## Lab Day02 EF

### Tasks:

#### Task 1: Referential Actions with Navigation Properties

1. Create a new Entity Framework Core project.
2. Define two entities, Author and Book, with a one-to-many relationship (one Author can write many Books).
3. Implement referential actions using navigation properties, ensuring cascading delete is enabled.
4. Test the referential actions by deleting an Author and verifying that related Books are also deleted.

#### Task 2: Referential Actions with Fluent API

1. Modify the project to configure the one-to-many relationship using Fluent API.
2. Ensure that referential actions (cascade delete) are correctly implemented.
3. Verify the cascading delete functionality as in Task 1.

#### Task 3: Include and Exclude Tables

1. Create two more entities, Publisher and Library, and set up relationships.
2. Configure Entity Framework Core to include Publisher in the model and exclude Library.

#### Task 4: Controlling Entity Shape with Data Annotation

1. Define a new entity Customer with various attributes.
2. Use data annotations to control the shape of the Customer entity (e.g., required fields, string length).
3. Verify that the constraints are enforced.

#### Task 5: Controlling Entity Shape with Fluent API

1. Refactor the Customer entity configuration to use Fluent API instead of data annotations.
2. Ensure the same constraints are applied as in Task 4.

#### Task 6: Default and Computed Columns

1. Add a Product entity with a Price and a computed TotalPrice column (Price + Tax).
2. Configure default values for some columns.
3. Verify that default and computed columns are correctly populated.

#### Task 7: Indexing in Entity Framework Core

1. Create an Order entity with attributes such as OrderDate and CustomerID.
2. Use Entity Framework Core to create indexes on OrderDate and CustomerID.
3. Verify that indexes are created by inspecting the database.

#### Task 8: Operations on Indexes

1. Perform operations such as adding, removing, and modifying indexes on the Order entity.
2. Verify that the changes are reflected in the database.

#### Task 9: Using Database Sequences

1. Define an Invoice entity with an InvoiceNumber generated by a database sequence.
2. Configure the sequence in Entity Framework Core.
3. Verify that the InvoiceNumber is generated using the sequence.

#### Task 10: Entity Framework Reverse Scaffold

1. Create a new database with a few tables.
2. Use Entity Framework Core reverse scaffolding to generate the models and DbContext from the existing database.
3. Verify the generated code and make any necessary adjustments.