

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

[Table("ADDRESSES")]
public class Address {
    [Column("ADDRESS_ID")]
    [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    public int Id { get; set; }

    [Required, StringLength((8))]
    [Column("POSTAL_CODE")]
    public string PostalCode { get; set; }

    [Required, StringLength(100)]
    [Column("LOCATION")]
    public string Location { get; set; }

    [Required, StringLength(100)]
    [Column("STREET")]
    public string Street { get; set; }

    public State State { get; set; }

    [Required]
    [Column("STATE")]
    public string StateCode { get; set; }

    public Country Country { get; set; }

    [Required]
    [Column("COUNTRY")]
    public string CountryName { get; set; }
}
```

```

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace Model.Entities;
[Table("CINEMAS")]
public class Cinema
{
    [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    [Column("CINEMA_ID")]
    public int Id { get; set; }

    [Required, StringLength(100)]
    [Column("LABEL")]
    public string Label { get; set; }

    public Address Address { get; set; }

    [Required]
    [Column("ADDRESS_ID")]
    public int AddressId { get; set; }
}

```

```

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

[Table("E_COUNTRIES")]
public class Country {
    [StringLength(100)]
    [Key, Column("NAME")]
    public string Name {get; set; }
}

```

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using Model.Entities;

[Table("HALLS")]
public class Hall{

    [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    [Column("HALL_ID")]
    public int Id { get; set; }

    [Required, Range(0, 500)]
    [Column("CAPACITY")]
    public int Capacity { get; set; }

    [Required, StringLength(45)]
    [Column("NAME")]
    public string Name { get; set; }

    [Required, StringLength(45)]
    [Column("CODE")]
    public string Code { get; set; }

    //relation
    public Cinema Cinema { get; set; }

    [Column("CINEMA_ID")]
    public int CinemaId { get; set; }
}
```

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

[Table("MOVIES")]
public class Movie {
    [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    [Key]
    [Column("MOVIE_ID")]
    public int Id { get; set; }

    [Required]
    [Column("TITLE")]
    [StringLength(100)]
    public string Title { get; set; }

    [Required]
    [Column("DIRECTOR")]
    [StringLength(50)]
    public string Director { get; set; }

    [Required]
    [Column("LENGTH")]
    [MinLength(0)]
    [MaxLength(900)]
    // [Range(0, 900)]
    public int Length { get; set; }

    [Required]
    [Column("SHORT_DESCRIPTION")]
    [StringLength(400)]
    public string ShortDescription { get; set; }
}
```

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

[Table("MOVIE_CREWS")]
public class Person {

    [Key, DatabaseGenerated(DatabaseGeneratedOption.Identity)]
    [Column("PERSON_ID")]
    public int Id { get; set; }

    [Required, StringLength(45)]
    [Column("FIRST_NAME")]
    public string FirstName { get; set; }

    [Required, StringLength(45)]
    [Column("LAST_NAME")]
    public string LastName { get; set; }
}
```

```
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

[Table("E_STATES")]
public class State {

    [StringLength(100)]
    [Key, Column("NAME")]
    public string Name { get; set; }
}
```

```

using Microsoft.EntityFrameworkCore;
using Model.Entities;

namespace Model.Configurations;

public class MovieContext : DbContext {
    public MovieContext(DbContextOptions<MovieContext> options) :
base(options) { }

    public DbSet<Movie> Movies{ get; set; }
    public DbSet<Person> People { get; set; }
    public DbSet<State> States { get; set; }
    public DbSet<Country> Countries { get; set; }
    public DbSet<Address> Addresses { get; set; }
    public DbSet<Cinema> Cinemas { get; set; }
    public DbSet<Hall> Halls { get; set; }

    protected override void OnModelCreating(ModelBuilder builder)
    {
        builder.Entity<Cinema>()
            .HasOne(c => c.Address)
            .WithOne()
            .HasForeignKey<Cinema>(c => c.AddressId);

        //Address
        builder.Entity<Address>()
            .HasOne(a => a.Country)
            .WithMany()
            .HasForeignKey(a => a.CountryName);

        builder.Entity<Address>()
            .HasOne(a => a.State)
            .WithMany()
            .HasForeignKey(a => a.StateCode);

        //hall
        builder.Entity<Hall>()
            .HasOne(h => h.Cinema)
            .WithMany()
            .HasForeignKey(h => h.CinemaId);
    }
}

