

Bibliothek - Entity Framework Vererbung

Teil 1 Ausleihbare Elemente in der Bibliothek (Single Table Inheritance)

Ausgangssituation

Du entwickelst ein **Bibliothekssystem**, das verschiedene Arten von ausleihbaren Elementen verwaltet. Die ausleihbaren Elemente umfassen **Bücher**. Jedes Buch hat spezifische Eigenschaften, die je nach Buchart variieren können.

Ziel der Aufgabe

Implementiere die Vererbungshierarchie für ausleihbare Bücher in Entity Framework Code First, indem du:

1. Eine Basisklasse **Item** mit den allgemeinen Eigenschaften für alle Bücher erstellst.
2. Abgeleitete Klassen für spezifische Bucharten definierst.

Anforderungen

Schritt 1: Definition der Basisklasse Item

Erstelle eine Basisklasse **Item** mit den folgenden Eigenschaften:

- **Id**: Primärschlüssel (int)
- **Title**: Titel des Buches (string)
- **Author**: Autor des Buches (string)
- **PublishedDate**: Veröffentlichungsdatum (DateTime)
- **ISBN**: ISBN-Nummer (string)
- **AvailableCopies**: Verfügbare Exemplare (int)

Schritt 2: Definition der abgeleiteten Klassen für Bücher

Erstelle die folgenden abgeleiteten Klassen für spezifische Bucharten:

1. **NonFiction** (Sachbuch)
2. **Novel** (Roman)
3. **Textbook** (Lehrbuch)
4. **Biography** (Biografie)

5. **ScienceFiction** (Science-Fiction)

6. **Fantasy** (Fantasy)

7. **Mystery** (Krimi)

Schritt 3: Implementierung in Entity Framework

1. Setze die oben definierten Klassen in Entity Framework um.

```
public class Item {
    public int Id { get; set; }
    public string Title { get; set; }
    public string Author { get; set; }
    public DateTime PublishedDate { get; set; }
    public string ISBN { get; set; }
    public int AvailableCopies { get; set; }
}
public class NonFiction : Item { }
public class Novel : Item { }
public class Textbook : Item { }
public class Biography : Item { }
public class ScienceFiction : Item { }
public class Fantasy : Item { }
public class Mystery : Item { }
```

2. Konfiguriere den DbContext entsprechend, sodass die gewählte Vererbungsstrategie in der Datenbankstruktur korrekt abgebildet wird.

Schritt 4: Beispielhafte Datenbank-Seeding

Füge Beispielbücher in die Datenbank ein, um sicherzustellen, dass die Vererbung korrekt umgesetzt wurde. Jede Buchart sollte vertreten sein.

```
-- Beispielhafte Buchdaten (Single Table Inheritance)
INSERT INTO Items (Id, Title, Author, PublishedDate, ISBN, AvailableCopies,
ItemType) VALUES (1, 'Das Kapital', 'Karl Marx', '1867-09-14', '978-3-16-148410-0', 5, 'NonFiction');
INSERT INTO Items (Id, Title, Author, PublishedDate, ISBN, AvailableCopies,
ItemType) VALUES (2, '1984', 'George Orwell', '1949-06-08', '978-0-452-28423-4', 3, 'Novel');
INSERT INTO Items (Id, Title, Author, PublishedDate, ISBN, AvailableCopies,
ItemType) VALUES (3, 'Einführung in die Programmierung', 'John Doe', '2020-01-01', '978-3-16-148410-1', 10, 'Textbook');
INSERT INTO Items (Id, Title, Author, PublishedDate, ISBN, AvailableCopies,
ItemType) VALUES (4, 'Die Biografie von Albert Einstein', 'Walter Isaacson', '2007-10-02', '978-1-59420-193-0', 4, 'Biography');
INSERT INTO Items (Id, Title, Author, PublishedDate, ISBN, AvailableCopies,
ItemType) VALUES (5, 'Der Marsianer', 'Andy Weir', '2011-02-11', '978-3-446-23559-6', 6, 'ScienceFiction');
INSERT INTO Items (Id, Title, Author, PublishedDate, ISBN, AvailableCopies,
ItemType) VALUES (6, 'Harry Potter und der Stein der Weisen', 'J.K. Rowling', '1997-06-26', '978-3-7459-9406-7', 8, 'Fantasy');
INSERT INTO Items (Id, Title, Author, PublishedDate, ISBN, AvailableCopies,
ItemType) VALUES (7, 'Der Hund von Baskerville', 'Arthur Conan Doyle', '1902-04-01', '978-3-16-148410-8', 2, 'Mystery');
```

Teil 2: Personen in der Bibliothek (Joint Table)

Ausgangssituation

Im Rahmen des Bibliothekssystems verwaltest du verschiedene Typen von Personen, die mit der Bibliothek verbunden sind. Diese umfassen Autoren, Kunden und Bibliothekare. Jede Person hat spezifische Eigenschaften, die in den abgeleiteten Klassen erfasst werden.

Ziel der Aufgabe

Implementiere die Vererbungshierarchie für Personen in Entity Framework Code First, indem du:

1. Eine Basisklasse Person mit den allgemeinen Eigenschaften für alle Personen in der Bibliothek erstellst.
2. Abgeleitete Klassen für spezifische Personentypen mit zusätzlichen Eigenschaften definierst.

