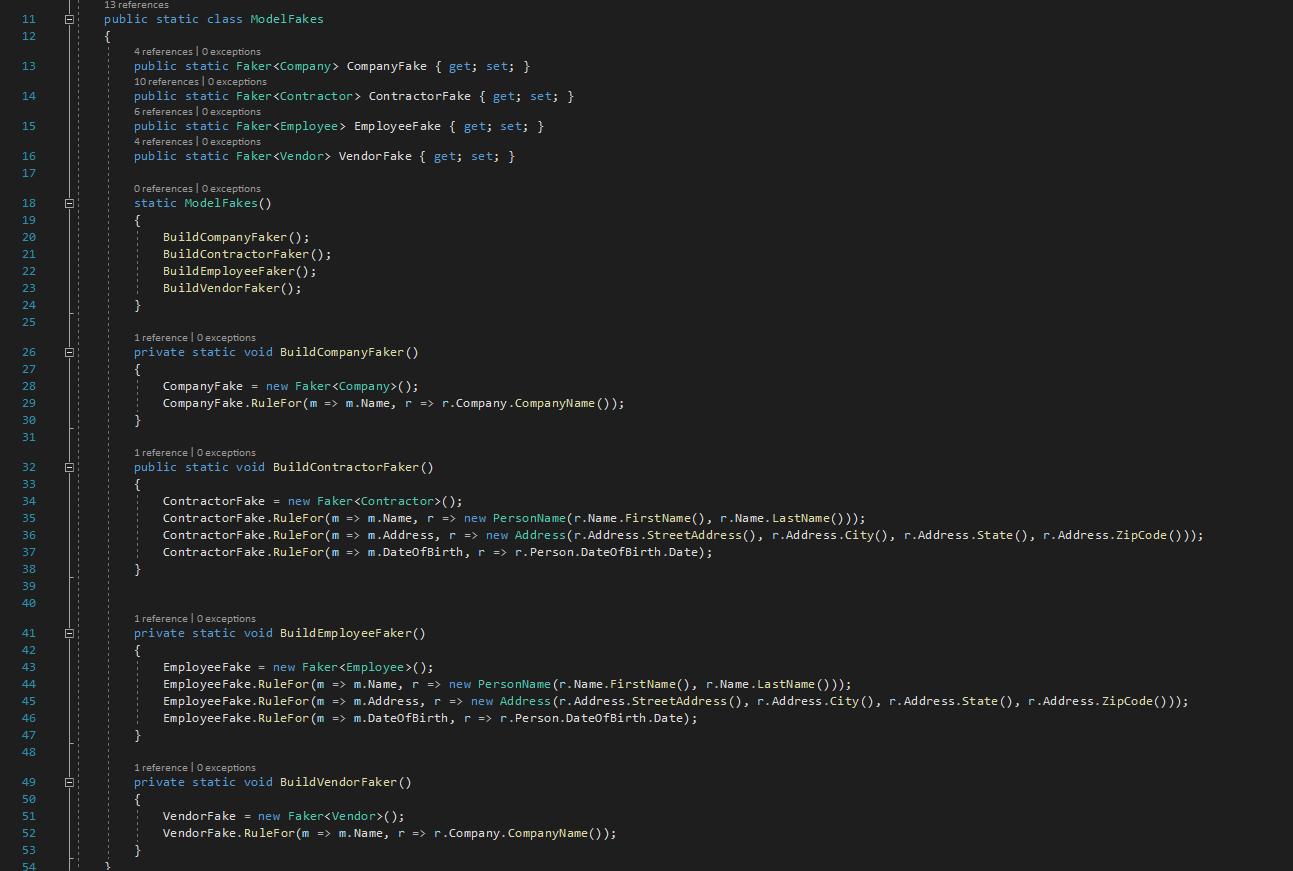
**Figure 71: g.) InMemoryDbTests.cs**



**Figure 72: h.) InMemoryDbTests(2).cs**



**Figure 73: i.) InMemoryDbTests(3).cs Figure 74: j.) ModelFakes.cs**

****



**Figure 75: k.) ValueObjectTests.cs**

****

**Figure 76: l.) ValueObjectTests(2).cs**

**Task 7: Summary**

* This project is pretty straightforward the use of common programming languages like SQL for developing and designing a database is required. You can also use PostgreSQL or Apache alternatively.
* Producing this project in Microsoft Visual Studio I needed to use C#, JSON, and JavaScript, etc. Entity Framework Core module made this project unchallenging and effortless.
* Project examples like this with the use of the conceptual company you're required to code and process minuo attributes like the entities, employees, contractors, and departments, etc.
* Alternatives from this project can range from citizen data collection for utilizing it into national, tax, real estate, vote and even investing equity market, etc. Complex and intricate projects can involve the use of Artificial Intelligence, Deep Learning, and Machine Learning.

**References:**

**Bibliography**

<https://docs.microsoft.com/en-us/ef/core/>

<https://docs.microsoft.com/en-us/ef/>

<https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx>

<https://github.com/dotnet/efcore>

<https://www.nuget.org/packages/Microsoft.EntityFrameworkCore>

Pages: 65

Words: 1,355

Characters (no spaces): 9,226

Characters (with spaces): 10,859

Paragraphs: 279

Lines: 1,558