```

```

INSERT INTO E_COUNTRIES (NAME)
VALUES ('Austria');
INSERT INTO E_COUNTRIES (NAME)
VALUES ('France');
INSERT INTO E_COUNTRIES (NAME)
VALUES ('Germany');
INSERT INTO E_STATES (NAME)
VALUES ('Burgenland');
INSERT INTO E_STATES (NAME)
VALUES ('Kärnten');
INSERT INTO E_STATES (NAME)
VALUES ('Niederösterreich');
INSERT INTO E_STATES (NAME)
VALUES ('Oberösterreich');
INSERT INTO E_STATES (NAME)
VALUES ('Salzburg');
INSERT INTO E_STATES (NAME)
VALUES ('Steiermark');
INSERT INTO E_STATES (NAME)
VALUES ('Tirol');
INSERT INTO E_STATES (NAME)
VALUES ('Vorarlberg');
INSERT INTO E_STATES (NAME)
VALUES ('Wien');

INSERT INTO ADDRESSES (ADDRESS_ID, POSTAL_CODE, LOCATION, STREET, STATE,
COUNTRY)
VALUES (1, '3500', 'Krems a.d. Donau', 'Gewerbestrasse 21',
'Niederösterreich', 'Austria');
INSERT INTO ADDRESSES (ADDRESS_ID, POSTAL_CODE, LOCATION, STREET, STATE,
COUNTRY)
VALUES (3, '3500', 'Krems a.d. Donau', 'Dr Karl Dorrek Strasse 30',
'Niederösterreich', 'Austria');
INSERT INTO CINEMAS (CINEMA_ID, LABEL, ADDRESS_ID)
VALUES (1, 'Cinemaplexx Krems', 1);
INSERT INTO CINEMAS (CINEMA_ID, LABEL, ADDRESS_ID)
VALUES (3, 'Kino im Kesselhaus', 3);
INSERT INTO HALLS (HALL_ID, CAPACITY, NAME, CODE, CINEMA_ID)
VALUES (1, 50, 'Saal A', 'krems_A', 1);
INSERT INTO HALLS (HALL_ID, CAPACITY, NAME, CODE, CINEMA_ID)
VALUES (2, 120, 'Saal B', 'krems_b', 1);
INSERT INTO HALLS (HALL_ID, CAPACITY, NAME, CODE, CINEMA_ID)
VALUES (3, 40, 'Kesselsaal', 'KL-34', 3);

```

	ADDRESS_ID	COUNTRY	LOCATION	POSTAL_CODE	STATE	STREET
1	1	Austria	Krems a.d. Donau	3500	Niederösterreich	Gewerbestrasse 21
2	3	Austria	Krems a.d. Donau	3500	Niederösterreich	Dr Karl Dörrek Strasse 30

	CINEMA_ID	ADDRESS_ID	LABEL
1	1	1	Cinemaplexx Krems
2	3	3	Kino im Kesselhaus

	NAME
1	Austria
2	France
3	Germany

	NAME
1	Burgenland
2	Kärnten
3	Niederösterreich
4	Oberösterreich
5	Salzburg
6	Steiermark
7	Tirol
8	Vorarlberg
9	Wien

	HALL_ID	CAPACITY	NAME	CODE	CINEMA_ID
1	1	50	Saal A	krems_A	1
2	2	120	Saal B	krems_b	1
3	3	40	Kesselsaal	KL-34	3

	MOVIE_ID	TITLE	DIRECTOR	LENGTH	SHORT_DESCRIPTION
1	1	Movietitel	Direktorname	200	Beschreibung des Moives
2	2	Avatar	James Cameron	162	Ein querschnittgelähmter US-Marinesoldat wird auf eine ganz...
3	3	Matrix	Lana Wachowski	136	Ein Computerhacker erfährt von mysteriösen Rebellen die Wah...
4	4	Lord OfThe Rings	Peter Jackson	178	Ein sanftmütiger Hobbit aus dem Auenland und seine acht Gef...
5	5	Avatar -Way ofWater	James Cameron	190	Jake Sully lebt mit seiner neu gegründeten Familie auf dem ...