Anforderungen

Schritt 1: Definition der Basisklasse Person

Erstelle eine Basisklasse Person mit den folgenden Eigenschaften:

- Id: Primärschlüssel (int)
- FirstName: Vorname (string)
- LastName: Nachname (string)
- DateOfBirth: Geburtsdatum (DateTime)

Schritt 2: Definition der abgeleiteten Klassen für Personen

Erstelle die folgenden abgeleiteten Klassen für spezifische Personentypen:

1. **Author** (Autor) mit der zusätzlichen Eigenschaft:
 - Biography: (string)
2. **Customer** (Kunde) mit der zusätzlichen Eigenschaft:
 - MembershipDate: (DateTime)
3. **Librarian** (Bibliothekar) mit der zusätzlichen Eigenschaft:
 - EmployeeId: (string)

Schritt 3: Implementierung in Entity Framework

1. Setze die oben definierten Klassen in Entity Framework um.

```
// Basisklasse für Personen
public class Person {
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public DateTime DateOfBirth { get; set; }
}

// Abgeleitete Klasse für Autoren
public class Author : Person {
    public string Biography { get; set; }
}

// Abgeleitete Klasse für Kunden
public class Customer : Person {
    public DateTime MembershipDate { get; set; }
}

// Abgeleitete Klasse für Bibliothekare
public class Librarian : Person {
    public string EmployeeId { get; set; }
}
```

2. Konfiguriere den DbContext entsprechend, sodass die gewählte Vererbungsstrategie in der Datenbankstruktur korrekt abgebildet wird.

Schritt 4: Beispielhafte Datenbank-Seeding

Füge Beispielpersonen in die Datenbank ein, um sicherzustellen, dass die Vererbung korrekt umgesetzt wurde. Jeder Personentyp sollte vertreten sein.

```
-- Beispielhafte Personendaten (Joint Table)
INSERT INTO Persons (Id, FirstName, LastName, DateOfBirth) VALUES (1, 'Johann', 'Wolfgang', '1756-01-27');
INSERT INTO Persons (Id, FirstName, LastName, DateOfBirth) VALUES (2, 'Jane', 'Austen', '1775-12-16');
INSERT INTO Persons (Id, FirstName, LastName, DateOfBirth) VALUES (3, 'Agatha', 'Christie', '1890-09-15');
INSERT INTO Persons (Id, FirstName, LastName, DateOfBirth) VALUES (4, 'George', 'Orwell', '1903-06-25');
INSERT INTO Persons (Id, FirstName, LastName, DateOfBirth) VALUES (5, 'Isaac', 'Asimov', '1920-01-02');

-- Beispielhafte Autorendaten
INSERT INTO Authors (Id, FirstName, LastName, DateOfBirth, Biography) VALUES (1, 'Johann', 'Wolfgang', '1756-01-27', 'Komponist und Pianist.');
INSERT INTO Authors (Id, FirstName, LastName, DateOfBirth, Biography) VALUES (2, 'Jane', 'Austen', '1775-12-16', 'Berühmte britische Romanautorin.');
INSERT INTO Authors (Id, FirstName, LastName, DateOfBirth, Biography) VALUES (3, 'Agatha', 'Christie', '1890-09-15', 'Meisterin des Krimis.');
INSERT INTO Authors (Id, FirstName, LastName, DateOfBirth, Biography) VALUES (4, 'George', 'Orwell', '1903-06-25', 'Autor von politischen Romanen.');
```

```
INSERT INTO Authors (Id, FirstName, LastName, DateOfBirth, Biography) VALUES (5, 'Isaac', 'Asimov', '1920-01-02', 'Berühmter Science-Fiction-Autor.');
```

```
-- Beispielhafte Kundendaten
```

```
INSERT INTO Customers (Id, FirstName, LastName, DateOfBirth, MembershipDate) VALUES (1, 'Max', 'Mustermann', '1990-01-15', '2020-05-01');
```

```
INSERT INTO Customers (Id, FirstName, LastName, DateOfBirth, MembershipDate) VALUES (2, 'Erika', 'Musterfrau', '1992-03-22', '2021-08-15');
```

```
INSERT INTO Customers (Id, FirstName, LastName, DateOfBirth, MembershipDate) VALUES (3, 'Hans', 'Meier', '1985-11-30', '2019-11-10');
```

```
INSERT INTO Customers (Id, FirstName, LastName, DateOfBirth, MembershipDate) VALUES (4, 'Anna', 'Schmidt', '1988-07-12', '2022-01-18');
```

```
INSERT INTO Customers (Id, FirstName, LastName, DateOfBirth, MembershipDate) VALUES (5, 'Peter', 'Müller', '1995-09-05', '2023-03-25');
```

```
-- Beispielhafte Bibliothekarendaten
```

```
INSERT INTO Librarians (Id, FirstName, LastName, DateOfBirth, EmployeeId) VALUES (1, 'Karl', 'Klein', '1980-02-20', 'LIB001');
```

```
INSERT INTO Librarians (Id, FirstName, LastName, DateOfBirth, EmployeeId) VALUES (2, 'Laura', 'Lange', '1978-05-11', 'LIB002');
```

```
INSERT INTO Librarians (Id, FirstName, LastName, DateOfBirth, EmployeeId) VALUES (3, 'Tom', 'Hansen', '1982-12-01', 'LIB003');
```

```
INSERT INTO Librarians (Id, FirstName, LastName, DateOfBirth, EmployeeId) VALUES (4, 'Lisa', 'Schneider', '1985-03-30', 'LIB004');
```

```
INSERT INTO Librarians (Id, FirstName, LastName, DateOfBirth, EmployeeId) VALUES (5, 'Nina', 'Fischer', '1990-06-16', 'LIB005');
```